i.MX RT1170 Processor Reference Manual

Document Number: IMXRT1170RM
Rev. 4, 05/2025

Contents

Section number **Title** **Page**

Chapter 1
About this Document

1.1 Audience 27

1.2 Organization 27

1.3 Suggested Reading 27

1.4 Conventions 28

1.5 Register Access 30

1.6 Acronyms and Abbreviations 32

Chapter 2
Introduction

2.1 Introduction 35

2.2 Features 37

2.3 Target Applications 40

2.4 Endianness Support 41

Chapter 3
Memory Maps

3.1 Memory system overview 43

3.2 System memory map (CM7) 43

3.3 System memory map (CM4) 45

3.4 AIPS-1 Memory Map 46

3.5 AIPS-2 Memory Map 50

3.6 AIPS-3 Memory Map 52

3.7 AIPS-4 Memory Map 52

3.8 AIPS M7 Memory Map 54

3.9 PPB M7 Memory Map 54

3.10 PPB M4 Memory Map 55

Chapter 4
Interrupts, DMA Events, and XBAR Assignments

4.1 Overview....57
4.2 CM7 interrupts....57
4.3 CM4 interrupts....72
4.4 DMA Mux....86
4.5 PIT Channel Assignments For Periodic DMA Triggering....100
4.6 XBAR Resource Assignments....100

Chapter 5
Direct Memory Access Multiplexer (DMAMUX)

5.1 Chip-specific DMAMUX information....117
5.2 Overview....117
5.3 Functional description....119
5.4 External signals....124
5.5 Application information....124
5.6 Initialization....127
5.7 Memory map/register definition....127

Chapter 6
Enhanced Direct Memory Access (eDMA)

6.1 Chip-specific eDMA information....131
6.2 Overview....131
6.3 Functional description....135
6.4 Initialization/application information....141
6.5 Memory map/register definition....156

Chapter 7
System Security

7.1 Chapter overview....219
7.2 Feature summary....219
7.3 Resource Management....222
7.4 Secure Key Manager (KEYMGR)....222
7.5 High-Assurance Boot (HAB)....223
7.6 Secure Non-Volatile Storage (SNVS) module....226

7.7 Cryptographic Acceleration and Assurance Module (CAAM)..... 228
7.8 On-Chip OTP Controller (OCOTP_CTRL).....229
7.9 Debug.....230
7.10 Inline Encryption Engine (IEE)..... 232
7.11 On-the-Fly AES Decryption (OTFAD)..... 232
7.12 Memory-Mapped Cryptographic Acceleration Unit (MMCAU)..... 233
7.13 Code Watchdog Timer (CDOG).....233

Chapter 8
System Debug

8.1 Overview.....235
8.2 Chip and Arm Platform Debug Architecture.....235
8.3 Miscellaneous..... 243

Chapter 9
JTAG Controller (JTAGC)

9.1 Chip-specific JTAG information..... 245
9.2 Overview.....245
9.3 Functional description.....246
9.4 External signal description.....258
9.5 Initialization/application information..... 259
9.6 Register description..... 260

Chapter 10
System Boot

10.1 Chip-specific Boot Information.....263
10.2 Overview.....271
10.3 Boot modes..... 273
10.4 Device configuration.....277
10.5 Device initialization.....279
10.6 Boot devices.....282
10.7 Program image.....327
10.8 External Memory Configuration Data (XMCD).....335

10.9 Serial Boot (Serial Downloader)....345
10.10 Recovery devices....358
10.11 SD/MMC manufacture mode....359
10.12 High-Assurance Boot (HAB)....359
10.13 ROM APIs....361

Chapter 11
External Signals and Pin Multiplexing

11.1 Overview....373

Chapter 12
IOMUX Controller (IOMUXC)

12.1 Chip-specific IOMUXC information....451
12.2 Overview....451
12.3 Functional description....453
12.4 Memory Map and register definition....456

Chapter 13
General Purpose Input/Output (GPIO)

13.1 Chip-specific GPIO information....1375
13.2 Overview....1376
13.3 Functional Description....1377
13.4 External Signals....1384
13.5 Application Information....1384
13.6 Memory Map/Register Definition....1385

Chapter 14
Clock and Power Management

14.1 Introduction....1403
14.2 Components of Clock and Power Management....1403
14.3 Power Management....1404
14.4 Clock Generation....1414

Chapter 15
Clock Controller Module (CCM)

15.1 Chip-specific CCM information.... 1423
15.2 Overview.... 1423
15.3 System Clocks.... 1425
15.4 Clock Tree.... 1441
15.5 Functional description.... 1447
15.6 External Signals.... 1482
15.7 Analog IP (AI) Interface.... 1482
15.8 Programming Guidelines.... 1484
15.9 Memory Map and register definition.... 1487

Chapter 16
Crystal Oscillator (XTALOSC)

16.1 Chip-specific XTALOSC information.... 1649
16.2 Overview.... 1649
16.3 Functional Description.... 1650
16.4 External Signals.... 1653
16.5 Memory Map and register definition.... 1654

Chapter 17
Power Management Unit (PMU)

17.1 Chip-specific PMU information.... 1675
17.2 Overview.... 1675
17.3 Functional Description.... 1676
17.4 External Signals.... 1682
17.5 Memory Map and register definition.... 1682

Chapter 18
PHY LDO (PHY_LDO)

18.1 Chip-specific PHY LDO information.... 1763
18.2 Overview.... 1763
18.3 Memory Map and register definition.... 1764

Chapter 19
General Power Controller (GPC)

19.1 Chip-specific General Power Controller (GPC) information.... 1769
19.2 Overview.... 1769
19.3 Functional description.... 1771
19.4 External Signals.... 1779
19.5 Initialization.... 1779
19.6 Memory Map and register definition.... 1780

Chapter 20
Power Gating and Memory Controller (PGMC)

20.1 Chip-specific Power Gating and Memory Controller (PGMC) information.... 1901
20.2 Overview.... 1901
20.3 Functional description.... 1903
20.4 External Signals.... 1908
20.5 Initialization.... 1908
20.6 Memory Map and register definition.... 1909

Chapter 21
DCDC Converter (DCDC)

21.1 Chip-specific DCDC information.... 1951
21.2 Overview.... 1951
21.3 Functional description.... 1953
21.4 External Signals.... 1957
21.5 Initialization.... 1957
21.6 Application Information.... 1958
21.7 Memory Map and register definition.... 1961

Chapter 22
Temperature Sensor (TMPSNS)

22.1 Chip-specific TMPSNS information.... 1999
22.2 Overview.... 1999
22.3 Functional description.... 2000
22.4 External Signals.... 2002
22.5 Initialization.... 2002

22.6 Application Information....2002
22.7 Memory Map and register definition....2003

Chapter 23
State Save and Restore Controller (SSARC)

23.1 Chip-specific State Save and Restore Controller (SSARC) information....2017
23.2 Overview....2017
23.3 Functional description....2018
23.4 External Signals....2021
23.5 Initialization....2021
23.6 Memory Map and register definition....2021

Chapter 24
Secure Non-Volatile Storage (SNVS)

24.1 Chip-specific SNVS information....2039
24.2 Overview....2039
24.3 SNVS functional description....2041
24.4 External Signals....2046
24.5 Initialization of SNVS....2047
24.6 Memory Map and register definition....2048

Chapter 25
System Reset Controller (SRC)

25.1 Chip-specific SRC information....2075
25.2 Overview....2075
25.3 Functional Description....2076
25.4 External Signals....2085
25.5 Initialization....2085
25.6 Memory Map and register definition....2086

Chapter 26
Fusemap

26.1 Overview....2109
26.2 Boot Fusemap....2111

26.3 Fusemap Descriptions Table....2120
26.4 FlexRAM Partition Table.... 2124

Chapter 27
On-Chip OTP Controller (OCOTP_CTRL)

27.1 Chip-specific OCOTP_CTRL information....2127
27.2 Overview....2127
27.3 Functional Description....2129
27.4 External Signals.... 2135
27.5 Fuse Map....2135
27.6 Memory Map/Register Definition....2135

Chapter 28
External Memory Controllers

28.1 Overview....2159
28.2 eMMC/eSD/SDIO....2159
28.3 Flexible Serial Peripheral Interface.... 2159

Chapter 29
Smart External Memory Controller (SEMC)

29.1 Chip-specific SEMC information.... 2161
29.2 Overview....2161
29.3 Functional Description....2164
29.4 External Signals.... 2212
29.5 Initialization.... 2217
29.6 Application Information....2217
29.7 Memory Map and Register Definition....2219

Chapter 30
FlexSPI Controller (FLEXSPI)

30.1 Chip-specific FlexSPI information.... 2289
30.2 Overview....2292
30.3 Functional description....2294
30.4 External signals....2353

30.5 Initialization..... 2354
30.6 Application information..... 2355
30.7 Memory map and register definition..... 2373
30.8 AHB memory map definition..... 2436

Chapter 31
External ECC Controller (XECC)

31.1 Chip-specific XECC information..... 2439
31.2 Overview..... 2439
31.3 Functional description..... 2441
31.4 External Signals..... 2445
31.5 Initialization..... 2445
31.6 Memory Map and register definition..... 2445

Chapter 32
Ultra Secured Digital Host Controller (uSDHC)

32.1 Chip-specific uSDHC information..... 2471
32.2 Overview..... 2471
32.3 Functional description..... 2474
32.4 External signals..... 2507
32.5 Application of uSDHC..... 2508
32.6 Software restrictions..... 2534
32.7 uSDHC memory map/register definition..... 2536

Chapter 33
ARM Cortex M7 Platform (M7)

33.1 Chip-specific Arm Cortex M7 information..... 2611
33.2 Overview..... 2613
33.3 Functional Description..... 2614
33.4 External Signals..... 2615
33.5 Memory Map and register definition..... 2615

Chapter 34
ARM Cortex M4 Platform (M4)

34.1 Chip-specific Arm Cortex M4 information.... 2619
34.2 Overview.... 2620
34.3 Configuration.... 2622
34.4 Performance.... 2622
34.5 Platform sub-blocks.... 2623
34.6 Debug.... 2625
34.7 Local Memory Controller (LMEM).... 2627
34.8 Miscellaneous Control Module (MCM).... 2652
34.9 MCM Memory Map/Register Definition.... 2653

Chapter 35
Messaging Unit (MU)

35.1 Chip-specific MU information.... 2677
35.2 Overview.... 2677
35.3 Functional Description.... 2679
35.4 External Signals.... 2691
35.5 Initialization.... 2691
35.6 Application Information.... 2692
35.7 MU Processor A-side Memory Map/Register Definition.... 2694
35.8 MU Processor B-side Memory Map/Register Definition.... 2708

Chapter 36
Network Interconnect Bus System (NIC-301)

36.1 Chip-specific NIC-301 information.... 2723
36.2 Overview.... 2723
36.3 External Signals.... 2724
36.4 Memory Map and Register Definition.... 2724

Chapter 37
On-Chip RAM Memory Controller (OCRAM)

37.1 Chip-specific OCRAM information.... 2731
37.2 Overview.... 2731
37.3 Functional Description.... 2733

37.4 Programmable Registers.....2734

Chapter 38
MECC64

38.1 Chip-specific MECC information.....2735
38.2 Overview.....2735
38.3 Functional description.....2737
38.4 Signals.....2739
38.5 Initialization.....2740
38.6 Application information.....2740
38.7 Memory Map and register definition.....2740

Chapter 39
FlexRAM

39.1 Chip-specific FlexRAM information.....2765
39.2 Overview.....2766
39.3 Functional description.....2768
39.4 Memory Map and Register Definition.....2774

Chapter 40
AHB to IP Bridge (AIPS Lite)

40.1 Chip-specific AIPS Lite information.....2833
40.2 Overview.....2833
40.3 Functional Description.....2833
40.4 Memory Map/Register Definition.....2834

Chapter 41
Semaphores (SEMA4)

41.1 Chip-specific SEMA4 information.....2835
41.2 Overview.....2835
41.3 Functional Description.....2837
41.4 External Signal Description.....2842
41.5 Initialization Information.....2842
41.6 Application Information.....2842

41.7 Memory map and register definition....2844

Chapter 42
Resource Domain Controller (RDC)

42.1 Chip-specific RDC information....2855
42.2 Overview....2856
42.3 Functional Description....2857
42.4 External Signals....2864
42.5 Programming Interface....2864
42.6 RDC Memory Map/Register Definition....2871
42.7 RDC SEMA42 Memory Map/Register Definition....2890

Chapter 43
Extended Resource Domain Controller 2 (XRDC2)

43.1 Chip-specific XRDC2 information....2899
43.2 Overview....2909
43.3 Functional description....2911
43.4 External Signals....2918
43.5 Initialization....2918
43.6 Application information....2918
43.7 Memory Map and register definition....2919

Chapter 44
Display And Camera Overview

44.1 Overview....2943

Chapter 45
CMOS Sensor Interface (CSI)

45.1 Chip-specific CSI information....2947
45.2 Overview....2947
45.3 Functional Description....2949
45.4 External Signals....2969
45.5 Initialization....2969
45.6 CSI Memory Map/Register Definition....2969

Chapter 46
MIPI CSI2 (MIPI_CSI2)

46.1 Chip-specific MIPI CSI-2 information....3007
46.2 Overview....3007
46.3 Functional description....3010
46.4 External Signals....3022
46.5 Initialization....3022
46.6 Memory Map and register definition....3023

Chapter 47
Enhanced LCD Interface (eLCDIF)

47.1 Chip-specific eLCDIF information....3039
47.2 Overview....3039
47.3 Functional Description....3041
47.4 External Signals....3058
47.5 Initialization....3058
47.6 eLCDIF Memory Map/Register Definition....3059

Chapter 48
LCDIF Interface v2 (LCDIF v2)

48.1 Chip-specific LCDIFv2 information....3089
48.2 Overview....3089
48.3 Functional description....3092
48.4 External Signals....3099
48.5 Application information....3099
48.6 Memory Map and register definition....3100

Chapter 49
MIPI DSI Host Controller (MIPI_DSI)

49.1 Chip-specific MIPI DSI information....3131
49.2 Overview....3131
49.3 Functional description....3134
49.4 External Signals....3155

49.5 Application Information....3155
49.6 Initialization....3156
49.7 Memory Map and register definition....3156

Chapter 50
Video Mux Controller (VIDEO_MUX)

50.1 Chip-specific VIDEO MUX information....3227
50.2 Overview....3227
50.3 Functional description....3229
50.4 External Signals....3230
50.5 Initialization....3231
50.6 Memory Map and register definition....3231

Chapter 51
Display Content Integrity Checker (DCIC)

51.1 Chip-specific DCIC information....3239
51.2 Overview....3239
51.3 Functional Description....3240
51.4 External Signals....3244
51.5 System Considerations....3244
51.6 DCIC Memory Map/Register Definition....3245

Chapter 52
Pixel Pipeline (PXP)

52.1 Chip-specific PXP information....3257
52.2 Overview....3257
52.3 Functional Description....3258
52.4 External Signals....3293
52.5 Initialization....3293
52.6 PXP Memory Map/Register Definition....3293

Chapter 53
Graphics Processing Unit (GPU2D)

53.1 Chip-specific GPU information....3331

53.2 Overview....3331
53.3 Functional description....3333
53.4 Software Application Programming Interface (API)....3334

Chapter 54
Audio Overview

54.1 Audio Overview....3337

Chapter 55
Asynchronous Sample Rate Converter (ASRC)

55.1 Chip-specific Asynchronous Sample Rate Converter (ASRC) information....3345
55.2 Overview....3346
55.3 Functional Description....3349
55.4 External Signals....3358
55.5 Application Information....3359
55.6 ASRC Memory Map/Register Definition....3360

Chapter 56
PDM Microphone Interface (PDM)

56.1 Chip-specific PDM Microphone Interface (PDM) information....3413
56.2 Overview....3413
56.3 Functional Description....3415
56.4 External Signals....3432
56.5 Initialization....3433
56.6 Application Information....3437
56.7 PDM register descriptions....3437

Chapter 57
Medium Quality Sound (MQS)

57.1 Chip-specific MQS information....3463
57.2 Overview....3463
57.3 Functional Description....3464
57.4 External Signals....3465
57.5 Application Information....3466

Chapter 58
Synchronous Audio Interface (SAI)

58.1 Chip-specific SAI information....3467
58.2 Overview....3468
58.3 Functional description....3470
58.4 External signals....3481
58.5 Memory map and register definition....3481

Chapter 59
Sony/Philips Digital Interface (SPDIF)

59.1 Chip-specific SPDIF information....3519
59.2 Overview....3519
59.3 Functional Description....3521
59.4 External Signals....3531
59.5 Application Information....3532
59.6 SPDIF Memory Map/Register Definition....3533

Chapter 60
10/100 /1000 -Mbps Ethernet MAC (ENET/ENET1G)

60.1 Chip-specific ENET information....3559
60.2 Overview....3559
60.3 Functional description....3564
60.4 External Signals....3630
60.5 Memory map/register definition....3632

Chapter 61
Ethernet Quality Of Service (ENET_QOS)

61.1 Chip-specific ENET QOS information....3765
61.2 Overview....3765
61.3 Functional description....3767
61.4 External signals....3924
61.5 Initialization....3927
61.6 Application Information....3931

61.7 Memory Map and register definition.....3951

Chapter 62
Universal Serial Bus Controller (USB)

62.1 Chip-specific USB information.....4839
62.2 Overview.....4839
62.3 Functional Description.....4841
62.4 External Signals.....4846
62.5 USB Operation Model.....4846
62.6 USB Non-Core Memory Map/Register Definition.....5013
62.7 USB Core Memory Map/Register Definition.....5019

Chapter 63
Universal Serial Bus 2.0 Integrated PHY (USB-PHY)

63.1 Chip-specific USB-PHY information.....5109
63.2 Overview.....5109
63.3 Functional Description.....5110
63.4 External Signals.....5117
63.5 Initialization and Application Information.....5117
63.6 Recommended Register Configuration for USB Certification.....5117
63.7 Register Macro Usage.....5118
63.8 USB PHY Memory Map/Register Definition.....5119

Chapter 64
USB Device Charger Detection Module (USBDCD)

64.1 Chip-specific USBDCD information.....5159
64.2 Overview.....5159
64.3 Functional description.....5161
64.4 External Signals.....5178
64.5 Initialization information.....5179
64.6 Application information.....5179
64.7 Memory map/Register definition.....5180

Chapter 65
Euro, MasterCard, Visa Subscriber Identification Module (EMVSIM)

65.1 Chip-specific EMVSIM information....5193
65.2 Overview....5193
65.3 Functional Description....5195
65.4 External Signals....5211
65.5 Initialization....5211
65.6 Application Information....5214
65.7 Memory Map and Registers....5219

Chapter 66
Flexible Controller Area Network (FLEXCAN)

66.1 Chip-specific FLEXCAN information....5251
66.2 Overview....5251
66.3 Functional description....5254
66.4 FlexCAN signal descriptions....5311
66.5 Initialization/application information....5311
66.6 Memory map/register definition....5313

Chapter 67
Flexible I/O (FlexIO)

67.1 Chip-specific FlexIO information....5393
67.2 Overview....5393
67.3 Functional description....5395
67.4 Application Information....5411
67.5 FlexIO Signal Descriptions....5431
67.6 Memory Map and Registers....5431

Chapter 68
Keypad Port (KPP)

68.1 Chip-specific KPP information....5467
68.2 Overview....5467
68.3 Functional Description....5469

68.4 External Signals.....5477
68.5 Initialization/Application Information.....5479
68.6 Memory Map and Register Definition.....5480

Chapter 69
Low Power Inter-Integrated Circuit (LPI2C)

69.1 Chip-specific LPI2C information.....5487
69.2 Overview.....5487
69.3 Functional description.....5490
69.4 Signal descriptions.....5502
69.5 Memory Map and Registers.....5503

Chapter 70
Low Power Serial Peripheral Interface (LPSPI)

70.1 Chip-specific LPSPI information.....5547
70.2 Overview.....5547
70.3 Functional description.....5549
70.4 Signals.....5560
70.5 Memory map and registers.....5561

Chapter 71
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

71.1 Chip-specific LPUART information.....5587
71.2 Overview.....5587
71.3 Functional description.....5590
71.4 External signals.....5608
71.5 Register definition.....5608

Chapter 72
Timers Overview

72.1 Overview.....5637

Chapter 73
Enhanced Flex Pulse Width Modulator (eFlexPWM)

73.1 Chip-specific FlexPWM information.....5641

73.2 Overview....5641
73.3 Functional description....5643
73.4 External signals....5685
73.5 PWM register descriptions....5687

Chapter 74
General Purpose Timer (GPT)

74.1 Chip-specific GPT information....5821
74.2 Overview....5821
74.3 External Signals....5823
74.4 Clocks....5824
74.5 Functional Description....5826
74.6 Initialization/ Application Information....5830
74.7 Memory map/register definition....5831

Chapter 75
Periodic Interrupt Timer (PIT)

75.1 Chip-specific PIT information....5843
75.2 Introduction....5843
75.3 Modes of operation....5845
75.4 Functional description....5845
75.5 Initialization and application information....5849
75.6 Memory Map and Registers....5850

Chapter 76
Quad Timer (TMR)

76.1 Chip-specific TMR information....5859
76.2 Overview....5859
76.3 Modes of Operation....5861
76.4 Functional Description....5861
76.5 Resets....5878
76.6 Clocks....5878
76.7 Interrupts....5879

76.8 DMA 5881
76.9 Memory map/register definition 5881

Chapter 77
Quadrature Decoder (QDC)

77.1 Chip-specific QDC information 5901
77.2 Overview 5901
77.3 Functional description 5906
77.4 Memory map and registers 5917

Chapter 78
System Watchdog Timer (WDOG 1,2)

78.1 Chip-specific System Watchdog information 5945
78.2 Overview 5945
78.3 Functional description 5947
78.4 External signals 5956
78.5 Initialization 5957
78.6 WDOG Memory Map/Register Definition 5957

Chapter 79
CPU Watchdog Timer (WDOG 3,4)

79.1 Chip-specific CPU Watchdog information 5965
79.2 Overview 5966
79.3 Functional description 5967
79.4 External signals 5973
79.5 Application Information 5974
79.6 Memory map and register definition 5975

Chapter 80
External Watchdog Module (EWM)

80.1 Chip-specific EWM information 5983
80.2 Overview 5983
80.3 Functional Description 5985
80.4 EWM Signal Descriptions 5990

80.5 Memory Map/Register Definition....5990

Chapter 81
On Chip Cross Triggers Overview

81.1 Overview....5999

Chapter 82
And-Or-Inverter (AOI)

82.1 Chip-specific AOI information....6001
82.2 Overview....6001
82.3 Functional Description....6004
82.4 External Signal Description....6007
82.5 Memory Map and Register Descriptions....6007

Chapter 83
Inter-Peripheral Crossbar Switch A (XBARA)

83.1 Chip-specific XBAR information....6013
83.2 Overview....6013
83.3 Functional Description....6015
83.4 External Signals....6016
83.5 Memory Map and Register Descriptions....6018

Chapter 84
Inter-Peripheral Crossbar Switch B (XBARB)

84.1 Chip-specific XBAR information....6063
84.2 Overview....6063
84.3 Functional Description....6063
84.4 External Signals....6064
84.5 Memory Map and Register Descriptions....6064

Chapter 85
Analog Overview

85.1 Overview....6071

Chapter 86
Analog Comparator (ACMP)

86.1 Chip-specific CMP information....6073
86.2 Overview....6074
86.3 CMP functional description....6080
86.4 DAC functional description....6096
86.5 CMP external signals....6100
86.6 Initialization....6100
86.7 Memory map/register definitions....6101

Chapter 87
Analog-to-Digital Converter (LPADC)

87.1 Chip-specific LPADC information....6115
87.2 Overview....6117
87.3 Functional description....6119
87.4 Signal descriptions....6129
87.5 Initialization....6131
87.6 Application information....6132
87.7 Memory map and register definition....6142

Chapter 88
ADC External Trigger Control (ADC_ETC)

88.1 Chip-specific ADC_ETC information....6175
88.2 Overview....6175
88.3 Functional description....6178
88.4 Initialization....6181
88.5 Application information....6181
88.6 Memory Map and register definition....6184

Chapter 89
Digital-to-Analog Converters (DAC)

89.1 Chip-specific DAC information....6217
89.2 Overview....6217
89.3 Functional description....6219
89.4 Memory map/register definition....6221

Chapter 1
About this Document

1.1 Audience

The reference manual is intended for the board-level product designers and product software developers. This manual assumes that the reader has a background in computer engineering and/or software engineering and understands the concepts of the digital system design, microprocessor architecture, input/output (I/O) devices, industry standard communication, and device interface protocols.

1.2 Organization

The reference manual describes the chip at a system level and provides an architectural overview. It also describes the system memory map, system-level interrupt events, external pins and pin multiplexing, external memory, system debug, system boot, multimedia subsystem, power management, and system security.

1.3 Suggested Reading

This section lists additional resources that provide background for the information in the reference manual, as well as general information about the architecture.

Table 1-1. Suggested Reading

Type	Description	Resource
Fact Sheet	The Fact Sheet is an overview of the product key features and its uses.	i.MX RT Series Crossover Processor Fact Sheet
Reference Manual	The Reference Manual contains a comprehensive description of the structure and function (operation) of a device.	
Data Sheet	The Data Sheet includes electrical characteristics and signal connections.	

Table continues on the next page...

Table 1-1. Suggested Reading (continued)

Type	Description	Resource
Chip Errata	The Chip mask set Errata provides additional or corrective information for a particular device mask set.	
Application Notes	Provides additional information about particular device feature or function.	
	AN12077: Using the i.MX RT FlexRAM	AN12077
	AN12879: How to Enable Spread Spectrum for RT Family	AN12879
	AN12940: Use Case of RT1170 LCD Display System based on MIPI DSI	AN12940
	Other Application Notes	
Web Sites	Product summary page on nxp.com with documentation, software, and resources for the device.	
	Product documentation page on nxp.com with a list of all documentation related to the device.	
Community Forum	i.MX RT Community support forum for questions, support, and information about the device.	i.MX RT Community

1.4 Conventions

The reference manual uses the following notational conventions:

cleared / set

When a bit has a value of zero, it is said to be cleared; when it has a value of one, it is said to be set.

mnemonics

Instruction mnemonics are shown in lowercase bold.

italics

Italics indicate variable command parameters, for example, **bcctr***x*.

The book titles in the text are set in italics.

15

An integer in decimal.

0x

the prefix to denote a hexadecimal number.

0b

The prefix to denote a binary number. Binary values of 0 and 1 are written without a prefix.

n'H4000CA00

The n-bit hexadecimal number.

BLK_REG_NAME

The register names are all uppercase. The block mnemonic is prepended with an underscore delimiter (_).

BLK_REG[FIELD]

The fields within registers appear in brackets. For example, ESR[RLS] refers to the Receive Last Slot field of the ESAI Status Register.

BLK_REG[*n*]

The bit number *n* within the BLK.REG register.

BLK_REG[*l:r*]

The register bit ranges. The ranges are indicated by the left-most bit number *l* and the right-most bit number *r*, separated by a colon (:). For example, ESR[15:0] refers to the lower half word in the ESAI Status Register.

x, U

In some contexts, such as signal encodings, an unitalicized x indicates a "don't care" or "uninitialized". The binary value can be 1 or 0.

x

An italicized *x* indicates an alphanumeric variable.

n, m

Italicized *n or m* represent integer variables.

!

Binary logic operator NOT.

&&

Binary logic operator AND.

||

Binary logic operator OR.

^ or <O+>

Binary logic operator XOR. For example, A <O+> B.

|

Bit-wise OR. For example, 0b0001 | 0b1000 yields the value of 0b1001.

&

Bit-wise AND. For example, 0b0001 & 0b1000 yields the value of 0b0000.

{A,B}

Concatenation, where the *n*-bit value A is prepended to the *m*-bit value B to form an (*n* +*m*)-bit value. For example, {0, REG*m* [14:0]} yeilds a 16-bit value with 0 in the most significant bit.

- or grey fill

Indicates a reserved bit field in a register. Although these bits can be written to with ones or zeros, they always read zeros.

>>

Shift right logical one position.

<<

Shift left logical one position.

<=
    Assignment.
==
    Compare equal.
!=
    Compare not equal.
>
    Greater than.
<
    Less than.

1.5 Register Access

1.5.1 Register Diagram Field Access Type Legend

This figure provides the interpretation of the notation used in the register diagrams for a number of common field access types:

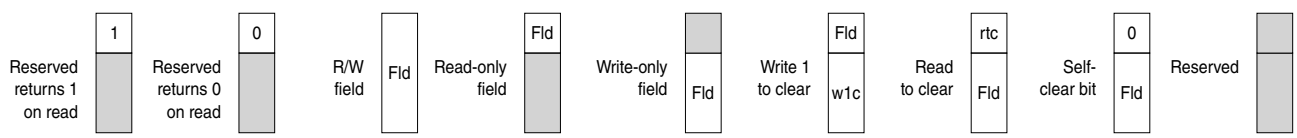


Figure 1-1. Register Field Conventions

NOTE

For reserved register fields, the software should mask off the data in the field after a read (the software can't rely on the contents of data read from a reserved field) and always write all zeros.

1.5.2 Register Macro Usage

A common operation is to update one field without disturbing the contents of the remaining fields in the register. Normally, this requires a read-modify-write (RMW) operation, where the CPU reads the register, modifies the target field, then writes the results back to the register. This is an expensive operation in terms of CPU cycles, because of the initial register read.

To address this issue, some hardware registers are implemented as a group, including registers that can be used to either set, clear, or toggle (SCT) individual bits of the primary register. When writing to an SCT register, all the bits set to 1 perform the associated operation on the primary register, while the bits set to 0 are not affected. The SCT registers always read back 0, and should be considered write-only. The SCT registers are not implemented if the primary register is read-only.

With this architecture, it is possible to update one or more fields using only register writes. First, all bits of the target fields are cleared by a write to the associated clear register, then the desired value of the target fields is written to the set register. This sequence of two writes is referred to as a clear-set (CS) operation.

A CS operation does have one potential drawback. Whenever a field is modified, the hardware sees a value of 0 before the final value is written. For most fields, passing through the 0 state is not a problem. Nonetheless, this behavior is something to consider when using a CS operation.

Also, a CS operation is not required for fields that are one-bit wide. While the CS operation works in this case, it is more efficient to simply set or clear the target bit (that is, one write instead of two). A simple set or clear operation is also atomic, while a CS operation is not.

Note that not all macros for set, clear, or toggle (SCT) are atomic. For registers that do not provide hardware support for this functionality, these macros are implemented as a sequence of read-modify-write operations. When an atomic operation is required, the developer should pay attention to this detail, because unexpected behavior might result if an interrupt occurs in the middle of the critical section comprising the update sequence.

A set of SCT registers is offered for registers in many modules on this device, as described in this manual. In a module memory map table, the suffix _SET, _CLR, or _TOG is added to the base name of the register. For example, the CCM_ANALOG_PLL_ARM register has three other registers called CCM_ANALOG_PLL_ARM_SET, CCM_ANALOG_PLL_ARM_CLR, and CCM_ANALOG_PLL_ARM_TOG.

In the sub-section that describes one of these sets of registers, a short-hand convention is used to denote that a register has the SCT register set. There is an italicized *n* appended to the end of the short register name. Using the above example, the name used for this register is CCM_ANALOG_PLL_ARM*n*. When you see this designation, there is a SCT register set associated with the register, and you can verify this by checking it in the memory map table. The address offset for each of these registers is given in the form of the following example:

Address: 20C_8000h base + 0h offset + (4d × i), where i=0d to 3d

In this example, the address for each of the base registers and their three SCT registers can be calculated as:

Register	Address
CCM_ANALOG_PLL_ARM	20C_8000h
CCM_ANALOG_PLL_ARM_SET	20C_8004h
CCM_ANALOG_PLL_ARM_CLR	20C_8008h
CCM_ANALOG_PLL_ARM_TOG	20C_800Ch

1.6 Acronyms and Abbreviations

The table below contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
ACMP	Analog Comparator
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AIPS	Arm IP Bus
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASRC	Asynchronous Sample Rate Converter
AXI	Advanced eXtensible Interface
CA/CM	Arm Cortex-A/Cortex-M
CAAM	Cryptographic Acceleration and Assurance Module
CAN	Controller Area Network
CCM	Clock Controller Module
CM7	Arm Cortex M7 Core
CPU	Central Processing Unit
CSI	CMOS Sensor Interface
CSU	Central Security Unit
CTI	Cross Trigger Interface
D-cache	Data cache
DAP	Debug Access Port
DDR	Double data rate
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory

Table continues on the next page...

Term	Meaning
ECC	Error correcting codes
LPSPI	Low-power SPI
EDMA	Enhanced Direct Memory Access
EIM	External Interface Module
ENET	Ethernet
EPIT	Enhanced Periodic Interrupt Timer
EPROM	Erasable Programmable Read-Only Memory
ETF	Embedded Trace FIFO
ETM	Embedded Trace Macrocell
FIFO	First-In-First-Out
GIC	General Interrupt Controller
GPC	General Power Controller
GPIO	General-Purpose I/O
GPR	General-Purpose Register
GPS	Global Positioning System
GPT	General-Purpose Timer
GPU	Graphics Processing Unit
GPV	Global Programmers View
HAB	High-Assurance Boot
I-cache	Instruction cache
I2C or I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IOMUX	Input-Output Multiplexer
IP	Intellectual Property
IrDA	Infrared Data Association
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
ELCDIF	Liquid Crystal Display Interface
LDO	Low-Dropout
LIFO	Last-In-First-Out
LRU	Least-Recently Used
LPSR	Low-Power State Retention
LSB	Least-Significant Byte
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MAC	Medium Access Control
MCM	Miscellaneous Control Module
MMC	Multimedia Card
MSB	Most-Significant Byte
MT/s	Mega Transfers per second
OCRAM	On-Chip Random-Access Memory

Table continues on the next page...

Term	Meaning
OCOTP	On-Chip One-Time Programmable Controller
PCI	Peripheral Component Interconnect
PCIe	PCI express
BPC	Basic Power Controller
PIC	Programmable Interrupt Controller
PMU	Power Management Unit
POR	Power-On Reset
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse Width Modulation
PXP	Pixel Pipeline
QoS	Quality of Service
R2D	Radians to Degrees
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RTOS	Real-Time Operating System
Rx	Receive
SAI	Synchronous Audio Interface
SD	Secure Digital
SDIO	Secure Digital Input/Output
SDLC	Synchronous Data Link Control
SDMA	Smart DMA
SIM	Subscriber Identification Module
SNVS	Secure Non-Volatile Storage
SoC	System-on-Chip
SPBA	Shared Peripheral Bus Arbiter
SPDIF	Sony Philips Digital Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRC	System Reset Controller
TFT	Thin-Film Transistor
TPIU	Trace Port Interface Unit
TSGEN	Time Stamp Generator
Tx	Transmit
TZASC	TrustZone Address Space Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USDHC	Ultra Secured Digital Host Controller
WDOG	Watchdog
WLAN	Wireless Local Area Network
WXGA	Wide Extended Graphics Array

Chapter 2
Introduction

2.1 Introduction

The i.MX RT1170 is a new processor family featuring NXP's advanced implementation of the high performance Arm Cortex®-M7 Core and a power efficient Arm Cortex®-M4 Core. It offers high-performance processing optimized for lowest power consumption and best real-time response.

The i.MX RT1170 has 2MB on-chip RAM in total, including a 512KB RAM which can be flexibly configured as TCM or general-purpose on-chip RAM. The i.MX RT1170 integrates advanced power management module with DCDC and LDO that reduces complexity of external power supply and simplifies power sequencing. The i.MX RT1170 also provides various memory interfaces, including SDRAM, Raw NAND FLASH, NOR FLASH, SD/eMMC, Quad SPI, HyperRAM/HyperFlash and a wide range of other interfaces for connecting peripherals, such as WLAN, Bluetooth®, GPS, displays, and camera sensors. Same as other i.MX processors, i.MX RT1170 also has rich audio and video features, including MIPI CSI/DSI, LCD display, graphics accelerator, camera interface, SPDIF and I2S audio interface.

2.1.1 Block Diagram

The functional block diagram is shown in the figure below. This diagram provides a view of the chip's major functional components and core complexes.

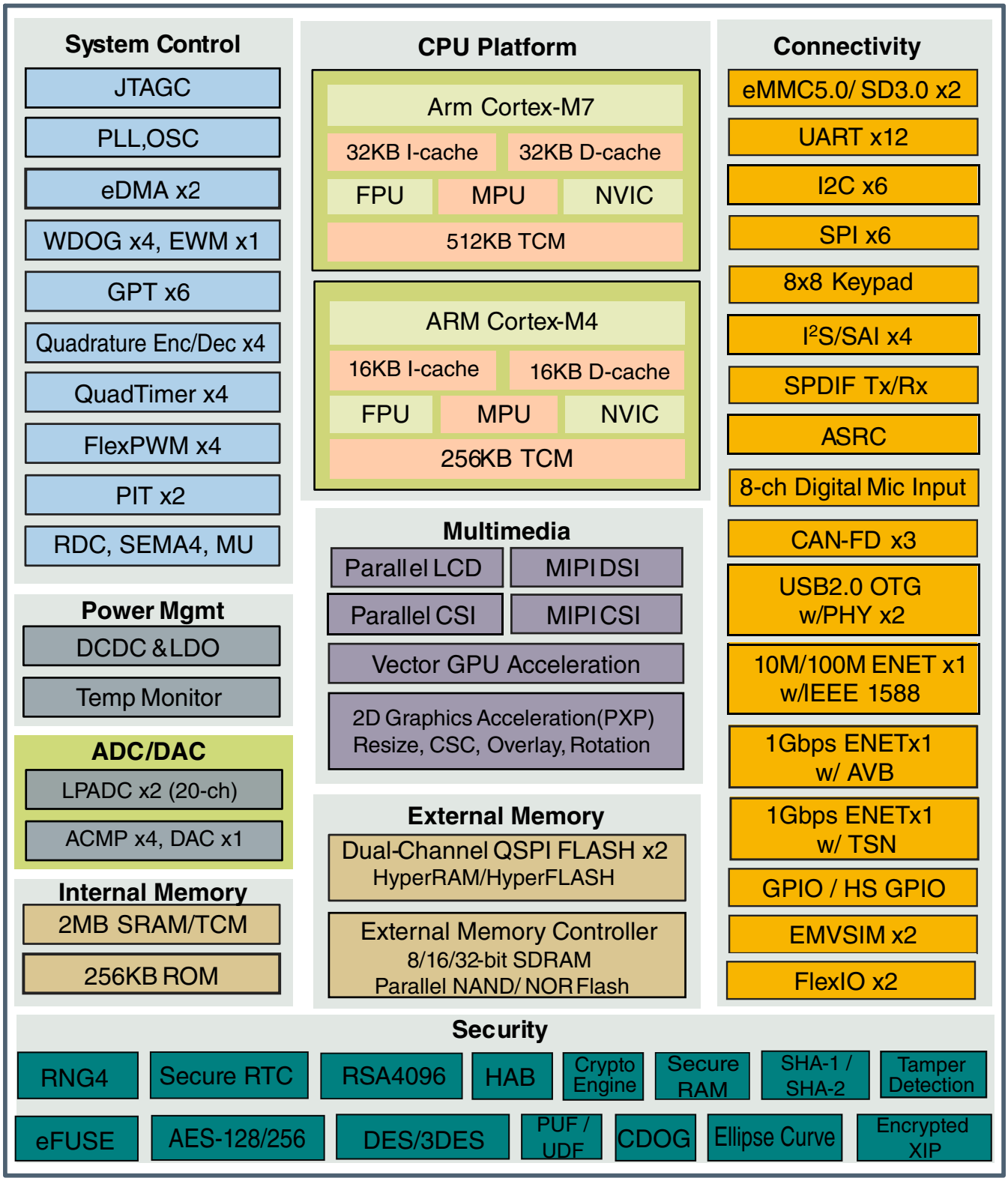


Figure 2-1. Simplified Block Diagram

2.1.2 System Bus Diagram

The system bus diagram is shown below.

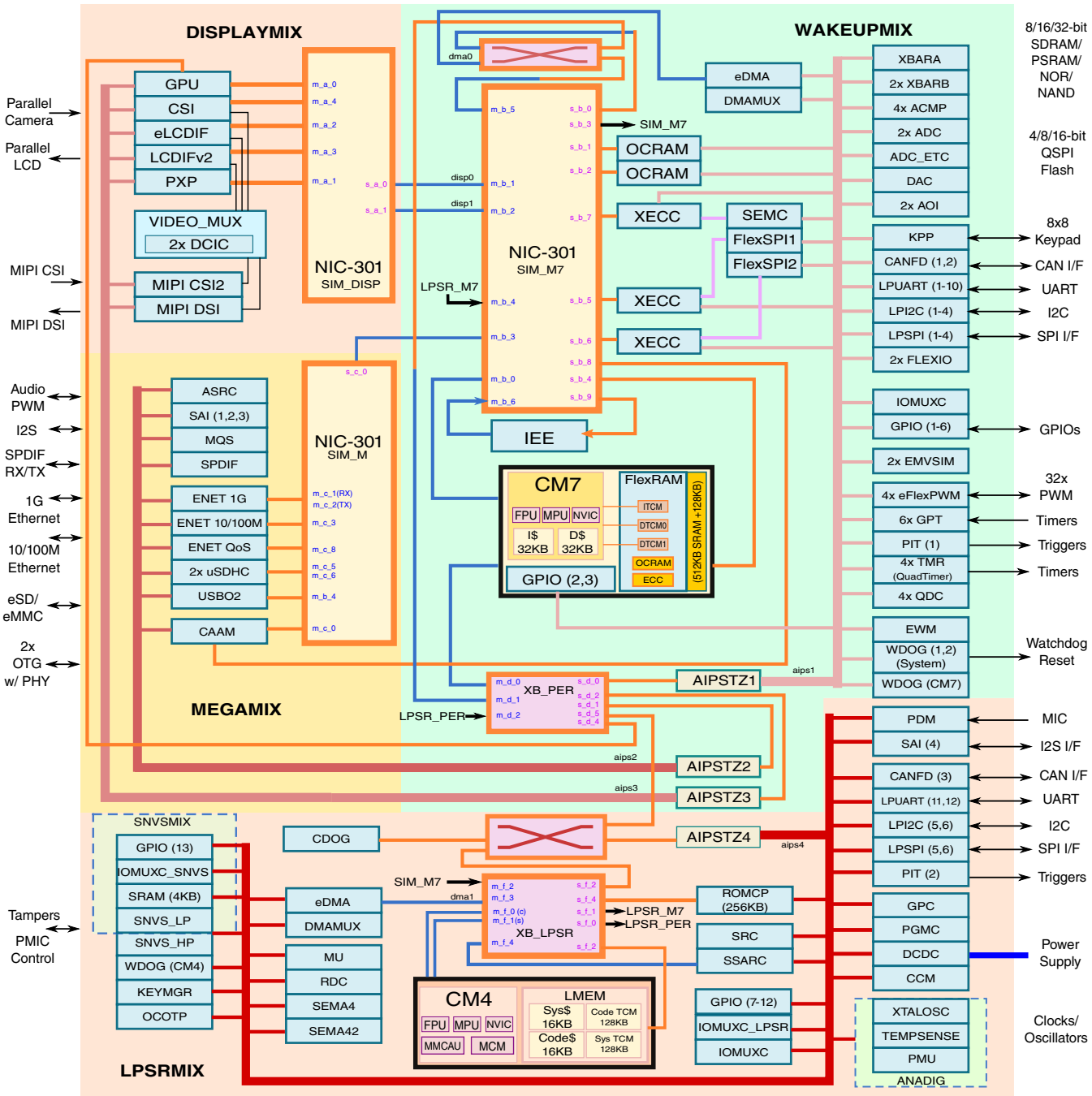


Figure 2-2. Bus Diagram

2.2 Features

The i.MX RT1170 processors are based on Arm®Cortex®-M7 Platform, and Arm®Cortex®-M4 which have the following features:

Single Arm Cortex-M7 with:
- 32KB L1 Instruction Cache
- 32KB L1 Data Cache
- Single-precision and double-precision FPU (Floating Point Unit)
- Integrated Memory Protection Unit (MPU), up to 16 individual protection regions
- Up to 512KB I-TCM and D-TCM in total
- ECC support for cache and TCM

Single Arm Cortex-M4 with:
- 16KB L1 Instruction Cache
- 16KB L1 Data Cache
- Single-precision FPU defined by ARMv7-M Architecture FPv4-SP
- ECC support for TCM
- Integrated MPU with 8 individual protection regions

On Chip Memory:
- Boot ROM (256KB)
- On-chip RAM (2MB in total)
  - FlexRAM - configurable 512KB RAM shared with M7 TCM and 256KB RAM shared with M4 TCM

NOTE

When using the M4 boot mode, FlexRAM will not be accessible until the M7 domain is released

  - OCRAM - dedicated 1.25MB RAM

External Memory Interfaces:
- 8/16/32-bit SDRAM, up to SDRAM-133/SDRAM-166/SDRAM-200
- 8/16-bit SLC NAND FLASH, with ECC handled in software
- SD/eMMC
- Single/Dual channel Quad SPI FLASH with XIP support
- Parallel NOR FLASH with XIP support
- SPI NOR/NAND FLASH
- HyperRAM/HyperFLASH
- Synchronization mode for all devices

Graphics:
- Generic 2D Graphics engine (PXP)
  - BitBlit

    - Flexible image composition options – alpha, chroma key
    - Image rotation (90, 180, 270 degree rotation)
    - Porter-Duff operation
    - Image resize
    - Color space conversion
    - Multiple pixel format support (RGB, YUV444, YUV422, YUV420, YUV400)
    - Standard 2D-DMA operation
- Vector Graphics Processing (GPU)
    - Real-time hardware curve tessellation of lines, quadratic and cubic Bezier curves
    - OpenVG 1.1 support
    - 16x Line Anti-aliasing
    - Vector Drawing

Display Interface:

- Parallel RGB LCD Interface (eLCDIF) that supports 8/16/24-bit I/F, up to WXGA with 60fps, and Index color with 256 entry x 24bit color LUT
- Parallel RGB LCD Interface Version 2 (LCDIFv2) that is enhanced based on the above version, supporting up to 8 layers of alpha blending
- MIPI Display Serial Interface (MIPI DSI) with 2-lane I/F and Smart LCD Display with 8080 interface through SEMC

Camera Sensor Interface:

- Parallel Camera - CMOS Sensor Interface (CSI) with 8/16/24-bit input
- MIPI Camera Sensor Interface (MIPI CSI2) with 2-lane I/F

Audio:

- SPDIF Input and Output
- 4x Synchronous Audio Interface (SAI) modules supporting I2S, AC97, TDM, and Codec/DSP interfaces
- MQS interface for medium quality audio via GPIO pads
- PDM microphone interface with 4 pairs inputs
- Asynchronous Sample Rate Converter (ASRC)

Connectivity:

- 2x USB 2.0 OTG controller with integrated PHY interfaces
- 2x Ultra Secure Digital Host Controller (uSDHC) interfaces
- 1x 10M/100M Ethernet controller with support for IEEE1588
- 1x Gigabit Ethernet controller with support for AVB
- 1x Gigabit Ethernet controller with Time Sensitive Networking (TSN) Capability
- 12x Universal asynchronous receiver/transmitter (UARTs) modules
- 6x I2C modules
- 6x SPI modules
- 3x CAN-FD modules

Timers:

- 6x General Programmable Timer (GPT)
- 2x Periodical Interrupt Timer (PIT)
- 4x Quad Timer (QTimer)
- 4x FlexPWM
- 4x Quadrature Encoder/Decoder
- 4x WatchDog modules (WDOG)

Analog:

- 2x Analog-Digital-Converters (ADC) (up to 20 channels)
- 4x Analog Comparators (ACMP)
- 1x 12-bit Digital-Analog-Converter (DAC)

Security:

- High Assurance Boot (HAB)
- Random Number Generator (RNG4)
- Secure Non-volatile Storage (SNVS)
  - Secure real-time clock (RTC)
  - Zero Master Key (ZMK)
  - Tamper Detection
- JTAG Controller (JTAGC)
- Cryptographic Acceleration and Assurance Module (CAAM)
- Inline Encryption Engine (IEE)
- On-the-Fly AES Decryption (OTFAD)
- Secure always-on RAM (4KB)
- Secure key management (KEYMGR) and protection
- Secure and trusted access control
- Code WatchDog Timer (CDOG)
- Manufacturing Protection

System Debug:

- Arm CoreSight debug and trace architecture
- Trace Port Interface Unit (TPIU) to support off-chip real-time trace
- Cross Triggering Interface (CTI)
- Support for 5-pin (JTAG) and SWD debug interfaces

Power Management:

- Full PMIC integration, including on-chip DCDC and LDO
- Temperature sensor with programmable trim points
- GPC hardware power management controller

2.3 Target Applications

This processor can be used in areas such as industrial HMI, IoT, high-end audio appliance, medium/low-end instrument cluster, Point-of-Sale(PoS), motor control and home appliances, etc.

2.4 Endianness Support

The chip supports only the Little Endian mode.

Chapter 3
Memory Maps

3.1 Memory system overview

This section introduces the memory architecture of the chip.

NOTE

Accessing the reserved memory regions can result in unpredictable behavior.

3.2 System memory map (CM7)

The table below shows the CM7 memory map:

Table 3-1. System memory map (CM7)

Start Address	End Address	Size	Description
E010_0000	FFFF_FFFF	511MB	Reserved
E000_0000	E00F_FFFF	1MB	PPB M7
C000_0000	DFFF_FFFF	512MB	SEMC3
A000_0000	BFFF_FFFF	512MB	SEMC2
9000_0000	9FFF_FFFF	256MB	SEMC1
8000_0000	8FFF_FFFF	256MB	SEMC0
7FC0_0000	7FFF_FFFF	4MB	FlexSPI2 RX FIFO
7F80_0000	7FBF_FFFF	4MB	FlexSPI2 TX FIFO
6000_0000	7F7F_FFFF	504MB	FlexSPI2/ FlexSPI2 ciphertext
4280_0000	5FFF_FFFF	472MB	Reserved
4210_0000	427F_FFFF	7MB	Reserved
4200_0000	420F_FFFF	1MB	AIPS M7 (Peripheral, Platform)
41A0_0000	41FF_FFFF	6MB	Reserved
4190_0000	419F_FFFF	1MB	CDOG (Peripheral, AHB)

Table continues on the next page...

Table 3-1. System memory map (CM7) (continued)

Start Address	End Address	Size	Description
4180_0000	418F_FFFF	1MB	GPU2D (Peripheral, AHB)
4170_0000	417F_FFFF	1MB	Reserved
4160_0000	416F_FFFF	1MB	Reserved
4150_0000	415F_FFFF	1MB	Reserved
4140_0000	414F_FFFF	1MB	SIM_M7 configuration port
4130_0000	413F_FFFF	1MB	Reserved
4120_0000	412F_FFFF	1MB	Reserved
4110_0000	411F_FFFF	1MB	SIM_M configuration port
4100_0000	410F_FFFF	1MB	SIM_DISP configuration port
40C0_0000	40FF_FFFF	4MB	AIPS-4
4080_0000	40BF_FFFF	4MB	AIPS-3
4040_0000	407F_FFFF	4MB	AIPS-2
4000_0000	403F_FFFF	4MB	AIPS-1
3000_0000	3FFF_FFFF	256MB	FlexSPI1/ FlexSPI1 cipher text
2FC0_0000	2FFF_FFFF	4MB	FlexSPI1 RX FIFO
2F80_0000	2FBF_FFFF	4MB	FlexSPI1 TX FIFO
2040_0000	2F7F_FFFF	244MB	Reserved
2036_0000	203F_FFFF	640KB	OCRAM M7 [1]
2035_0000	2035_FFFF	64KB	OCRAM2 ECC
2034_0000	2034_FFFF	64KB	OCRAM1 ECC
202C_0000	2033_FFFF	512KB	OCRAM2
2024_0000	202B_FFFF	512KB	OCRAM1
2020_0000	2023_FFFF	256KB	OCRAM M4 (LMEM 128KB SRAM_L + 128KB SRAM_U backdoor) [2, 3]
2010_0000	201F_FFFF	1MB	Reserved
2008_0000	200F_FFFF	512KB	Reserved
2000_0000	2007_FFFF	512KB	DTCM (FlexRAM)
1000_0000	1FFF_FFFF	256MB	Reserved
0800_0000	0FFF_FFFF	128MB	Reserved
0040_0000	07FF_FFFF	124MB	Reserved
0029_0000	003F_FFFF	1472KB	Reserved
0028_0000	0028_FFFF	64KB	CAAM Secure RAM
0024_0000	0027_FFFF	256KB	Reserved
0020_0000	0023_FFFF	256KB	Reserved
0010_0000	001F_FFFF	1MB	Reserved
0008_0000	000F_FFFF	512KB	Reserved
0000_0000	0007_FFFF	512KB	ITCM (FlexRAM)

1. When ECC is not enabled, the size is OCRAM allocated from FLEXRAM + 128KB OCRAM. When ECC is enabled, the size is OCRAM allocated from FLEXRAM.
2. This is the remapping address for CM4 TCM. CM7 can access CM4 TCM through this aliased region.

3. For dual core part, this memory space cannot be accessed by CM7 or other masters while CM4 is powered down. The result is unpredictable in this condition.

NOTE

When the ECC feature is enabled, the users cannot utilise the ECC memory region in RAM.

3.3 System memory map (CM4)

The table below shows the CM4 memory map:

Table 3-2. System memory map (CM4)

Start address	End address	Size	Description
E010_0000	FFFF_FFFF	511MB	Reserved
E000_0000	E00F_FFFF	1MB	PPB M4
C000_0000	DFFF_FFFF	512MB	SEMC3
A000_0000	BFFF_FFFF	512MB	SEMC2
9000_0000	9FFF_FFFF	256MB	SEMC1
8000_0000	8FFF_FFFF	256MB	SEMC0
7FC0_0000	7FFF_FFFF	4MB	FlexSPI2 RX FIFO
7F80_0000	7FBF_FFFF	4MB	FlexSPI2 TX FIFO
6000_0000	7F7F_FFFF	504MB	FlexSPI2/FlexSPI2 ciphertext
4280_0000	5FFF_FFFF	472MB	Reserved
4210_0000	427F_FFFF	7MB	Reserved
4200_0000	420F_FFFF	1MB	Reserved
41A0_0000	41FF_FFFF	6MB	Reserved
4190_0000	419F_FFFF	1MB	CDOG (Peripheral, AHB)
4180_0000	418F_FFFF	1MB	GPU2D (Peripheral, AHB)
4170_0000	417F_FFFF	1MB	Reserved
4160_0000	416F_FFFF	1MB	Reserved
4150_0000	415F_FFFF	1MB	Reserved
4140_0000	414F_FFFF	1MB	SIM_M7 configuration port
4130_0000	413F_FFFF	1MB	Reserved
4120_0000	412F_FFFF	1MB	Reserved
4110_0000	411F_FFFF	1MB	SIM_M configuration port
4100_0000	410F_FFFF	1MB	SIM_DISP configuration port
40C0_0000	40FF_FFFF	4MB	AIPS-4
4080_0000	40BF_FFFF	4MB	AIPS-3
4040_0000	407F_FFFF	4MB	AIPS-2
4000_0000	403F_FFFF	4MB	AIPS-1
3000_0000	3FFF_FFFF	256MB	FlexSPI1/ FlexSPI1 ciphertext

Table continues on the next page...

Table 3-2. System memory map (CM4) (continued)

Start address	End address	Size	Description
2FC0_0000	2FFF_FFFF	4MB	FlexSPI1 RX FIFO
2F80_0000	2FBF_FFFF	4MB	FlexSPI1 TX FIFO
2040_0000	2F7F_FFFF	244MB	Reserved
2036_0000	203F_FFFF	640KB	OCRAM M7 [1]
2035_0000	2035_FFFF	64KB	OCRAM2 ECC
2034_0000	2034_FFFF	64KB	OCRAM1 ECC
202C_0000	2033_FFFF	512KB	OCRAM2
2024_0000	202B_FFFF	512KB	OCRAM1
2020_0000	2023_FFFF	256KB	OCRAM M4 (LMEM 128KB RAM_L + 128KB RAM_U backdoor) [2]
2002_0000	201F_FFFF	1920KB	Reserved
2000_0000	2001_FFFF	128KB	System TCM (LMEM RAM_U)
1FFE_0000	1FFF_FFFF	128KB	Code TCM (LMEM RAM_L)
1800_0000	1FFD_FFFF	128MB	Reserved
0800_0000	17FF_FFFF	256MB	FlexSPI1 (alias)
0029_0000	07FF_FFFF	128448KB	Reserved
0028_0000	0028_FFFF	64KB	CAAM Secure RAM
0024_0000	0027_FFFF	256KB	Reserved
0020_0000	0023_FFFF	256KB	Reserved
0008_0000	001F_FFFF	1536KB	Reserved
0004_0000	0007_FFFF	256KB	Reserved
0000_0000	0003_FFFF	256KB	Reserved

1. When ECC is not enabled, the size is OCRAM allocated from FLEXRAM + 128KB OCRAM. When ECC is enabled, the size is OCRAM allocated from FLEXRAM.
2. This is the remapping address for CM4 TCM. CM7 can access CM4 TCM through this aliased region.

NOTE

When the ECC feature is enabled, the users cannot utilize the ECC memory region in RAM.

3.4 AIPS-1 Memory Map

The table below shows the AIPS-1 memory map (This memory map applies to CM4 and CM7):

Table 3-3. AIPS-1 memory map

Start Address	End Address	Size	NIC Port
4030_0000	403F_FFFF	1MB	Reserved

Table continues on the next page...

Table 3-3. AIPS-1 memory map (continued)

Start Address	End Address	Size	NIC Port
4020_0000	402F_FFFF	1MB	Reserved
401F_C000	401F_FFFF	16KB	Reserved
401F_8000	401F_BFFF	16KB	Reserved
401F_4000	401F_7FFF	16KB	Reserved
401F_0000	401F_3FFF	16KB	Reserved
401E_C000	401E_FFFF	16KB	Reserved
401E_8000	401E_BFFF	16KB	Reserved
401E_4000	401E_7FFF	16KB	Reserved
401E_0000	401E_3FFF	16KB	Reserved
401D_C000	401D_FFFF	16KB	Reserved
401D_8000	401D_BFFF	16KB	Reserved
401D_4000	401D_7FFF	16KB	Reserved
401D_0000	401D_3FFF	16KB	Reserved
401C_C000	401C_FFFF	16KB	Reserved
401C_8000	401C_BFFF	16KB	Reserved
401C_4000	401C_7FFF	16KB	Reserved
401C_0000	401C_3FFF	16KB	Reserved
401B_C000	401B_FFFF	16KB	Reserved
401B_8000	401B_BFFF	16KB	Reserved
401B_4000	401B_7FFF	16KB	Reserved
401B_0000	401B_3FFF	16KB	ACMP4
401A_C000	401A_FFFF	16KB	ACMP3
401A_8000	401A_BFFF	16KB	ACMP2
401A_4000	401A_7FFF	16KB	ACMP1
401A_0000	401A_3FFF	16KB	Reserved
4019_C000	4019_FFFF	16KB	Reserved
4019_8000	4019_BFFF	16KB	FLEXPWM4
4019_4000	4019_7FFF	16KB	FLEXPWM3
4019_0000	4019_3FFF	16KB	FLEXPWM2
4018_C000	4018_FFFF	16KB	FLEXPWM1
4018_8000	4018_BFFF	16KB	Reserved
4018_4000	4018_7FFF	16KB	Reserved
4018_0000	4018_3FFF	16KB	QDC4
4017_C000	4017_FFFF	16KB	QDC3
4017_8000	4017_BFFF	16KB	QDC2
4017_4000	4017_7FFF	16KB	QDC1
4017_0000	4017_3FFF	16KB	Reserved
4016_C000	4016_FFFF	16KB	Reserved
4016_8000	4016_BFFF	16KB	TMR4

Table continues on the next page...

Table 3-3. AIPS-1 memory map (continued)

Start Address	End Address	Size	NIC Port
4016_4000	4016_7FFF	16KB	TMR3
4016_0000	4016_3FFF	16KB	TMR2
4015_C000	4015_FFFF	16KB	TMR1
4015_8000	4015_BFFF	16KB	EMVSIM2
4015_4000	4015_7FFF	16KB	EMVSIM1
4015_0000	4015_3FFF	16KB	Reserved
4014_C000	4014_FFFF	16KB	Reserved
4014_8000	4014_BFFF	16KB	Reserved
4014_4000	4014_7FFF	16KB	Reserved
4014_0000	4014_3FFF	16KB	GPIO6
4013_C000	4013_FFFF	16KB	GPIO5
4013_8000	4013_BFFF	16KB	GPIO4
4013_4000	4013_7FFF	16KB	GPIO3
4013_0000	4013_3FFF	16KB	GPIO2
4012_C000	4012_FFFF	16KB	GPIO1
4012_8000	4012_BFFF	16KB	Reserved
4012_4000	4012_7FFF	16KB	Reserved
4012_0000	4012_3FFF	16KB	LPSPI4
4011_C000	4011_FFFF	16KB	LPSPI3
4011_8000	4011_BFFF	16KB	LPSPI2
4011_4000	4011_7FFF	16KB	LPSPI1
4011_0000	4011_3FFF	16KB	LPI2C4
4010_C000	4010_FFFF	16KB	LPI2C3
4010_8000	4010_BFFF	16KB	LPI2C2
4010_4000	4010_7FFF	16KB	LPI2C1
4010_0000	4010_3FFF	16KB	GPT6
400F_C000	400F_FFFF	16KB	GPT5
400F_8000	400F_BFFF	16KB	GPT4
400F_4000	400F_7FFF	16KB	GPT3
400F_0000	400F_3FFF	16KB	GPT2
400E_C000	400E_FFFF	16KB	GPT1
400E_8000	400E_BFFF	16KB	IOMUXC
400E_4000	400E_7FFF	16KB	IOMUXC_GPR
400E_0000	400E_3FFF	16KB	KPP
400D_C000	400D_FFFF	16KB	Reserved
400D_8000	400D_BFFF	16KB	PIT1
400D_4000	400D_7FFF	16KB	SEMC
400D_0000	400D_3FFF	16KB	FlexSPI2
400C_C000	400C_FFFF	16KB	FlexSPI1

Table continues on the next page...

Table 3-3. AIPS-1 memory map (continued)

Start Address	End Address	Size	NIC Port
400C_8000	400C_BFFF	16KB	CAN2
400C_4000	400C_7FFF	16KB	CAN1
400C_0000	400C_3FFF	16KB	Reserved
400B_C000	400B_FFFF	16KB	AOI2
400B_8000	400B_BFFF	16KB	AOI1
400B_4000	400B_7FFF	16KB	Reserved
400B_0000	400B_3FFF	16KB	FlexIO2
400A_C000	400A_FFFF	16KB	FlexIO1
400A_8000	400A_BFFF	16KB	Reserved
400A_4000	400A_7FFF	16KB	Reserved
400A_0000	400A_3FFF	16KB	LPUART10
4009_C000	4009_FFFF	16KB	LPUART9
4009_8000	4009_BFFF	16KB	LPUART8
4009_4000	4009_7FFF	16KB	LPUART7
4009_0000	4009_3FFF	16KB	LPUART6
4008_C000	4008_FFFF	16KB	LPUART5
4008_8000	4008_BFFF	16KB	LPUART4
4008_4000	4008_7FFF	16KB	LPUART3
4008_0000	4008_3FFF	16KB	LPUART2
4007_C000	4007_FFFF	16KB	LPUART1
4007_8000	4007_BFFF	16KB	Reserved
4007_4000	4007_7FFF	16KB	DMAMUX0
4007_0000	4007_3FFF	16KB	EDMA
4006_C000	4006_FFFF	16KB	IEE
4006_8000	4006_BFFF	16KB	IEE_APC
4006_4000	4006_7FFF	16KB	DAC
4006_0000	4006_3FFF	16KB	Reserved
4005_C000	4005_FFFF	16KB	Reserved
4005_8000	4005_BFFF	16KB	Reserved
4005_4000	4005_7FFF	16KB	LPADC2
4005_0000	4005_3FFF	16KB	LPADC1
4004_C000	4004_FFFF	16KB	Reserved
4004_8000	4004_BFFF	16KB	ADC_ETC
4004_4000	4004_7FFF	16KB	XBAR3
4004_0000	4004_3FFF	16KB	XBAR2
4003_C000	4003_FFFF	16KB	XBAR1
4003_8000	4003_BFFF	16KB	WDOG3
4003_4000	4003_7FFF	16KB	WDOG2
4003_0000	4003_3FFF	16KB	WDOG1

Table continues on the next page...

Table 3-3. AIPS-1 memory map (continued)

Start Address	End Address	Size	NIC Port
4002_C000	4002_FFFF	16KB	EWM
4002_8000	4002_BFFF	16KB	CM7 (FLEXRAM)
4002_4000	4002_7FFF	16KB	XECC_SEMC
4002_0000	4002_3FFF	16KB	XECC_FLEXSPI2
4001_C000	4001_FFFF	16KB	XECC_FLEXSPI1
4001_8000	4001_BFFF	16KB	MECC2
4001_4000	4001_7FFF	16KB	MECC1
4001_0000	4001_3FFF	16KB	Reserved
4000_C000	4000_FFFF	16KB	Reserved
4000_8000	4000_BFFF	16KB	Reserved
4000_4000	4000_7FFF	16KB	Reserved
4000_0000	4000_3FFF	16KB	Reserved

3.5 AIPS-2 Memory Map

The table below shows the AIPS-2 memory map (This memory map applies to CM4 and CM7):

Table 3-4. AIPS-2 memory map

Start Address	End Address	Size	NIC Port
4070_0000	407F_FFFF	1MB	Reserved
4060_0000	406F_FFFF	1MB	Reserved
4054_0000	405F_FFFF	768KB	Reserved
404D_0000	4053_FFFF	448KB	CAAM(Reserved)
404C_C000	404C_FFFF	16KB	CAAM(Debug)
404C_8000	404C_BFFF	16KB	CAAM(Debug)
404C_4000	404C_7FFF	16KB	CAAM(Debug)
404C_0000	404C_3FFF	16KB	CAAM(Debug)
404B_C000	404B_FFFF	16KB	CAAM(Reserved)
404B_8000	404B_BFFF	16KB	CAAM(Reserved)
404B_4000	404B_7FFF	16KB	CAAM(Reserved)
404B_0000	404B_3FFF	16KB	CAAM(Reserved)
404A_C000	404A_FFFF	16KB	CAAM(RTIC)
404A_8000	404A_BFFF	16KB	CAAM(RTIC)
404A_4000	404A_7FFF	16KB	CAAM(RTIC)
404A_0000	404A_3FFF	16KB	CAAM(RTIC)
4049_C000	4049_FFFF	16KB	CAAM(Reserved)

Table continues on the next page...

Table 3-4. AIPS-2 memory map (continued)

Start Address	End Address	Size	NIC Port
4049_8000	4049_BFFF	16KB	CAAM(Reserved)
4049_4000	4049_7FFF	16KB	CAAM(Reserved)
4049_0000	4049_3FFF	16KB	CAAM(Reserved)
4048_C000	4048_FFFF	16KB	CAAM(JR3)
4048_8000	4048_BFFF	16KB	CAAM(JR3)
4048_4000	4048_7FFF	16KB	CAAM(JR3)
4048_0000	4048_3FFF	16KB	CAAM(JR3)
4047_C000	4047_FFFF	16KB	CAAM(JR2)
4047_8000	4047_BFFF	16KB	CAAM(JR2)
4047_4000	4047_7FFF	16KB	CAAM(JR2)
4047_0000	4047_3FFF	16KB	CAAM(JR2)
4046_C000	4046_FFFF	16KB	CAAM(JR1)
4046_8000	4046_BFFF	16KB	CAAM(JR1)
4046_4000	4046_7FFF	16KB	CAAM(JR1)
4046_0000	4046_3FFF	16KB	CAAM(JR1)
4045_C000	4045_FFFF	16KB	CAAM(JR0)
4045_8000	4045_BFFF	16KB	CAAM(JR0)
4045_4000	4045_7FFF	16KB	CAAM(JR0)
4045_0000	4045_3FFF	16KB	CAAM(JR0)
4044_C000	4044_FFFF	16KB	CAAM(General)
4044_8000	4044_BFFF	16KB	CAAM(General)
4044_4000	4044_7FFF	16KB	CAAM(General)
4044_0000	4044_3FFF	16KB	CAAM(General)
4043_C000	4043_FFFF	16KB	ENET_QOS
4043_8000	4043_BFFF	16KB	USBPHY2
4043_4000	4043_7FFF	16KB	USBPHY1
4043_0000	4043_3FFF	16KB	USB OTG1
4042_C000	4042_FFFF	16KB	USB OTG2
4042_8000	4042_BFFF	16KB	Reserved
4042_4000	4042_7FFF	16KB	ENET
4042_0000	4042_3FFF	16KB	ENET_1G
4041_C000	4041_FFFF	16KB	USDHC2
4041_8000	4041_BFFF	16KB	USDHC1
4041_4000	4041_7FFF	16KB	ASRC
4041_0000	4041_3FFF	16KB	Reserved
4040_C000	4040_FFFF	16KB	SAI3
4040_8000	4040_BFFF	16KB	SAI2
4040_4000	4040_7FFF	16KB	SAI1
4040_0000	4040_3FFF	16KB	SPDIF

3.6 AIPS-3 Memory Map

The table below shows the AIPS-3 memory map (This memory map applies to CM4 and CM7):

Table 3-5. AIPS-3 memory map

Start Address	End Address	Size	NIC Port
40B0_0000	40BF_FFFF	1MB	Reserved
40A0_0000	40AF_FFFF	1MB	Reserved
4081_C000	4081_FFFF	16KB	Reserved
4081_8000	4081_BFFF	16KB	VIDEO_MUX
4081_4000	4081_7FFF	16KB	PXP
4081_0000	4081_3FFF	16KB	MIPI_CSI
4080_C000	4080_FFFF	16KB	MIPI_DSI
4080_8000	4080_BFFF	16KB	LCDIFv2
4080_4000	4080_7FFF	16KB	eLCDIF
4080_0000	4080_3FFF	16KB	CSI

3.7 AIPS-4 Memory Map

The table below shows the AIPS-4 memory map (This memory map applies to CM4 and CM7):

Table 3-6. AIPS-4 memory map

Start Address	End Address	Size	NIC Port
40F0_0000	40FF_FFFF	1MB	Reserved
40E0_0000	40EF_FFFF	1MB	Reserved
40CF_0000	40DF_FFFF	1088KB	Reserved
40CE_C000	40CE_FFFF	16KB	XRDC2 MGR M7
40CE_8000	40CE_BFFF	16KB	XRDC2 MGR M7
40CE_4000	40CE_7FFF	16KB	XRDC2 MGR M7
40CE_0000	40CE_3FFF	16KB	XRDC2 MGR M7
40CD_C000	40CD_FFFF	16KB	XRDC2 MGR M4
40CD_8000	40CD_BFFF	16KB	XRDC2 MGR M4
40CD_4000	40CD_7FFF	16KB	XRDC2 MGR M4
40CD_0000	40CD_3FFF	16KB	XRDC2 MGR M4
40CC_C000	40CC_FFFF	16KB	RDC_SEMAPHORE2

Table continues on the next page...

Table 3-6. AIPS-4 memory map (continued)

Start Address	End Address	Size	NIC Port
40CC_8000	40CC_BFFF	16KB	SEMA4
40CC_0000	40CC_7FFF	32KB	CCM
40CB_C000	40CB_FFFF	16KB	Reserved
40CB_8000	40CB_BFFF	16KB	SSARC_LP
40CB_4000	40CB_7FFF	16KB	SSARC_HP (SRAM)
40CB_0000	40CB_3FFF	16KB	PIT2
40CA_C000	40CA_FFFF	16KB	OCOTP
40CA_8000	40CA_BFFF	16KB	DCDC
40CA_4000	40CA_7FFF	16KB	ROMCP
40CA_0000	40CA_3FFF	16KB	GPIO13
40C9_C000	40C9_FFFF	16KB	SNVS_SRAM
40C9_8000	40C9_BFFF	16KB	IOMUXC_SNVS_GPR
40C9_4000	40C9_7FFF	16KB	IOMUXC_SNVS
40C9_0000	40C9_3FFF	16KB	SNVS
40C8_C000	40C8_FFFF	16KB	Reserved
40C8_8000	40C8_BFFF	16KB	PGMC
40C8_4000	40C8_7FFF	16KB	ANALOG/ANADIG
40C8_0000	40C8_3FFF	16KB	KEYMGR
40C7_C000	40C7_FFFF	16KB	Reserved
40C7_8000	40C7_BFFF	16KB	RDC
40C7_4000	40C7_7FFF	16KB	Reserved
40C7_0000	40C7_3FFF	16KB	GPIO12
40C6_C000	40C6_FFFF	16KB	GPIO11
40C6_8000	40C6_BFFF	16KB	GPIO10
40C6_4000	40C6_7FFF	16KB	GPIO9
40C6_0000	40C6_3FFF	16KB	GPIO8
40C5_C000	40C5_FFFF	16KB	GPIO7
40C5_8000	40C5_BFFF	16KB	Reserved
40C5_4000	40C5_7FFF	16KB	Reserved
40C5_0000	40C5_3FFF	16KB	Reserved
40C4_C000	40C4_FFFF	16KB	MU-B
40C4_8000	40C4_BFFF	16KB	MU-A
40C4_4000	40C4_7FFF	16KB	RDC_SEMAPHORE1
40C4_0000	40C4_3FFF	16KB	SAI4
40C3_C000	40C3_FFFF	16KB	CAN3
40C3_8000	40C3_BFFF	16KB	LPI2C6
40C3_4000	40C3_7FFF	16KB	LPI2C5
40C3_0000	40C3_3FFF	16KB	LPSPI6
40C2_C000	40C2_FFFF	16KB	LPSPI5

Table continues on the next page...

Table 3-6. AIPS-4 memory map (continued)

Start Address	End Address	Size	NIC Port
40C2_8000	40C2_BFFF	16KB	LPUART12
40C2_4000	40C2_7FFF	16KB	LPUART11
40C2_0000	40C2_3FFF	16KB	PDM
40C1_C000	40C1_FFFF	16KB	Reserved
40C1_8000	40C1_BFFF	16KB	DMAMUX1 (LPSR)
40C1_4000	40C1_7FFF	16KB	EDMA_LPSR
40C1_0000	40C1_3FFF	16KB	WDOG4
40C0_C000	40C0_FFFF	16KB	IOMUXC_LPSR_GPR
40C0_8000	40C0_BFFF	16KB	IOMUXC_LPSR
40C0_4000	40C0_7FFF	16KB	SRC
40C0_0000	40C0_3FFF	16KB	GPC

3.8 AIPS M7 Memory Map

The table below shows the AIPS M7 memory map:

Table 3-7. AIPS M7 memory map

Start Address	End Address	Region	Size	Description
4201_0000	420F_FFFF	AIPS M7	960KB	Reserved
4200_C000	4200_FFFF		16KB	GPIO_M7_3
4200_8000	4200_BFFF		16KB	GPIO_M7_2
4200_0000	4200_7FFF		32KB	Reserved

3.9 PPB M7 Memory Map

The table below shows the detailed PPB M7 memory map.

Table 3-8. PPB memory map for M7

Start Address	End Address	Region	Size	Allocation
E00F_F000	E00F_FFFF	M7 Platform	4KB	PPB ROM
E00F_E000	E00F_EFFF		4KB	Processeor ROM
E00F_D000	E00F_DFFF		4KB	SYS ROM
E008_3000	E00F_CFFF		488KB	PPB Reserved
E008_2000	E008_2FFF		4KB	PPB Reserved

Table continues on the next page...

Table 3-8. PPB memory map for M7 (continued)

Start Address	End Address	Region	Size	Allocation
E008_1000	E008_1FFF		4KB	PPB Reserved
E008_0000	E008_0FFF		4KB	MCM
E004_9000	E007_FFFF	CSSYS	220KB	PPB Reserved
E004_8000	E004_8FFF		4KB	CSSYS SWO
E004_7000	E004_7FFF		4KB	CSSYS TSGEN
E004_6000	E004_6FFF		4KB	CSSYS TPIU
E004_5000	E004_5FFF		4KB	CSSYS ATB Funnel
E004_4000	E004_4FFF		4KB	CSSYS CTI
E004_3000	E004_3FFF	M7 Platform	4KB	ATB Funnel
E004_2000	E004_2FFF		4KB	CTI
E004_1000	E004_1FFF		4KB	ETM
E004_0000	E004_0FFF		4KB	PPB RESERVED
E000_0000	E003_FFFF	M7	256KB	M7 Internal Use

3.10 PPB M4 Memory Map

The table below shows the detailed PPB M4 memory map.

Table 3-9. PPB memory map for M4

Start Address	End Address	Region	Size	Allocation
E00F_F000	E00F_FFFF	M4 Platform	4KB	PPB ROM
E00F_E000	E00F_EFFF		4KB	PPB Reserved
E00F_D000	E00F_DFFF		4KB	PPB Reserved
E008_3000	E00F_CFFF		488KB	PPB Reserved
E008_2000	E008_2FFF		4KB	AHB_LMEM
E008_1000	E008_1FFF		4KB	MMCAU
E008_0000	E008_0FFF		4KB	MCM
E004_9000	E007_FFFF	CSSYS	220KB	PPB Reserved
E004_8000	E004_8FFF		4KB	CSSYS SWO
E004_7000	E004_7FFF		4KB	CSSYS TSGEN
E004_6000	E004_6FFF		4KB	CSSYS TPIU
E004_5000	E004_5FFF		4KB	CSSYS ATB Funnel
E004_4000	E004_4FFF		4KB	CSSYS CTI
E004_3000	E004_3FFF	M4 Platform	4KB	ATB Funnel
E004_2000	E004_2FFF		4KB	CTI
E004_1000	E004_1FFF		4KB	ETM
E004_0000	E004_0FFF		4KB	PPB Reserved

Table continues on the next page...

Table 3-9. PPB memory map for M4 (continued)

Start Address	End Address	Region	Size	Allocation
E000_0000	E003_FFFF	M4	256KB	M4 Internal Use

Chapter 4
Interrupts, DMA Events, and XBAR Assignments

4.1 Overview

This section describes the Interrupt assignments, DMA events, and XBAR resource assignments.

4.2 CM7 interrupts

The Nested Vectored Interrupt Controller (NVIC) collects interrupt request sources and provides an interface to the Cortex-M7 core.

The table below describes the Cortex-M7 interrupt sources:

Table 4-1. CM7 domain interrupt summary

IRQ	Interrupt Source	LOGIC	Description
0	EDMA	OR	eDMA Channel 0 Transfer Complete
0		OR	eDMA Channel 16 Transfer Complete
1		OR	eDMA Channel 1 Transfer Complete
1		OR	eDMA Channel 17 Transfer Complete
2		OR	eDMA Channel 2 Transfer Complete
2		OR	eDMA Channel 18 Transfer Complete
3		OR	eDMA Channel 3 Transfer Complete
3		OR	eDMA Channel 19 Transfer Complete

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
4		OR	eDMA Channel 4 Transfer Complete
4		OR	eDMA Channel 20 Transfer Complete
5		OR	eDMA Channel 5 Transfer Complete
5		OR	eDMA Channel 21 Transfer Complete
6		OR	eDMA Channel 6 Transfer Complete
6		OR	eDMA Channel 22 Transfer Complete
7		OR	eDMA Channel 7 Transfer Complete
7		OR	eDMA Channel 23 Transfer Complete
8		OR	eDMA Channel 8 Transfer Complete
8		OR	eDMA Channel 24 Transfer Complete
9		OR	eDMA Channel 9 Transfer Complete
9		OR	eDMA Channel 25 Transfer Complete
10		OR	eDMA Channel 10 Transfer Complete
10		OR	eDMA Channel 26 Transfer Complete
11		OR	eDMA Channel 11 Transfer Complete
11		OR	eDMA Channel 27 Transfer Complete
12		OR	eDMA Channel 12 Transfer Complete
12		OR	eDMA Channel 28 Transfer Complete
13		OR	eDMA Channel 13 Transfer Complete
13		OR	eDMA Channel 29 Transfer Complete
14		OR	eDMA Channel 14 Transfer Complete
14		OR	eDMA Channel 30 Transfer Complete
15		OR	eDMA Channel 15 Transfer Complete

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
15		OR	eDMA Channel 31 Transfer Complete
16		-	Error Interrupt, Channels 0-15 / 16-31
17	CM7	-	CTI trigger output0 interrupt
18		-	CTI trigger output1 interrupt
19		-	Core Platform exception IRQ
20	LPUART1	OR	UART1 TX interrupt
20		OR	UART1 RX interrupt
20		OR	UART1 TX async interrupt
20		OR	UART1 RX async interrupt
21	LPUART2	OR	UART2 TX interrupt
21		OR	UART2 RX interrupt
21		OR	UART2 TX async interrupt
21		OR	UART2 RX async interrupt
22	LPUART3	OR	UART3 TX interrupt
22		OR	UART3 RX interrupt
22		OR	UART3 TX async interrupt
22		OR	UART3 RX async interrupt
23	LPUART4	OR	UART4 TX interrupt
23		OR	UART4 RX interrupt
23		OR	UART4 TX async interrupt
23		OR	UART4 RX async interrupt
24	LPUART5	OR	UART5 TX interrupt
24		OR	UART5 RX interrupt
24		OR	UART5 TX async interrupt
24		OR	UART5 RX async interrupt
25	LPUART6	OR	UART6 TX interrupt
25		OR	UART6 RX interrupt
25		OR	UART6 TX async interrupt
25		OR	UART6 RX async interrupt
26	LPUART7	OR	UART7 TX interrupt
26		OR	UART7 RX interrupt
26		OR	UART7 TX async interrupt
26		OR	UART7 RX async interrupt
27	LPUART8	OR	UART8 TX interrupt
27		OR	UART8 RX interrupt
27		OR	UART8 TX async interrupt
27		OR	UART8 RX async interrupt
28	LPUART9	OR	UART9 TX interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
28		OR	UART9 RX interrupt
28		OR	UART9 TX async interrupt
28		OR	UART9 RX async interrupt
29	LPUART10	OR	UART10 TX interrupt
29		OR	UART10 RX interrupt
29		OR	UART10 TX async interrupt
29		OR	UART10 RX async interrupt
30	LPUART11	OR	UART11 TX interrupt
30		OR	UART11 RX interrupt
30		OR	UART11 TX async interrupt
30		OR	UART11 RX async interrupt
31	LPUART12	OR	UART12 TX interrupt
31		OR	UART12 RX interrupt
31		OR	UART12 TX async interrupt
31		OR	UART12 RX async interrupt
32	LPI2C1	OR	LPI2C1 interrupt master async
32		OR	LPI2C1 interrupt slave async
32		OR	LPI2C1 interrupt master
32		OR	LPI2C1 interrupt slave
33	LPI2C2	OR	LPI2C2 interrupt master async
33		OR	LPI2C2 interrupt slave async
33		OR	LPI2C2 interrupt master
33		OR	LPI2C2 interrupt slave
34	LPI2C3	OR	LPI2C3 interrupt master async
34		OR	LPI2C3 interrupt slave async
34		OR	LPI2C3 interrupt master
34		OR	LPI2C3 interrupt slave
35	LPI2C4	OR	LPI2C4 interrupt master async
35		OR	LPI2C4 interrupt slave async
35		OR	LPI2C4 interrupt master
35		OR	LPI2C5 interrupt slave
36	LPI2C5	OR	LPI2C5 interrupt master async
36		OR	LPI2C5 interrupt slave async
36		OR	LPI2C5 interrupt master
36		OR	LPI2C5 interrupt slave

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
37	LPI2C6	OR	LPI2C6 interrupt master async
37		OR	LPI2C6 interrupt slave async
37		OR	LPI2C6 interrupt master
37		OR	LPI2C6 interrupt slave
38	LPSPI1	-	LPSPI1 interrupt request
39	LPSPI2	-	LPSPI2 interrupt request
40	LPSPI3	-	LPSPI3 interrupt request
41	LPSPI4	-	LPSPI4 interrupt request
42	LPSPI5	-	LPSPI5 interrupt request
43	LPSPI6	-	LPSPI6 interrupt request
44	CAN1	OR	Interrupt from Bus off
44		OR	Interrupt from CAN line error
44		OR	ORed interrupts from ipi_int_MB
44		OR	Rx warning interrupt
44		OR	Tx warning interrupt
44		OR	Interrupt from wake up
44		OR	Bus off done interrupt
44		OR	FD error interrupt
45		OR	Correctable error interrupt
45		OR	Non correctable error int host
45		OR	Non correctable error int internal
46	CAN2	OR	Interrupt from Bus off
46		OR	Interrupt from CAN line error
46		OR	ORed interrupts from ipi_int_MB
46		OR	Rx warning interrupt
46		OR	Tx warning interrupt
46		OR	Interrupt from wake up
46		OR	Bus off done interrupt
46		OR	FD error interrupt
47		OR	Correctable error interrupt
47		OR	Non correctable error int host
47		OR	Non correctable error int internal
48	CAN3	OR	Interrupt from Bus off
48		OR	Interrupt from CAN line error
48		OR	ORed interrupts from ipi_int_MB

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
48		OR	Rx warning interrupt
48		OR	Tx warning interrupt
48		OR	Interrupt from wake up
48		OR	Bus off done interrupt
48		OR	FD error interrupt
49		OR	Correctable error interrupt
49		OR	Non correctable error int host
49		OR	Non correctable error int internal
50	FlexRAM	-	FlexRAM address out of range or access hit IRQ
51	KPP	-	Keypad Interrupt
52	Reserved	-	N/A
53	GPR_IRQ	-	SW interrupt IOMUXC_GPR_GPR7[GINT]
54	eLCDIF	-	eLCDIF Interrupt
55	LCDIFv2	-	LCDIFv2 Interrupt
56	CSI	-	CSI interrupt
57	PXP	-	PXP interrupt
58	MIPI_CSI	-	MIPI CSI interrupt
59	MIPI_DSI	-	MIPI DSI interrupt
60	GPU2D	-	GPU2D interrupt
61	GPIO6	-	Combined interrupt indication for GPIO6 signal 0 through 15
62	Reserved	-	-
63	DAC	OR	DAC interrupt
63		OR	DAC async interrupt
64	KEYMGR	-	PUF interrupt
65	WDOG2	-	Watchdog2 Timer reset
66	SNVS_HP	-	SRTC Consolidated Interrupt. Non TZ.
67		-	SRTC Security Interrupt. TZ.
68	SNVS_LP	OR	ON-OFF button press shorter than 5 secs (pulse event)
68	SNVS_HP	OR	ON-OFF button press shorter than 5 secs (pulse event)
69	CAAM	-	CAAM interrupt queue for JQ0
70		-	CAAM interrupt queue for JQ1
71		-	CAAM interrupt queue for JQ2
72		-	CAAM interrupt queue for JQ3
73		-	CAAM interrupt for recoverable error

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
74		-	CAAM interrupt for RTIC
75	CDOG	-	CDOG interrupt
76	SAI1	OR	SAI RX interrupt
76		OR	SAI RX async interrupt
76		OR	SAI TX interrupt
76		OR	SAI TX async interrupt
77	SAI2	OR	SAI RX interrupt
77		OR	SAI RX async interrupt
77		OR	SAI TX interrupt
77		OR	SAI TX async interrupt
78	SAI3	-	SAI RX interrupt
79		-	SAI TX interrupt
80	SAI4	-	SAI RX interrupt
81		-	SAI TX interrupt
82	SPDIF	OR	SPDIF RX interrupt
82		OR	SPDIF TX interrupt
83	TMPSNS (Temperature Sensor)	-	Global Ored interrupt of all TMPSNS individual interrupts
84	TMPSNS (Temperature Sensor)	OR	TempSensor low
84		OR	TempSensor high
85		-	TempSensor panic
86	PMU	-	LPSR 1p8 brownout interrupt
87		-	LPSR 1p0 brownout interrupt
88	LPADC1	OR	ADC1 interrupt
88		OR	ADC1 async interrupt
89	LPADC2	OR	ADC2 interrupt
89		OR	ADC2 async interrupt
90	USBPHY1	-	USB1 wakeup interrupt when USB1 core is power down
91	USBPHY2	-	USB2 wakeup interrupt when USB2 core is power down
92	RDC	-	RDC reconfiguration complete interrupt
93	GPIO13	-	Combined interrupt indication for GPIO13 signal 0 through 31
94	Reserved	-	-
95	VIDEO_MUX	OR	DCIC1 Error Interrupt
95		OR	DCIC1 Function Interrupt
96		OR	DCIC2 Error Interrupt
96		OR	DCIC2 Function Interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
97	ASRC	-	ASRC interrupt
98	CM7	-	FlexRAM ECC fatal interrupt
99	CM7_GPIO2	OR	Combined interrupt indication for CM7_GPIO2 signal 0 through 31
99	CM7_GPIO3	OR	Combined interrupt indication for CM7_GPIO3 signal 0 through 31
100	GPIO1	-	Combined interrupt indication for GPIO1 signal 0 through 15
101		-	Combined interrupt indication for GPIO1 signal 16 through 31
102	GPIO2	-	Combined interrupt indication for GPIO2 signal 0 through 15
103		-	Combined interrupt indication for GPIO2 signal 16 through 31
104	GPIO3	-	Combined interrupt indication for GPIO3 signal 0 through 15
105		-	Combined interrupt indication for GPIO3 signal 16 through 31
106	GPIO4	-	Combined interrupt indication for GPIO4 signal 0 through 15
107		-	Combined interrupt indication for GPIO4 signal 16 through 31
108	GPIO5	-	Combined interrupt indication for GPIO5 signal 0 through 15
109	Reserved	-	-
110	FLEXIO1	OR	IPI compare interrupt
110		OR	IPI async compare interrupt
111	FLEXIO2	OR	IPI compare interrupt
111		OR	IPI async compare interrupt
112	WDOG1	-	Watchdog1 Timer reset
113	WDOG3	OR	Watchdog3 Timer reset
113		OR	Watchdog Timer Async reset
114	EWM	-	EWM interrupt
115	OCOTP	-	Read FUSE error interrupt
116		-	Read FUSE done interrupt
117	GPC	-	GPC interrupt
118	MU	-	ORed of all (tx,rx and gp) interrupt
119	GPT1	-	All GPT1 interrupts

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
120	GPT2	-	All GPT2 interrupts
121	GPT3	-	All GPT3 interrupts
122	GPT4	-	All GPT4 interrupts
123	GPT5	-	All GPT5 interrupts
124	GPT6	-	All GPT6 interrupts
125	FLEXPWM1	OR	capture 0 interrupt
125		OR	compare 0 interrupt
125		OR	reload 0 interrupt
126		OR	capture 1 interrupt
126		OR	compare 1 interrupt
126		OR	reload 1 interrupt
127		OR	capture 2 interrupt
127		OR	compare 2 interrupt
127		OR	reload 2 interrupt
128		OR	capture 3 interrupt
128		OR	compare 3 interrupt
128		OR	reload 3 interrupt
129		OR	fault interrupt
129		OR	reload error interrupt
130	FLEXSPI1	-	FlexSPI1 interrupt
131	FLEXSPI2	-	FlexSPI2 interrupt
132	SEMC	-	SEMC interrupt
133	USDHC1	-	uSDHC1 Enhanced SDHC Interrupt Request
134	USDHC2	-	uSDHC2 Enhanced SDHC Interrupt Request
135	USB OTG2	-	USB OTG2 interrupt
136	USB OTG1	-	USB OTG1 interrupt
137	ENET	OR	MAC 0 Periodic Timer Overflow
137		OR	MAC 0 Time Stamp Available
137		OR	MAC 0 Payload Receive Error
137		OR	MAC 0 Transmit FIFO Underrun
137		OR	MAC 0 Collision Retry Limit
137		OR	MAC 0 Late Collision
137		OR	MAC 0 Ethernet Bus Error
137		OR	MAC 0 MII Data Transfer Done
137		OR	MAC 0 Receive Buffer Done
137		OR	MAC 0 Receive Frame Done

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
137		OR	MAC 0 Transmit Buffer Done
137		OR	MAC 0 Transmit Frame Done
137		OR	MAC 0 Graceful Stop
137		OR	MAC 0 Babbling Transmit Error
137		OR	MAC 0 Babbling Receive Error
137		OR	MAC 0 Wakeup Request (sync)
137		OR	MAC 0 Wakeup Request (async)
138		-	MAC 0 1588 Timer Interrupt – synchronous
139	ENET_1G	OR	MAC 0 Receive Buffer 1 Done
139		OR	MAC 0 Receive Frame 1 Done
139		OR	MAC 0 Transmit Buffer 1 Done
139		OR	MAC 0 Transmit Frame 1 Done
140		OR	MAC 0 Receive Buffer 2 Done
140		OR	MAC 0 Receive Frame 2 Done
140		OR	MAC 0 Transmit Buffer 2 Done
140		OR	MAC 0 Transmit Frame 2 Done
141		OR	MAC 0 Periodic Timer Overflow
141		OR	MAC 0 Time Stamp Available
141		OR	MAC 0 Payload Receive Error
141		OR	MAC 0 Transmit FIFO Underrun
141		OR	MAC 0 Collision Retry Limit
141		OR	MAC 0 Late Collision
141		OR	MAC 0 Ethernet Bus Error
141		OR	MAC 0 MII Data Transfer Done
141		OR	MAC 0 Receive Buffer Done
141		OR	MAC 0 Receive Frame Done
141		OR	MAC 0 Transmit Buffer Done
141		OR	MAC 0 Transmit Frame Done
141		OR	MAC 0 Graceful Stop

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
141		OR	MAC 0 Babbling Transmit Error
141		OR	MAC 0 Babbling Receive Error
141		OR	MAC 0 Receive Flush Frame0
141		OR	MAC 0 Receive Flush Frame1
141		OR	MAC 0 Receive Flush Frame2
141		OR	MAC 0 Wakeup Request (sync)
141		OR	MAC 0 Wakeup Request (async)
141		OR	MAC 0 Babbling Receive Error
141		OR	MAC 0 Wakeup Request (err)
142		-	MAC 0 1588 Timer Interrupt – synchronous
143	XBAR1	OR	XBAR IRQ (XBAR_OUT0)
143		OR	XBAR IRQ (XBAR_OUT1)
144		OR	XBAR IRQ (XBAR_OUT2)
144		OR	XBAR IRQ (XBAR_OUT3)
145	ADC_ETC	-	ADC_ETC done0 interrupt
146		-	ADC_ETC done1 interrupt
147		-	ADC_ETC done2 interrupt
148		-	ADC_ETC done3 interrupt
149		-	ADC_ETC err interrupt
150	Reserved	-	N/A
150	Reserved	-	N/A
150	Reserved	-	N/A
151	Reserved	-	N/A
151	Reserved	-	N/A
151	Reserved	-	N/A
152	Reserved	-	N/A
152	Reserved	-	N/A
152	Reserved	-	N/A
153	Reserved	-	N/A
153	Reserved	-	N/A
153	Reserved	-	N/A
154	Reserved	-	N/A
154	Reserved	-	N/A
155	PIT1	OR	PIT1 channel 0 interrupt
155		OR	PIT1 channel 1 interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
155		OR	PIT1 channel 2 interrupt
155		OR	PIT1 channel 3 interrupt
156	PIT2	OR	PIT2 channel 0 interrupt
156		OR	PIT2 channel 1 interrupt
156		OR	PIT2 channel 2 interrupt
156		OR	PIT2 channel 3 interrupt
157	ACMP1	OR	ACMP1 interrupt
157		OR	ACMP1 async interrupt for STOP mode wake up
158	ACMP2	OR	ACMP2 interrupt
158		OR	ACMP2 async interrupt for STOP mode wake up
159	ACMP3	OR	ACMP3 interrupt
159		OR	ACMP3 async interrupt for STOP mode wake up
160	ACMP4	OR	ACMP4 interrupt
160		OR	ACMP4 async interrupt for STOP mode wake up
161	Reserved	-	N/A
162	Reserved	-	N/A
163	Reserved	-	N/A
164	Reserved	-	N/A
165	QDC1	OR	Index marker interrupt
165		OR	Home marker interrupt
165		OR	Watchdog timeout interrupt
165		OR	Compare interrupt
165		OR	Simultaneous input switching interrupt
166	QDC2	OR	Index marker interrupt
166		OR	Home marker interrupt
166		OR	Watchdog timeout interrupt
166		OR	Compare interrupt
166		OR	Simultaneous input switching interrupt
167	QDC3	OR	Index marker interrupt
167		OR	Home marker interrupt
167		OR	Watchdog timeout interrupt
167		OR	Compare interrupt
167		OR	Simultaneous input switching interrupt
168	QDC4	OR	Index marker interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
168		OR	Home marker interrupt
168		OR	Watchdog timeout interrupt
168		OR	Compare interrupt
168		OR	Simultaneous input switching interrupt
169	Reserved	-	N/A
170	Reserved	-	N/A
171	TMR1	OR	Interrupt request for timer #0
171		OR	Interrupt request for timer #1
171		OR	Interrupt request for timer #2
171		OR	Interrupt request for timer #3
172	TMR2	OR	Interrupt request for timer #0
172		OR	Interrupt request for timer #1
172		OR	Interrupt request for timer #2
172		OR	Interrupt request for timer #3
173	TMR3	OR	Interrupt request for timer #0
173		OR	Interrupt request for timer #1
173		OR	Interrupt request for timer #2
173		OR	Interrupt request for timer #3
174	TMR4	OR	Interrupt request for timer #0
174		OR	Interrupt request for timer #1
174		OR	Interrupt request for timer #2
174		OR	Interrupt request for timer #3
175	SEMA4	-	SEMA4 CP0 interrupt
176		-	SEMA CP1 interrupt
177	FLEXPWM2	OR	capture 0 interrupt
177		OR	compare 0 interrupt
177		OR	reload 0 interrupt
178		OR	capture 1 interrupt
178		OR	compare 1 interrupt
178		OR	reload 1 interrupt
179		OR	capture 2 interrupt
179		OR	compare 2 interrupt
179		OR	reload 2 interrupt
180		OR	capture 3 interrupt
180		OR	compare 3 interrupt
180		OR	reload 3 interrupt
181		OR	fault interrupt
181		OR	reload error interrupt
182	FLEXPWM3	OR	capture 0 interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
182		OR	compare 0 interrupt
182		OR	reload 0 interrupt
183		OR	capture 1 interrupt
183		OR	compare 1 interrupt
183		OR	reload 1 interrupt
184		OR	capture 2 interrupt
184		OR	compare 2 interrupt
184		OR	reload 2 interrupt
185		OR	capture 3 interrupt
185		OR	compare 3 interrupt
185		OR	reload 3 interrupt
186		OR	fault interrupt
186		OR	reload error interrupt
187	FLEXPWM4	OR	capture 0 interrupt
187		OR	compare 0 interrupt
187		OR	reload 0 interrupt
188		OR	capture 1 interrupt
188		OR	compare 1 interrupt
188		OR	reload 1 interrupt
189		OR	capture 2 interrupt
189		OR	compare 2 interrupt
189		OR	reload 2 interrupt
190		OR	capture 3 interrupt
190		OR	compare 3 interrupt
190		OR	reload 3 interrupt
191		OR	fault interrupt
191		OR	reload error interrupt
192	Reserved	-	N/A
193	Reserved	-	N/A
194	Reserved	-	N/A
195	Reserved	-	N/A
196	Reserved	-	N/A
197	Reserved	-	N/A
198	Reserved	-	N/A
199	Reserved	-	N/A
200	PDM	OR	HWVAD event interrupt
200		OR	HWVAD event async interrupt
201		OR	HWVAD error interrupt
201		OR	HWVAD error async interrupt

Table continues on the next page...

Table 4-1. CM7 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Description
202		OR	filter interrupt
202		OR	filter async interrupt
203		OR	error interrupt
203		OR	error async interrupt
204	EMVSIM1	OR	EMVSIM1 interrupt (data_irq)
204		OR	EMVSIM1 interrupt
205	EMVSIM2	OR	EMVSIM2 interrupt (data_irq)
205		OR	EMVSIM2 interrupt
206	MECC1	-	MECC1 interrupt
207	MECC1	-	MECC1 fatal interrupt
208	MECC2	-	MECC2 interrupt
209	MECC2	-	MECC2 fatal interrupt
210	XECC_FLEXSPI1	-	XECC_FLEXSPI1 interrupt
211	XECC_FLEXSPI1	-	XECC_FLEXSPI1 fatal interrupt
212	XECC_FLEXSPI2	-	XECC_FLEXSPI2 interrupt
213	XECC_FLEXSPI2	-	XECC_FLEXSPI2 fatal interrupt
214	XECC_SEMC	-	XECC_SEMC interrupt
215	XECC_SEMC	-	XECC_SEMC fatal interrupt
216	ENET_QOS	OR	LPI Rx exit interrupt output
216		OR	Interrupt signal to host system
216		OR	Channel 0 Transmit Interrupt signal to host system
216		OR	Channel 1 Transmit Interrupt signal to host system
216		OR	Channel 2 Transmit Interrupt signal to host system
216		OR	Channel 3 Transmit Interrupt signal to host system
216		OR	Channel 4 Transmit Interrupt signal to host system
216		OR	Channel 0 Receive Interrupt signal to host system
216		OR	Channel 1 Receive Interrupt signal to host system
216		OR	Channel 2 Receive Interrupt signal to host system
216		OR	Channel 3 Receive Interrupt signal to host system
216		OR	Channel 4 Receive Interrupt signal to host system
217		-	Interrupt from PMT

4.3 CM4 interrupts

The Nested Vectored Interrupt Controller (NVIC) collects up to 218 interrupt requests from all i.MX sources and provides an interface to the Cortex M4 core.

The table below describes the M4 interrupt sources:

Table 4-2. CM4 domain interrupt summary

IRQ	Interrupt Source	LOGIC	Interrupt Description
0	EDMA_LPSR	OR	eDMA_LPSR Channel 0 Transfer Complete
0		OR	eDMA_LPSR Channel 16 Transfer Complete
1		OR	eDMA_LPSR Channel 1 Transfer Complete
1		OR	eDMA_LPSR Channel 17 Transfer Complete
2		OR	eDMA_LPSR Channel 2 Transfer Complete
2		OR	eDMA_LPSR Channel 18 Transfer Complete
3		OR	eDMA_LPSR Channel 3 Transfer Complete
3		OR	eDMA_LPSR Channel 19 Transfer Complete
4		OR	eDMA_LPSR Channel 4 Transfer Complete
4		OR	eDMA_LPSR Channel 20 Transfer Complete
5		OR	eDMA_LPSR Channel 5 Transfer Complete
5		OR	eDMA_LPSR Channel 21 Transfer Complete
6		OR	eDMA_LPSR Channel 6 Transfer Complete
6		OR	eDMA_LPSR Channel 22 Transfer Complete
7		OR	eDMA_LPSR Channel 7 Transfer Complete
7		OR	eDMA_LPSR Channel 23 Transfer Complete
8		OR	eDMA_LPSR Channel 8 Transfer Complete
8		OR	eDMA_LPSR Channel 24 Transfer Complete

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
9		OR	eDMA_LPSR Channel 9 Transfer Complete
9		OR	eDMA_LPSR Channel 25 Transfer Complete
10		OR	eDMA_LPSR Channel 10 Transfer Complete
10		OR	eDMA_LPSR Channel 26 Transfer Complete
11		OR	eDMA_LPSR Channel 11 Transfer Complete
11		OR	eDMA_LPSR Channel 27 Transfer Complete
12		OR	eDMA_LPSR Channel 12 Transfer Complete
12		OR	eDMA_LPSR Channel 28 Transfer Complete
13		OR	eDMA_LPSR Channel 13 Transfer Complete
13		OR	eDMA_LPSR Channel 29 Transfer Complete
14		OR	eDMA_LPSR Channel 14 Transfer Complete
14		OR	eDMA_LPSR Channel 30 Transfer Complete
15		OR	eDMA_LPSR Channel 15 Transfer Complete
15		OR	eDMA_LPSR Channel 31 Transfer Complete
16		-	Error Interrupt, Channels 0-15 / 16-31
17	Reserved	-	N/A
18	Reserved	-	N/A
19	CM4	-	Correctable TCM ECC error(Uncorrectable error goes to NMI)
20	LPUART1	OR	UART1 TX interrupt
20		OR	UART1 RX interrupt
20		OR	UART1 TX async interrupt
20		OR	UART1 RX async interrupt
21	LPUART2	OR	UART2 TX interrupt
21		OR	UART2 RX interrupt
21		OR	UART2 TX async interrupt
21		OR	UART2 RX async interrupt
22	LPUART3	OR	UART3 TX interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
22		OR	UART3 RX interrupt
22		OR	UART3 TX async interrupt
22		OR	UART3 RX async interrupt
23	LPUART4	OR	UART4 TX interrupt
23		OR	UART4 RX interrupt
23		OR	UART4 TX async interrupt
23		OR	UART4 RX async interrupt
24	LPUART5	OR	UART5 TX interrupt
24		OR	UART5 RX interrupt
24		OR	UART5 TX async interrupt
24		OR	UART5 RX async interrupt
25	LPUART6	OR	UART6 TX interrupt
25		OR	UART6 RX interrupt
25		OR	UART6 TX async interrupt
25		OR	UART6 RX async interrupt
26	LPUART7	OR	UART7 TX interrupt
26		OR	UART7 RX interrupt
26		OR	UART7 TX async interrupt
26		OR	UART7 RX async interrupt
27	LPUART8	OR	UART8 TX interrupt
27		OR	UART8 RX interrupt
27		OR	UART8 TX async interrupt
27		OR	UART8 RX async interrupt
28	LPUART9	OR	UART9 TX interrupt
28		OR	UART9 RX interrupt
28		OR	UART9 TX async interrupt
28		OR	UART9 RX async interrupt
29	LPUART10	OR	UART10 TX interrupt
29		OR	UART10 RX interrupt
29		OR	UART10 TX async interrupt
29		OR	UART10 RX async interrupt
30	LPUART11	OR	UART11 TX interrupt
30		OR	UART11 RX interrupt
30		OR	UART11 TX async interrupt
30		OR	UART11 RX async interrupt
31	LPUART12	OR	UART12 TX interrupt
31		OR	UART12 RX interrupt
31		OR	UART12 TX async interrupt
31		OR	UART12 RX async interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
32	LPI2C1	OR	LPI2C1 interrupt master async
32		OR	LPI2C1 interrupt slave async
32		OR	LPI2C1 interrupt master
32		OR	LPI2C1 interrupt slave
33	LPI2C2	OR	LPI2C2 interrupt master async
33		OR	LPI2C2 interrupt slave async
33		OR	LPI2C2 interrupt master
33		OR	LPI2C2 interrupt slave
34	LPI2C3	OR	LPI2C3 interrupt master async
34		OR	LPI2C3 interrupt slave async
34		OR	LPI2C3 interrupt master
34		OR	LPI2C3 interrupt slave
35	LPI2C4	OR	LPI2C4 interrupt master async
35		OR	LPI2C4 interrupt slave async
35		OR	LPI2C4 interrupt master
35		OR	LPI2C5 interrupt slave
36	LPI2C5	OR	LPI2C5 interrupt master async
36		OR	LPI2C5 interrupt slave async
36		OR	LPI2C5 interrupt master
36		OR	LPI2C5 interrupt slave
37	LPI2C6	OR	LPI2C6 interrupt master async
37		OR	LPI2C6 interrupt slave async
37		OR	LPI2C6 interrupt master
37		OR	LPI2C6 interrupt slave
38	LPSPI1	-	LPSPI1 interrupt request
39	LPSPI2	-	LPSPI2 interrupt request
40	LPSPI3	-	LPSPI3 interrupt request
41	LPSPI4	-	LPSPI4 interrupt request
42	LPSPI5	-	LPSPI5 interrupt request
43	LPSPI6	-	LPSPI6 interrupt request
44	CAN1	OR	Interrupt from Bus off
44		OR	Interrupt from CAN line error
44		OR	Ored interrupts from ipi_int_MB
44		OR	Rx warning interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
44		OR	Tx warning interrupt
44		OR	Interrupt from wake up
44		OR	Bus off done interrupt
44		OR	FD error interrupt
45		OR	Correctable error interrupt
45		OR	Non correctable error int host
45		OR	Non correctable error int internal
46	CAN2	OR	Interrupt from Bus off
46		OR	Interrupt from CAN line error
46		OR	Ored interrupts from ipi_int_MB
46		OR	Rx warning interrupt
46		OR	Tx warning interrupt
46		OR	Interrupt from wake up
46		OR	Bus off done interrupt
46		OR	FD error interrupt
47		OR	Correctable error interrupt
47		OR	Non correctable error int host
47		OR	Non correctable error int internal
48	CAN3	OR	Interrupt from Bus off
48		OR	Interrupt from CAN line error
48		OR	Ored interrupts from ipi_int_MB
48		OR	Rx warning interrupt
48		OR	Tx warning interrupt
48		OR	Interrupt from wake up
48		OR	Bus off done interrupt
48		OR	FD error interrupt
49		OR	Correctable error interrupt
49		OR	Non correctable error int host
49		OR	Non correctable error int internal
50	Reserved	-	N/A
51	KPP	-	Keypad Interrupt
52	Reserved	-	N/A
53	GPR_IRQ	-	SW interrupt IOMUXC_GPR_GPR7[GINT]
54	eLCDIF	-	eLCDIF Interrupt
55	LCDIFv2	-	LCDIFv2 Interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
56	CSI	-	CSI interrupt
57	PXP	-	PXP interrupt
58	MIPI_CSI	-	MIPI CSI interrupt
59	MIPI_DSI	-	MIPI DSI interrupt
60	GPU2D	-	GPU2D interrupt
61	GPIO12	-	Combined interrupt indication for GPIO12 signal 0 through 15
62	Reserved	-	-
63	DAC	OR	DAC interrupt
63		OR	DAC async interrupt
64	KEYMGR	-	PUF interrupt
65	WDOG2	-	Watchdog2 Timer reset
66	SNVS_HP	-	SRTC Consolidated Interrupt. Non TZ.
67		-	SRTC Security Interrupt. TZ.
68	SNVS_LP	OR	ON-OFF button press (source SPOF)
68	SNVS_HP	OR	ON-OFF button press (source BI)
69	CAAM	-	CAAM interrupt queue for JQ0
70		-	CAAM interrupt queue for JQ1
71		-	CAAM interrupt queue for JQ2
72		-	CAAM interrupt queue for JQ3
73		-	CAAM interrupt for recoverable error
74		-	CAAM interrupt for RTIC
75	CDOG	-	CDOG interrupt
76	SAI1	OR	SAI RX interrupt
76		OR	SAI RX async interrupt
76		OR	SAI TX interrupt
76		OR	SAI TX async interrupt
77	SAI2	OR	SAI RX interrupt
77		OR	SAI RX async interrupt
77		OR	SAI TX interrupt
77		OR	SAI TX async interrupt
78	SAI3	-	SAI RX interrupt
79		-	SAI TX interrupt
80	SAI4	-	SAI RX interrupt
81		-	SAI TX interrupt
82	SPDIF	OR	SPDIF Rx interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
82		OR	SPDIF Tx interrupt
83	TMPSNS (Temperature Sensor)	-	Global ORed interrupt of all Tempsense individual interrupts
84	TMPSNS (Temperature Sensor)	OR	TempSensor low
84		OR	TempSensor high
85		-	TempSensor panic
86	PMU	-	LPSR 1p8 brownout interrupt
87		-	LPSR 1p0 brownout interrupt
88	LPADC1	OR	ADC1 interrupt
88		OR	ADC1 async interrupt
89	LPADC2	OR	ADC2 interrupt
89		OR	ADC2 async interrupt
90	USBPHY1	-	USB1 wakeup interrupt when USB1 core is powered down
91	USBPHY2	-	USB2 wakeup interrupt when USB2 core is powered down
92	RDC	-	RDC reconfiguration complete interrupt
93	GPIO13	-	Combined interrupt indication for GPIO13 signal 0 through 31
94	Reserved	-	-
95	VIDEO_MUX	OR	DCIC1 Error Interrupt
95		OR	DCIC1 Function Interrupt
96		OR	DCIC2 Error Interrupt
96		OR	DCIC2 Function Interrupt
97	ASRC	-	ASRC interrupt
98	CM7_IMXRT	-	FlexRAM ECC fatal interrupt
99	GPIO7	OR	Combined interrupt indication for GPIO7 signal 0 through 31
99	GPIO8	OR	Combined interrupt indication for GPIO8 signal 0 through 31
99	GPIO9	OR	Combined interrupt indication for GPIO9 signal 0 through 31
99	GPIO10	OR	Combined interrupt indication for GPIO10 signal 0 through 31
99	GPIO11	OR	Combined interrupt indication for GPIO11 signal 0 through 31
100	GPIO1	-	Combined interrupt indication for GPIO1 signal 0 through 15

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
101		-	Combined interrupt indication for GPIO1 signal 16 through 31
102	GPIO2	-	Combined interrupt indication for GPIO2 signal 0 through 15
103		-	Combined interrupt indication for GPIO2 signal 16 through 31
104	GPIO3	-	Combined interrupt indication for GPIO3 signal 0 through 15
105		-	Combined interrupt indication for GPIO3 signal 16 through 31
106	GPIO4	-	Combined interrupt indication for GPIO4 signal 0 through 15
107		-	Combined interrupt indication for GPIO4 signal 16 through 31
108	GPIO5	-	Combined interrupt indication for GPIO5 signal 0 through 15
109		-	Combined interrupt indication for GPIO5 signal 16 through 31
110	FLEXIO1	OR	IPI compare interrupt
110		OR	IPI async compare interrupt
111	FLEXIO2	OR	IPI compare interrupt
111		OR	IPI async compare interrupt
112	WDOG1	-	Watchdog1 Timer reset
113	WDOG4	OR	Watchdog4 Timer reset
113		OR	Watchdog4 Timer Async reset
114	EWM	-	EWM IRQ
115	OCOTP	-	Read FUSE error interrupt
116		-	Read FUSE done interrupt
117	GPC	-	GPC interrupt
118	MU	-	ORed of all (tx,rx and gp) interrupt
119	GPT1	-	All GPT1 interrupts
120	GPT2	-	All GPT2 interrupts
121	GPT3	-	All GPT3 interrupts
122	GPT4	-	All GPT4 interrupts
123	GPT5	-	All GPT5 interrupts
124	GPT6	-	All GPT6 interrupts
125	FLEXPWM1	OR	capture 0 interrupt
125		OR	compare 0 interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
125		OR	reload 0 interrupt
126		OR	capture 1 interrupt
126		OR	compare 1 interrupt
126		OR	reload 1 interrupt
127		OR	capture 2 interrupt
127		OR	compare 2 interrupt
127		OR	reload 2 interrupt
128		OR	capture 3 interrupt
128		OR	compare 3 interrupt
128		OR	reload 3 interrupt
129		OR	fault interrupt
129		OR	reload error interrupt
130	FLEXSPI1	-	FlexSPI1 interrupt
131	FLEXSPI2	-	FlexSPI2 interrupt
132	SEMC	-	SEMC interrupt
133	USDHC1	-	uSDHC1 Enhanced SDHC Interrupt Request
134	USDHC2	-	uSDHC2 Enhanced SDHC Interrupt Request
135	USB OTG2	-	USB OTG2 interrupt
136	USB OTG1	-	USB OTG1 interrupt
137	ENET	OR	MAC 0 Periodic Timer Overflow
137		OR	MAC 0 Time Stamp Available
137		OR	MAC 0 Payload Receive Error
137		OR	MAC 0 Transmit FIFO Underrun
137		OR	MAC 0 Collision Retry Limit
137		OR	MAC 0 Late Collision
137		OR	MAC 0 Ethernet Bus Error
137		OR	MAC 0 MII Data Transfer Done
137		OR	MAC 0 Receive Buffer Done
137		OR	MAC 0 Receive Frame Done
137		OR	MAC 0 Transmit Buffer Done
137		OR	MAC 0 Transmit Frame Done
137		OR	MAC 0 Graceful Stop
137		OR	MAC 0 Babbling Transmit Error
137		OR	MAC 0 Babbling Receive Error

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
137		OR	MAC 0 Wakeup Request (sync)
137		OR	MAC 0 Wakeup Request (async)
138		-	MAC 0 1588 Timer Interrupt – synchronous
139	ENET_1G	OR	MAC 0 Receive Buffer1 Done
139		OR	MAC 0 Receive Frame1 Done
139		OR	MAC 0 Transmit Buffer1 Done
139		OR	MAC 0 Transmit Frame1 Done
140		OR	MAC 0 Receive Buffer2 Done
140		OR	MAC 0 Receive Frame2 Done
140		OR	MAC 0 Transmit Buffer2 Done
140		OR	MAC 0 Transmit Frame2 Done
141		OR	MAC 0 Periodic Timer Overflow
141		OR	MAC 0 Time Stamp Available
141		OR	MAC 0 Payload Receive Error
141		OR	MAC 0 Transmit FIFO Underrun
141		OR	MAC 0 Collision Retry Limit
141		OR	MAC 0 Late Collision
141		OR	MAC 0 Ethernet Bus Error
141		OR	MAC 0 MII Data Transfer Done
141		OR	MAC 0 Receive Buffer Done
141		OR	MAC 0 Receive Frame Done
141		OR	MAC 0 Transmit Buffer Done
141		OR	MAC 0 Transmit Frame Done
141		OR	MAC 0 Graceful Stop
141		OR	MAC 0 Babbling Transmit Error
141		OR	MAC 0 Babbling Receive Error
141		OR	MAC 0 Receive Flush Frame0
141		OR	MAC 0 Receive Flush Frame1
141		OR	MAC 0 Receive Flush Frame2
141		OR	MAC 0 Wakeup Request (sync)
141		OR	MAC 0 Wakeup Request (async)

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
141		OR	MAC 0 Babbling Receive Error
141		OR	MAC 0 Wakeup Request (err)
142		-	MAC 0 1588 Timer Interrupt – synchronous
143	XBAR1	OR	XBAR IRQ (XBAR_OUT0)
143		OR	XBAR IRQ (XBAR_OUT1)
144		OR	XBAR IRQ (XBAR_OUT2)
144		OR	XBAR IRQ (XBAR_OUT3)
145	ADC_ETC	-	ADC_ETC done0 interrupt
146		-	ADC_ETC done1 interrupt
147		-	ADC_ETC done2 interrupt
148		-	ADC_ETC done3 interrupt
149		-	ADC_ETC err interrupt
150	Reserved	-	N/A
151	Reserved	-	N/A
152	Reserved	-	N/A
153	Reserved	-	N/A
154	Reserved	-	N/A
155	PIT1	OR	PIT1 channel 0 interrupt
155		OR	PIT1 channel 1 interrupt
155		OR	PIT1 channel 2 interrupt
155		OR	PIT1 channel 3 interrupt
156	PIT2	OR	PIT2 channel 0 interrupt
156		OR	PIT2 channel 1 interrupt
156		OR	PIT2 channel 2 interrupt
156		OR	PIT2 channel 3 interrupt
157	ACMP1	OR	ACMP1 interrupt
157		OR	ACMP1 async interrupt for STOP mode wake up
158	ACMP2	OR	ACMP2 interrupt
158		OR	ACMP2 async interrupt for STOP mode wake up
159	ACMP3	OR	ACMP3 interrupt
159		OR	ACMP3 async interrupt for STOP mode wake up
160	ACMP4	OR	ACMP4 interrupt
160		OR	ACMP4 async interrupt for STOP mode wake up
161	Reserved	-	N/A
162	Reserved	-	N/A

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
163	Reserved	-	N/A
164	Reserved	-	N/A
165	QDC1	OR	Index marker interrupt
165		OR	Home marker interrupt
165		OR	Watchdog timeout interrupt
165		OR	Compare interrupt
165		OR	Simultaneous input switching interrupt
166	QDC2	OR	Index marker interrupt
166		OR	Home marker interrupt
166		OR	Watchdog timeout interrupt
166		OR	Compare interrupt
166		OR	Simultaneous input switching interrupt
167	QDC3	OR	Index marker interrupt
167		OR	Home marker interrupt
167		OR	Watchdog timeout interrupt
167		OR	Compare interrupt
167		OR	Simultaneous input switching interrupt
168	QDC4	OR	Index marker interrupt
168		OR	Home marker interrupt
168		OR	Watchdog timeout interrupt
168		OR	Compare interrupt
168		OR	Simultaneous input switching interrupt
169	Reserved	-	N/A
170	Reserved	-	N/A
171	TMR1	OR	Interrupt request for timer #0
171		OR	Interrupt request for timer #1
171		OR	Interrupt request for timer #2
171		OR	Interrupt request for timer #3
172	TMR2	OR	Interrupt request for timer #0
172		OR	Interrupt request for timer #1
172		OR	Interrupt request for timer #2
172		OR	Interrupt request for timer #3
173	TMR3	OR	Interrupt request for timer #0
173		OR	Interrupt request for timer #1
173		OR	Interrupt request for timer #2
173		OR	Interrupt request for timer #3
174	TMR4	OR	Interrupt request for timer #0

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
174		OR	Interrupt request for timer #1
174		OR	Interrupt request for timer #2
174		OR	Interrupt request for timer #3
175	SEMA4	-	SEMA4 CP0 interrupt
176		-	SEMA4 CP1 interrupt
177	FLEXPWM2	OR	capture 0 interrupt
177		OR	compare 0 interrupt
177		OR	reload 0 interrupt
178		OR	capture 1 interrupt
178		OR	compare 1 interrupt
178		OR	reload 1 interrupt
179		OR	capture 2 interrupt
179		OR	compare 2 interrupt
179		OR	reload 2 interrupt
180		OR	capture 3 interrupt
180		OR	compare 3 interrupt
180		OR	reload 3 interrupt
181		OR	fault interrupt
181		OR	reload error interrupt
182	FLEXPWM3	OR	capture 0 interrupt
182		OR	compare 0 interrupt
182		OR	reload 0 interrupt
183		OR	capture 1 interrupt
183		OR	compare 1 interrupt
183		OR	reload 1 interrupt
184		OR	capture 2 interrupt
184		OR	compare 2 interrupt
184		OR	reload 2 interrupt
185		OR	capture 3 interrupt
185		OR	compare 3 interrupt
185		OR	reload 3 interrupt
186		OR	fault interrupt
186		OR	reload error interrupt
187	FLEXPWM4	OR	capture 0 interrupt
187		OR	compare 0 interrupt
187		OR	reload 0 interrupt
188		OR	capture 1 interrupt
188		OR	compare 1 interrupt
188		OR	reload 1 interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
189		OR	capture 2 interrupt
189		OR	compare 2 interrupt
189		OR	reload 2 interrupt
190		OR	capture 3 interrupt
190		OR	compare 3 interrupt
190		OR	reload 3 interrupt
191		OR	fault interrupt
191		OR	reload error interrupt
192	Reserved	-	N/A
193	Reserved	-	N/A
194	Reserved	-	N/A
195	Reserved	-	N/A
196	Reserved	-	N/A
197	Reserved	-	N/A
198	Reserved	-	N/A
199	Reserved	-	N/A
200	PDM	OR	HWVAD event interrupt
200		OR	HWVAD event async interrupt
201		OR	HWVAD error interrupt
201		OR	HWVAD error async interrupt
202		OR	filter interrupt
202		OR	filter async interrupt
203		OR	error interrupt
203		OR	error async interrupt
204	EMVSIM1	OR	EMVSIM1 interrupt (data_irq)
204		OR	EMVSIM1 interrupt
205	EMVSIM2	OR	EMVSIM2 interrupt (data_irq)
205		OR	EMVSIM2 interrupt
206	MECC1	-	MECC1 interrupt
207		-	MECC1 Fatal interrupt
208	MECC2	-	MECC2 interrupt
209		-	MECC2 Fatal interrupt
210	XECC_FLEXSPI1	-	XECC_FLEXSPI1 interrupt
211		-	XECC_FLEXSPI1 Fatal interrupt
212	XECC_FLEXSPI2	-	XECC_FLEXSPI2 interrupt
213		-	XECC_FLEXSPI2 Fatal interrupt
214	XECC_SEMC	-	XECC_SEMC interrupt
215		-	XECC_SEMC Fatal interrupt

Table continues on the next page...

Table 4-2. CM4 domain interrupt summary (continued)

IRQ	Interrupt Source	LOGIC	Interrupt Description
216	ENET_QOS	OR	LPI Rx exit interrupt output
216		OR	Interrupt signal to host system
216		OR	Channel 0 Transmit Interrupt signal to host system
216		OR	Channel 1 Transmit Interrupt signal to host system
216		OR	Channel 2 Transmit Interrupt signal to host system
216		OR	Channel 3 Transmit Interrupt signal to host system
216		OR	Channel 4 Transmit Interrupt signal to host system
216		OR	Channel 0 Receive Interrupt signal to host system
216		OR	Channel 1 Receive Interrupt signal to host system
216		OR	Channel 2 Receive Interrupt signal to host system
216		OR	Channel 3 Receive Interrupt signal to host system
216		OR	Channel 4 Receive Interrupt signal to host system
217		-	Interrupt from PMT

4.4 DMA Mux

The table below shows the DMA request signals for the peripherals in the chip:

NOTE

The DMA Mux mapping is applicable to EDMA and EDMA_LPSR.

Table 4-3. DMA Mux Mapping

Channel	Module	Gate	Description
0	Reserved	-	Reserved
1	FLEXIO1	OR	FlexIO1 shifter2 DMA Request
		OR	FlexIO1 shifter2 Async DMA Request
		OR	FlexIO1 shifter3 DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	FlexIO1 shifter3 Async DMA Request
		OR	FlexIO1 timer2 DMA Request
		OR	FlexIO1 timer2 Async DMA Request
		OR	FlexIO1 timer3 DMA Request
		OR	FlexIO1 timer3 Async DMA Request
2	FLEXIO1	OR	FlexIO1 shifter4 DMA Request
		OR	FlexIO1 shifter4 Async DMA Request
		OR	FlexIO1 shifter5 DMA Request
		OR	FlexIO1 shifter5 Async DMA Request
		OR	FlexIO1 timer4 DMA Request
		OR	FlexIO1 timer4 Async DMA Request
		OR	FlexIO1 timer5 DMA Request
		OR	FlexIO1 timer5 Async DMA Request
3	FLEXIO1	OR	FlexIO1 shifter6 DMA Request
		OR	FlexIO1 shifter6 Async DMA Request
		OR	FlexIO1 shifter7 DMA Request
		OR	FlexIO1 shifter7 Async DMA Request
		OR	FlexIO1 timer6 DMA Request
		OR	FlexIO1 timer6 Async DMA Request
		OR	FlexIO1 timer7 DMA Request
		OR	FlexIO1 timer7 Async DMA Request
4	FLEXIO2	OR	FlexIO2 shifter0 DMA Request
		OR	FlexIO2 shifter0 Async DMA Request
		OR	FlexIO2 shifter1 DMA Request
		OR	FlexIO2 shifter1 Async DMA Request
		OR	FlexIO2 timer0 DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	FlexIO2 timer0 Async DMA Request
		OR	FlexIO2 timer1 DMA Request
		OR	FlexIO2 timer1 Async DMA Request
5	FLEXIO2	OR	FlexIO2 shifter2 DMA Request
		OR	FlexIO2 shifter2 Async DMA Request
		OR	FlexIO2 shifter3 DMA Request
		OR	FlexIO2 shifter3 Async DMA Request
		OR	FlexIO2 timer2 DMA Request
		OR	FlexIO2 timer2 Async DMA Request
		OR	FlexIO2 timer3 DMA Request
		OR	FlexIO2 timer3 Async DMA Request
6	FLEXIO2	OR	FlexIO2 shifter4 DMA Request
		OR	FlexIO2 shifter4 Async DMA Request
		OR	FlexIO2 shifter5 DMA Request
		OR	FlexIO2 shifter5 Async DMA Request
		OR	FlexIO2 timer4 DMA Request
		OR	FlexIO2 timer4 Async DMA Request
		OR	FlexIO2 timer5 DMA Request
		OR	FlexIO2 timer5 Async DMA Request
7	FLEXIO2	OR	FlexIO2 shifter6 DMA Request
		OR	FlexIO2 shifter6 Async DMA Request
		OR	FlexIO2 shifter7 DMA Request
		OR	FlexIO2 shifter7 Async DMA Request
		OR	FlexIO2 timer6 DMA Request
		OR	FlexIO2 timer6 Async DMA Request
		OR	FlexIO2 timer7 DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	FlexIO2 timer7 Async DMA Request
8	LPUART1	OR	UART1 Tx FIFO DMA Request
		OR	UART1 Tx FIFO Async DMA Request
9	LPUART1	OR	UART1 Rx FIFO DMA Request
		OR	UART1 Rx FIFO Async DMA Request
10	LPUART2	OR	UART2 Tx FIFO DMA Request
		OR	UART2 Tx FIFO Async DMA Request
11	LPUART2	OR	UART2 Rx FIFO DMA Request
		OR	UART2 Rx FIFO Async DMA Request
12	LPUART3	OR	UART3 Tx FIFO Async DMA Request
		OR	UART3 Tx FIFO DMA Request
13	LPUART3	OR	UART3 Rx FIFO DMA Request
		OR	UART3 Rx FIFO Async DMA Request
14	LPUART4	OR	UART4 Tx FIFO DMA Request
		OR	UART4 Tx FIFO Async DMA Request
15	LPUART4	OR	UART4 Rx FIFO DMA Request
		OR	UART4 Rx FIFO Async DMA Request
16	LPUART5	OR	UART5 Tx FIFO DMA Request
		OR	UART5 Tx FIFO Async DMA Request
17	LPUART5	OR	UART5 Rx FIFO DMA Request
		OR	UART5 Rx FIFO Async DMA Request
18	LPUART6	OR	UART6 Tx FIFO DMA Request
		OR	UART6 Tx FIFO Async DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
19	LPUART6	OR	UART6 Rx FIFO DMA Request
		OR	UART6 Rx FIFO Async DMA Request
20	LPUART7	OR	UART7 Tx FIFO DMA Request
		OR	UART7 Tx FIFO Async DMA Request
21	LPUART7	OR	UART7 Rx FIFO DMA Request
		OR	UART7 Rx FIFO Async DMA Request
22	LPUART8	OR	UART8 Tx FIFO DMA Request
		OR	UART8 Tx FIFO Async DMA Request
23	LPUART8	OR	UART8 Rx FIFO DMA Request
		OR	UART8 Rx FIFO Async DMA Request
24	LPUART9	OR	UART9 Tx FIFO DMA Request
		OR	UART9 Tx FIFO Async DMA Request
25	LPUART9	OR	UART9 Rx FIFO DMA Request
		OR	UART9 Rx FIFO Async DMA Request
26	LPUART10	OR	UART10 Tx FIFO DMA Request
		OR	UART10 Tx FIFO Async DMA Request
27	LPUART10	OR	UART10 Rx FIFO DMA Request
		OR	UART10 Rx FIFO Async DMA Request
28	LPUART11	OR	UART11 Tx FIFO DMA Request
		OR	UART11 Tx FIFO Async DMA Request
29	LPUART11	OR	UART11 Rx FIFO DMA Request
		OR	UART11 Rx FIFO Async DMA Request
30	LPUART12	OR	UART12 Tx FIFO DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	UART12 Tx FIFO Async DMA Request
31	LPUART12	OR	UART12 Rx FIFO DMA Request
		OR	UART12 Rx FIFO Async DMA Request
32	CSI	-	CSI Write DMA Request
33	PXP	-	PXP DMA Event
34	eLCDIF	-	eLCDIF DMA Event
35	LCDIFv2	-	LCDIFv2 DMA Event
36	LPSPI1	OR	LPSPI1 Rx FIFO DMA Request
		OR	LPSPI1 Rx FIFO Async DMA Request
37	LPSPI1	OR	LPSPI1 Tx FIFO DMA Request
		OR	LPSPI1 Tx FIFO Async DMA Request
38	LPSPI2	OR	LPSPI2 Rx FIFO DMA Request
		OR	LPSPI2 Rx FIFO Async DMA Request
39	LPSPI2	OR	LPSPI2 Tx FIFO DMA Request
		OR	LPSPI2 Tx FIFO Async DMA Request
40	LPSPI3	OR	LPSPI3 Rx FIFO DMA Request
		OR	LPSPI3 Rx FIFO Async DMA Request
41	LPSPI3	OR	LPSPI3 Tx FIFO DMA Request
		OR	LPSPI3 Tx FIFO Async DMA Request
42	LPSPI4	OR	LPSPI4 Rx FIFO DMA Request
		OR	LPSPI4 Rx FIFO Async DMA Request
43	LPSPI4	OR	LPSPI4 Tx FIFO DMA Request
		OR	LPSPI4 Tx FIFO Async DMA Request
44	LPSPI5	OR	LPSPI5 Rx FIFO DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	LPSPI5 Rx FIFO Async DMA Request
45	LPSPI5	OR	LPSPI5 Tx FIFO DMA Request
		OR	LPSPI5 Tx FIFO Async DMA Request
46	LPSPI6	OR	LPSPI6 Rx FIFO DMA Request
		OR	LPSPI6 Rx FIFO Async DMA Request
47	LPSPI6	OR	LPSPI6 Tx FIFO DMA Request
		OR	LPSPI6 Tx FIFO Async DMA Request
48	LPI2C1	OR	I2C1 Master Rx FIFO DMA Request
		OR	I2C1 Master Rx FIFO Async DMA Request
		OR	I2C1 Slave Rx FIFO DMA Request
		OR	I2C1 Slave Rx FIFO Async DMA Request
		OR	I2C1 Master Tx FIFO DMA Request
		OR	I2C1 Master Tx FIFO Async DMA Request
		OR	I2C1 Slave Tx FIFO DMA Request
		OR	I2C1 Slave Tx FIFO Async DMA Request
49	LPI2C2	OR	I2C2 Master Rx FIFO DMA Request
		OR	I2C2 Master Rx FIFO Async DMA Request
		OR	I2C2 Slave Rx FIFO DMA Request
		OR	I2C2 Slave Rx FIFO Async DMA Request
		OR	I2C2 Master Tx FIFO DMA Request
		OR	I2C2 Master Tx FIFO Async DMA Request
		OR	I2C2 Slave Tx FIFO DMA Request
		OR	I2C2 Slave Tx FIFO Async DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
50	LPI2C3	OR	I2C3 Master Rx FIFO DMA Request
		OR	I2C3 Master Rx FIFO Async DMA Request
		OR	I2C3 Slave Rx FIFO DMA Request
		OR	I2C3 Slave Rx FIFO Async DMA Request
		OR	I2C3 Master Tx FIFO DMA Request
		OR	I2C3 Master Tx FIFO Async DMA Request
		OR	I2C3 Slave Tx FIFO DMA Request
		OR	I2C3 Slave Tx FIFO Async DMA Request
51	LPI2C4	OR	I2C4 Master Rx FIFO DMA Request
		OR	I2C4 Master Rx FIFO Async DMA Request
		OR	I2C4 Slave Rx FIFO DMA Request
		OR	I2C4 Slave Rx FIFO Async DMA Request
		OR	I2C4 Master Tx FIFO DMA Request
		OR	I2C4 Master Tx FIFO Async DMA Request
		OR	I2C4 Slave Tx FIFO DMA Request
		OR	I2C4 Slave Tx FIFO Async DMA Request
52	LPI2C5	OR	I2C5 Master Rx FIFO DMA Request
		OR	I2C5 Master Rx FIFO Async DMA Request
		OR	I2C5 Slave Rx FIFO DMA Request
		OR	I2C5 Slave Rx FIFO Async DMA Request
		OR	I2C5 Master Tx FIFO DMA Request
		OR	I2C5 Master Tx FIFO Async DMA Request
		OR	I2C5 Slave Tx FIFO DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	I2C5 Slave Tx FIFO Async DMA Request
53	LPI2C6	OR	I2C6 Master Rx FIFO DMA Request
		OR	I2C6 Master Rx FIFO Async DMA Request
		OR	I2C6 Slave Rx FIFO DMA Request
		OR	I2C6 Slave Rx FIFO Async DMA Request
		OR	I2C6 Master Tx FIFO DMA Request
		OR	I2C6 Master Tx FIFO Async DMA Request
		OR	I2C6 Slave Tx FIFO DMA Request
		OR	I2C6 Slave Tx FIFO Async DMA Request
54	SAI1	-	SAI1 Rx FIFO DMA Request
55	SAI1	-	SAI1 Tx FIFO DMA Request
56	SAI2	-	SAI2 Rx FIFO DMA Request
57	SAI2	-	SAI2 Tx FIFO DMA Request
58	SAI3	-	SAI3 Rx FIFO DMA Request
59	SAI3	-	SAI3 Tx FIFO DMA Request
60	SAI4	-	SAI4 Rx FIFO DMA Request
61	SAI4	-	SAI4 Tx FIFO DMA Request
62	SPDIF	-	SPDIF RX DMA Request
63	SPDIF	-	SPDIF TX DMA Request
64	ADC_ETC	-	ADC_ETC DMA Request
65	FLEXIO1	OR	FlexIO1 shifter0 DMA Request
		OR	FlexIO1 shifter0 Async DMA Request
		OR	FlexIO1 shifter1 DMA Request
		OR	FlexIO1 shifter1 Async DMA Request
		OR	FlexIO1 timer0 DMA Request
		OR	FlexIO1 timer0 Async DMA Request
		OR	FlexIO1 timer1 DMA Request
		OR	FlexIO1 timer1 Async DMA Request
66	LPADC1	-	ADC1 DMA Request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
67	LPADC2	-	ADC2 DMA Request
68	Reserved	-	Reserved
69	ACMP1	-	ACMP1 DMA Request
70	ACMP2	-	ACMP2 DMA Request
71	ACMP3	-	ACMP3 DMA Request
72	ACMP4	-	ACMP4 DMA Request
73-76	Reserved	-	Reserved
77	FLEXSPI1	-	FlexSPI1 Rx FIFO DMA Request
78	FLEXSPI1	-	FlexSPI1 Tx FIFO DMA Request
79	FLEXSPI2	-	FlexSPI2 Rx FIFO DMA Request
80	FLEXSPI2	-	FlexSPI2 Tx FIFO DMA Request
81	XBAR1	-	XBAR1_OUT0 DMA request
82	XBAR1	-	XBAR1_OUT1 DMA request
83	XBAR1	-	XBAR1_OUT2 DMA request
84	XBAR1	-	XBAR1_OUT3 DMA request
85	FLEXPWM1	-	FlexPWM1 sub-module0 capture register read DMA request
86	FLEXPWM1	-	FlexPWM1 sub-module1 capture register read DMA request
87	FLEXPWM1	-	FlexPWM1 sub-module2 capture register read DMA request
88	FLEXPWM1	-	FlexPWM1 sub-module3 capture register read DMA request
89	FLEXPWM1	-	FlexPWM1 sub-module0 value registers write DMA request
90	FLEXPWM1	-	FlexPWM1 sub-module1 value registers write DMA request
91	FLEXPWM1	-	FlexPWM1 sub-module2 value registers write DMA request
92	FLEXPWM1	-	FlexPWM1 sub-module3 value registers write DMA request
93	FLEXPWM2	-	FlexPWM2 sub-module0 capture register read DMA request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
94	FLEXPWM2	-	FlexPWM2 sub-module1 capture register read DMA request
95	FLEXPWM2	-	FlexPWM2 sub-module2 capture register read DMA request
96	FLEXPWM2	-	FlexPWM2 sub-module3 capture register read DMA request
97	FLEXPWM2	-	FlexPWM2 sub-module0 value registers write DMA request
98	FLEXPWM2	-	FlexPWM2 sub-module1 value registers write DMA request
99	FLEXPWM2	-	FlexPWM2 sub-module2 value registers write DMA request
100	FLEXPWM2	-	FlexPWM2 sub-module3 value registers write DMA request
101	FLEXPWM3	-	FlexPWM3 sub-module0 capture register read DMA request
102	FLEXPWM3	-	FlexPWM3 sub-module1 capture register read DMA request
103	FLEXPWM3	-	FlexPWM3 sub-module2 capture register read DMA request
104	FLEXPWM3	-	FlexPWM3 sub-module3 capture register read DMA request
105	FLEXPWM3	-	FlexPWM3 sub-module0 value registers write DMA request
106	FLEXPWM3	-	FlexPWM3 sub-module1 value registers write DMA request
107	FLEXPWM3	-	FlexPWM3 sub-module2 value registers write DMA request
108	FLEXPWM3	-	FlexPWM3 sub-module3 value registers write DMA request
109	FLEXPWM4	-	FlexPWM4 sub-module0 capture register read DMA request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
110	FLEXPWM4	-	FlexPWM4 sub-module1 capture register read DMA request
111	FLEXPWM4	-	FlexPWM4 sub-module2 capture register read DMA request
112	FLEXPWM4	-	FlexPWM4 sub-module3 capture register read DMA request
113	FLEXPWM4	-	FlexPWM4 sub-module0 value registers write DMA request
114	FLEXPWM4	-	FlexPWM4 sub-module1 value registers write DMA request
115	FLEXPWM4	-	FlexPWM4 sub-module2 value registers write DMA request
116	FLEXPWM4	-	FlexPWM4 sub-module3 value registers write DMA request
117-132	Reserved	-	Reserved
133	QTIMER1	-	QTIMER1 timer0 capture register read DMA request
134	QTIMER1	-	QTIMER1 timer1 capture register read DMA request
135	QTIMER1	-	QTIMER1 timer2 capture register read DMA request
136	QTIMER1	-	QTIMER1 timer3 capture register read DMA request
137	QTIMER1	OR	QTimer1 timer0 cmpld1 register write DMA request
		OR	QTimer1 timer1 cmpld2 register write DMA request
138	QTIMER1	OR	QTimer1 timer1 cmpld1 register write DMA request
		OR	QTimer1 timer0 cmpld2 register write DMA request
139	QTIMER1	OR	QTimer1 timer2 cmpld1 register write DMA request
		OR	QTimer1 timer3 cmpld2 register write DMA request
140	QTIMER1	OR	QTimer1 timer3 cmpld1 register write DMA request
		OR	QTimer1 timer2 cmpld2 register write DMA request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
141	QTIMER2	-	QTIMER2 timer0 capture register read DMA request
142	QTIMER2	-	QTIMER2 timer1 capture register read DMA request
143	QTIMER2	-	QTIMER2 timer2 capture register read DMA request
144	QTIMER2	-	QTIMER2 timer3 capture register read DMA request
145	QTIMER2	OR	QTimer2 timer0 cmpld1 register write DMA request
		OR	QTimer2 timer1 cmpld2 register write DMA request
146	QTIMER2	OR	QTimer2 timer1 cmpld1 register write DMA request
		OR	QTimer2 timer0 cmpld2 register write DMA request
147	QTIMER2	OR	QTimer2 timer2 cmpld1 register write DMA request
		OR	QTimer2 timer3 cmpld2 register write DMA request
148	QTIMER2	OR	QTimer2 timer3 cmpld1 register write DMA request
		OR	QTimer2 timer2 cmpld2 register write DMA request
149	QTIMER3	-	QTIMER3 timer0 capture register read DMA request
150	QTIMER3	-	QTIMER3 timer1 capture register read DMA request
151	QTIMER3	-	QTIMER3 timer2 capture register read DMA request
152	QTIMER3	-	QTIMER3 timer3 capture register read DMA request
153	QTIMER3	OR	QTimer3 timer0 cmpld1 register write DMA request
		OR	QTimer3 timer1 cmpld2 register write DMA request
154	QTIMER3	OR	QTimer3 timer1 cmpld1 register write DMA request
		OR	QTimer3 timer0 cmpld2 register write DMA request
155	QTIMER3	OR	QTimer3 timer2 cmpld1 register write DMA request
		OR	QTimer3 timer3 cmpld2 register write DMA request
156	QTIMER3	OR	QTimer3 timer3 cmpld1 register write DMA request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
		OR	QTimer3 timer2 cmpld2 register write DMA request
157	QTIMER4	-	QTIMER4 timer0 capture register read DMA request
158	QTIMER4	-	QTIMER4 timer1 capture register read DMA request
159	QTIMER4	-	QTIMER4 timer2 capture register read DMA request
160	QTIMER4	-	QTIMER4 timer3 capture register read DMA request
161	QTIMER4	OR	QTimer4 timer0 cmpld1 register write DMA request
		OR	QTimer4 timer1 cmpld2 register write DMA request
162	QTIMER4	OR	QTimer4 timer1 cmpld1 register write DMA request
		OR	QTimer4 timer0 cmpld2 register write DMA request
163	QTIMER4	OR	QTimer4 timer2 cmpld1 register write DMA request
		OR	QTimer4 timer3 cmpld2 register write DMA request
164	QTIMER4	OR	QTimer4 timer3 cmpld1 register write DMA request
		OR	QTimer4 timer2 cmpld2 register write DMA request
165-180	Reserved	-	Reserved
181	PDM	OR	PDM DMA request
		OR	PDM DMA async request
182	ENET	-	ENET Timer0 DMA Request
183	ENET	-	ENET Timer1 DMA Request
184	ENET_1G	-	ENET_1G Timer0 DMA Request
185	ENET_1G	-	ENET_1G Timer1 DMA Request
186	CAN1	-	CAN1 DMA Request
187	CAN2	-	CAN2 DMA Request
188	CAN3	-	CAN3 DMA Request
189	DAC	-	DAC DMA Request
190	Reserved	-	Reserved
191	ASRC	-	ASRC pair A input DMA request
192	ASRC	-	ASRC pair B input DMA request

Table continues on the next page...

Table 4-3. DMA Mux Mapping (continued)

Channel	Module	Gate	Description
193	ASRC	-	ASRC pair C input DMA request
194	ASRC	-	ASRC pair A output DMA request
195	ASRC	-	ASRC pair B output DMA request
196	ASRC	-	ASRC pair C output DMA request
197	EMVSIM1	-	EMVSIM1 TX DMA Request
198	EMVSIM1	-	EMVSIM1 RX DMA Request
199	EMVSIM2	-	EMVSIM2 TX DMA Request
200	EMVSIM2	-	EMVSIM2 RX DMA Request
201	ENET_QOS	-	ENET_QOS PPS0 output DMA Request
202	ENET_QOS	-	ENET_QOS PPS1 output DMA Request
203-207	Reserved	-	Reserved

4.5 PIT Channel Assignments For Periodic DMA Triggering

This table shows the PIT channel to DMA channel correspondence.

Table 4-4. PIT Channel Assignments For Periodic DMA Triggering

DMA Channel Number	PIT Channel
eDMA Channel 0	PIT1 Channel 0
eDMA Channel 1	PIT1 Channel 1
eDMA Channel 2	PIT1 Channel 2
eDMA Channel 3	PIT1 Channel 3
eDMA_LPSR Channel 0	PIT2 Channel 0
eDMA_LPSR Channel 1	PIT2 Channel 1
eDMA_LPSR Channel 2	PIT2 Channel 2
eDMA_LPSR Channel 3	PIT2 Channel 3

4.6 XBAR Resource Assignments

This table shows the XBAR resource assignments in the chip.

NOTE

ADC_ETC0_COCO0 ... ADC_ETC1_COCO3 in the table are corresponding to coco0 ... coco7, also ADC_ETC0_TRIG0 ... ADC_ETC1_TRIG3 corresponding to trg0 ... trg7, see the figure "ADC_ETC block diagram" in the ADC_ETC chapter.

Table 4-5. XBAR1 Input Assignments

Assigned Input	XBAR1 Input	Gate
LOGIC LOW	XBAR1_IN00	-
LOGIC HIGH	XBAR1_IN01	-
GND	XBAR1_IN02	-
GND	XBAR1_IN03	-
IOMUX_XBAR_INOUT04	XBAR1_IN04	-
IOMUX_XBAR_INOUT05	XBAR1_IN05	-
IOMUX_XBAR_INOUT06	XBAR1_IN06	-
IOMUX_XBAR_INOUT07	XBAR1_IN07	-
IOMUX_XBAR_INOUT08	XBAR1_IN08	-
IOMUX_XBAR_INOUT09	XBAR1_IN09	-
IOMUX_XBAR_INOUT10	XBAR1_IN10	-
IOMUX_XBAR_INOUT11	XBAR1_IN11	-
IOMUX_XBAR_INOUT12	XBAR1_IN12	-
IOMUX_XBAR_INOUT13	XBAR1_IN13	-
IOMUX_XBAR_INOUT14	XBAR1_IN14	-
IOMUX_XBAR_INOUT15	XBAR1_IN15	-
IOMUX_XBAR_INOUT16	XBAR1_IN16	-
IOMUX_XBAR_INOUT17	XBAR1_IN17	-
IOMUX_XBAR_INOUT18	XBAR1_IN18	-
IOMUX_XBAR_INOUT19	XBAR1_IN19	-
IOMUX_XBAR_INOUT20	XBAR1_IN20	-
IOMUX_XBAR_INOUT21	XBAR1_IN21	-
IOMUX_XBAR_INOUT22	XBAR1_IN22	-
IOMUX_XBAR_INOUT23	XBAR1_IN23	-
IOMUX_XBAR_INOUT24	XBAR1_IN24	-
IOMUX_XBAR_INOUT25	XBAR1_IN25	-
IOMUX_XBAR_INOUT26	XBAR1_IN26	-
IOMUX_XBAR_INOUT27	XBAR1_IN27	-
IOMUX_XBAR_INOUT28	XBAR1_IN28	-
IOMUX_XBAR_INOUT29	XBAR1_IN29	-
IOMUX_XBAR_INOUT30	XBAR1_IN30	-
IOMUX_XBAR_INOUT31	XBAR1_IN31	-
IOMUX_XBAR_INOUT32	XBAR1_IN32	-
IOMUX_XBAR_INOUT33	XBAR1_IN33	-

Table continues on the next page...

Table 4-5. XBAR1 Input Assignments (continued)

Assigned Input	XBAR1 Input	Gate
IOMUX_XBAR_INOUT34	XBAR1_IN34	-
IOMUX_XBAR_INOUT35	XBAR1_IN35	-
IOMUX_XBAR_INOUT36	XBAR1_IN36	-
IOMUX_XBAR_INOUT37	XBAR1_IN37	-
IOMUX_XBAR_INOUT38	XBAR1_IN38	-
IOMUX_XBAR_INOUT39	XBAR1_IN39	-
IOMUX_XBAR_INOUT40	XBAR1_IN40	-
GND	XBAR1_IN41	-
ACMP1_OUT	XBAR1_IN42	-
ACMP2_OUT	XBAR1_IN43	-
ACMP3_OUT	XBAR1_IN44	-
ACMP4_OUT	XBAR1_IN45	-
GND	XBAR1_IN46-49	-
QTIMER1_TIMER0	XBAR1_IN50	-
QTIMER1_TIMER1	XBAR1_IN51	-
QTIMER1_TIMER2	XBAR1_IN52	-
QTIMER1_TIMER3	XBAR1_IN53	-
QTIMER2_TIMER0	XBAR1_IN54	-
QTIMER2_TIMER1	XBAR1_IN55	-
QTIMER2_TIMER2	XBAR1_IN56	-
QTIMER2_TIMER3	XBAR1_IN57	-
QTIMER3_TIMER0	XBAR1_IN58	-
QTIMER3_TIMER1	XBAR1_IN59	-
QTIMER3_TIMER2	XBAR1_IN60	-
QTIMER3_TIMER3	XBAR1_IN61	-
QTIMER4_TIMER0	XBAR1_IN62	-
QTIMER4_TIMER1	XBAR1_IN63	-
QTIMER4_TIMER2	XBAR1_IN64	-
QTIMER4_TIMER3	XBAR1_IN65	-
GND	XBAR1_IN66-73	-
FLEXPWM1_PWM0_OUT_TRIG0	XBAR1_IN74	-
FLEXPWM1_PWM0_OUT_TRIG1	XBAR1_IN75	-
FLEXPWM1_PWM1_OUT_TRIG0	XBAR1_IN76	-
FLEXPWM1_PWM1_OUT_TRIG1	XBAR1_IN77	-
FLEXPWM1_PWM2_OUT_TRIG0	XBAR1_IN78	-
FLEXPWM1_PWM2_OUT_TRIG1	XBAR1_IN79	-
FLEXPWM1_PWM3_OUT_TRIG0	XBAR1_IN80	-
FLEXPWM1_PWM3_OUT_TRIG1	XBAR1_IN81	-
FLEXPWM2_PWM0_OUT_TRIG0	XBAR1_IN82	OR

Table continues on the next page...

Table 4-5. XBAR1 Input Assignments (continued)

Assigned Input	XBAR1 Input	Gate
FLEXPWM2_PWM0_OUT_TRIG1	XBAR1_IN82	OR
FLEXPWM2_PWM1_OUT_TRIG0	XBAR1_IN83	OR
FLEXPWM2_PWM1_OUT_TRIG1	XBAR1_IN83	OR
FLEXPWM2_PWM2_OUT_TRIG0	XBAR1_IN84	OR
FLEXPWM2_PWM2_OUT_TRIG1	XBAR1_IN84	OR
FLEXPWM2_PWM3_OUT_TRIG0	XBAR1_IN85	OR
FLEXPWM2_PWM3_OUT_TRIG1	XBAR1_IN85	OR
FLEXPWM3_PWM0_OUT_TRIG0	XBAR1_IN86	OR
FLEXPWM3_PWM0_OUT_TRIG1	XBAR1_IN86	OR
FLEXPWM3_PWM1_OUT_TRIG0	XBAR1_IN87	OR
FLEXPWM3_PWM1_OUT_TRIG1	XBAR1_IN87	OR
FLEXPWM3_PWM2_OUT_TRIG0	XBAR1_IN88	OR
FLEXPWM3_PWM2_OUT_TRIG1	XBAR1_IN88	OR
FLEXPWM3_PWM3_OUT_TRIG0	XBAR1_IN89	OR
FLEXPWM3_PWM3_OUT_TRIG1	XBAR1_IN89	OR
FLEXPWM4_PWM0_OUT_TRIG0	XBAR1_IN90	OR
FLEXPWM4_PWM0_OUT_TRIG1	XBAR1_IN90	OR
FLEXPWM4_PWM1_OUT_TRIG0	XBAR1_IN91	OR
FLEXPWM4_PWM1_OUT_TRIG1	XBAR1_IN91	OR
FLEXPWM4_PWM2_OUT_TRIG0	XBAR1_IN92	OR
FLEXPWM4_PWM2_OUT_TRIG1	XBAR1_IN92	OR
FLEXPWM4_PWM3_OUT_TRIG0	XBAR1_IN93	OR
FLEXPWM4_PWM3_OUT_TRIG1	XBAR1_IN93	OR
GND	XBAR1_IN94-101	-
PIT1_TRIGGER0	XBAR1_IN102	-
PIT1_TRIGGER1	XBAR1_IN103	-
PIT1_TRIGGER2	XBAR1_IN104	-
PIT1_TRIGGER3	XBAR1_IN105	-
QDC1_POSMATCH	XBAR1_IN106	-
QDC2_POSMATCH	XBAR1_IN107	-
QDC3_POSMATCH	XBAR1_IN108	-
QDC4_POSMATCH	XBAR1_IN109	-
GND	XBAR1_IN110-111	-
EDMA_DONE0	XBAR1_IN112	-
EDMA_DONE1	XBAR1_IN113	-
EDMA_DONE2	XBAR1_IN114	-
EDMA_DONE3	XBAR1_IN115	-
EDMA_DONE4	XBAR1_IN116	-
EDMA_DONE5	XBAR1_IN117	-

Table continues on the next page...

Table 4-5. XBAR1 Input Assignments (continued)

Assigned Input	XBAR1 Input	Gate
EDMA_DONE6	XBAR1_IN118	-
EDMA_DONE7	XBAR1_IN119	-
EDMA_LPSR_DONE0	XBAR1_IN120	-
EDMA_LPSR_DONE1	XBAR1_IN121	-
EDMA_LPSR_DONE2	XBAR1_IN122	-
EDMA_LPSR_DONE3	XBAR1_IN123	-
EDMA_LPSR_DONE4	XBAR1_IN124	-
EDMA_LPSR_DONE5	XBAR1_IN125	-
EDMA_LPSR_DONE6	XBAR1_IN126	-
EDMA_LPSR_DONE7	XBAR1_IN127	-
AOI1_OUT0	XBAR1_IN128	-
AOI1_OUT1	XBAR1_IN129	-
AOI1_OUT2	XBAR1_IN130	-
AOI1_OUT3	XBAR1_IN131	-
AOI2_OUT0	XBAR1_IN132	-
AOI2_OUT1	XBAR1_IN133	-
AOI2_OUT2	XBAR1_IN134	-
AOI2_OUT3	XBAR1_IN135	-
ADC_ETC0_COCO0	XBAR1_IN136	-
ADC_ETC0_COCO1	XBAR1_IN137	-
ADC_ETC0_COCO2	XBAR1_IN138	-
ADC_ETC0_COCO3	XBAR1_IN139	-
ADC_ETC1_COCO0	XBAR1_IN140	-
ADC_ETC1_COCO1	XBAR1_IN141	-
ADC_ETC1_COCO2	XBAR1_IN142	-
ADC_ETC1_COCO3	XBAR1_IN143	-
GND	XABR1_IN144-155	-

Table 4-6. XBAR2 Input Assignments

Assigned Input	XBAR2 Input	Gate
LOGIC LOW	XBAR2_IN00	-
LOGIC HIGH	XBAR2_IN01	-
ACMP1_OUT	XBAR2_IN02	-
ACMP2_OUT	XBAR2_IN03	-
ACMP3_OUT	XBAR2_IN04	-
ACMP4_OUT	XBAR2_IN05	-
GND	XBAR2_IN06-09	-
QTIMER1_TIMER0	XBAR2_IN10	-

Table continues on the next page...

Table 4-6. XBAR2 Input Assignments (continued)

Assigned Input	XBAR2 Input	Gate
QTIMER1_TIMER1	XBAR2_IN11	-
QTIMER1_TIMER2	XBAR2_IN12	-
QTIMER1_TIMER3	XBAR2_IN13	-
QTIMER2_TIMER0	XBAR2_IN14	-
QTIMER2_TIMER1	XBAR2_IN15	-
QTIMER2_TIMER2	XBAR2_IN16	-
QTIMER2_TIMER3	XBAR2_IN17	-
QTIMER3_TIMER0	XBAR2_IN18	-
QTIMER3_TIMER1	XBAR2_IN19	-
QTIMER3_TIMER2	XBAR2_IN20	-
QTIMER3_TIMER3	XBAR2_IN21	-
QTIMER4_TIMER0	XBAR2_IN22	-
QTIMER4_TIMER1	XBAR2_IN23	-
QTIMER4_TIMER2	XBAR2_IN24	-
QTIMER4_TIMER3	XBAR2_IN25	-
GND	XBAR2_IN26-33	-
FLEXPWM1_PWM0_OUT_TRIG0	XBAR2_IN34	OR
FLEXPWM1_PWM0_OUT_TRIG1	XBAR2_IN34	OR
FLEXPWM1_PWM1_OUT_TRIG0	XBAR2_IN35	OR
FLEXPWM1_PWM1_OUT_TRIG1	XBAR2_IN35	OR
FLEXPWM1_PWM2_OUT_TRIG0	XBAR2_IN36	OR
FLEXPWM1_PWM2_OUT_TRIG1	XBAR2_IN36	OR
FLEXPWM1_PWM3_OUT_TRIG0	XBAR2_IN37	OR
FLEXPWM1_PWM3_OUT_TRIG1	XBAR2_IN37	OR
FLEXPWM2_PWM0_OUT_TRIG0	XBAR2_IN38	OR
FLEXPWM2_PWM0_OUT_TRIG1	XBAR2_IN38	OR
FLEXPWM2_PWM1_OUT_TRIG0	XBAR2_IN39	OR
FLEXPWM2_PWM1_OUT_TRIG1	XBAR2_IN39	OR
FLEXPWM2_PWM2_OUT_TRIG0	XBAR2_IN40	OR
FLEXPWM2_PWM2_OUT_TRIG1	XBAR2_IN40	OR
FLEXPWM2_PWM3_OUT_TRIG0	XBAR2_IN41	OR
FLEXPWM2_PWM3_OUT_TRIG1	XBAR2_IN41	OR
FLEXPWM3_PWM0_OUT_TRIG0	XBAR2_IN42	OR
FLEXPWM3_PWM0_OUT_TRIG1	XBAR2_IN42	OR
FLEXPWM3_PWM1_OUT_TRIG0	XBAR2_IN43	OR
FLEXPWM3_PWM1_OUT_TRIG1	XBAR2_IN43	OR
FLEXPWM3_PWM2_OUT_TRIG0	XBAR2_IN44	OR
FLEXPWM3_PWM2_OUT_TRIG1	XBAR2_IN44	OR
FLEXPWM3_PWM3_OUT_TRIG0	XBAR2_IN45	OR

Table continues on the next page...

Table 4-6. XBAR2 Input Assignments (continued)

Assigned Input	XBAR2 Input	Gate
FLEXPWM3_PWM3_OUT_TRIG1	XBAR2_IN45	OR
FLEXPWM4_PWM0_OUT_TRIG0	XBAR2_IN46	OR
FLEXPWM4_PWM0_OUT_TRIG1	XBAR2_IN46	OR
FLEXPWM4_PWM1_OUT_TRIG0	XBAR2_IN47	OR
FLEXPWM4_PWM1_OUT_TRIG1	XBAR2_IN47	OR
FLEXPWM4_PWM2_OUT_TRIG0	XBAR2_IN48	OR
FLEXPWM4_PWM2_OUT_TRIG1	XBAR2_IN48	OR
FLEXPWM4_PWM3_OUT_TRIG0	XBAR2_IN49	OR
FLEXPWM4_PWM3_OUT_TRIG1	XBAR2_IN49	OR
GND	XBAR2_IN50-57	-
PIT1_TRIGGER0	XBAR2_IN58	-
PIT1_TRIGGER1	XBAR2_IN59	-
ADC_ETC0_COCO0	XBAR2_IN60	-
ADC_ETC0_COCO1	XBAR2_IN61	-
ADC_ETC0_COCO2	XBAR2_IN62	-
ADC_ETC0_COCO3	XBAR2_IN63	-
ADC_ETC1_COCO0	XBAR2_IN64	-
ADC_ETC1_COCO1	XBAR2_IN65	-
ADC_ETC1_COCO2	XBAR2_IN66	-
ADC_ETC1_COCO3	XBAR2_IN67	-
GND	XBAR2_IN68-75	-
QDC1_POSMATCH	XBAR2_IN76	-
QDC2_POSMATCH	XBAR2_IN77	-
QDC3_POSMATCH	XBAR2_IN78	-
QDC4_POSMATCH	XBAR2_IN79	-
GND	XBAR2_IN80-81	-
EDMA_DONE0	XBAR2_IN82	-
EDMA_DONE1	XBAR2_IN83	-
EDMA_DONE2	XBAR2_IN84	-
EDMA_DONE3	XBAR2_IN85	-
EDMA_DONE4	XBAR2_IN86	-
EDMA_DONE5	XBAR2_IN87	-
EDMA_DONE6	XBAR2_IN88	-
EDMA_DONE7	XBAR2_IN89	-
EDMA_LPSR_DONE0	XBAR2_IN90	-
EDMA_LPSR_DONE1	XBAR2_IN91	-
EDMA_LPSR_DONE2	XBAR2_IN92	-
EDMA_LPSR_DONE3	XBAR2_IN93	-
EDMA_LPSR_DONE4	XBAR2_IN94	-

Table continues on the next page...

Table 4-6. XBAR2 Input Assignments (continued)

Assigned Input	XBAR2 Input	Gate
EDMA_LPSR_DONE5	XBAR2_IN95	-
EDMA_LPSR_DONE6	XBAR2_IN96	-
EDMA_LPSR_DONE7	XBAR2_IN97	-
GND	XBAR2_IN98-99	-

Table 4-7. XBAR3 Input Assignments

Assigned Input	XBAR3 Input	Gate
LOGIC LOW	XBAR3_IN00	-
LOGIC HIGH	XBAR3_IN01	-
ACMP1_OUT	XBAR3_IN02	-
ACMP2_OUT	XBAR3_IN03	-
ACMP3_OUT	XBAR3_IN04	-
ACMP4_OUT	XBAR3_IN05	-
GND	XBAR3_IN06-09	-
QTIMER1_TIMER0	XBAR3_IN10	-
QTIMER1_TIMER1	XBAR3_IN11	-
QTIMER1_TIMER2	XBAR3_IN12	-
QTIMER1_TIMER3	XBAR3_IN13	-
QTIMER2_TIMER0	XBAR3_IN14	-
QTIMER2_TIMER1	XBAR3_IN15	-
QTIMER2_TIMER2	XBAR3_IN16	-
QTIMER2_TIMER3	XBAR3_IN17	-
QTIMER3_TIMER0	XBAR3_IN18	-
QTIMER3_TIMER1	XBAR3_IN19	-
QTIMER3_TIMER2	XBAR3_IN20	-
QTIMER3_TIMER3	XBAR3_IN21	-
QTIMER4_TIMER0	XBAR3_IN22	-
QTIMER4_TIMER1	XBAR3_IN23	-
QTIMER4_TIMER2	XBAR3_IN24	-
QTIMER4_TIMER3	XBAR3_IN25	-
GND	XBAR3_IN26-33	-
FLEXPWM1_PWM0_OUT_TRIG0	XBAR3_IN34	OR
FLEXPWM1_PWM0_OUT_TRIG1	XBAR3_IN34	OR
FLEXPWM1_PWM1_OUT_TRIG0	XBAR3_IN35	OR
FLEXPWM1_PWM1_OUT_TRIG1	XBAR3_IN35	OR
FLEXPWM1_PWM2_OUT_TRIG0	XBAR3_IN36	OR
FLEXPWM1_PWM2_OUT_TRIG1	XBAR3_IN36	OR
FLEXPWM1_PWM3_OUT_TRIG0	XBAR3_IN37	OR

Table continues on the next page...

Table 4-7. XBAR3 Input Assignments (continued)

Assigned Input	XBAR3 Input	Gate
FLEXPWM1_PWM3_OUT_TRIG1	XBAR3_IN37	OR
FLEXPWM2_PWM0_OUT_TRIG0	XBAR3_IN38	OR
FLEXPWM2_PWM0_OUT_TRIG1	XBAR3_IN38	OR
FLEXPWM2_PWM1_OUT_TRIG0	XBAR3_IN39	OR
FLEXPWM2_PWM1_OUT_TRIG1	XBAR3_IN39	OR
FLEXPWM2_PWM2_OUT_TRIG0	XBAR3_IN40	OR
FLEXPWM2_PWM2_OUT_TRIG1	XBAR3_IN40	OR
FLEXPWM2_PWM3_OUT_TRIG0	XBAR3_IN41	OR
FLEXPWM2_PWM3_OUT_TRIG1	XBAR3_IN41	OR
FLEXPWM3_PWM0_OUT_TRIG0	XBAR3_IN42	OR
FLEXPWM3_PWM0_OUT_TRIG1	XBAR3_IN42	OR
FLEXPWM3_PWM1_OUT_TRIG0	XBAR3_IN43	OR
FLEXPWM3_PWM1_OUT_TRIG1	XBAR3_IN43	OR
FLEXPWM3_PWM2_OUT_TRIG0	XBAR3_IN44	OR
FLEXPWM3_PWM2_OUT_TRIG1	XBAR3_IN44	OR
FLEXPWM3_PWM3_OUT_TRIG0	XBAR3_IN45	OR
FLEXPWM3_PWM3_OUT_TRIG1	XBAR3_IN45	OR
FLEXPWM4_PWM0_OUT_TRIG0	XBAR3_IN46	OR
FLEXPWM4_PWM0_OUT_TRIG1	XBAR3_IN46	OR
FLEXPWM4_PWM1_OUT_TRIG0	XBAR3_IN47	OR
FLEXPWM4_PWM1_OUT_TRIG1	XBAR3_IN47	OR
FLEXPWM4_PWM2_OUT_TRIG0	XBAR3_IN48	OR
FLEXPWM4_PWM2_OUT_TRIG1	XBAR3_IN48	OR
FLEXPWM4_PWM3_OUT_TRIG0	XBAR3_IN49	OR
FLEXPWM4_PWM3_OUT_TRIG1	XBAR3_IN49	OR
GND	XBAR3_IN50-57	-
PIT1_TRIGGER0	XBAR3_IN58	-
PIT1_TRIGGER1	XBAR3_IN59	-
ADC_ETC0_COCO0	XBAR3_IN60	-
ADC_ETC0_COCO1	XBAR3_IN61	-
ADC_ETC0_COCO2	XBAR3_IN62	-
ADC_ETC0_COCO3	XBAR3_IN63	-
ADC_ETC1_COCO0	XBAR3_IN64	-
ADC_ETC1_COCO1	XBAR3_IN65	-
ADC_ETC1_COCO2	XBAR3_IN66	-
ADC_ETC1_COCO3	XBAR3_IN67	-
GND	XBAR3_IN68-75	-
QDC1_POSMATCH	XBAR3_IN76	-
QDC2_POSMATCH	XBAR3_IN77	-

Table continues on the next page...

Table 4-7. XBAR3 Input Assignments (continued)

Assigned Input	XBAR3 Input	Gate
QDC3_POSMATCH	XBAR3_IN78	-
QDC4_POSMATCH	XBAR3_IN79	-
GND	XBAR3_IN80-81	-
EDMA_DONE0	XBAR3_IN82	-
EDMA_DONE1	XBAR3_IN83	-
EDMA_DONE2	XBAR3_IN84	-
EDMA_DONE3	XBAR3_IN85	-
EDMA_DONE4	XBAR3_IN86	-
EDMA_DONE5	XBAR3_IN87	-
EDMA_DONE6	XBAR3_IN88	-
EDMA_DONE7	XBAR3_IN89	-
EDMA_LPSR_DONE0	XBAR3_IN90	-
EDMA_LPSR_DONE1	XBAR3_IN91	-
EDMA_LPSR_DONE2	XBAR3_IN92	-
EDMA_LPSR_DONE3	XBAR3_IN93	-
EDMA_LPSR_DONE4	XBAR3_IN94	-
EDMA_LPSR_DONE5	XBAR3_IN95	-
EDMA_LPSR_DONE6	XBAR3_IN96	-
EDMA_LPSR_DONE7	XBAR3_IN97	-
GND	XBAR3_IN98-99	-

Table 4-8. XBAR1 Output Assignments

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT00	DMA_CH_MUX_REQ_81	-
XBAR1_OUT01	DMA_CH_MUX_REQ_82	-
XBAR1_OUT02	DMA_CH_MUX_REQ_83	-
XBAR1_OUT03	DMA_CH_MUX_REQ_84	-
XBAR1_OUT04	IOMUX_XBAR_INOUT04	-
XBAR1_OUT05	IOMUX_XBAR_INOUT05	-
XBAR1_OUT06	IOMUX_XBAR_INOUT06	-
XBAR1_OUT07	IOMUX_XBAR_INOUT07	-
XBAR1_OUT08	IOMUX_XBAR_INOUT08	-
XBAR1_OUT09	IOMUX_XBAR_INOUT09	-
XBAR1_OUT10	IOMUX_XBAR_INOUT10	-
XBAR1_OUT11	IOMUX_XBAR_INOUT11	-
XBAR1_OUT12	IOMUX_XBAR_INOUT12	-
XBAR1_OUT13	IOMUX_XBAR_INOUT13	-
XBAR1_OUT14	IOMUX_XBAR_INOUT14	-

Table continues on the next page...

Table 4-8. XBAR1 Output Assignments (continued)

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT15	IOMUX_XBAR_INOUT15	-
XBAR1_OUT16	IOMUX_XBAR_INOUT16	-
XBAR1_OUT17	IOMUX_XBAR_INOUT17	-
XBAR1_OUT18	IOMUX_XBAR_INOUT18	-
XBAR1_OUT19	IOMUX_XBAR_INOUT19	-
XBAR1_OUT20	IOMUX_XBAR_INOUT20	-
XBAR1_OUT21	IOMUX_XBAR_INOUT21	-
XBAR1_OUT22	IOMUX_XBAR_INOUT22	-
XBAR1_OUT23	IOMUX_XBAR_INOUT23	-
XBAR1_OUT24	IOMUX_XBAR_INOUT24	-
XBAR1_OUT25	IOMUX_XBAR_INOUT25	-
XBAR1_OUT26	IOMUX_XBAR_INOUT26	-
XBAR1_OUT27	IOMUX_XBAR_INOUT27	-
XBAR1_OUT28	IOMUX_XBAR_INOUT28	-
XBAR1_OUT29	IOMUX_XBAR_INOUT29	-
XBAR1_OUT30	IOMUX_XBAR_INOUT30	-
XBAR1_OUT31	IOMUX_XBAR_INOUT31	-
XBAR1_OUT32	IOMUX_XBAR_INOUT32	-
XBAR1_OUT33	IOMUX_XBAR_INOUT33	-
XBAR1_OUT34	IOMUX_XBAR_INOUT34	-
XBAR1_OUT35	IOMUX_XBAR_INOUT35	-
XBAR1_OUT36	IOMUX_XBAR_INOUT36	-
XBAR1_OUT37	IOMUX_XBAR_INOUT37	-
XBAR1_OUT38	IOMUX_XBAR_INOUT38	-
XBAR1_OUT39	IOMUX_XBAR_INOUT39	-
XBAR1_OUT40	IOMUX_XBAR_INOUT40	-
XBAR1_OUT41	ACMP1_SAMPLE	-
XBAR1_OUT42	ACMP2_SAMPLE	-
XBAR1_OUT43	ACMP3_SAMPLE	-
XBAR1_OUT44	ACMP4_SAMPLE	-
XBAR1_OUT45 - XBAR1_OUT48	Reserved	-
XBAR1_OUT49	FLEXPWM1_PWM0_EXTA	-
XBAR1_OUT50	FLEXPWM1_PWM1_EXTA	-
XBAR1_OUT51	FLEXPWM1_PWM2_EXTA	-
XBAR1_OUT52	FLEXPWM1_PWM3_EXTA	-
XBAR1_OUT53	FLEXPWM1_PWM0_EXT_SYNC	-
XBAR1_OUT54	FLEXPWM1_PWM1_EXT_SYNC	-
XBAR1_OUT55	FLEXPWM1_PWM2_EXT_SYNC	-
XBAR1_OUT56	FLEXPWM1_PWM3_EXT_SYNC	-

Table continues on the next page...

Table 4-8. XBAR1 Output Assignments (continued)

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT57	FLEXPWM1_EXT_CLK	-
XBAR1_OUT58	FLEXPWM1_FAULT0	-
XBAR1_OUT59	FLEXPWM1_FAULT1	-
XBAR_OUT60	FLEXPWM1_FAULT2	OR
	FLEXPWM2_FAULT2	OR
	FLEXPWM3_FAULT2	OR
	FLEXPWM4_FAULT2	OR
XBAR_OUT61	FLEXPWM1_FAULT3	OR
	FLEXPWM2_FAULT3	OR
	FLEXPWM3_FAULT3	OR
	FLEXPWM4_FAULT3	OR
XBAR1_OUT62	FLEXPWM1_EXT_FORCE	-
XBAR1_OUT63	FLEXPWM2_PWM0_EXTA	-
XBAR1_OUT64	FLEXPWM2_PWM1_EXTA	-
XBAR1_OUT65	FLEXPWM2_PWM2_EXTA	-
XBAR1_OUT66	FLEXPWM2_PWM3_EXTA	-
XBAR1_OUT67	FLEXPWM2_PWM0_EXT_SYNC	-
XBAR1_OUT68	FLEXPWM2_PWM1_EXT_SYNC	-
XBAR1_OUT69	FLEXPWM2_PWM2_EXT_SYNC	-
XBAR1_OUT70	FLEXPWM2_PWM3_EXT_SYNC	-
XBAR1_OUT71	FLEXPWM2_EXT_CLK	-
XBAR1_OUT72	FLEXPWM2_FAULT0	-
XBAR1_OUT73	FLEXPWM2_FAULT1	-
XBAR1_OUT74	FLEXPWM2_EXT_FORCE	-
XBAR1_OUT75	FLEXPWM3_PWM0_EXTA	OR
	FLEXPWM4_PWM0_EXTA	OR
XBAR1_OUT76	FLEXPWM3_PWM1_EXTA	OR
	FLEXPWM4_PWM1_EXTA	OR
XBAR1_OUT77	FLEXPWM3_PWM2_EXTA	OR
	FLEXPWM4_PWM2_EXTA	OR
XBAR1_OUT78	FLEXPWM3_PWM3_EXTA	OR
	FLEXPWM4_PWM3_EXTA	OR
XBAR1_OUT79	FLEXPWM3_EXT_CLK	OR
	FLEXPWM4_EXT_CLK	OR
XBAR1_OUT80	FLEXPWM3_PWM0_EXT_SYNC	-
XBAR1_OUT81	FLEXPWM3_PWM1_EXT_SYNC	-
XBAR1_OUT82	FLEXPWM3_PWM2_EXT_SYNC	-
XBAR1_OUT83	FLEXPWM3_PWM3_EXT_SYNC	-
XBAR1_OUT84	FLEXPWM3_FAULT0	-

Table continues on the next page...

Table 4-8. XBAR1 Output Assignments (continued)

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT85	FLEXPWM3_FAULT1	-
XBAR1_OUT86	FLEXPWM3_EXT_FORCE	-
XBAR1_OUT87	FLEXPWM4_PWM0_EXT_SYNC	-
XBAR1_OUT88	FLEXPWM4_PWM1_EXT_SYNC	-
XBAR1_OUT89	FLEXPWM4_PWM2_EXT_SYNC	-
XBAR1_OUT90	FLEXPWM4_PWM3_EXT_SYNC	-
XBAR1_OUT91	FLEXPWM4_FAULT0	-
XBAR1_OUT92	FLEXPWM4_FAULT1	-
XBAR1_OUT93	FLEXPWM4_EXT_FORCE	-
XBAR1_OUT94 -XBAR1_OUT107	Reserved	-
XBAR1_OUT108	QDC1_PHASEA	-
XBAR1_OUT109	QDC1_PHASEB	-
XBAR1_OUT110	QDC1_INDEX	-
XBAR1_OUT111	QDC1_HOME	-
XBAR1_OUT112	QDC1_TRIGGER	-
XBAR1_OUT113	QDC2_PHASEA	-
XBAR1_OUT114	QDC2_PHASEB	-
XBAR1_OUT115	QDC2_INDEX	-
XBAR1_OUT116	QDC2_HOME	-
XBAR1_OUT117	QDC2_TRIGGER	-
XBAR1_OUT118	QDC3_PHASEA	-
XBAR1_OUT119	QDC3_PHASEB	-
XBAR1_OUT120	QDC3_INDEX	-
XBAR1_OUT121	QDC3_HOME	-
XBAR1_OUT122	QDC3_TRIGGER	-
XBAR1_OUT123	QDC4_PHASEA	-
XBAR1_OUT124	QDC4_PHASEB	-
XBAR1_OUT125	QDC4_INDEX	-
XBAR1_OUT126	QDC4_HOME	-
XBAR1_OUT127	QDC4_TRIGGER	-
XBAR1_OUT128 - XBAR1_OUT131	Reserved	-
XBAR1_OUT132	CAN1	CAN1 external timer tick
XBAR1_OUT133	CAN2	CAN2 external timer tick
XBAR1_OUT134 - XBAR1_OUT137	Reserved	-
XBAR1_OUT138	QTIMER1_TIMER0	Selectable by setting IOMUXC_GPR_GPR12[QTIMER1_TRM0_INPUT_SEL]
XBAR1_OUT139	QTIMER1_TIMER1	Selectable by setting IOMUXC_GPR_GPR12[QTIMER1_TRM1_INPUT_SEL]

Table continues on the next page...

Table 4-8. XBAR1 Output Assignments (continued)

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT140	QTIMER1_TIMER2	Selectable by setting IOMUXC_GPR_GPR12[QTIMER1_TRM2_INPUT_SEL]
XBAR1_OUT141	QTIMER1_TIMER3	Selectable by setting IOMUXC_GPR_GPR12[QTIMER1_TRM3_INPUT_SEL]
XBAR1_OUT142	QTIMER2_TIMER0	Selectable by setting IOMUXC_GPR_GPR13[QTIMER2_TRM0_INPUT_SEL]
XBAR1_OUT143	QTIMER2_TIMER1	Selectable by setting IOMUXC_GPR_GPR13[QTIMER2_TRM1_INPUT_SEL]
XBAR1_OUT144	QTIMER2_TIMER2	Selectable by setting IOMUXC_GPR_GPR13[QTIMER2_TRM2_INPUT_SEL]
XBAR1_OUT145	QTIMER2_TIMER3	Selectable by setting IOMUXC_GPR_GPR13[QTIMER2_TRM3_INPUT_SEL]
XBAR1_OUT146	QTIMER3_TIMER0	Selectable by setting IOMUXC_GPR_GPR14[QTIMER3_TRM0_INPUT_SEL]
XBAR1_OUT147	QTIMER3_TIMER1	Selectable by setting IOMUXC_GPR_GPR14[QTIMER3_TRM1_INPUT_SEL]
XBAR1_OUT148	QTIMER3_TIMER2	Selectable by setting IOMUXC_GPR_GPR14[QTIMER3_TRM2_INPUT_SEL]
XBAR1_OUT149	QTIMER3_TIMER3	Selectable by setting IOMUXC_GPR_GPR14[QTIMER3_TRM3_INPUT_SEL]
XBAR1_OUT150	QTIMER4_TIMER0	Selectable by setting IOMUXC_GPR_GPR15[QTIMER4_TRM0_INPUT_SEL]
XBAR1_OUT151	QTIMER4_TIMER1	Selectable by setting IOMUXC_GPR_GPR15[QTIMER4_TRM1_INPUT_SEL]
XBAR1_OUT152	QTIMER4_TIMER2	Selectable by setting IOMUXC_GPR_GPR15[QTIMER4_TRM2_INPUT_SEL]
XBAR1_OUT153	QTIMER4_TIMER3	Selectable by setting IOMUXC_GPR_GPR15[QTIMER4_TRM3_INPUT_SEL]
XBAR1_OUT154	EWM_IN	-
XBAR1_OUT155	ADC_ETC0_TRIG0	-
XBAR1_OUT156	ADC_ETC0_TRIG1	-
XBAR1_OUT157	ADC_ETC0_TRIG2	-
XBAR1_OUT158	ADC_ETC0_TRIG3	-

Table continues on the next page...

Table 4-8. XBAR1 Output Assignments (continued)

XBAR1 Output	Assigned Output	Gate
XBAR1_OUT159	ADC_ETC1_TRIG0	-
XBAR1_OUT160	ADC_ETC1_TRIG1	-
XBAR1_OUT161	ADC_ETC1_TRIG2	-
XBAR1_OUT162	ADC_ETC1_TRIG3	-
XBAR1_OUT163-170	Reserved	-
XBAR1_OUT171	FLEXIO1_TRIG_IN0	-
XBAR1_OUT172	FLEXIO1_TRIG_IN1	-
XBAR1_OUT173	FLEXIO2_TRIG_IN0	-
XBAR1_OUT174	FLEXIO2_TRIG_IN1	-
XBAR1_OUT175-176	Reserved	-

Table 4-9. XBAR2 Output Assignments

XBAR2 Output	Assigned Output	Gate
XBAR2_OUT00	AOI1_IN0	-
XBAR2_OUT01	AOI1_IN1	-
XBAR2_OUT02	AOI1_IN2	-
XBAR2_OUT03	AOI1_IN3	-
XBAR2_OUT04	AOI1_IN4	-
XBAR2_OUT05	AOI1_IN5	-
XBAR2_OUT06	AOI1_IN6	-
XBAR2_OUT07	AOI1_IN7	-
XBAR2_OUT08	AOI1_IN8	-
XBAR2_OUT09	AOI1_IN9	-
XBAR2_OUT10	AOI1_IN10	-
XBAR2_OUT11	AOI1_IN11	-
XBAR2_OUT12	AOI1_IN12	-
XBAR2_OUT13	AOI1_IN13	-
XBAR2_OUT14	AOI1_IN14	-
XBAR2_OUT15	AOI1_IN15	-

Table 4-10. XBAR3 Output Assignments

XBAR3 Output	Assigned Output	Gate
XBAR3_OUT0	AOI2_IN0	-
XBAR3_OUT1	AOI2_IN1	-
XBAR3_OUT2	AOI2_IN2	-
XBAR3_OUT3	AOI2_IN3	-
XBAR3_OUT4	AOI2_IN4	-

Table continues on the next page...

Table 4-10. XBAR3 Output Assignments (continued)

XBAR3 Output	Assigned Output	Gate
XBAR3_OUT5	AOI2_IN5	-
XBAR3_OUT6	AOI2_IN6	-
XBAR3_OUT7	AOI2_IN7	-
XBAR3_OUT8	AOI2_IN8	-
XBAR3_OUT9	AOI2_IN9	-
XBAR3_OUT10	AOI2_IN10	-
XBAR3_OUT11	AOI2_IN11	-
XBAR3_OUT12	AOI2_IN12	-
XBAR3_OUT13	AOI2_IN13	-
XBAR3_OUT14	AOI2_IN14	-
XBAR3_OUT15	AOI2_IN15	-

Chapter 5
Direct Memory Access Multiplexer (DMAMUX)

5.1 Chip-specific DMAMUX information

The chip has two DMA controllers and correspondingly two DMAMUX. EDMA is in CM7 domain and co-works with CM7, EDMA_LPSR is in CM4 domain and co-works with CM4. DMA_CH_MUX serves EDMA and DMA_CH_MUX_LPSR serves EDMA_LPSR.

Table 5-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
DMA Mux Mapping	DMAMUX	DMA Mux

5.2 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 32 DMA channels.

5.2.1 Block diagram

The following figure illustrates the block diagram of DMAMUX.

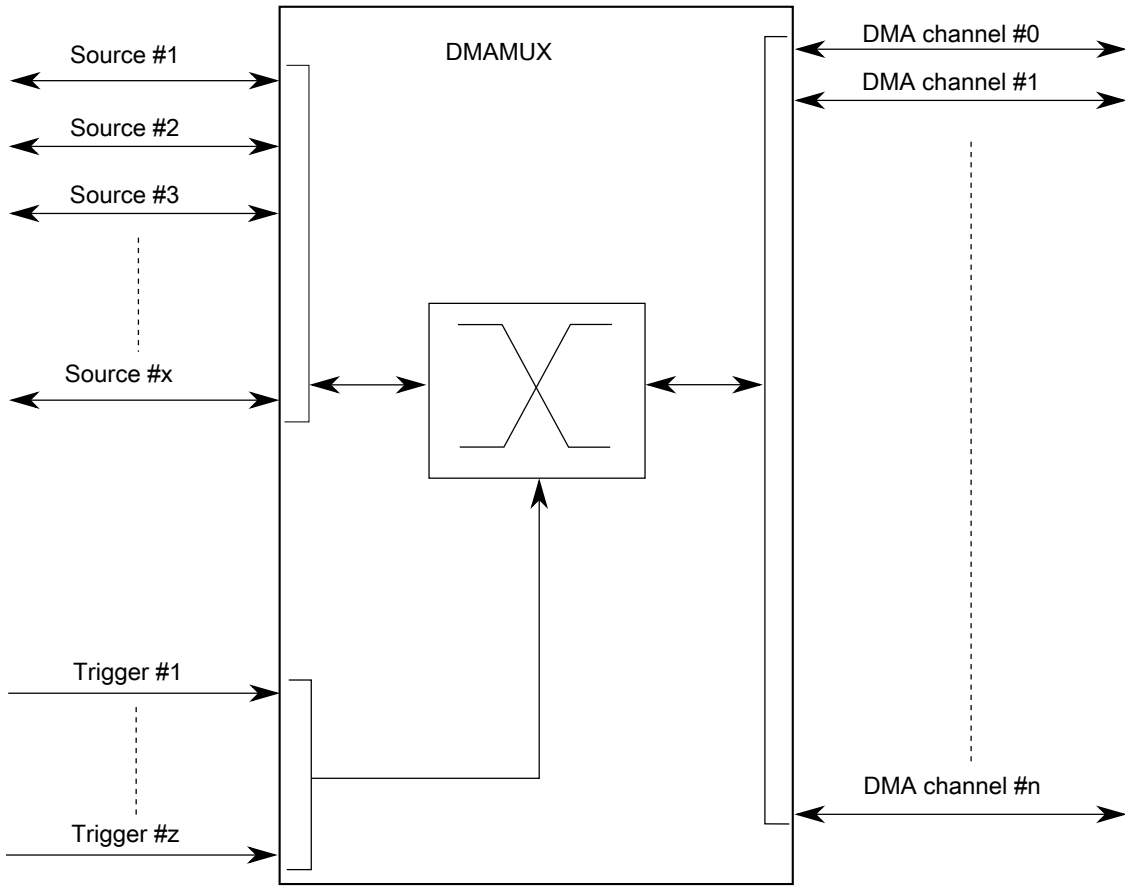


Figure 5-1. DMAMUX block diagram

5.2.2 Features

The DMAMUX module provides these features:

- Up to 208 peripheral slots can be routed to 32 channels.
- 32 independently selectable DMA channel routers.
  - Each channel output can be individually configured to be Always On and not depend on any of the peripheral slots.
  - The first 4 channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots.
- On every memory map configuration change for a any channel, this module signals to the DMA Controller to reset the internal state machine for that channel and it can accept a new request based on the new configuration.

5.3 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels.

As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

5.3.1 Modes of operation

The following operating modes are available:

- Disabled mode

  In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically.

  Configuration of the period is done in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0 to 3.

5.3.2 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 4 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention.

The trigger is generated by the periodic interrupt timer (PIT); as such, the configuration of the periodic triggering interval is done via configuration registers in the PIT. See the section on periodic interrupt timer for more information on this topic.

Note

Because of the dynamic nature of the system (due to DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

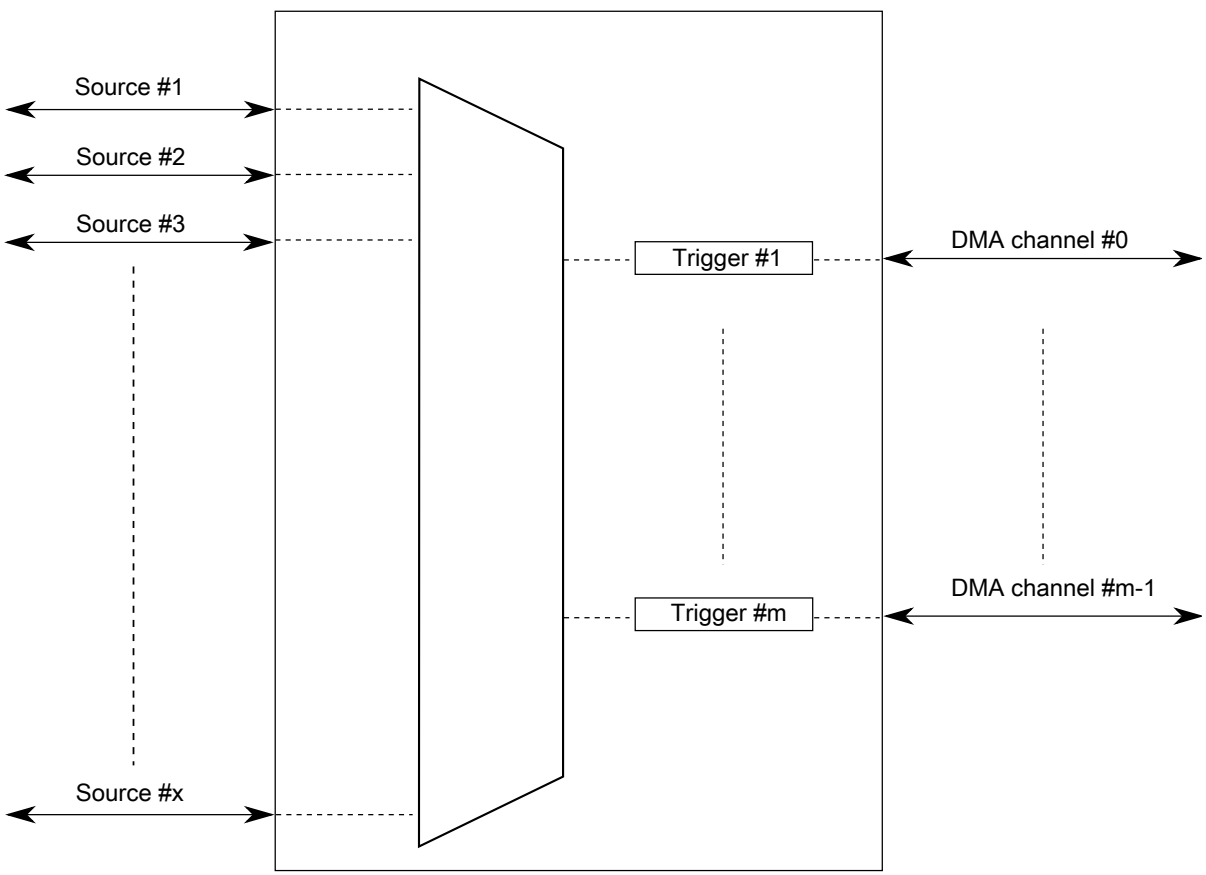


Figure 5-2. DMAMUX triggered channels

The DMA channel triggering capability allows the system to schedule regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

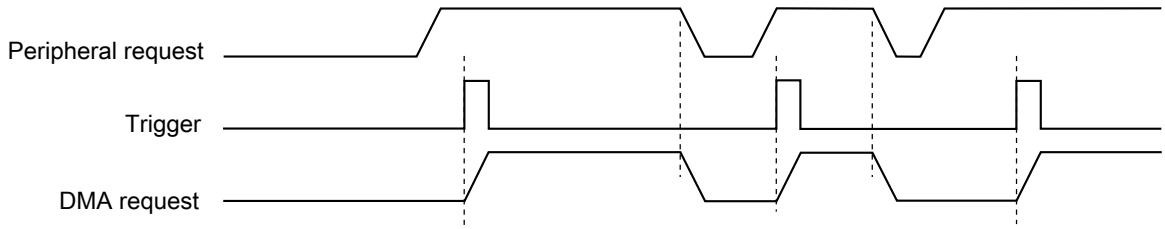


Figure 5-3. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

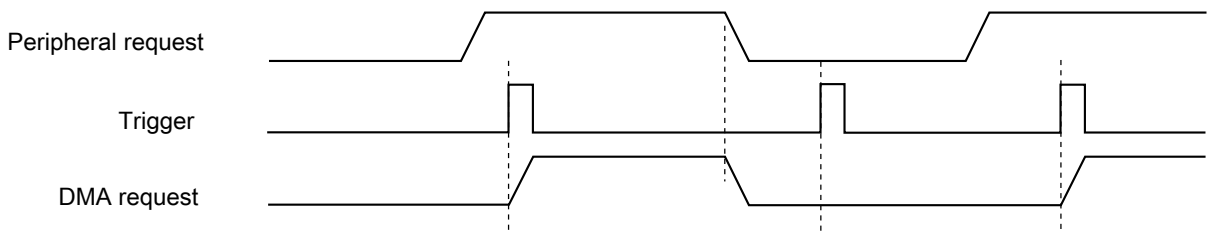


Figure 5-4. DMAMUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

  As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5 µs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

5.3.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, each DMA Channel can be individually configured to function as an Always Enabled source. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the always enabled channel provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA channel can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use explicit software reactivation.

  In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use an always-enabled DMA source.

  In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and configured as "always enabled" channel. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

NOTE

When a channel is configured as "Always Enabled", then the peripheral DMA sources for that channel are ignored; i.e. SOURCE field has no effect.

5.3.4 Clocks

The following table describes the clock sources for DMAMUX. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 5-2. DMAMUX clock

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock

5.3.5 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

5.4 External signals

The DMAMUX has no external pins.

5.5 Application information

This section provides instructions for initializing the DMA channel MUX.

5.5.1 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00000000 to CHCFG1.
2. Configure channel 1 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC0000005 to CHCFG1.

The following code example illustrates steps 1 and 4 above:

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

To enable a source without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is setwhile CHCFG[TRIG] is cleared.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1 with no periodic triggering capability :

1. Write 0x00000000 to CHCFG1.
2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x80000005 to CHCFG1.

The following code example illustrates steps 1 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0x40021000/* Example only ! */
/* Following example assumes long is 32-bits */
volatile unsigned long *CHCFG0 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned long *CHCFG1 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned long *CHCFG2 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned long *CHCFG3 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned long *CHCFG4 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0010);
volatile unsigned long *CHCFG5 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0014);
volatile unsigned long *CHCFG6 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0018);
volatile unsigned long *CHCFG7 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x001C);
volatile unsigned long *CHCFG8 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0020);
volatile unsigned long *CHCFG9 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0024);
volatile unsigned long *CHCFG10= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0028);
volatile unsigned long *CHCFG11= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x002C);
volatile unsigned long *CHCFG12= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0030);
volatile unsigned long *CHCFG13= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0034);
volatile unsigned long *CHCFG14= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0038);
volatile unsigned long *CHCFG15= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x003C);

In File main.c:
#include "registers.h"
:
:
*CHCFG1 = 0x00000000;
*CHCFG1 = 0x80000005;
```

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00000000 to CHCFG8.
3. Write 0x80000007 to CHCFG8. (In this example, setting CHCFG[TRIG] would have no effect due to the assumption that channel 8 does not support the periodic triggering functionality.)

The following code example illustrates steps 2 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0x40021000/* Example only ! */
/* Following example assumes long is 32-bits */
volatile unsigned long *CHCFG0 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned long *CHCFG1 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned long *CHCFG2 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned long *CHCFG3 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned long *CHCFG4 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0010);
volatile unsigned long *CHCFG5 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0014);
volatile unsigned long *CHCFG6 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0018);
volatile unsigned long *CHCFG7 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x001C);
volatile unsigned long *CHCFG8 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0020);
volatile unsigned long *CHCFG9 = (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0024);
volatile unsigned long *CHCFG10= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0028);
volatile unsigned long *CHCFG11= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x002C);
volatile unsigned long *CHCFG12= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0030);
volatile unsigned long *CHCFG13= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0034);
volatile unsigned long *CHCFG14= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x0038);
volatile unsigned long *CHCFG15= (volatile unsigned long *) (DMAMUX_BASE_ADDR+0x003C);

In File main.c:
#include "registers.h"
:
:
*CHCFG8 = 0x00000000;
*CHCFG8 = 0x80000007;
```

5.6 Initialization

5.6.1 Configuration options

Table 5-3. Channel Configuration Options

ENBL	TRIG	A_ON	Function	Mode
0	X	X	DMA channel is disabled	Disabled Mode
1	0	0	DMA channel is enabled with no triggering (transparent)	Normal Mode
1	1	0	DMA channel is enabled with triggering	Periodic Trigger Mode
1	0	1	DMA channel is always enabled	Always On Mode
1	1	1	DMA channel is always enabled with triggering	Always On Trigger Mode

5.7 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

5.7.1 DMAMUX register descriptions

5.7.1.1 DMAMUX memory map

DMAMUX0 base address: 4007_4000h

DMAMUX1 base address: 40C1_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0 - 7C	Channel a Configuration Register (CHCFG0 - CHCFG31)	32	RW	0000_0000

5.7.1.2 Channel a Configuration Register (CHCFG0 - CHCFG31)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

NOTE

Setting multiple CHCFG registers with the same source value will result in unpredictable behavior. This is true, even if a channel is disabled (ENBL==0).

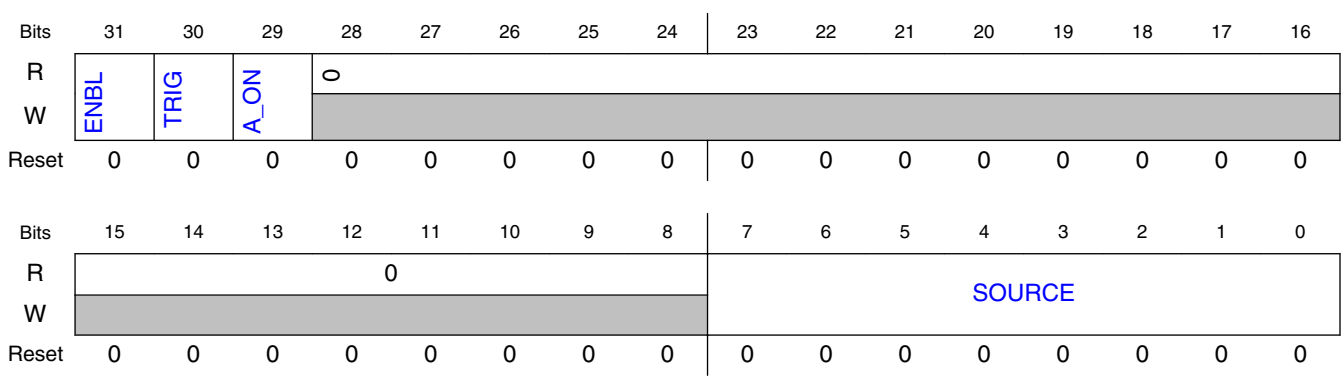
Before changing the trigger or source settings, a DMA channel must be disabled via CHCFGn[ENBL].

5.7.1.2.1 Offset

For a = 0 to 31:

Register	Offset
CHCFGa	0h + (a × 4h)

5.7.1.2.2 Diagram



5.7.1.2.3 Fields

Field	Description
31 ENBL	DMA Mux Channel Enable Enables the channel for DMA Mux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 0 - DMA Mux channel is disabled

Table continues on the next page...

Field	Description
	1 - DMA Mux channel is enabled
30 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0 - Triggering is disabled. If triggering is disabled and ENBL is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode) 1 - Triggering is enabled. If triggering is enabled and ENBL is set, the DMA_CH_MUX is in Periodic Trigger mode.
29 A_ON	DMA Channel Always Enable Enables the DMA Channel to be always ON. If TRIG bit is set, the module will assert request on every trigger. 0 - DMA Channel Always ON function is disabled 1 - DMA Channel Always ON function is enabled
28-8 —	Reserved field
7-0 SOURCE	DMA Channel Source (Slot Number) Specifies which DMA source, if any, is routed to a particular DMA channel. See the chip-specific DMA_CH_MUX information for details about the peripherals and their slot numbers.

Chapter 6
Enhanced Direct Memory Access (eDMA)

6.1 Chip-specific eDMA information

Table 6-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

6.2 Overview

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 32 channels

6.2.1 Block diagram

The following figure illustrates the components of the eDMA system, including the eDMA module ("engine").

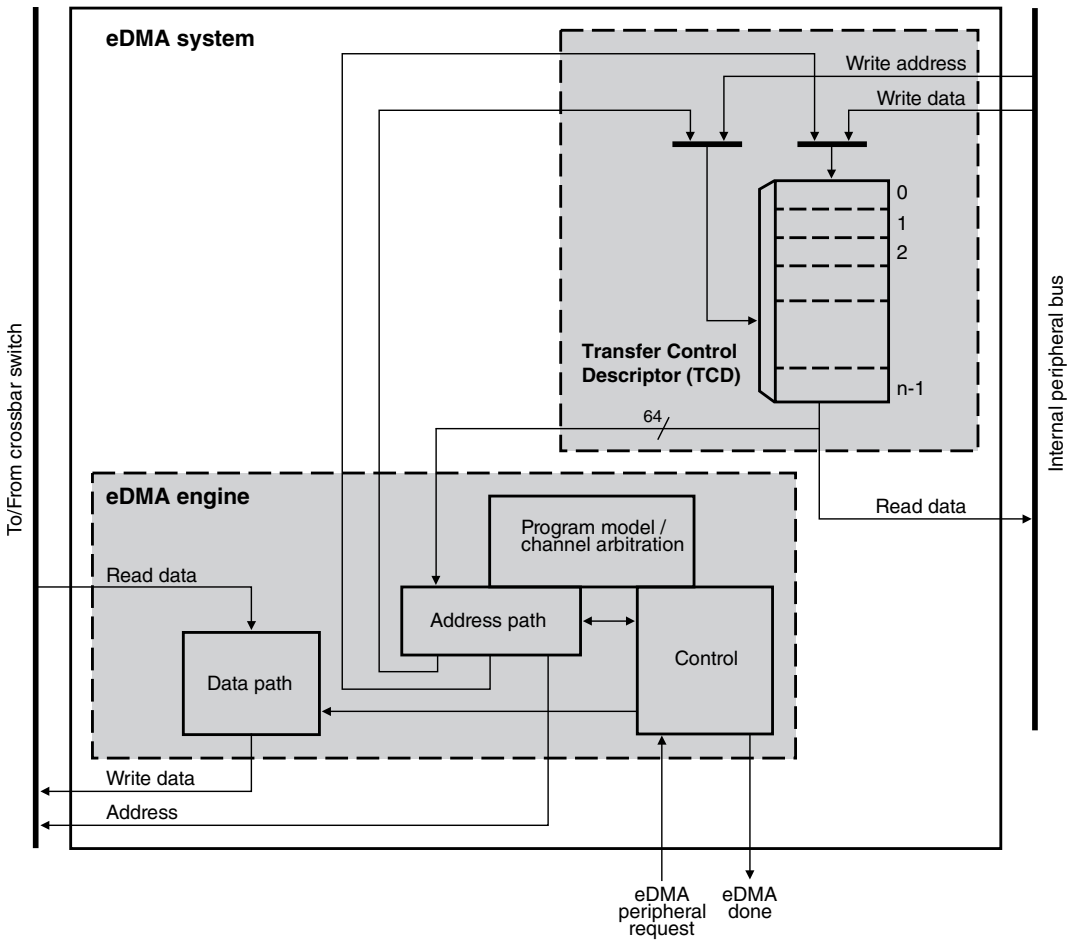


Figure 6-1. eDMA system block diagram

6.2.2 Block parts

The eDMA module comprises two major modules: the eDMA engine and the transfer-control descriptor local memory.

Table 6-2 describes the eDMA engine submodules.

Table 6-2. eDMA engine submodules

Submodule	Function
Address path	The address path block: • Provides registered versions of two channel Transfer Control Descriptors (TCDs)—channel x (normal start) and channel y (preemption start) • Manages all master bus-address calculations

Table continues on the next page...

Table 6-2. eDMA engine submodules (continued)

Submodule	Function
	All channels provide the same functionality. This structure enables preemption of data transfers associated with an active channel (after completion of a read/write sequence) if the eDMA engine asserts a higher priority channel activation. After eDMA activates a channel, it runs until the minor loop completes, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI*n*[ECP]) in which the eDMA engine can preempt a large data move operation to minimize the time another channel stalls. When the eDMA engine selects a channel to execute, it reads the contents of the channel TCD from local memory and loads it into one of the following: • The address path channel x registers (normal start) • The address path channel y registers (preemption start) After the minor loop execution completes, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count completes, the eDMA engine performs additional processing, including: • Final address pointer updates • Reloading the TCD*n*_CITER field • A possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	The data path block implements the bus master read/write data path. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the two-stage pipelined internal bus. The address path module represents the first stage of the bus pipeline (address phase). The data path module implements the second stage of the pipeline (data phase).
Programming model/ channel arbitration	This block implements: • The first section of the eDMA programming model • Channel arbitration logic The programming model registers connect to the chip's internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs also connect to this block (via control logic).
Control	The control block provides all control functions for the eDMA engine. For data transfers in which the source size (SSIZE) and destination size (DSIZE) are equal, the eDMA engine performs a series of source read/destination write operations until it has transferred the number of bytes specified in the minor loop byte count (NBYTES). For TCDs in which the source and destination sizes are not equal, the eDMA engine executes multiple accesses of the smaller size data for each reference of the larger size. For example, if the source size (SSIZE) references 16-bit data and the destination size (DSIZE) is 32-bit data, eDMA performs two reads, then one 32-bit write.

Table 6-3 explains the partitioning of the TCD local memory.

Table 6-3. Transfer control descriptor memory

Submodule	Description
Memory controller	The memory controller logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. If simultaneous accesses occur, the eDMA engine receives priority and the peripheral transaction stalls.
Memory array	The memory array provides TCD storage for the transfer profile for each channel.

6.2.3 Features

The eDMA module is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. Use it for applications where you statically know the size of the data to be transferred and do not define the size within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for enhanced addressing modes
- 32-channel implementation performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage to support 16- and 32-byte transfers
  - Connections to the crossbar switch (AXBS) for bus mastering the data movement
- TCD supports two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion notification via programmable interrupt requests
  - One interrupt per channel. eDMA engine can generate an interrupt when major iteration count completes
  - Programmable error terminations per channel and logically summed together to form one error interrupt to the interrupt controller

- Programmable support for scatter/gather DMA processing
- Support for complex data structures

NOTE

In the discussion of this module, *n* is the channel number.

6.3 Functional description

The operation of the eDMA is described in the following subsections.

6.3.1 Modes of operation

eDMA operates in the following modes:

Table 6-4. Modes of operation

Mode	Description
Normal	In Normal mode, eDMA transfers data from a source to a destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A *service request* initiates a transfer of a specific number of bytes (NBYTES) as specified in the TCD. • The *minor loop* is the sequence of read and write operations that transfers the NBYTES of data for a service request. • Each service request executes one iteration of the major loop, transferring NBYTES of data.
Debug	DMA operation is configurable in Debug mode via Control (CR) • If CR[EDBG] = 0, eDMA continues to operate normally when the chip is in debug mode. • If CR[EDBG] = 1, eDMA stops transferring data when the chip enters debug mode. If a channel is active when eDMA enters Debug mode, eDMA continues operation until the channel retires.
Wait	Before entering Wait mode, eDMA attempts to complete any transfer that is in progress. After the transfer completes, the chip enters Wait mode.

6.3.2 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

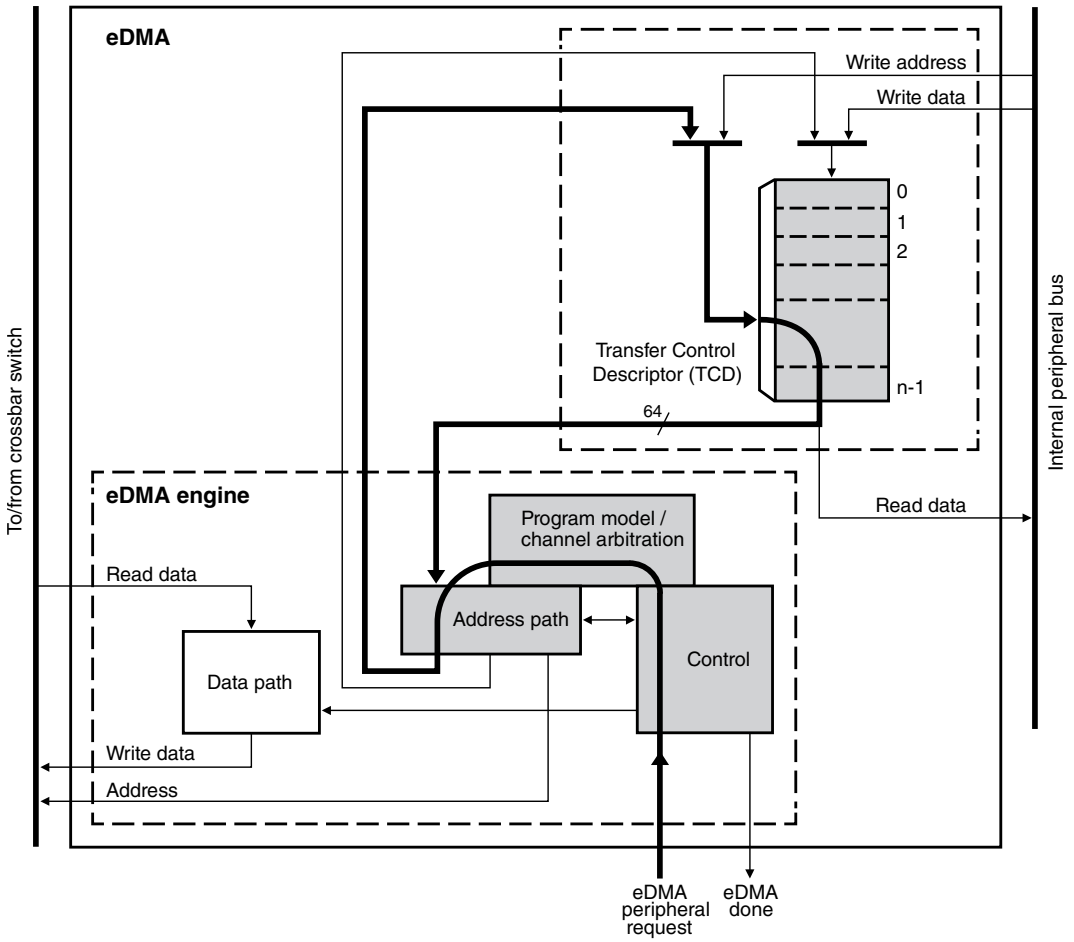


Figure 6-2. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration executes, using either the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the eDMA engine's internal register file. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the internal register file.

The following diagram illustrates the second part of the basic data flow:

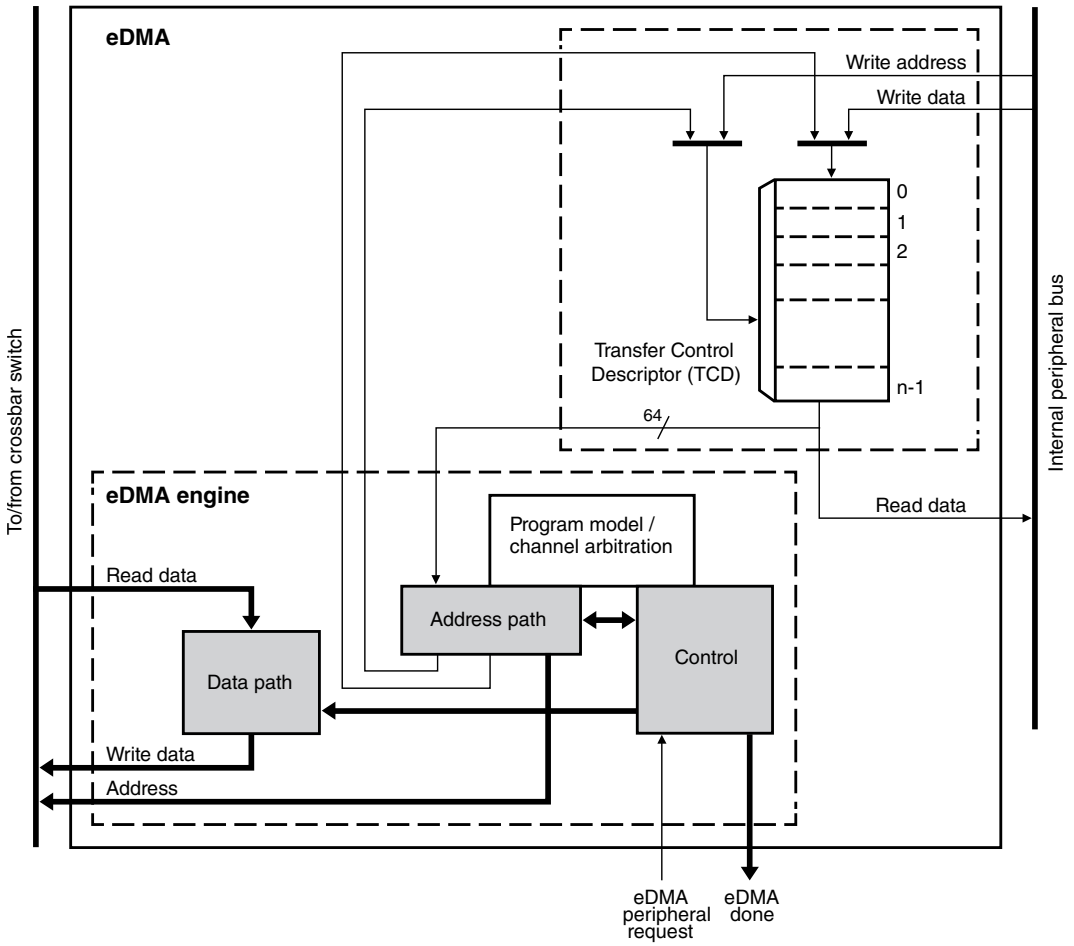


Figure 6-3. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) execute sequentially through the required source reads and destination writes to perform the data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, for example, SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

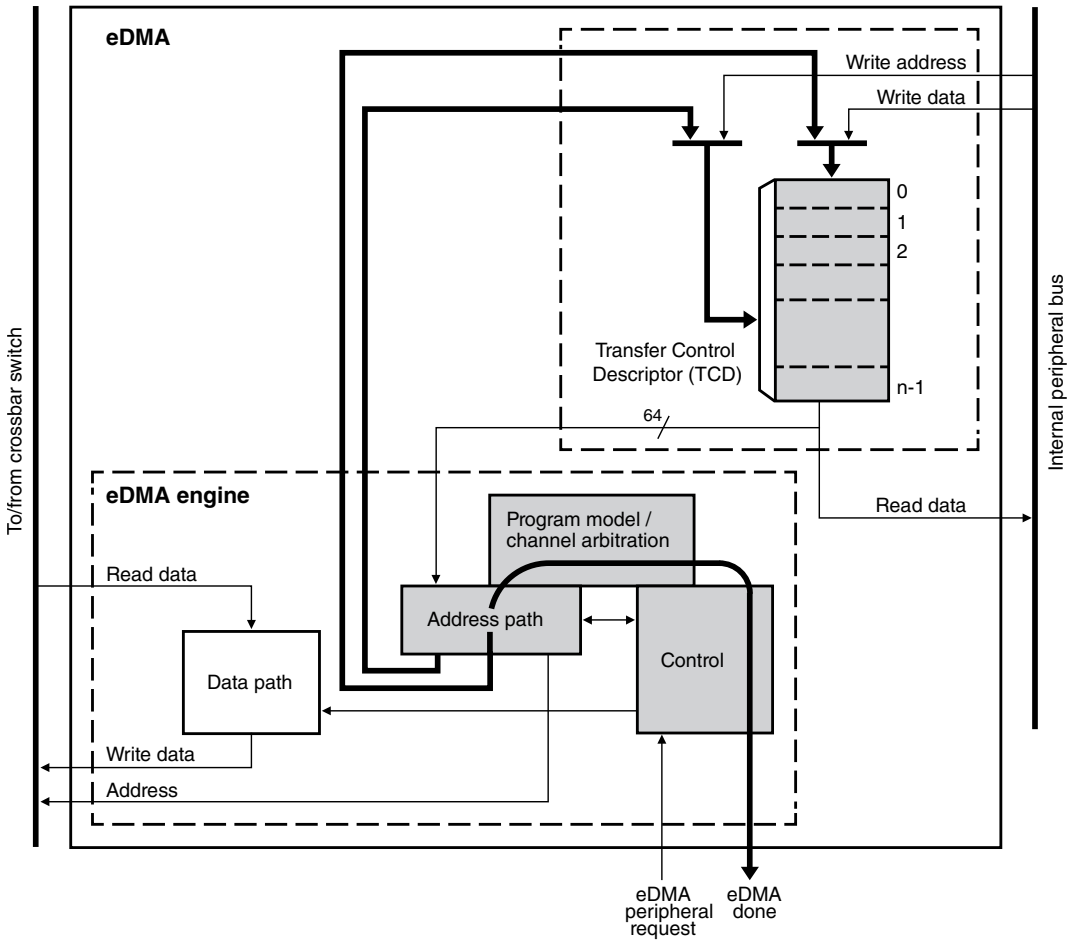


Figure 6-4. eDMA operation, part 3

6.3.3 Fault reporting and handling

Channel errors are reported in the Error Status register (DMAx_ES) and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes is detailed below:

- The addresses and offsets must be aligned on 0-modulo transfer size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.

- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal. All channel priority levels must be unique when fixed arbitration mode is enabled.

NOTE

When two channels have the same priority, a channel priority error exists and is reported in the Error Status register. However, the channel number is not reported in the Error Status register. When all of the channel priorities within a group are not unique, the channel number selected by arbitration is undetermined.

To aid in Channel Priority Error (CPE) debug, set the Halt On Error bit in the DMA's Control register. If all channel priorities within a group are not unique, the DMA is halted after the CPE error is recorded. The DMA remains halted and does not process any channel service requests. After all of the channel priorities are set to unique numbers, the DMA may be enabled again by clearing the HALT bit.

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[ELINK] bit does not equal the TCDn_BITER[ELINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion, when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the

data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

A transfer may be canceled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the Error Status register (DMAx_ES) is updated with the canceled channel number and ECX is set. The TCD of a canceled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is canceled by the error cancel transfer mechanism, the channel number is loaded into DMA_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

NOTE

The cancel transfer request enables you to stop a large data transfer when the full data transfer is no longer needed. The cancel transfer bit does not abort the channel. It simply stops the transferring of data and then retires the channel through its normal shutdown sequence. The application software must manage the context of the cancel. If an interrupt is desired (or not), then the interrupt should be enabled (or disabled) before the cancel request. The application software must clean up the transfer control descriptor because the full transfer did not occur.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the Error Status register (DMAx_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

6.3.4 Channel preemption

Channel preemption is enabled on a per-channel basis by setting DCHPRIn[ECP]. Channel preemption enables the executing channel's data transfers to temporarily be suspended in favor of starting a higher priority channel. After the preempting channel has completed its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preemption ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This enables a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

6.3.5 Clocks

The following table describes the clock sources for eDMA. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 6-5. eDMA clocks

Clock name	Description
edma_hclk	Module clock
ipg_clk	Peripheral clock

6.4 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

6.4.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI*n* registers if a configuration other than the default is desired.
3. Enable error interrupts in the EEI register if desired.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the ERQ register.
6. Request channel service via either:
   - Software: setting TCD*n*_CSR[START]
   - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels in the programming model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 6-6, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the system bus, unless a configuration error is detected. Transfers from the source, as defined by TCD*n*_SADDR, to the destination, as defined by TCD*n*_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES have been transferred.

When the transfer is complete, the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post-processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 6-6. TCD control and status fields

TCD*n*_CSR field name	Description
START	Control bit to start channel explicitly when using a software-initiated DMA service (automatically cleared by hardware)
ACTIVE	Status bit indicating the channel is currently in execution
DONE	Status bit indicating major loop completion (cleared by software when using a software-initiated DMA service)
DREQ	Control bit to disable DMA request at end of major loop completion when using a hardware-initiated DMA service
BWC	Control bits for throttling bandwidth control of a channel
ESG	Control bit to enable scatter/gather feature
INTHALF	Control bit to enable interrupt when major loop is half complete
INTMAJOR	Control bit to enable interrupt when major loop completes

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

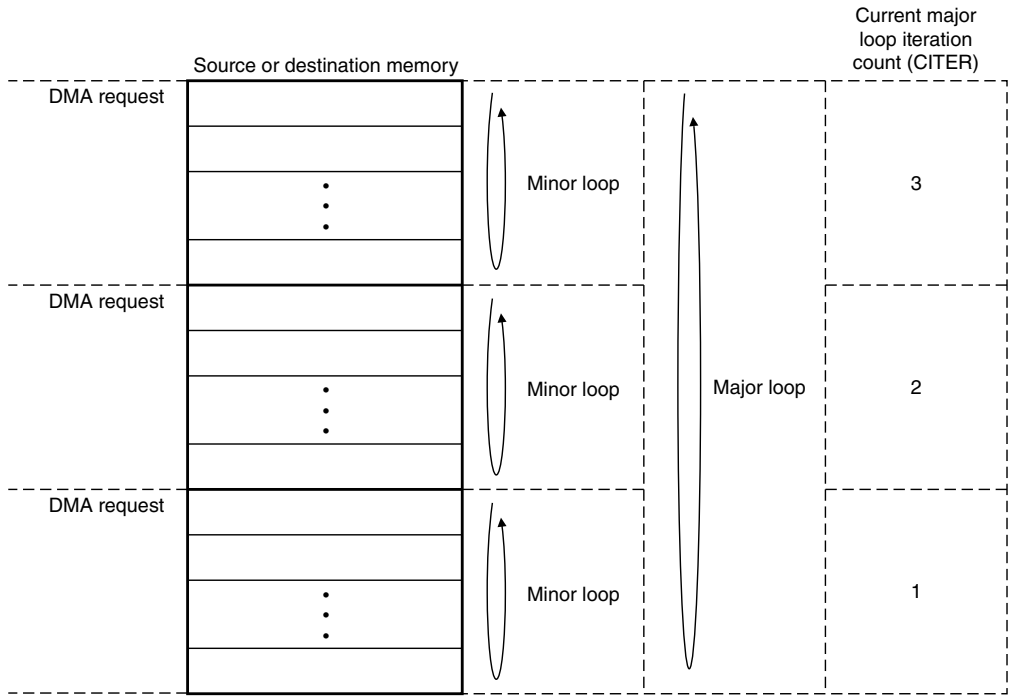


Figure 6-5. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

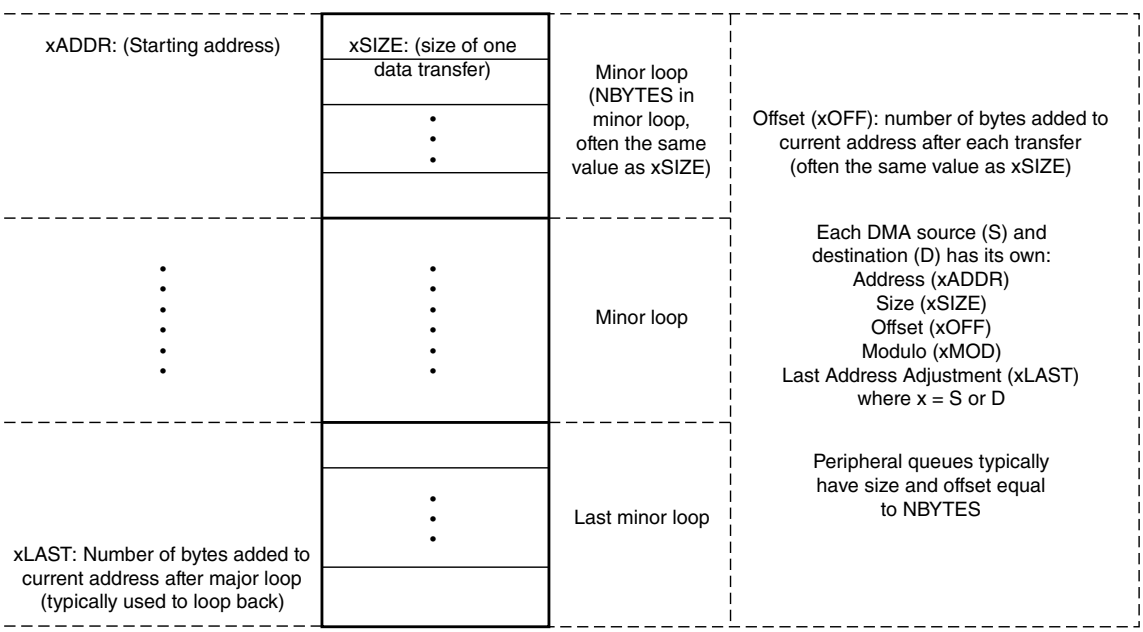


Figure 6-6. Memory array terms

6.4.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per-channel basis with the exception of channel priority error (ES[CPE]).

For all error types other than group or channel priority errors, the channel number causing the error is recorded in the Error Status register (DMAx_ES). If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

Channel priority errors are identified within a group after that group has been selected as the active group. For example:

1. The eDMA is configured for fixed group and fixed channel arbitration modes.
2. Group 1 is the highest priority and all channels are unique in that group.
3. Group 0 is the next highest priority and has two channels with the same priority level.
4. If Group 1 has any service requests, those requests will be executed.
5. After all of Group 1 requests have completed, Group 0 will be the next active group.
6. If Group 0 has a service request, then an undefined channel in Group 0 will be selected and a channel priority error will occur.
7. This repeats until the all of Group 0 requests have been removed or a higher priority Group 1 request comes in.

In this sequence, for item 2, the eDMA acknowledge lines will assert only if the selected channel is requesting service via the eDMA peripheral request signal. If interrupts are enabled for all channels, the user will get an error interrupt, but the channel number for the ERR register and the error interrupt request line may be wrong because they reflect the selected channel. A group priority error is global and any request in any group will cause a group priority error.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel/group priority with an active request is selected, but the lowest numbered channel with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting are associated with the selected channel.

6.4.3 Arbitration mode considerations

This section discusses arbitration considerations for the eDMA.

6.4.3.1 Fixed group arbitration, Fixed channel arbitration

In this mode, the channel service request from the highest priority channel in the highest priority group is selected to execute. If the eDMA is programmed so that the channels within one group use "fixed" priorities, and that group is assigned the highest "fixed" priority of all groups, that group can take all the bandwidth of the eDMA controller. That is, no other groups will be serviced if there is always at least one DMA request pending on a channel in the highest priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly. Preemption is available in this scenario only.

6.4.3.2 Fixed group arbitration, Round-robin channel arbitration

The highest priority group with a request will be serviced. Lower priority groups will be serviced if no pending requests exist in the higher priority groups.

Within each group, c hannels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels assigned within the group.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, Fixed channel arbitration, but all the channels in the highest priority group will be serviced. Service latency will be short on the highest priority group, but could potentially be very much longer as the group priority decreases.

6.4.4 DMA transfer examples

This section presents examples of how to perform DMA transfers with the eDMA.

6.4.4.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, TCD*n*_CSR[DONE] is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has an 8-bit memory port located at 0x1000. The

destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJOR] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] bit requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32 bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[*n*] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

6.4.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop, transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, the eDMA peripheral, requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32 bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0.
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:

a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
b. Write 32 bits to location 0x2010 → first iteration of the minor loop.
c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
d. Write 32 bits to location 0x2014 → second iteration of the minor loop.
e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
f. Write 32 bits to location 0x2018 → third iteration of the minor loop.
g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
h. Write 32 bits to location 0x201C → last iteration of the minor loop → major loop complete.

14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[n] = 1.
16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

6.4.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of zero for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567*x*) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes, and the MOD field is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 6-7. Modulo example

Transfer number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C

Table continues on the next page...

Table 6-7. Modulo example (continued)

Transfer number	Address
5	0x12345670
6	0x12345674

6.4.5 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

6.4.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software-initiated service requests. The first is to read the TCD*n*_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test TCD*n*_CSR[START] and TCD*n*_CSR[ACTIVE]. The minor-loop-complete condition is indicated by both bits reading zero after TCD*n*_CSR[START] was set. Polling the TCD*n*_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request via software
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

The best method to test for minor-loop completion when using service requests initiated by hardware, that is, peripherals, is to read the TCD*n*_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request via hardware (peripheral request asserted)
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

For both activation types, the major-loop-complete status is explicitly indicated via the TCD*n*_CSR[DONE] bit.

The TCD*n*_CSR[START] bit is cleared automatically when the channel begins execution regardless of how the channel activates.

6.4.5.2 transfer descriptors of active channels

The eDMA reads back the true TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_NBYTES values if read when a channel executes. The true values of SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

6.4.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected for both group and channel arbitration modes. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed group, fixed channel arbitration mode, determination of the actively running relative priority outstanding requests become undefined. Channel and/or group priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD*n*_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD*n*_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

6.4.6 Channel linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD*n*_CSR[START] bit of another channel (or itself), thus initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD*n*_CITER[ELINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_ELINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x3
```

executes as:

1. Minor loop done → set TCD2_CSR[START] bit.
2. Minor loop done → set TCD2_CSR[START] bit.
3. Minor loop done → set TCD2_CSR[START] bit.
4. Minor loop done, major loop done → set TCD3_CSR[START] bit.

When minor loop linking is enabled (TCD*n*_CITER[ELINK] = 1), the TCD*n*_CITER[CITER] field uses a nine-bit vector to form the current iteration count. When minor loop linking is disabled (TCD*n*_CITER[ELINK] = 0), the TCD*n*_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD*n*_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

The TCD*n*_CITER[ELINK] bit and the TCD*n*_BITER[ELINK] bit must be equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD, at the end of a loop.

Table 6-8. Channel linking parameters

Desired link behavior	TCD control field name	Description
Link at end of minor loop	CITER[ELINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of major loop	CSR[MAJOR_ELINK]	Enable channel-to-channel linking on major loop completion
	CSR[MAJOR_LINKCH]	Link channel number when linking at end of major loop

6.4.7 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

6.4.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode
2. Disable all the channels, change the channel priorities, then enable the appropriate channels.

6.4.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting TCDn_CSR[MAJORELINK] during channel execution (see the diagram in TCD structure). This field is read from the TCD local memory at the end of channel execution, thus enabling you to enable the feature during channel execution.

Because you can change the configuration during execution, a coherency model is needed. Consider the scenario where you attempt to execute a dynamic channel link by enabling TCDn_CSR[MAJORELINK] at the same time the eDMA engine is retiring the channel. TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

1. Write one to TCDn_CSR[MAJORELINK].
2. Read back TCDn_CSR[MAJORELINK].
3. Test the TCDn_CSR[MAJORELINK] request status:
   - If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
   - If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces TCDn_CSR[MAJORELINK] to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

NOTE

You must clear TCDn_CSR[DONE] before writing TCDn_CSR[MAJORELINK]. The eDMA engine automatically clears TCDn_CSR[DONE] after a channel begins execution.

6.4.7.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It enables a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in eDMA programmer's model, thus replacing the current descriptor.

Because you are able to change the configuration during execution, a coherency model is needed. Consider the scenario where you attempt to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] bit at the same time the eDMA engine is retiring the channel. The ESG bit would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the MAJORLINKCH field and the ESG bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD MAJOR[ELINK] and ESG bits to zero on any writes to a channel's TCD word 7 if that channel's TCD[DONE] bit is set, indicating the major loop is complete.

NOTE

The user must clear the TCDn_CSR[DONE] bit before writing the MAJORELINK or ESG bits. The TCDn_CSR[DONE] bit is cleared automatically by the eDMA engine after a channel begins execution.

6.4.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When TCDn_CSR[MAJORELINK] is zero, TCDn_CSR[MAJORLINKCH] is not used by the eDMA. In this case, MAJORLINKCH may be used for other purposes. This method uses the MAJORLINKCH field as a TCD identification (ID).

1. When the descriptors are built, write a unique TCD ID in TCDn_CSR[MAJORLINKCH] for each TCD associated with a channel using dynamic scatter/gather.
2. Write one to TCDn_CSR[DREQ].

   Should a dynamic scatter/gather attempt fail, setting the DREQ bit prevents a future hardware activation of the channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST_SGA final offset value.

3. Write the TCDn_DLASTSGA register with the scatter/gather address.
4. Write one to TCDn_CSR[ESG].
5. Read back the 16-bit TCD control/status field.
6. Test the ESG request status and MAJORLINKCH value in the TCDn_CSR register:

   If ESG = 1, the dynamic link attempt was successful.

   If ESG = 0 and MAJORLINKCH (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If ESG = 0 and MAJORLINKCH (ID) changed, the dynamic link attempt was successful (the new TCD's ESG value cleared the ESG bit).

6.4.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD[DLAST_SGA] field as a TCD identification (ID).

1. Write one to TCDn_CSR[DREQ].

Should a dynamic scatter/gather attempt fail, setting TCDn_CSR[DREQ] prevents a future hardware activation of the channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST_SGA final offset value.

2. Write the TCDn_DLAST_SGA register with the scatter/gather address.
3. Write one to TCDn_CSR[ESG].
4. Read back TCDn_CSR[ESG].
5. Test the ESG request status:

   If ESG = 1, the dynamic link attempt was successful.

   If ESG = 0, read the 32-bit TCDn_DLAST_SGA field.

   If ESG = 0 and TCDn_DLAST_SGA did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If ESG = 0 and TCDn_DLAST_SGA changed, the dynamic link attempt was successful (the new TCD's ESG value cleared the ESG bit).

6.4.8 Suspend/resume a DMA channel with active hardware service requests

The DMA enables you to move data from memory or peripheral registers to another location in memory or peripheral registers without CPU interaction. After the DMA and peripherals have been configured and are active, it is rare to suspend a peripheral's service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, a specific procedure must be followed. This section provides guidance on how to coherently suspend and resume a Direct Memory Access (DMA) channel when the DMA is triggered by a slave module such as the Serial Peripheral Interface (SPI), ADC, or other module.

6.4.8.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check the DMA's Hardware Request Status register (DMA_HRSn) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ bit on appropriate DMA channel.

6.4.8.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting the relevant ERQ bit.
2. Enable the DMA service request at the peripheral.

For example, assume the SPI is set as a master for transmitting data via a DMA service request when the SPI_TXFIFO has an empty slot. The DMA transfers the next command and data to the TXFIFO upon the request. You must suspend the DMA/SPI transfer loop and perform the following steps:

1. Disable the DMA service request at the source by writing zero to SPI_RSER[TFFF_RE]. Confirm that SPI_RSER[TFFF_RE] is zero.
2. Ensure there is no DMA service request from the SPI by verifying that DMA_HRS[HRS*n*] is zero for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ bit. If a service request is present, wait until the request has been processed and the HRS bit reads zero.

6.5 Memory map/register definition

The eDMA programming model consists of registers that provide:

- Control and status functions
- Channel configuration functions
- TCD definition functions

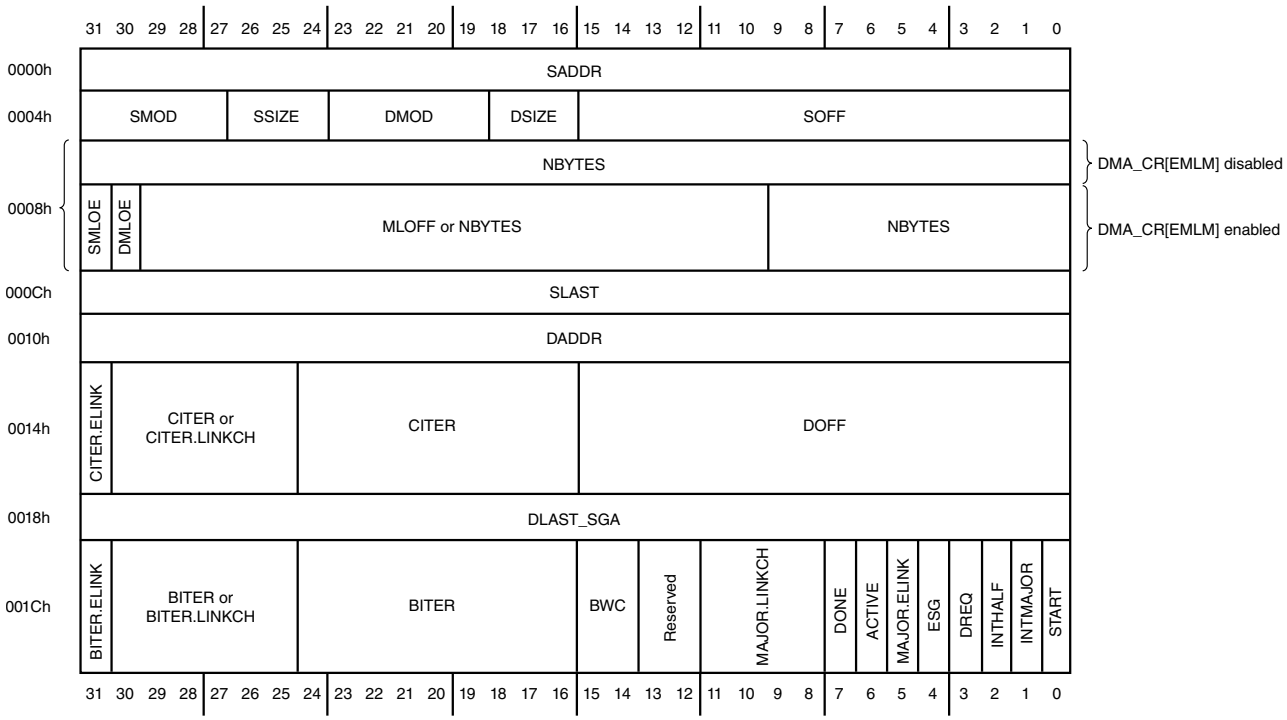
6.5.1 TCD memory

Each channel requires a 32-byte TCD to define the desired data movement operation. The channel descriptors are in local memory in sequential order: channel 0, channel 1, ... channel 31. Each TCD*n* definition comprises 11 registers of 16 or 32 bits.

6.5.2 TCD initialization

Before activating a channel, you must initialize its TCD with the appropriate transfer profile.

6.5.3 TCD structure



6.5.4 Reserved memory and fields

- Reading reserved fields in a register returns the value of zero.
- The eDMA ignores writes to reserved bits in a register.
- Reading or writing a reserved memory location generates a bus error.

6.5.5 DMA register descriptions

6.5.5.1 DMA memory map

DMA0 base address: 4007_0000h

DMA1 base address: 40C1_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control (CR)	32	RW	Table 6-8
4	Error Status (ES)	32	RO	0000_0000
C	Enable Request (ERQ)	32	RW	0000_0000
14	Enable Error Interrupt (EEI)	32	RW	0000_0000
18	Clear Enable Error Interrupt (CEEI)	8	WORZ	00
19	Set Enable Error Interrupt (SEEI)	8	WORZ	00
1A	Clear Enable Request (CERQ)	8	WORZ	00
1B	Set Enable Request (SERQ)	8	WORZ	00
1C	Clear DONE Status Bit (CDNE)	8	WORZ	00
1D	Set START Bit (SSRT)	8	WORZ	00
1E	Clear Error (CERR)	8	WORZ	00
1F	Clear Interrupt Request (CINT)	8	WORZ	00
24	Interrupt Request (INT)	32	W1C	0000_0000
2C	Error (ERR)	32	W1C	0000_0000
34	Hardware Request Status (HRS)	32	RO	0000_0000
44	Enable Asynchronous Request in Stop (EARS)	32	RW	0000_0000
100	Channel Priority (DCHPRI3)	8	RW	03
101	Channel Priority (DCHPRI2)	8	RW	02
102	Channel Priority (DCHPRI1)	8	RW	01
103	Channel Priority (DCHPRI0)	8	RW	00
104	Channel Priority (DCHPRI7)	8	RW	07
105	Channel Priority (DCHPRI6)	8	RW	06
106	Channel Priority (DCHPRI5)	8	RW	05
107	Channel Priority (DCHPRI4)	8	RW	04
108	Channel Priority (DCHPRI11)	8	RW	0B
109	Channel Priority (DCHPRI10)	8	RW	0A
10A	Channel Priority (DCHPRI9)	8	RW	09
10B	Channel Priority (DCHPRI8)	8	RW	08
10C	Channel Priority (DCHPRI15)	8	RW	0F
10D	Channel Priority (DCHPRI14)	8	RW	0E
10E	Channel Priority (DCHPRI13)	8	RW	0D
10F	Channel Priority (DCHPRI12)	8	RW	0C
110	Channel Priority (DCHPRI19)	8	RW	13
111	Channel Priority (DCHPRI18)	8	RW	12
112	Channel Priority (DCHPRI17)	8	RW	11
113	Channel Priority (DCHPRI16)	8	RW	10
114	Channel Priority (DCHPRI23)	8	RW	17
115	Channel Priority (DCHPRI22)	8	RW	16
116	Channel Priority (DCHPRI21)	8	RW	15
117	Channel Priority (DCHPRI20)	8	RW	14

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
118	Channel Priority (DCHPRI27)	8	RW	1B
119	Channel Priority (DCHPRI26)	8	RW	1A
11A	Channel Priority (DCHPRI25)	8	RW	19
11B	Channel Priority (DCHPRI24)	8	RW	18
11C	Channel Priority (DCHPRI31)	8	RW	1F
11D	Channel Priority (DCHPRI30)	8	RW	1E
11E	Channel Priority (DCHPRI29)	8	RW	1D
11F	Channel Priority (DCHPRI28)	8	RW	1C
1000 - 13E0	TCD Source Address (TCD0_SADDR - TCD31_SADDR)	32	RW	Table 6-8
1004 - 13E4	TCD Signed Source Address Offset (TCD0_SOFF - TCD31_SOFF)	16	RW	Table 6-8
1006 - 13E6	TCD Transfer Attributes (TCD0_ATTR - TCD31_ATTR)	16	RW	Table 6-8
1008 - 13E8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD0_NBYTES_MLNO - TCD31_NBYTES_MLNO)	32	RW	Table 6-8
1008 - 13E8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD0_NBYTES_MLOFFNO - TCD31_NBYTES_MLOFFNO)	32	RW	Table 6-8
1008 - 13E8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD0_NBYTES_MLOFFYES - TCD31_NBYTES_MLOFFYES)	32	RW	Table 6-8
100C - 13EC	TCD Last Source Address Adjustment (TCD0_SLAST - TCD31_SLAST)	32	RW	Table 6-8
1010 - 13F0	TCD Destination Address (TCD0_DADDR - TCD31_DADDR)	32	RW	Table 6-8
1014 - 13F4	TCD Signed Destination Address Offset (TCD0_DOFF - TCD31_DOFF)	16	RW	Table 6-8
1016 - 13F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD31_CITER_ELINKNO)	16	RW	Table 6-8
1016 - 13F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD31_CITER_ELINKYES)	16	RW	Table 6-8
1018 - 13F8	TCD Last Destination Address Adjustment/Scatter Gather Address (TCD0_DLASTSGA - TCD31_DLASTSGA)	32	RW	Table 6-8
101C - 13FC	TCD Control and Status (TCD0_CSR - TCD31_CSR)	16	RW	Table 6-8
101E - 13FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD31_BITER_ELINKNO)	16	RW	Table 6-8
101E - 13FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD31_BITER_ELINKYES)	16	RW	Table 6-8

6.5.5.2 Control (CR)

This register defines the basic operating configuration of the eDMA module. eDMA arbitrates channel service requests in two groups of 16 channels each:

- Group 1 contains channels 31-16
- Group 0 contains channels 15-0

You can configure arbitration within a group to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, eDMA selects and executes the highest-priority channel that requests service. The channel priority registers assign the priorities (see the Channel Priority (DCHPRI0 - DCHPRI31) registers). For round-robin arbitration, the eDMA engine ignores channel priorities and cycles through channels within each group from high to low channel number without regard to priority.

NOTE

For correct operation, you must write to this register only when the eDMA channels are inactive—that is, when TCD*n*_CSR[ACTIVE] = 0.

The group priorities operate in a similar fashion. In group fixed priority arbitration mode, channel service requests in the highest priority group are executed first, where priority level 1 is the highest and priority level 0 is the lowest. The group priorities are assigned in the GRP*n*PRI fields of the Control register (CR). All group priorities must have unique values prior to any channel service requests occurring; otherwise, a configuration error is reported. For group round robin arbitration, eDMA ignores the group priorities and the groups are cycled through (from high to low group number) without regard to priority.

Minor loop offsets are address-offset values to be added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) when the minor loop completes. When you enable minor loop offsets, eDMA adds the minor loop offset (MLOFF) value to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, before it writes the addresses back into the TCD. If the major loop is complete, eDMA ignores the minor loop offset, and uses the major loop address offsets (TCDn_SLAST and TCDn_DLAST_SGA) to compute the next TCDn_SADDR and TCDn_DADDR values.

Enabling minor loop mapping (EMLM = 1) redefines TCDn word2. eDMA uses a portion of TCDn word2 for multiple fields:

- A source enable field (SMLOE) to specify the minor loop offset is to be applied to the source address (TCDn_SADDR) when the minor loop completes
- A destination enable field (DMLOE) to specify the minor loop offset to be applied to the destination address (TCDn_DADDR) when the minor loop completes
- The sign extended minor loop offset value (MLOFF).

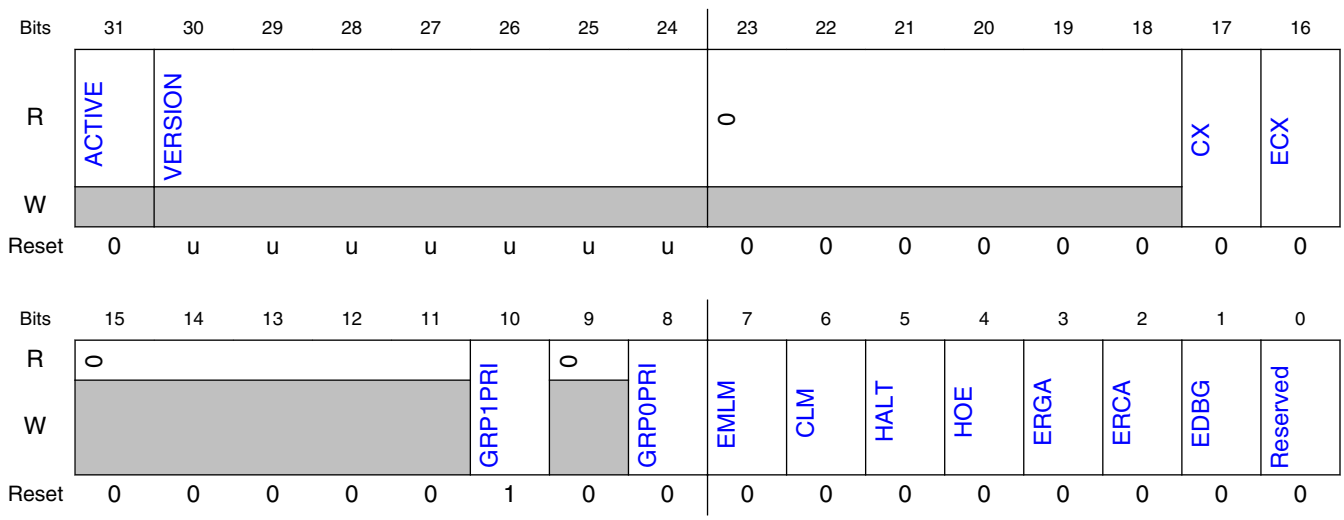
eDMA uses the same offset value (MLOFF) for both source and destination minor loop offsets. When you enable either minor loop offset (SMLOE = 1 or DMLOE = 1), the NBYTES field reduces in size to 10 bits. When you disable both minor loop offsets (SMLOE = 0 and and DMLOE = 0), the NBYTES field is a 30-bit vector.

When you disable minor loop mapping (EMLM = 0), the NBYTES field contains all 32 bits of TCDn word2.

6.5.5.2.1 Offset

Register	Offset
CR	0h

6.5.5.2.2 Diagram



6.5.5.2.3 Fields

Field	Description
31 ACTIVE	eDMA Active Status 0 - eDMA is idle 1 - eDMA is executing a channel
30-24 VERSION	eDMA version number
23-18 —	Reserved
17 CX	Cancel Transfer When you write 1 to this field, the following actions take place: • Stop the executing channel • Force the minor loop to finish. The cancellation takes effect after the last write of the current read/write sequence. This field is automatically written with 0 after the cancellation completes. The cancellation retires the channel normally as if the minor loop completed. 0 - Normal operation

Table continues on the next page...

Field	Description
	1 - Cancel the remaining data transfer
16 ECX	Error Cancel Transfer When you write a 1 to this field, the following actions take place: • Stop the executing channel • Force the minor loop to finish. The cancellation takes effect after the last write of the current read/write sequence. This field is automatically reset to 0 after the cancellation completes. In addition to cancelling the transfer, eDMA: • Treats the cancel as an error condition • Updates the Error Status register (DMAx_ES) • Optionally generates an error interrupt. 0 - Normal operation 1 - Cancel the remaining data transfer
15-11 —	Reserved
10 GRP1PRI	Channel Group 1 Priority Group 1 priority level when fixed priority group arbitration is enabled.
9 —	Reserved
8 GRP0PRI	Channel Group 0 Priority Group 0 priority level when fixed priority group arbitration is enabled.
7 EMLM	Enable Minor Loop Mapping When the value of this field is 0, TCDn.word2 is a 32-bit NBYTES field. When the value of this field is 1, TCDn.word2 includes: • Individual enable fields • An offset field • The NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field reduces in size when either offset is enabled. 0 - Disabled 1 - Enabled
6 CLM	Continuous Link Mode When the value of this field is 0, a minor loop channel link made to itself goes through channel arbitration before being activated again. When the value of this field is 1, a minor loop channel link made to itself does not go through channel arbitration before being activated again. When the minor loop completes, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop. **NOTE:** Do not use continuous link mode with a channel linking to itself if there is only one minor loop iteration per service request, for example, if the channel's NBYTES value is the same as either the source or destination size. The same data transfer profile can be achieved by simply increasing the NBYTES value, which provides more efficient, faster processing. 0 - Continuous link mode is off 1 - Continuous link mode is on
5 HALT	Halt eDMA Operations When this field is 1 the following actions take place:

Table continues on the next page...

Field	Description
	• eDMA stalls the start of any new channels • Executing channels are allowed to complete. When you write 0 to this field, channel execution resumes. 0 - Normal operation 1 - eDMA operations halted
4 HOE	Halt On Error When this field is 1, any error causes the eDMA engine to write 1 to the HALT field. Subsequently, the eDMA engine ignores all service requests until you write 0 to the HALT field. 0 - Normal operation 1 - Error causes HALT field to be automatically set to 1
3 ERGA	Enable Round Robin Group Arbitration When you write 1 to this field, eDMA uses round robin arbitration for selection among the groups. Otherwise, eDMA uses fixed priority arbitration. 0 - Fixed priority arbitration 1 - Round robin arbitration
2 ERCA	Enable Round Robin Channel Arbitration When you write 1 to this field, eDMA uses round robin arbitration for channel selection. Otherwise, eDMA uses fixed priority arbitration for channel selection. 0 - Fixed priority arbitration within each group 1 - Round robin arbitration within each group
1 EDBG	Enable Debug When this field is 0 and the chip enters Debug mode, eDMA continues operation. When this field is 1, entry of the chip into Debug mode causes the eDMA to stall the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the chip exits Debug mode or you write 0 to this field. 0 - When the chip is in Debug mode, the eDMA continues to operate. 1 - When the chip is in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete.
0 —	Reserved

6.5.5.3 Error Status (ES)

The ES register provides information concerning the most-recently recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor
  - An illegal priority register setting in fixed arbitration

- An error termination to a bus master read or write cycle
- A cancel transfer with error field that is 1 when a transfer is canceled via the corresponding cancel transfer control field

See Fault reporting and handling for more details.

6.5.5.3.1 Offset

Register	Offset
ES	4h

6.5.5.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPE	CPE	0	ERRCHN					SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.3.3 Fields

Field	Description
31 VLD	Logical OR of all ERR status fields 0 - No ERR fields are 1 1 - At least one ERR field has a value of 1, indicating a valid error exists that has not been cleared
30-17 —	Reserved
16 ECX	Transfer Canceled 0 - No canceled transfers 1 - The most-recently recorded entry was a canceled transfer initiated by the error cancel transfer field
15 GPE	Group Priority Error 0 - No group priority error. 1 - The most-recently recorded error was a configuration error among the group priorities. All group priorities are not unique.
14	Channel Priority Error

Table continues on the next page...

Field	Description
CPE	0 - No channel priority error. 1 - The most-recently recorded error was a configuration error in the channel priorities within a group. Channel priorities within a group are not unique.
13 —	Reserved
12-8 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the most-recently recorded error, excluding GPE and CPE errors or most-recently recorded error canceled transfer.
7 SAE	Source Address Error 0 - No source address configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].
6 SOE	Source Offset Error 0 - No source offset configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].
5 DAE	Destination Address Error 0 - No destination address configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].
4 DOE	Destination Offset Error 0 - No destination offset configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].
3 NCE	NBYTES/CITER Configuration Error 0 - No NBYTES/CITER configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields. TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], or TCDn_CITER[CITER] = 0, or TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK].
2 SGE	Scatter/Gather Configuration Error When 1, this field indicates the most-recently recorded error was a configuration error detected in the TCDn_DLASTSGA field. eDMA checks This field at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32-byte boundary. 0 - No scatter/gather configuration error. 1 - The most-recently recorded error was a configuration error detected in the TCDn_DLASTSGA field.
1 SBE	Source Bus Error 0 - No source bus error. 1 - The most-recently recorded error was a bus error on a source read.
0 DBE	Destination Bus Error 0 - No destination bus error. 1 - The most-recently recorded error was a bus error on a destination write.

6.5.5.4 Enable Request (ERQ)

The ERQ register provides a bit map for the 32 channels to enable the request signal for each channel. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ registers. These registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to this register.

DMA request input signals and this enable request field must be set to 1 before a channel's hardware service request is accepted. The state of the DMA enable request field does not affect a channel service request made explicitly through software or a linked channel request.

NOTE

Disable a channel's hardware service request at the source before writing 0 to the channel's ERQ field.

6.5.5.4.1 Offset

Register	Offset
ERQ	Ch

6.5.5.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERQ31	ERQ30	ERQ29	ERQ28	ERQ27	ERQ26	ERQ25	ERQ24	ERQ23	ERQ22	ERQ21	ERQ20	ERQ19	ERQ18	ERQ17	ERQ16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERQ15	ERQ14	ERQ13	ERQ12	ERQ11	ERQ10	ERQ9	ERQ8	ERQ7	ERQ6	ERQ5	ERQ4	ERQ3	ERQ2	ERQ1	ERQ0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.4.3 Fields

Field	Description
31 ERQ31	Enable DMA Request 31 0 - The DMA request signal for channel 31 is disabled 1 - The DMA request signal for channel 31 is enabled
30 ERQ30	Enable DMA Request 30 0 - The DMA request signal for channel 30 is disabled 1 - The DMA request signal for channel 30 is enabled
29 ERQ29	Enable DMA Request 29 0 - The DMA request signal for channel 29 is disabled 1 - The DMA request signal for channel 29 is enabled
28 ERQ28	Enable DMA Request 28 0 - The DMA request signal for channel 28 is disabled 1 - The DMA request signal for channel 28 is enabled
27 ERQ27	Enable DMA Request 27 0 - The DMA request signal for channel 27 is disabled 1 - The DMA request signal for channel 27 is enabled
26 ERQ26	Enable DMA Request 26 0 - The DMA request signal for channel 26 is disabled 1 - The DMA request signal for channel 26 is enabled
25 ERQ25	Enable DMA Request 25 0 - The DMA request signal for channel 25 is disabled 1 - The DMA request signal for channel 25 is enabled
24 ERQ24	Enable DMA Request 24 0 - The DMA request signal for channel 24 is disabled 1 - The DMA request signal for channel 24 is enabled
23 ERQ23	Enable DMA Request 23 0 - The DMA request signal for channel 23 is disabled 1 - The DMA request signal for channel 23 is enabled
22 ERQ22	Enable DMA Request 22 0 - The DMA request signal for channel 22 is disabled 1 - The DMA request signal for channel 22 is enabled
21 ERQ21	Enable DMA Request 21 0 - The DMA request signal for channel 21 is disabled 1 - The DMA request signal for channel 21 is enabled
20 ERQ20	Enable DMA Request 20 0 - The DMA request signal for channel 20 is disabled 1 - The DMA request signal for channel 20 is enabled
19	Enable DMA Request 19

Table continues on the next page...

Field	Description
ERQ19	0 - The DMA request signal for channel 19 is disabled 1 - The DMA request signal for channel 19 is enabled
18 ERQ18	Enable DMA Request 18 0 - The DMA request signal for channel 18 is disabled 1 - The DMA request signal for channel 18 is enabled
17 ERQ17	Enable DMA Request 17 0 - The DMA request signal for channel 17 is disabled 1 - The DMA request signal for channel 17 is enabled
16 ERQ16	Enable DMA Request 16 0 - The DMA request signal for channel 16 is disabled 1 - The DMA request signal for channel 16 is enabled
15 ERQ15	Enable DMA Request 15 0 - The DMA request signal for channel 15 is disabled 1 - The DMA request signal for channel 15 is enabled
14 ERQ14	Enable DMA Request 14 0 - The DMA request signal for channel 14 is disabled 1 - The DMA request signal for channel 14 is enabled
13 ERQ13	Enable DMA Request 13 0 - The DMA request signal for channel 13 is disabled 1 - The DMA request signal for channel 13 is enabled
12 ERQ12	Enable DMA Request 12 0 - The DMA request signal for channel 12 is disabled 1 - The DMA request signal for channel 12 is enabled
11 ERQ11	Enable DMA Request 11 0 - The DMA request signal for channel 11 is disabled 1 - The DMA request signal for channel 11 is enabled
10 ERQ10	Enable DMA Request 10 0 - The DMA request signal for channel 10 is disabled 1 - The DMA request signal for channel 10 is enabled
9 ERQ9	Enable DMA Request 9 0 - The DMA request signal for channel 9 is disabled 1 - The DMA request signal for channel 9 is enabled
8 ERQ8	Enable DMA Request 8 0 - The DMA request signal for channel 8 is disabled 1 - The DMA request signal for channel 8 is enabled
7 ERQ7	Enable DMA Request 7 0 - The DMA request signal for channel 7 is disabled 1 - The DMA request signal for channel 7 is enabled
6	Enable DMA Request 6

Table continues on the next page...

Field	Description
ERQ6	0 - The DMA request signal for channel 6 is disabled 1 - The DMA request signal for channel 6 is enabled
5 ERQ5	Enable DMA Request 5 0 - The DMA request signal for channel 5 is disabled 1 - The DMA request signal for channel 5 is enabled
4 ERQ4	Enable DMA Request 4 0 - The DMA request signal for channel 4 is disabled 1 - The DMA request signal for channel 4 is enabled
3 ERQ3	Enable DMA Request 3 0 - The DMA request signal for channel 3 is disabled 1 - The DMA request signal for channel 3 is enabled
2 ERQ2	Enable DMA Request 2 0 - The DMA request signal for channel 2 is disabled 1 - The DMA request signal for channel 2 is enabled
1 ERQ1	Enable DMA Request 1 0 - The DMA request signal for channel 1 is disabled 1 - The DMA request signal for channel 1 is enabled
0 ERQ0	Enable DMA Request 0 0 - The DMA request signal for channel 0 is disabled 1 - The DMA request signal for channel 0 is enabled

6.5.5.5 Enable Error Interrupt (EEI)

The EEI register provides a bit map for the 32 channels to enable the error interrupt signal for each channel. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI registers. These registers are provided so that the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI register.

The DMA error indicator and the error interrupt enable field must be set to 1 before an error interrupt request for a given channel is sent to the interrupt controller.

6.5.5.5.1 Offset

Register	Offset
EEI	14h

6.5.5.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EEI31	EEI30	EEI29	EEI28	EEI27	EEI26	EEI25	EEI24	EEI23	EEI22	EEI21	EEI20	EEI19	EEI18	EEI17	EEI16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EEI15	EEI14	EEI13	EEI12	EEI11	EEI10	EEI9	EEI8	EEI7	EEI6	EEI5	EEI4	EEI3	EEI2	EEI1	EEI0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.5.3 Fields

Field	Description
31 EEI31	Enable Error Interrupt 31 0 - An error on channel 31 does not generate an error interrupt 1 - An error on channel 31 generates an error interrupt request
30 EEI30	Enable Error Interrupt 30 0 - An error on channel 30 does not generate an error interrupt 1 - An error on channel 30 generates an error interrupt request
29 EEI29	Enable Error Interrupt 29 0 - An error on channel 29 does not generate an error interrupt 1 - An error on channel 29 generates an error interrupt request
28 EEI28	Enable Error Interrupt 28 0 - An error on channel 28 does not generate an error interrupt 1 - An error on channel 28 generates an error interrupt request
27 EEI27	Enable Error Interrupt 27 0 - An error on channel 27 does not generate an error interrupt 1 - An error on channel 27 generates an error interrupt request
26 EEI26	Enable Error Interrupt 26 0 - An error on channel 26 does not generate an error interrupt 1 - An error on channel 26 generates an error interrupt request
25 EEI25	Enable Error Interrupt 25 0 - An error on channel 25 does not generate an error interrupt 1 - An error on channel 25 generates an error interrupt request
24 EEI24	Enable Error Interrupt 24 0 - An error on channel 24 does not generate an error interrupt

Table continues on the next page...

Field	Description
	1 - An error on channel 24 generates an error interrupt request
23 EEI23	Enable Error Interrupt 23 0 - An error on channel 23 does not generate an error interrupt 1 - An error on channel 23 generates an error interrupt request
22 EEI22	Enable Error Interrupt 22 0 - An error on channel 22 does not generate an error interrupt 1 - An error on channel 22 generates an error interrupt request
21 EEI21	Enable Error Interrupt 21 0 - An error on channel 21 does not generate an error interrupt 1 - An error on channel 21 generates an error interrupt request
20 EEI20	Enable Error Interrupt 20 0 - An error on channel 20 does not generate an error interrupt 1 - An error on channel 20 generates an error interrupt request
19 EEI19	Enable Error Interrupt 19 0 - An error on channel 19 does not generate an error interrupt 1 - An error on channel 19 generates an error interrupt request
18 EEI18	Enable Error Interrupt 18 0 - An error on channel 18 does not generate an error interrupt 1 - An error on channel 18 generates an error interrupt request
17 EEI17	Enable Error Interrupt 17 0 - An error on channel 17 does not generate an error interrupt 1 - An error on channel 17 generates an error interrupt request
16 EEI16	Enable Error Interrupt 16 0 - An error on channel 16 does not generate an error interrupt 1 - An error on channel 16 generates an error interrupt request
15 EEI15	Enable Error Interrupt 15 0 - An error on channel 15 does not generate an error interrupt 1 - An error on channel 15 generates an error interrupt request
14 EEI14	Enable Error Interrupt 14 0 - An error on channel 14 does not generate an error interrupt 1 - An error on channel 14 generates an error interrupt request
13 EEI13	Enable Error Interrupt 13 0 - An error on channel 13 does not generate an error interrupt 1 - An error on channel 13 generates an error interrupt request
12 EEI12	Enable Error Interrupt 12 0 - An error on channel 12 does not generate an error interrupt 1 - An error on channel 12 generates an error interrupt request
11 EEI11	Enable Error Interrupt 11 0 - An error on channel 11 does not generate an error interrupt

Table continues on the next page...

Field	Description
	1 - An error on channel 11 generates an error interrupt request
10 EEI10	Enable Error Interrupt 10 0 - An error on channel 10 does not generate an error interrupt 1 - An error on channel 10 generates an error interrupt request
9 EEI9	Enable Error Interrupt 9 0 - An error on channel 9 does not generate an error interrupt 1 - An error on channel 9 generates an error interrupt request
8 EEI8	Enable Error Interrupt 8 0 - An error on channel 8 does not generate an error interrupt 1 - An error on channel 8 generates an error interrupt request
7 EEI7	Enable Error Interrupt 7 0 - An error on channel 7 does not generate an error interrupt 1 - An error on channel 7 generates an error interrupt request
6 EEI6	Enable Error Interrupt 6 0 - An error on channel 6 does not generate an error interrupt 1 - An error on channel 6 generates an error interrupt request
5 EEI5	Enable Error Interrupt 5 0 - An error on channel 5 does not generate an error interrupt 1 - An error on channel 5 generates an error interrupt request
4 EEI4	Enable Error Interrupt 4 0 - An error on channel 4 does not generate an error interrupt 1 - An error on channel 4 generates an error interrupt request
3 EEI3	Enable Error Interrupt 3 0 - An error on channel 3 does not generate an error interrupt 1 - An error on channel 3 generates an error interrupt request
2 EEI2	Enable Error Interrupt 2 0 - An error on channel 2 does not generate an error interrupt 1 - An error on channel 2 generates an error interrupt request
1 EEI1	Enable Error Interrupt 1 0 - An error on channel 1 does not generate an error interrupt 1 - An error on channel 1 generates an error interrupt request
0 EEI0	Enable Error Interrupt 0 0 - An error on channel 0 does not generate an error interrupt 1 - An error on channel 0 generates an error interrupt request

6.5.5.6 Clear Enable Error Interrupt (CEEI)

The CEEI provides a simple memory-mapped mechanism to write 0 to a given field in the EEI register to disable the error interrupt for a given channel. The data value on a register write causes the corresponding field in the EEI register to be written to 0. Writing 1 to the CAEE field provides a global clear to 0 function, forcing the EEI contents to be written to 0, disabling all DMA request inputs.

If the NOP field is written with 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field set to 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

6.5.5.6.1 Offset

Register	Offset
CEEI	18h

6.5.5.6.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0		0				
W	NOP	CAEE	0	CEEI				
Reset	0	0	0	0	0	0	0	0

6.5.5.6.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation, ignore the other fields in this register
6 CAEE	Clear All Enable Error Interrupts 0 - Write 0 only to the EEI field specified in the CEEI field 1 - Write 0 to all fields in EEI
5 —	Reserved
4-0 CEEI	Clear Enable Error Interrupt Writes 0 to the corresponding field in EEI

6.5.5.7 Set Enable Error Interrupt (SEEI)

The SEEI register provides a simple memory-mapped mechanism to write 1 to a given field in the EEI register to enable the error interrupt for a given channel. The data value on a register write causes the corresponding field in the EEI to be written to 1. Writing 1 to the SAEE field provides a global set to 1 function, forcing the entire EEI register contents to be written with 1.

If the NOP field is 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field set to 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

6.5.5.7.1 Offset

Register	Offset
SEEI	19h

6.5.5.7.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0		0				
W	NOP	SAEE	0	SEEI				
Reset	0	0	0	0	0	0	0	0

6.5.5.7.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation, ignore the other fields in this register
6 SAEE	Set All Enable Error Interrupts 0 - Write 1 only to the EEI field specified in the SEEI field 1 - Writes 1 to all fields in EEI

Table continues on the next page...

Field	Description
5 —	Reserved
4-0 SEEI	Set Enable Error Interrupt Writes 1 to the corresponding field in EEI

6.5.5.8 Clear Enable Request (CERQ)

The CERQ provides a simple memory-mapped mechanism to write 0 to a given field in the ERQ register to disable the DMA request for a given channel. The data value on a register write causes the corresponding field in the ERQ register to be written with 0. Setting the CAER field provides a global clear to 0 function, forcing the entire contents of the ERQ register to be written with 0, disabling all DMA request inputs.

If the NOP field is 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field written with 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

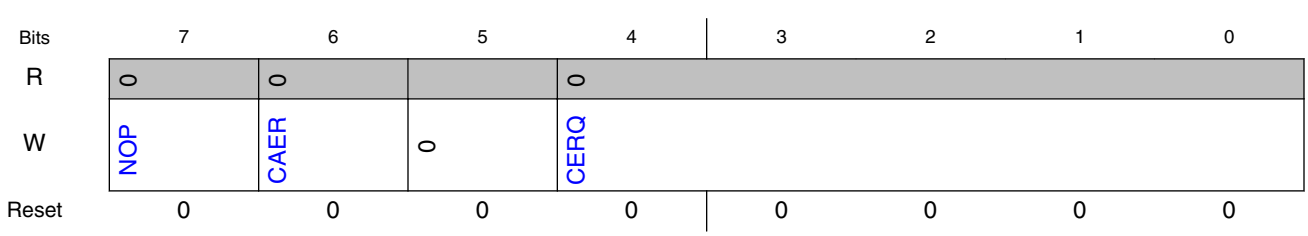
NOTE

Disable a channel's hardware service request at the source before writing 0 to the channel's ERQ field.

6.5.5.8.1 Offset

Register	Offset
CERQ	1Ah

6.5.5.8.2 Diagram



6.5.5.8.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation, ignore the other fields in this register
6 CAER	Clear All Enable Requests 0 - Write 0 to only the ERQ field specified in the CERQ field 1 - Write 0 to all fields in ERQ
5 —	Reserved
4-0 CERQ	Clear Enable Request Writes 0 to the corresponding field in ERQ.

6.5.5.9 Set Enable Request (SERQ)

The SERQ provides a simple memory-mapped mechanism to write 1 to a given field in the ERQ register to enable the DMA request for a given channel. The data value on a register write causes the corresponding field in the ERQ register to be set. Writing 1 to the SAER field provides a global set to 1 function, forcing the entire contents of ERQ register to be 1.

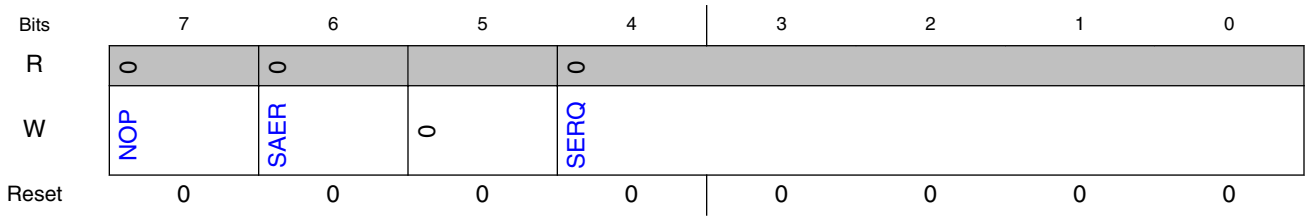
If the NOP field is 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field written with 1 so that these registers are not affected by the write.

Reads of this register returns all zeroes.

6.5.5.9.1 Offset

Register	Offset
SERQ	1Bh

6.5.5.9.2 Diagram



6.5.5.9.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation, ignore the other fields in this register
6 SAER	Set All Enable Requests 0 - Write 1 to only the ERQ field specified in the SERQ field 1 - Write 1 to all fields in ERQ
5 —	Reserved
4-0 SERQ	Set Enable Request Writes 1 to the corresponding field in ERQ.

6.5.5.10 Clear DONE Status Bit (CDNE)

The CDNE provides a simple memory-mapped mechanism to write 0 to the DONE field in the TCD of the given channel. The data value on a register write causes the DONE field in the corresponding TCD to be written with 0. Writing 1 to the CADN field provides a global clear function, forcing all DONE fields to be written with 0.

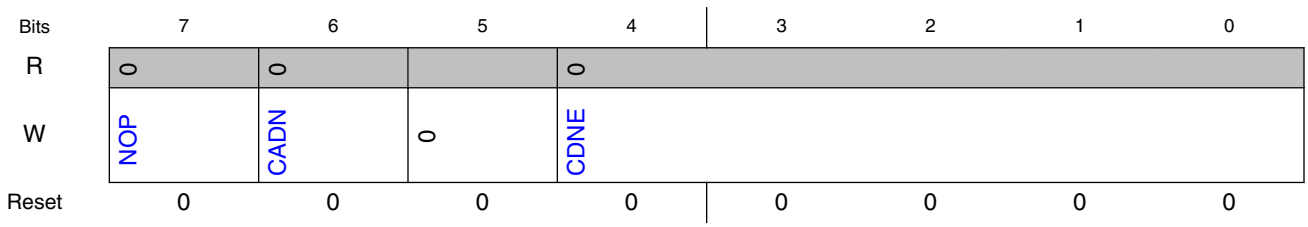
If the NOP field is 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field written with 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

6.5.5.10.1 Offset

Register	Offset
CDNE	1Ch

6.5.5.10.2 Diagram



6.5.5.10.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation; all other fields in this register are ignored.
6 CADN	Clears All DONE fields 0 - Writes 0 to only the TCDn_CSR[DONE] field specified in the CDNE field 1 - Writes 0 to all bits in TCDn_CSR[DONE]
5 —	Reserved
4-0 CDNE	Clear DONE field Writes 0 to the corresponding field in TCDn_CSR[DONE]

6.5.5.11 Set START Bit (SSRT)

The SSRT register provides a simple memory-mapped mechanism to write 1 to the START field in the TCD of the given channel. The data value on a register write causes the START field in the corresponding TCD to be written with 1. Writing 1 to the SAST field provides a global set to 1 function, forcing all START fields to be written with 1.

If the NOP field is 1, the command is ignored. This enables you to write 1 to a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word must all have their NOP field written with 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

6.5.5.11.1 Offset

Register	Offset
SSRT	1Dh

6.5.5.11.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0		0				
W	NOP	SAST	0	SSRT				
Reset	0	0	0	0	0	0	0	0

6.5.5.11.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation; all other fields in this register are ignored.
6 SAST	Set All START fields (activates all channels) 0 - Write 1 to only the TCDn_CSR[START] field specified in the SSRT field 1 - Write 1 to all bits in TCDn_CSR[START]
5 —	Reserved
4-0 SSRT	Set START field Sets the corresponding field in TCDn_CSR[START]

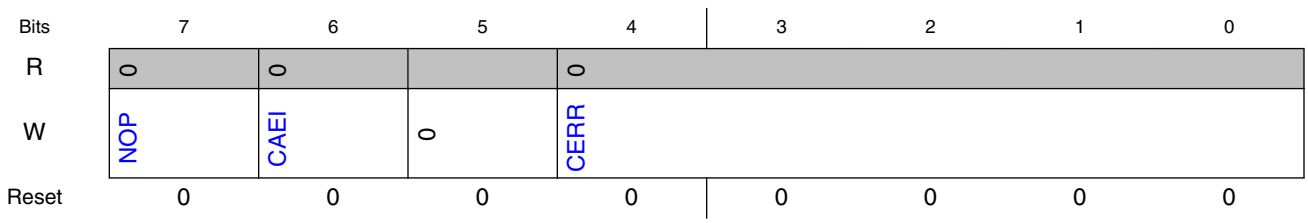
6.5.5.12 Clear Error (CERR)

The CERR provides a simple memory-mapped mechanism to write 0 to a given field in the ERR register to disable the error condition field for a given channel. The given value on a register write causes the corresponding field in the ERR register to be written with 0. Writing 1 to the CAEI field provides a global clear to 0 function, forcing the ERR register contents to be written with 0, clearing all channel error indicators. If the NOP field is 1, the command is ignored. This enables you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

6.5.5.12.1 Offset

Register	Offset
CERR	1Eh

6.5.5.12.2 Diagram



6.5.5.12.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation; all other fields in this register are ignored.
6 CAEI	Clear All Error Indicators 0 - Write 0 to only the ERR field specified in the CERR field 1 - Write 0 to all fields in ERR
5 —	Reserved
4-0 CERR	Clear Error Indicator Writes 0 to the corresponding field in ERR

6.5.5.13 Clear Interrupt Request (CINT)

The CINT register provides a simple, memory-mapped mechanism to clear a given field in the INT register to disable the interrupt request for a given channel. The given value on a register write causes the corresponding field in the INT register to be cleared. Setting the CAIR field provides a global clear function, forcing the entire contents of the INT to be cleared, disabling all DMA interrupt requests.

If the NOP field is 1, the command is ignored. This enables you to set a single, byte-wide register with a 32-bit write that does not affect the other registers addressed in the write. In such a case the other three bytes of the word would all have their NOP field set to 1 so that these registers are not affected by the write.

Reads of this register return all zeroes.

6.5.5.13.1 Offset

Register	Offset
CINT	1Fh

6.5.5.13.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	0	0		0				
W	NOP	CAIR	0	CINT				
Reset	0	0	0	0	0	0	0	0

6.5.5.13.3 Fields

Field	Description
7 NOP	No Op Enable 0 - Normal operation 1 - No operation; all other fields in this register are ignored.
6 CAIR	Clear All Interrupt Requests 0 - Clear only the INT field specified in the CINT field 1 - Clear all bits in INT
5 —	Reserved

Table continues on the next page...

Field	Description
4-0 CINT	Clear Interrupt Request Clears the corresponding field in INT

6.5.5.14 Interrupt Request (INT)

The INT register provides a bit map for the 32 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt-service routine associated with any given channel, it is the software's responsibility to write 0 to the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A 0 in any bit position has no effect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT register.

6.5.5.14.1 Offset

Register	Offset
INT	24h

6.5.5.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.14.3 Fields

Field	Description
31 INT31	Interrupt Request 31 0 - The interrupt request for channel 31 is cleared 1 - The interrupt request for channel 31 is active
30 INT30	Interrupt Request 30 0 - The interrupt request for channel 30 is cleared 1 - The interrupt request for channel 30 is active
29 INT29	Interrupt Request 29 0 - The interrupt request for channel 29 is cleared 1 - The interrupt request for channel 29 is active
28 INT28	Interrupt Request 28 0 - The interrupt request for channel 28 is cleared 1 - The interrupt request for channel 28 is active
27 INT27	Interrupt Request 27 0 - The interrupt request for channel 27 is cleared 1 - The interrupt request for channel 27 is active
26 INT26	Interrupt Request 26 0 - The interrupt request for channel 26 is cleared 1 - The interrupt request for channel 26 is active
25 INT25	Interrupt Request 25 0 - The interrupt request for channel 25 is cleared 1 - The interrupt request for channel 25 is active

Table continues on the next page...

Field	Description
24 INT24	Interrupt Request 24 0 - The interrupt request for channel 24 is cleared 1 - The interrupt request for channel 24 is active
23 INT23	Interrupt Request 23 0 - The interrupt request for channel 23 is cleared 1 - The interrupt request for channel 23 is active
22 INT22	Interrupt Request 22 0 - The interrupt request for channel 22 is cleared 1 - The interrupt request for channel 22 is active
21 INT21	Interrupt Request 21 0 - The interrupt request for channel 21 is cleared 1 - The interrupt request for channel 21 is active
20 INT20	Interrupt Request 20 0 - The interrupt request for channel 20 is cleared 1 - The interrupt request for channel 20 is active
19 INT19	Interrupt Request 19 0 - The interrupt request for channel 19 is cleared 1 - The interrupt request for channel 19 is active
18 INT18	Interrupt Request 18 0 - The interrupt request for channel 18 is cleared 1 - The interrupt request for channel 18 is active
17 INT17	Interrupt Request 17 0 - The interrupt request for channel 17 is cleared 1 - The interrupt request for channel 17 is active
16 INT16	Interrupt Request 16 0 - The interrupt request for channel 16 is cleared 1 - The interrupt request for channel 16 is active
15 INT15	Interrupt Request 15 0 - The interrupt request for channel 15 is cleared 1 - The interrupt request for channel 15 is active
14 INT14	Interrupt Request 14 0 - The interrupt request for channel 14 is cleared 1 - The interrupt request for channel 14 is active
13 INT13	Interrupt Request 13 0 - The interrupt request for channel 13 is cleared 1 - The interrupt request for channel 13 is active
12 INT12	Interrupt Request 12 0 - The interrupt request for channel 12 is cleared 1 - The interrupt request for channel 12 is active

Table continues on the next page...

Field	Description
11 INT11	Interrupt Request 11 0 - The interrupt request for channel 11 is cleared 1 - The interrupt request for channel 11 is active
10 INT10	Interrupt Request 10 0 - The interrupt request for channel 10 is cleared 1 - The interrupt request for channel 10 is active
9 INT9	Interrupt Request 9 0 - The interrupt request for channel 9 is cleared 1 - The interrupt request for channel 9 is active
8 INT8	Interrupt Request 8 0 - The interrupt request for channel 8 is cleared 1 - The interrupt request for channel 8 is active
7 INT7	Interrupt Request 7 0 - The interrupt request for channel 7 is cleared 1 - The interrupt request for channel 7 is active
6 INT6	Interrupt Request 6 0 - The interrupt request for channel 6 is cleared 1 - The interrupt request for channel 6 is active
5 INT5	Interrupt Request 5 0 - The interrupt request for channel 5 is cleared 1 - The interrupt request for channel 5 is active
4 INT4	Interrupt Request 4 0 - The interrupt request for channel 4 is cleared 1 - The interrupt request for channel 4 is active
3 INT3	Interrupt Request 3 0 - The interrupt request for channel 3 is cleared 1 - The interrupt request for channel 3 is active
2 INT2	Interrupt Request 2 0 - The interrupt request for channel 2 is cleared 1 - The interrupt request for channel 2 is active
1 INT1	Interrupt Request 1 0 - The interrupt request for channel 1 is cleared 1 - The interrupt request for channel 1 is active
0 INT0	Interrupt Request 0 0 - The interrupt request for channel 0 is cleared 1 - The interrupt request for channel 0 is active

6.5.5.15 Error (ERR)

The ERR register provides a bit map for the 32 channels, signaling the presence of an error for each channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the EEI register, then logically summed across groups of 16 and 32 channels to form several group error interrupt requests, which are then routed to the interrupt controller. During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to reset the appropriate bit to 0, negating the error-interrupt request. Typically, a write to the CERR in the interrupt service routine is used for this purpose. The normal DMA channel completion indicators (setting the TCD DONE field to 1 and the possible generation of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI fields. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a 1 in any bit position clears the corresponding channel's error status. A 0 in any bit position has no effect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be reset to 0.

6.5.5.15.1 Offset

Register	Offset
ERR	2Ch

6.5.5.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR31	ERR30	ERR29	ERR28	ERR27	ERR26	ERR25	ERR24	ERR23	ERR22	ERR21	ERR20	ERR19	ERR18	ERR17	ERR16
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR15	ERR14	ERR13	ERR12	ERR11	ERR10	ERR9	ERR8	ERR7	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.15.3 Fields

Field	Description
31 ERR31	Error In Channel 31 0 - No error in this channel has occurred 1 - An error in this channel has occurred
30 ERR30	Error In Channel 30 0 - No error in this channel has occurred 1 - An error in this channel has occurred
29 ERR29	Error In Channel 29 0 - No error in this channel has occurred 1 - An error in this channel has occurred
28 ERR28	Error In Channel 28 0 - No error in this channel has occurred 1 - An error in this channel has occurred
27 ERR27	Error In Channel 27 0 - No error in this channel has occurred 1 - An error in this channel has occurred
26 ERR26	Error In Channel 26 0 - No error in this channel has occurred 1 - An error in this channel has occurred
25 ERR25	Error In Channel 25 0 - No error in this channel has occurred 1 - An error in this channel has occurred

Table continues on the next page...

Field	Description
24 ERR24	Error In Channel 24 0 - No error in this channel has occurred 1 - An error in this channel has occurred
23 ERR23	Error In Channel 23 0 - No error in this channel has occurred 1 - An error in this channel has occurred
22 ERR22	Error In Channel 22 0 - No error in this channel has occurred 1 - An error in this channel has occurred
21 ERR21	Error In Channel 21 0 - No error in this channel has occurred 1 - An error in this channel has occurred
20 ERR20	Error In Channel 20 0 - No error in this channel has occurred 1 - An error in this channel has occurred
19 ERR19	Error In Channel 19 0 - No error in this channel has occurred 1 - An error in this channel has occurred
18 ERR18	Error In Channel 18 0 - No error in this channel has occurred 1 - An error in this channel has occurred
17 ERR17	Error In Channel 17 0 - No error in this channel has occurred 1 - An error in this channel has occurred
16 ERR16	Error In Channel 16 0 - No error in this channel has occurred 1 - An error in this channel has occurred
15 ERR15	Error In Channel 15 0 - No error in this channel has occurred 1 - An error in this channel has occurred
14 ERR14	Error In Channel 14 0 - No error in this channel has occurred 1 - An error in this channel has occurred
13 ERR13	Error In Channel 13 0 - No error in this channel has occurred 1 - An error in this channel has occurred
12 ERR12	Error In Channel 12 0 - No error in this channel has occurred 1 - An error in this channel has occurred

Table continues on the next page...

Field	Description
11 ERR11	Error In Channel 11 0 - No error in this channel has occurred 1 - An error in this channel has occurred
10 ERR10	Error In Channel 10 0 - No error in this channel has occurred 1 - An error in this channel has occurred
9 ERR9	Error In Channel 9 0 - No error in this channel has occurred 1 - An error in this channel has occurred
8 ERR8	Error In Channel 8 0 - No error in this channel has occurred 1 - An error in this channel has occurred
7 ERR7	Error In Channel 7 0 - No error in this channel has occurred 1 - An error in this channel has occurred
6 ERR6	Error In Channel 6 0 - No error in this channel has occurred 1 - An error in this channel has occurred
5 ERR5	Error In Channel 5 0 - No error in this channel has occurred 1 - An error in this channel has occurred
4 ERR4	Error In Channel 4 0 - No error in this channel has occurred 1 - An error in this channel has occurred
3 ERR3	Error In Channel 3 0 - No error in this channel has occurred 1 - An error in this channel has occurred
2 ERR2	Error In Channel 2 0 - No error in this channel has occurred 1 - An error in this channel has occurred
1 ERR1	Error In Channel 1 0 - No error in this channel has occurred 1 - An error in this channel has occurred
0 ERR0	Error In Channel 0 0 - No error in this channel has occurred 1 - An error in this channel has occurred

6.5.5.16 Hardware Request Status (HRS)

The HRS register provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals, as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

Each HRS field for its respective channel is 1 when a hardware request is present on the channel. After the request is completed and channel is free, the HRS field is automatically changed to 0 by hardware.

6.5.5.16.1 Offset

Register	Offset
HRS	34h

6.5.5.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HRS31	HRS30	HRS29	HRS28	HRS27	HRS26	HRS25	HRS24	HRS23	HRS22	HRS21	HRS20	HRS19	HRS18	HRS17	HRS16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS15	HRS14	HRS13	HRS12	HRS11	HRS10	HRS9	HRS8	HRS7	HRS6	HRS5	HRS4	HRS3	HRS2	HRS1	HRS0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.16.3 Fields

Field	Description
31 HRS31	Hardware Request Status Channel 31 0 - A hardware service request for channel 31 is not present 1 - A hardware service request for channel 31 is present
30 HRS30	Hardware Request Status Channel 30 0 - A hardware service request for channel 30 is not present 1 - A hardware service request for channel 30 is present
29 HRS29	Hardware Request Status Channel 29 0 - A hardware service request for channel 29 is not preset 1 - A hardware service request for channel 29 is present
28 HRS28	Hardware Request Status Channel 28 0 - A hardware service request for channel 28 is not present 1 - A hardware service request for channel 28 is present
27 HRS27	Hardware Request Status Channel 27 0 - A hardware service request for channel 27 is not present 1 - A hardware service request for channel 27 is present
26 HRS26	Hardware Request Status Channel 26 0 - A hardware service request for channel 26 is not present 1 - A hardware service request for channel 26 is present
25 HRS25	Hardware Request Status Channel 25 0 - A hardware service request for channel 25 is not present 1 - A hardware service request for channel 25 is present
24 HRS24	Hardware Request Status Channel 24 0 - A hardware service request for channel 24 is not present 1 - A hardware service request for channel 24 is present
23 HRS23	Hardware Request Status Channel 23 0 - A hardware service request for channel 23 is not present 1 - A hardware service request for channel 23 is present
22 HRS22	Hardware Request Status Channel 22 0 - A hardware service request for channel 22 is not present 1 - A hardware service request for channel 22 is present
21 HRS21	Hardware Request Status Channel 21 0 - A hardware service request for channel 21 is not present 1 - A hardware service request for channel 21 is present
20 HRS20	Hardware Request Status Channel 20 0 - A hardware service request for channel 20 is not present 1 - A hardware service request for channel 20 is present
19	Hardware Request Status Channel 19

Table continues on the next page...

Field	Description
HRS19	0 - A hardware service request for channel 19 is not present 1 - A hardware service request for channel 19 is present
18 HRS18	Hardware Request Status Channel 18 0 - A hardware service request for channel 18 is not present 1 - A hardware service request for channel 18 is present
17 HRS17	Hardware Request Status Channel 17 0 - A hardware service request for channel 17 is not present 1 - A hardware service request for channel 17 is present
16 HRS16	Hardware Request Status Channel 16 0 - A hardware service request for channel 16 is not present 1 - A hardware service request for channel 16 is present
15 HRS15	Hardware Request Status Channel 15 0 - A hardware service request for channel 15 is not present 1 - A hardware service request for channel 15 is present
14 HRS14	Hardware Request Status Channel 14 0 - A hardware service request for channel 14 is not present 1 - A hardware service request for channel 14 is present
13 HRS13	Hardware Request Status Channel 13 0 - A hardware service request for channel 13 is not present 1 - A hardware service request for channel 13 is present
12 HRS12	Hardware Request Status Channel 12 0 - A hardware service request for channel 12 is not present 1 - A hardware service request for channel 12 is present
11 HRS11	Hardware Request Status Channel 11 0 - A hardware service request for channel 11 is not present 1 - A hardware service request for channel 11 is present
10 HRS10	Hardware Request Status Channel 10 0 - A hardware service request for channel 10 is not present 1 - A hardware service request for channel 10 is present
9 HRS9	Hardware Request Status Channel 9 0 - A hardware service request for channel 9 is not present 1 - A hardware service request for channel 9 is present
8 HRS8	Hardware Request Status Channel 8 0 - A hardware service request for channel 8 is not present 1 - A hardware service request for channel 8 is present
7 HRS7	Hardware Request Status Channel 7 0 - A hardware service request for channel 7 is not present 1 - A hardware service request for channel 7 is present
6	Hardware Request Status Channel 6

Table continues on the next page...

Field	Description
HRS6	0 - A hardware service request for channel 6 is not present 1 - A hardware service request for channel 6 is present
5 HRS5	Hardware Request Status Channel 5 0 - A hardware service request for channel 5 is not present 1 - A hardware service request for channel 5 is present
4 HRS4	Hardware Request Status Channel 4 0 - A hardware service request for channel 4 is not present 1 - A hardware service request for channel 4 is present
3 HRS3	Hardware Request Status Channel 3 0 - A hardware service request for channel 3 is not present 1 - A hardware service request for channel 3 is present
2 HRS2	Hardware Request Status Channel 2 0 - A hardware service request for channel 2 is not present 1 - A hardware service request for channel 2 is present
1 HRS1	Hardware Request Status Channel 1 0 - A hardware service request for channel 1 is not present 1 - A hardware service request for channel 1 is present
0 HRS0	Hardware Request Status Channel 0 0 - A hardware service request for channel 0 is not present 1 - A hardware service request for channel 0 is present

6.5.5.17 Enable Asynchronous Request in Stop (EARS)

The EARS register is used to enable or disable the DMA requests in Enable Request (ERQ) by AND'ing the bits of these two registers.

6.5.5.17.1 Offset

Register	Offset
EARS	44h

6.5.5.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EDREQ_31	EDREQ_30	EDREQ_29	EDREQ_28	EDREQ_27	EDREQ_26	EDREQ_25	EDREQ_24	EDREQ_23	EDREQ_22	EDREQ_21	EDREQ_20	EDREQ_19	EDREQ_18	EDREQ_17	EDREQ_16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDREQ_15	EDREQ_14	EDREQ_13	EDREQ_12	EDREQ_11	EDREQ_10	EDREQ_9	EDREQ_8	EDREQ_7	EDREQ_6	EDREQ_5	EDREQ_4	EDREQ_3	EDREQ_2	EDREQ_1	EDREQ_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.5.5.17.3 Fields

Field	Description
31 EDREQ_31	Enable asynchronous DMA request in stop mode for channel 31. 0 - Disable asynchronous DMA request for channel 31 1 - Enable asynchronous DMA request for channel 31
30 EDREQ_30	Enable asynchronous DMA request in stop mode for channel 30. 0 - Disable asynchronous DMA request for channel 30 1 - Enable asynchronous DMA request for channel 30
29 EDREQ_29	Enable asynchronous DMA request in stop mode for channel 29. 0 - Disable asynchronous DMA request for channel 29 1 - Enable asynchronous DMA request for channel 29
28 EDREQ_28	Enable asynchronous DMA request in stop mode for channel 28. 0 - Disable asynchronous DMA request for channel 28 1 - Enable asynchronous DMA request for channel 28
27 EDREQ_27	Enable asynchronous DMA request in stop mode for channel 27. 0 - Disable asynchronous DMA request for channel 27 1 - Enable asynchronous DMA request for channel 27
26 EDREQ_26	Enable asynchronous DMA request in stop mode for channel 26. 0 - Disable asynchronous DMA request for channel 26 1 - Enable asynchronous DMA request for channel 26
25 EDREQ_25	Enable asynchronous DMA request in stop mode for channel 25. 0 - Disable asynchronous DMA request for channel 25 1 - Enable asynchronous DMA request for channel 25
24	Enable asynchronous DMA request in stop mode for channel 24.

Table continues on the next page...

Field	Description
EDREQ_24	0 - Disable asynchronous DMA request for channel 24 1 - Enable asynchronous DMA request for channel 24
23 EDREQ_23	Enable asynchronous DMA request in stop mode for channel 23. 0 - Disable asynchronous DMA request for channel 23 1 - Enable asynchronous DMA request for channel 23
22 EDREQ_22	Enable asynchronous DMA request in stop mode for channel 22. 0 - Disable asynchronous DMA request for channel 22 1 - Enable asynchronous DMA request for channel 22
21 EDREQ_21	Enable asynchronous DMA request in stop mode for channel 21. 0 - Disable asynchronous DMA request for channel 21 1 - Enable asynchronous DMA request for channel 21
20 EDREQ_20	Enable asynchronous DMA request in stop mode for channel 20. 0 - Disable asynchronous DMA request for channel 20 1 - Enable asynchronous DMA request for channel 20
19 EDREQ_19	Enable asynchronous DMA request in stop mode for channel 19. 0 - Disable asynchronous DMA request for channel 19 1 - Enable asynchronous DMA request for channel 19
18 EDREQ_18	Enable asynchronous DMA request in stop mode for channel 18. 0 - Disable asynchronous DMA request for channel 18 1 - Enable asynchronous DMA request for channel 18
17 EDREQ_17	Enable asynchronous DMA request in stop mode for channel 17. 0 - Disable asynchronous DMA request for channel 17 1 - Enable asynchronous DMA request for channel 17
16 EDREQ_16	Enable asynchronous DMA request in stop mode for channel 16. 0 - Disable asynchronous DMA request for channel 16 1 - Enable asynchronous DMA request for channel 16
15 EDREQ_15	Enable asynchronous DMA request in stop mode for channel 15. 0 - Disable asynchronous DMA request for channel 15 1 - Enable asynchronous DMA request for channel 15
14 EDREQ_14	Enable asynchronous DMA request in stop mode for channel 14. 0 - Disable asynchronous DMA request for channel 14 1 - Enable asynchronous DMA request for channel 14
13 EDREQ_13	Enable asynchronous DMA request in stop mode for channel 13. 0 - Disable asynchronous DMA request for channel 13 1 - Enable asynchronous DMA request for channel 13
12 EDREQ_12	Enable asynchronous DMA request in stop mode for channel 12. 0 - Disable asynchronous DMA request for channel 12 1 - Enable asynchronous DMA request for channel 12
11	Enable asynchronous DMA request in stop mode for channel 11.

Table continues on the next page...

Field	Description
EDREQ_11	0 - Disable asynchronous DMA request for channel 11 1 - Enable asynchronous DMA request for channel 11
10 EDREQ_10	Enable asynchronous DMA request in stop mode for channel 10. 0 - Disable asynchronous DMA request for channel 10 1 - Enable asynchronous DMA request for channel 10
9 EDREQ_9	Enable asynchronous DMA request in stop mode for channel 9. 0 - Disable asynchronous DMA request for channel 9 1 - Enable asynchronous DMA request for channel 9
8 EDREQ_8	Enable asynchronous DMA request in stop mode for channel 8. 0 - Disable asynchronous DMA request for channel 8 1 - Enable asynchronous DMA request for channel 8
7 EDREQ_7	Enable asynchronous DMA request in stop mode for channel 7. 0 - Disable asynchronous DMA request for channel 7 1 - Enable asynchronous DMA request for channel 7
6 EDREQ_6	Enable asynchronous DMA request in stop mode for channel 6. 0 - Disable asynchronous DMA request for channel 6 1 - Enable asynchronous DMA request for channel 6
5 EDREQ_5	Enable asynchronous DMA request in stop mode for channel 5. 0 - Disable asynchronous DMA request for channel 5 1 - Enable asynchronous DMA request for channel 5
4 EDREQ_4	Enable asynchronous DMA request in stop mode for channel 4. 0 - Disable asynchronous DMA request for channel 4 1 - Enable asynchronous DMA request for channel 4
3 EDREQ_3	Enable asynchronous DMA request in stop mode for channel 3. 0 - Disable asynchronous DMA request for channel 3 1 - Enable asynchronous DMA request for channel 3
2 EDREQ_2	Enable asynchronous DMA request in stop mode for channel 2. 0 - Disable asynchronous DMA request for channel 2 1 - Enable asynchronous DMA request for channel 2
1 EDREQ_1	Enable asynchronous DMA request in stop mode for channel 1. 0 - Disable asynchronous DMA request for channel 1 1 - Enable asynchronous DMA request for channel 1
0 EDREQ_0	Enable asynchronous DMA request in stop mode for channel 0. 0 - Disable asynchronous DMA request for channel 0 1 - Enable asynchronous DMA request for channel 0

6.5.5.18 Channel Priority (DCHPRI0 - DCHPRI31)

When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel within a group. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 15. When read, the GRPPRI bits of the DCHPRIn register reflect the current priority level of the group of channels in which the corresponding channel resides. GRPPRI bits are not affected by writes to the DCHPRIn registers. The group priority is assigned in the DMA control register.

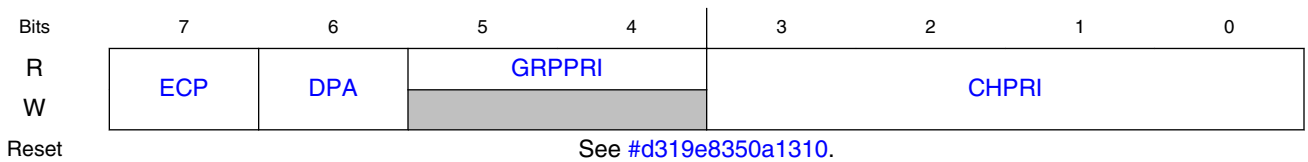
6.5.5.18.1 Offset

Register	Offset
DCHPRI3	100h
DCHPRI2	101h
DCHPRI1	102h
DCHPRI0	103h
DCHPRI7	104h
DCHPRI6	105h
DCHPRI5	106h
DCHPRI4	107h
DCHPRI11	108h
DCHPRI10	109h
DCHPRI9	10Ah
DCHPRI8	10Bh
DCHPRI15	10Ch
DCHPRI14	10Dh
DCHPRI13	10Eh
DCHPRI12	10Fh
DCHPRI19	110h
DCHPRI18	111h
DCHPRI17	112h
DCHPRI16	113h
DCHPRI23	114h
DCHPRI22	115h
DCHPRI21	116h
DCHPRI20	117h
DCHPRI27	118h

Table continues on the next page...

Register	Offset
DCHPRI26	119h
DCHPRI25	11Ah
DCHPRI24	11Bh
DCHPRI31	11Ch
DCHPRI30	11Dh
DCHPRI29	11Eh
DCHPRI28	11Fh

6.5.5.18.2 Diagram



See #d319e8350a1310.

6.5.5.18.3 Register reset values

Register	Reset value
DCHPRI0	DMA0,DMA1: 00h
DCHPRI1	DMA0,DMA1: 01h
DCHPRI2	DMA0,DMA1: 02h
DCHPRI3	DMA0,DMA1: 03h
DCHPRI4	DMA0,DMA1: 04h
DCHPRI5	DMA0,DMA1: 05h
DCHPRI6	DMA0,DMA1: 06h
DCHPRI7	DMA0,DMA1: 07h
DCHPRI8	DMA0,DMA1: 08h
DCHPRI9	DMA0,DMA1: 09h
DCHPRI10	DMA0,DMA1: 0Ah
DCHPRI11	DMA0,DMA1: 0Bh
DCHPRI12	DMA0,DMA1: 0Ch
DCHPRI13	DMA0,DMA1: 0Dh
DCHPRI14	DMA0,DMA1: 0Eh
DCHPRI15	DMA0,DMA1: 0Fh
DCHPRI16	DMA0,DMA1: 10h
DCHPRI17	DMA0,DMA1: 11h
DCHPRI18	DMA0,DMA1: 12h

Table continues on the next page...

Register	Reset value
DCHPRI19	DMA0,DMA1: 13h
DCHPRI20	DMA0,DMA1: 14h
DCHPRI21	DMA0,DMA1: 15h
DCHPRI22	DMA0,DMA1: 16h
DCHPRI23	DMA0,DMA1: 17h
DCHPRI24	DMA0,DMA1: 18h
DCHPRI25	DMA0,DMA1: 19h
DCHPRI26	DMA0,DMA1: 1Ah
DCHPRI27	DMA0,DMA1: 1Bh
DCHPRI28	DMA0,DMA1: 1Ch
DCHPRI29	DMA0,DMA1: 1Dh
DCHPRI30	DMA0,DMA1: 1Eh
DCHPRI31	DMA0,DMA1: 1Fh

6.5.5.18.4 Fields

Field	Description
7 ECP	Enable Channel Preemption. This field resets to 0. 0 - Channel n cannot be suspended by a higher priority channel's service request 1 - Channel n can be temporarily suspended by the service request of a higher priority channel
6 DPA	Disable Preempt Ability. This field resets to 0. 0 - Channel n can suspend a lower priority channel 1 - Channel n cannot suspend any channel, regardless of channel priority
5-4 GRPPRI	Channel n Current Group Priority Group priority assigned to this channel group when fixed-priority arbitration is enabled. This field is read-only; writes are ignored.
3-0 CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled.

6.5.5.19 TCD Source Address (TCD0_SADDR - TCD31_SADDR)

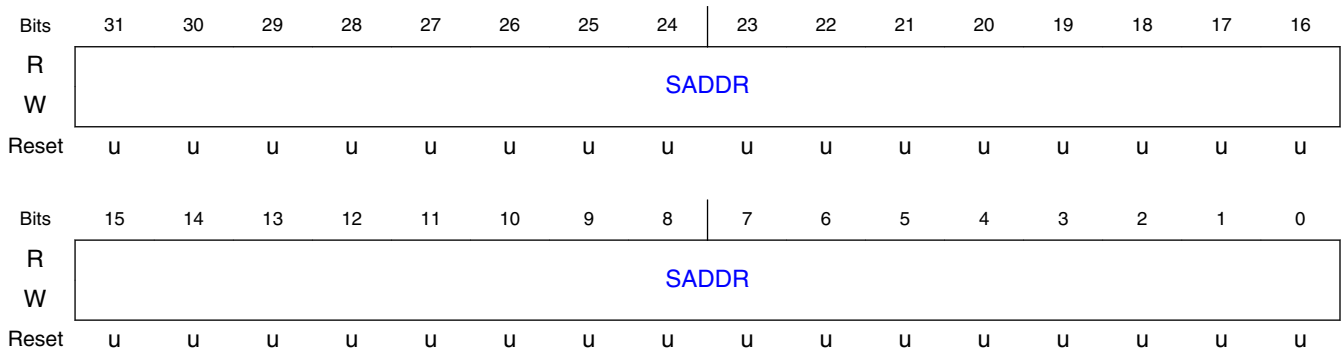
This register contains the source address of the transfer.

6.5.5.19.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SADDR	1000h + (n × 20h)

6.5.5.19.2 Diagram



6.5.5.19.3 Fields

Field	Description
31-0 SADDR	Source Address Memory address pointing to the source data.

6.5.5.20 TCD Signed Source Address Offset (TCD0_SOFF - TCD31_SOFF)

6.5.5.20.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SOFF	1004h + (n × 20h)

6.5.5.20.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.20.3 Fields

Field	Description
15-0 SOFF	Source address signed offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

6.5.5.21 TCD Transfer Attributes (TCD0_ATTR - TCD31_ATTR)

6.5.5.21.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_ATTR	1006h + (n × 20h)

6.5.5.21.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SMOD					SSIZE			DMOD					DSIZE		
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.21.3 Fields

Field	Description
15-11 SMOD	Source Address Modulo

Table continues on the next page...

Field	Description
	Any non-zero value in this field defines a specific address range specified to be the value after SADDR + SOFF calculation is performed on the original register value. Setting this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range. 00000 - Source address modulo feature is disabled 00001-11111 - Value defines address range used to set up circular data queue
10-8 SSIZE	Source data transfer size **NOTE:** 1. Using a reserved value causes a configuration error. 2. The eDMA defaults to privileged data access for all transactions. 000 - 8-bit 001 - 16-bit 010 - 32-bit 011 - 64-bit 100 - Reserved 101 - 32-byte burst (4 beats of 64 bits) 110 - Reserved 111 - Reserved
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination data transfer size See the SSIZE definition.

6.5.5.22 TCD Minor Byte Count (Minor Loop Mapping Disabled) (TCD0_NBYTES_MLNO - TCD31_NBYTES_MLNO)

This register, or one of the next two registers (TCD_NBYTES_MLOFFNO, TCD_NBYTES_MLOFFYES), that defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, is enabled but not used for this channel, or is enabled and used.

TCD word 2 is defined as follows if minor loop mapping is disabled (CR[EMLM] = 0).

If minor loop mapping is enabled, see the TCD_NBYTES_MLOFFNO and TCD_NBYTES_MLOFFYES register descriptions for the definition of TCD word 2.

6.5.5.22.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_NBYTES_MLNO	1008h + (n × 20h)

6.5.5.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NBYTES															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NBYTES															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.22.3 Fields

Field	Description
31-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes are performed until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. **NOTE:** An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.

6.5.5.23 TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (TCD0_NBYTES_MLOFFNO - TCD31_NBYTES_MLOFFNO)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFYES), that defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, is enabled but not used for this channel, or is enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

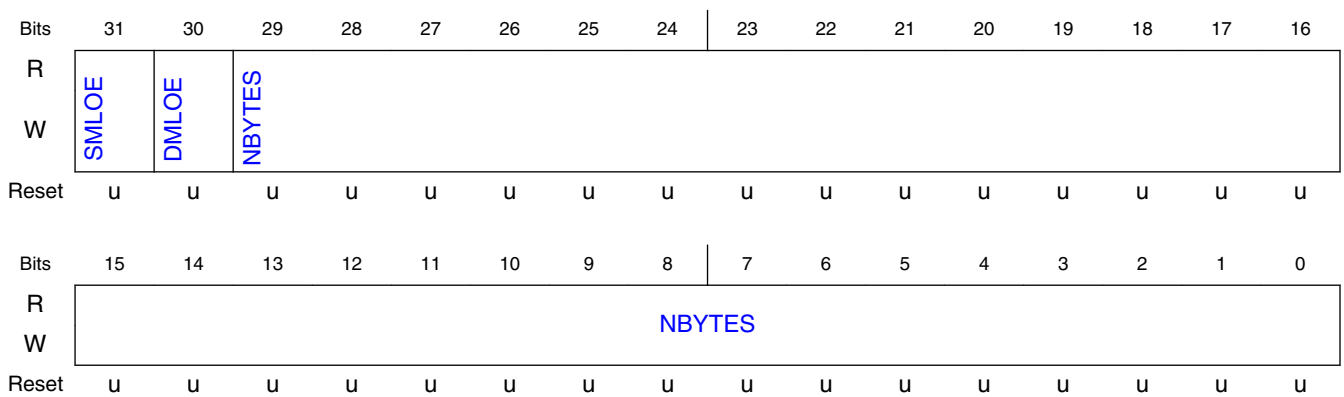
If minor loop mapping is enabled and SMLOE = 1 or DMLOE = 1, refer to the TCD_NBYTES_MLOFFYES register description. If minor loop mapping is disabled, refer to the TCD_NBYTES_MLNO register description.

6.5.5.23.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_NBYTES_MLOFFNO	1008h + (n × 20h)

6.5.5.23.2 Diagram



6.5.5.23.3 Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Specifies whether the minor loop offset is applied to the source address when the minor loop completes. 0 - The minor loop offset is not applied to the SADDR

Table continues on the next page...

Field	Description
	1 - The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset Enable Specifies whether the minor loop offset is applied to the destination address when the minor loop completes. 0 - The minor loop offset is not applied to the DADDR 1 - The minor loop offset is applied to the DADDR
29-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes are performed until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

6.5.5.24 TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (TCD0_NBYTES_MLOFFYES - TCD31_NBYTES_MLOFFYES)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFNO), that defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, is enabled but not used for this channel, or is enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset is enabled (SMLOE or DMLOE = 1)

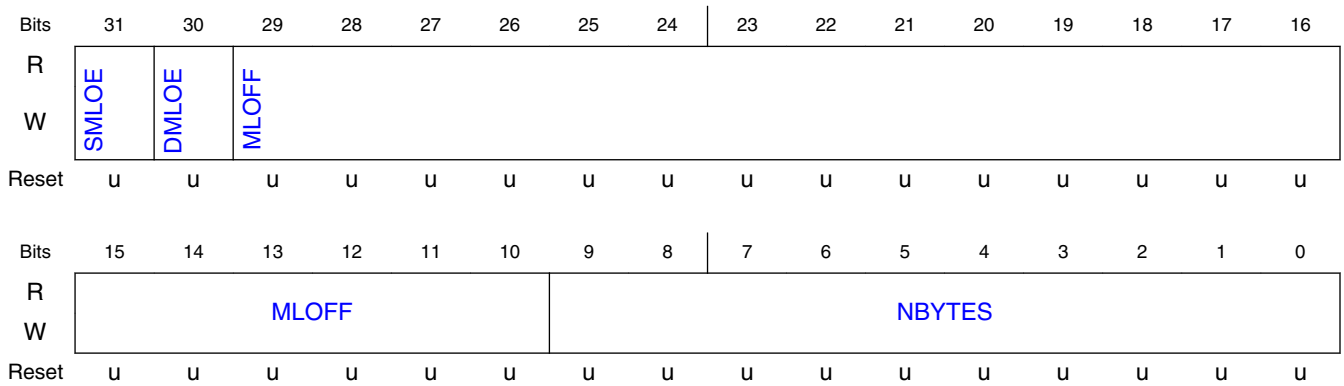
If minor loop mapping is enabled and SMLOE = 0 and DMLOE = 0, refer to the TCD_NBYTES_MLOFFNO register description. If minor loop mapping is disabled, refer to the TCD_NBYTES_MLNO register description.

6.5.5.24.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_NBYTES_MLOFFYES	1008h + (n × 20h)

6.5.5.24.2 Diagram



6.5.5.24.3 Fields

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Specifies whether the minor loop offset is applied to the source address when the minor loop completes. 0 - The minor loop offset is not applied to the SADDR 1 - The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset Enable Specifies whether the minor loop offset is applied to the destination address when the minor loop completes. 0 - The minor loop offset is not applied to the DADDR 1 - The minor loop offset is applied to the DADDR
29-10 MLOFF	If SMLOE = 1 or DMLOE = 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes are performed until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

6.5.5.25 TCD Last Source Address Adjustment (TCD0_SLAST - TCD31_SLAST)

6.5.5.25.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_SLAST	100Ch + (n × 20h)

6.5.5.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SLAST															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLAST															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.25.3 Fields

Field	Description
31-0 SLAST	Last Source Address Adjustment Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure. This register uses two's complement notation; the overflow bit is discarded.

6.5.5.26 TCD Destination Address (TCD0_DADDR - TCD31_DADDR)

This register contains the destination address of the transfer.

6.5.5.26.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DADDR	1010h + (n × 20h)

6.5.5.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.26.3 Fields

Field	Description
31-0 DADDR	Destination Address Memory address pointing to the destination data.

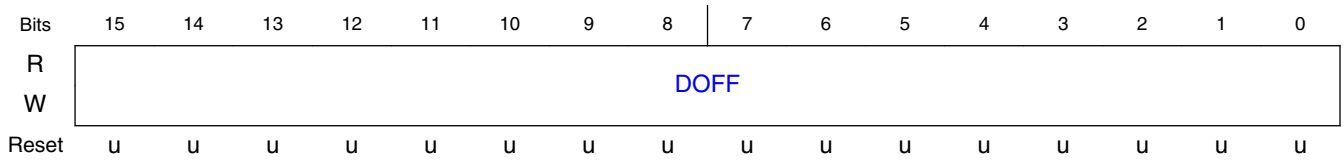
6.5.5.27 TCD Signed Destination Address Offset (TCD0_DOFF - TCD31_DOFF)

6.5.5.27.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DOFF	1014h + (n × 20h)

6.5.5.27.2 Diagram



6.5.5.27.3 Fields

Field	Description
15-0 DOFF	Destination Address Signed Offset Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.

6.5.5.28 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD31_CITER_ELINKNO)

This register contains the minor-loop channel-linking configuration and the channel's current iteration count. It is the same register as TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD31_CITER_ELINKYES), but its fields are defined differently based on the state of the ELINK field. If the ELINK field is 0, this register is defined as follows.

6.5.5.28.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_CITER_ELINKNO	1016h + (n × 20h)

6.5.5.28.2 Diagram

6.5.5.28.3 Fields

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this field enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets TCDn_CSR[START] of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This field must be equal to BITER[ELINK]; otherwise, a configuration error is reported. 0 - Channel-to-channel linking is disabled 1 - Channel-to-channel linking is enabled
14-0 CITER	Current Major Iteration Count This field is the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations. It optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** 1. When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. 2. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

6.5.5.29 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD31_CITER_ELINKYES)

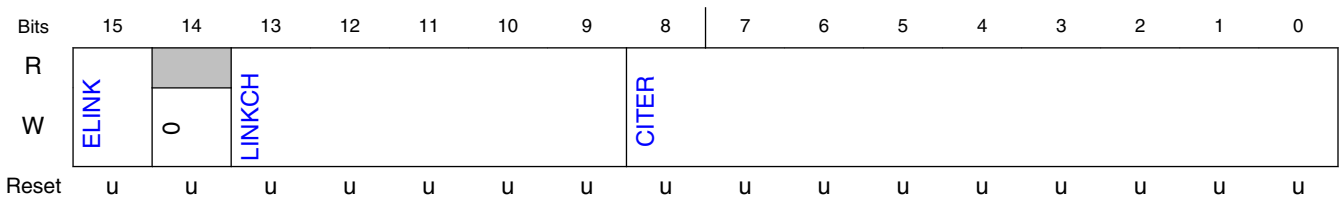
This register contains the minor-loop channel-linking configuration and the channel's current iteration count. It is the same register as TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD31_CITER_ELINKNO), but its fields are defined differently based on the state of the ELINK field. If the ELINK field is 1, this register is defined as follows.

6.5.5.29.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_CITER_ELINKYES	1016h + (n × 20h)

6.5.5.29.2 Diagram



6.5.5.29.3 Fields

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this field enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets TCDn_CSR[START] of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This field must be equal to BITER[ELINK]; otherwise, a configuration error is reported. 0 - Channel-to-channel linking is disabled 1 - Channel-to-channel linking is enabled
14 —	Reserved
13-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by this field, by setting that channel's TCDn_CSR[START].
8-0 CITER	Current Major Iteration Count This field is the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations. It optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Field	Description
	NOTE: 1. When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. 2. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

6.5.5.30 TCD Last Destination Address Adjustment/Scatter Gather Address (TCD0_DLASTSGA - TCD31_DLASTSGA)

6.5.5.30.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_DLASTSGA	1018h + (n × 20h)

6.5.5.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLASTSGA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLASTSGA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

6.5.5.30.3 Fields

Field	Description
31-0 DLASTSGA	Destination last address adjustment, or next memory address TCD for channel (scatter/gather) If (TCDn_CSR[ESG] = 0) then: • This is the adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise:

Field	Description
	• This address points to the beginning of a 0-modulo 32-byte region containing the next TCD to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte; otherwise a configuration error is reported.

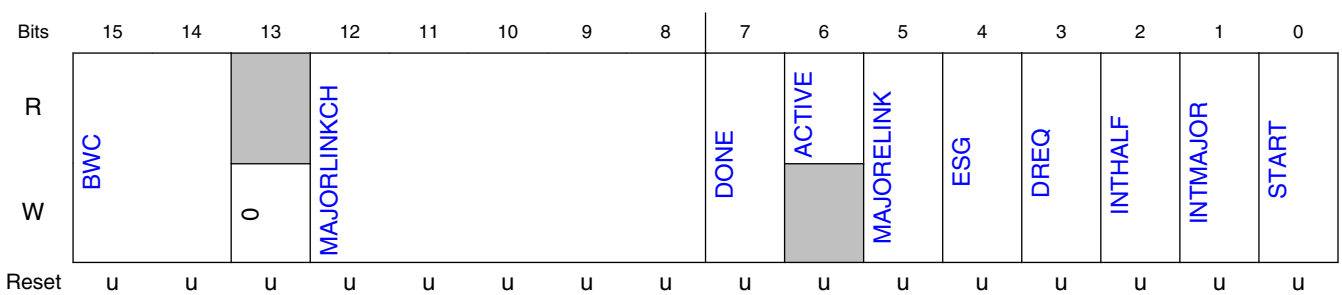
6.5.5.31 TCD Control and Status (TCD0_CSR - TCD31_CSR)

6.5.5.31.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_CSR	101Ch + (n × 20h)

6.5.5.31.2 Diagram



6.5.5.31.3 Fields

Field	Description
15-14	Bandwidth Control
BWC	Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00 - No eDMA engine stalls 01 - Reserved

Table continues on the next page...

Field	Description
	10 - eDMA engine stalls for 4 cycles after each R/W 11 - eDMA engine stalls for 8 cycles after each R/W
13 —	Reserved
12-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's START bit.
7 DONE	Channel Done This field indicates whether the eDMA has completed the major loop. The eDMA engine sets the value of this field to 1 when the CITER count reaches zero. The value of this field is reset to 0 by the hardware (when the channel is activated) or by software. **NOTE:** This field must be 0 to write the MAJORELINK or ESG fields.
6 ACTIVE	Channel Active This field indicates whether the channel is currently in execution. The eDMA sets the value of this field to 1 when channel service begins, and resets it to 0 as the minor loop completes or when any error condition is detected.
5 MAJORELINK	Enable channel-to-channel linking on major loop complete As the channel completes the major loop, this field controls linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets TCDn_CSR[START] of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to zero when written to when TCDn_CSR[DONE] is set. 0 - Channel-to-channel linking is disabled 1 - Channel-to-channel linking is enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this field controls scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure loaded as the TCD into local memory. **NOTE:** To support the dynamic scatter/gather coherency model, this field is forced to zero when written to when TCDn_CSR[DONE] is set. 0 - The current channel's TCD is normal format 1 - The current channel's TCD specifies a scatter gather format
3 DREQ	Disable Request If the value of this field is 1, eDMA hardware automatically writes 0 to the corresponding ERQ field when the current major iteration count reaches zero. 0 - The channel's ERQ field is not affected 1 - The channel's ERQ field value changes to 0 when the major loop is complete
2 INTHALF	Enable an interrupt when major counter is half complete. If the value of this field is 1, the channel generates an interrupt request by setting the appropriate field in the INT register when the current major iteration count reaches the halfway point. Specifically, the

Table continues on the next page...

Field	Description
	comparison performed by the eDMA engine is (CITER == (BITER >> 1)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE:** If BITER = 1, do not use INTHALF. Use INTMAJOR instead. 0 - Half-point interrupt is disabled 1 - Half-point interrupt is enabled
1 INTMAJOR	Enable an interrupt when major iteration count completes. If the value of this field is 1, the channel generates an interrupt request by setting the appropriate field in the INT when the current major iteration count reaches zero. 0 - End of major loop interrupt is disabled 1 - End of major loop interrupt is enabled
0 START	Channel Start If the value of this field is 1, the channel is requesting service. eDMA hardware automatically writes 0 to this field after the channel begins execution. 0 - Channel is not explicitly started 1 - Channel is explicitly started via a software initiated service request

6.5.5.32 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD31_BITER_ELINKNO)

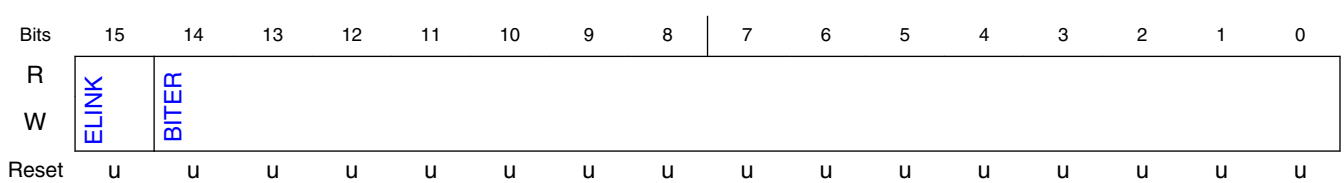
If TCDn_BITER[ELINK] is 0, the TCDn_BITER register is defined as follows.

6.5.5.32.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_BITER_ELINKNO	101Eh + (n × 20h)

6.5.5.32.2 Diagram



6.5.5.32.3 Fields

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this field enables linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets TCDn_CSR[START] of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0 - Channel-to-channel linking is disabled 1 - Channel-to-channel linking is enabled
14-0 BITER	Starting Major Iteration Count As the TCD is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

6.5.5.33 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD31_BITER_ELINKYES)

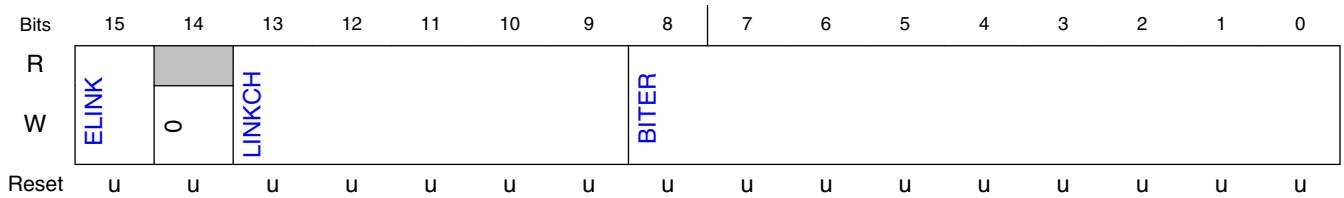
If TCDn_BITER[ELINK] is 1, the TCDn_BITER register is defined as follows.

6.5.5.33.1 Offset

For n = 0 to 31:

Register	Offset
TCDn_BITER_ELINKYES	101Eh + (n × 20h)

6.5.5.33.2 Diagram



6.5.5.33.3 Fields

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this field enables linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets TCDn_CSR[START] of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0 - Channel-to-channel linking is disabled 1 - Channel-to-channel linking is enabled
14 —	Reserved
13-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field, by setting that channel's TCDn_CSR[START]. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.
8-0 BITER	Starting major iteration count As the TCD is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

Chapter 7
System Security

7.1 Chapter overview

This chapter provides an overview of the following chip security components, explaining the purpose and features of each of them.

- System Boot ROM including High Assurance Boot (HAB)
- Secure Non-Volatile Storage (SNVS) with real-time clock, zero master key, and tamper protection
- Random Number Generator (RNG)
- Cryptographic Acceleration and Assurance Module (CAAM)
- On-chip One-Time Programmable Element Controller (OCOTP_CTRL) with on-chip electrical fuse arrays
- Secure key manager (KEYMGR) and protection
    - Physical Unclonable Function (PUF)
    - UnDocumented Function (UDF)
- JTAG Controller (JTAGC) and SWD debug interfaces
- Inline Encryption Engine (IEE)
- On-the-Fly AES Decryption (OTFAD)
- Resource Domain Controller (RDC) and Extended Resource Domain Controller (XRDC) are used to support both single-core and multi-core use cases with maximum flexible resource management
- Code WatchDog timer (CDOG)

The detailed description of each component can be found in the Security Reference Manual for this chip.

7.2 Feature summary

This figure shows a simplified diagram of the security subsystem:

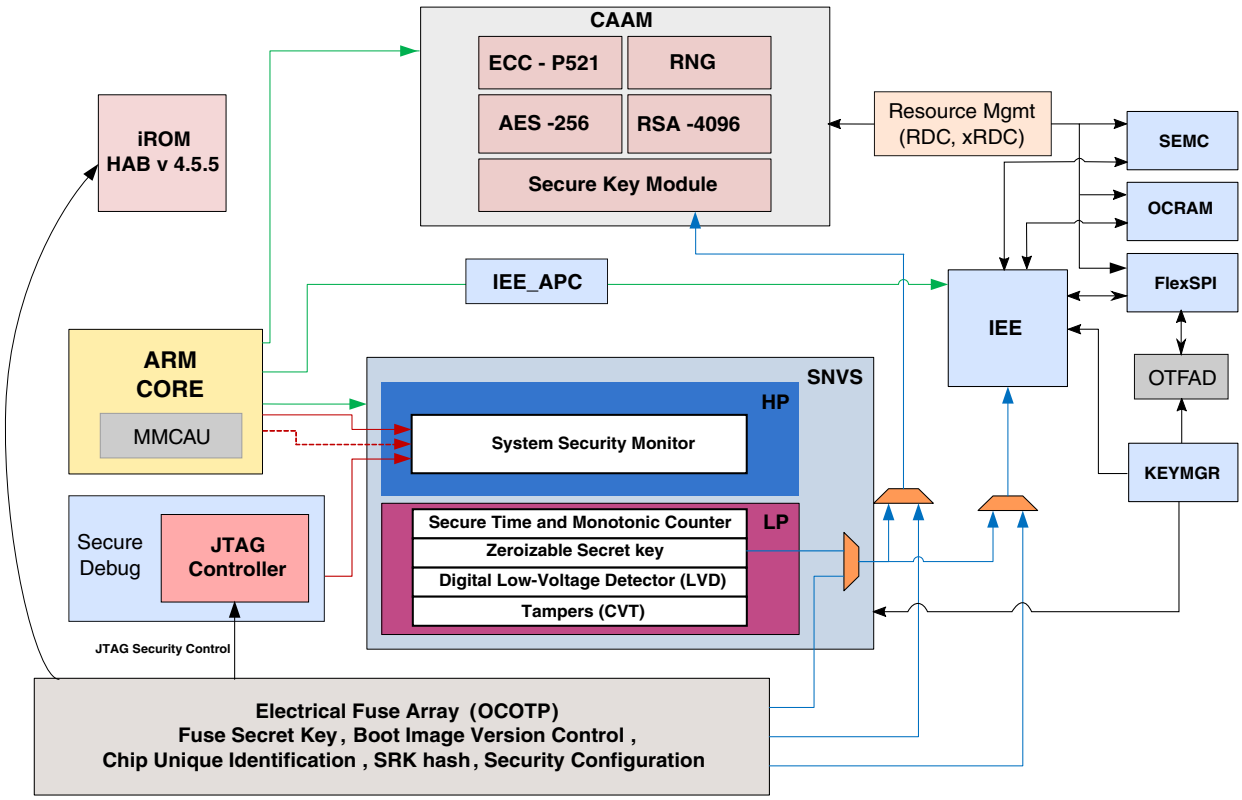


Figure 7-1. Security subsystem (simplified)

All platforms built using this chip share a general need for security, though the specific security requirements vary greatly from platform to platform. For example, portable consumer devices need to protect a different type and cost of assets than automotive or industrial platforms. Each market must be protected against different kinds of attacks. The platform designers need an appropriate set of countermeasures to meet the security needs of their specific platform.

To help the platform designers to meet the requirements of each market, the chip incorporates a range of security features. Most of these features provide protection against specific kinds of attack, and can be configured for different levels according to the required degree of protection. These features are designed to work together or independently. They can be also integrated with the appropriate software to create defensive layers. In addition, the chip includes a general-purpose accelerator that enhances the performance of selected industry-standard cryptographic algorithms.

The security features include:

- High Assurance Boot and encrypted boot
- Secure storage

    - Off-chip storage protection using AES-256 and the chip's unique hardware-only key
    - 4KB Secure RAM (SRAM) in SNVS
    - Zeroizable Master Key (256 bits)
    - SNVS General Purpose Registers (256 bits)
- Cryptographic Acceleration and Assurance Module (CAAM)
    - Symmetric Engines - AES 128, 256 with baseline modes (additional modes include GCM, CMAC), 3DES, DES
    - Public Key Cryptography Engine (PKHA): RSA up to 4096 key length, elliptic curve (supporting NIST, Brainpool)
    - Manufacturing protection
    - 64-bit multiplier for V2X performance requirements (500+ NIST P-256 signatures/sec)
    - Cryptographic Hash Engine: SHA-1, SHA-2 224/256/384/512, MD5, and HMAC
    - Random Number Generator (RNG)
        - True random entropy source
        - NIST-certified Deterministic Random Bit Generator (Hash-based)
    - Secure Hardware-only Cryptographic Key Management
    - Encrypted boot
    - Revision control check based on fuse values
    - Data Encryption Key (DEK) includes IV
    - Real-Time Integrity Checker (RTIC)
- Secure debug
    - 128-bit OTP debug authentication key
- Electrical fuses (OTP Memory)
- Inline Encryption Engine (IEE)
    - SDRAM encryption/decryption
    - Secure scan
    - OCRAM encryption/decryption
    - I/O direct encrypted storage and retrieval (Stream support)
    - FlexSPI decryption only (256-bit AES-XTS mode XIP)
    - Transparency to software during encrypted access (no configuration, control, or interrupts)
    - Secure on-chip key loading using private bus
- AES-128 counter mode On-The-Fly Decryption (OTFAD)
    - 128-bit key and 128-bit data sizes
    - Receives 64-bit encrypted data from FlexSPI
    - Hardware support for unwrapping key blobs
    - Acts as a slave sub-module to FlexSPI
- Secure Non-Volatile Storage (SNVS)

- Secure real-time clock (SRTC)
- Tamper protection
- DryICE

7.3 Resource Management

To support both single-core and multi-core use cases with max flexible resource management, both Resource Domain Controller (RDC) and Extended Resource Domain Controller 2 (XRDC) are used.

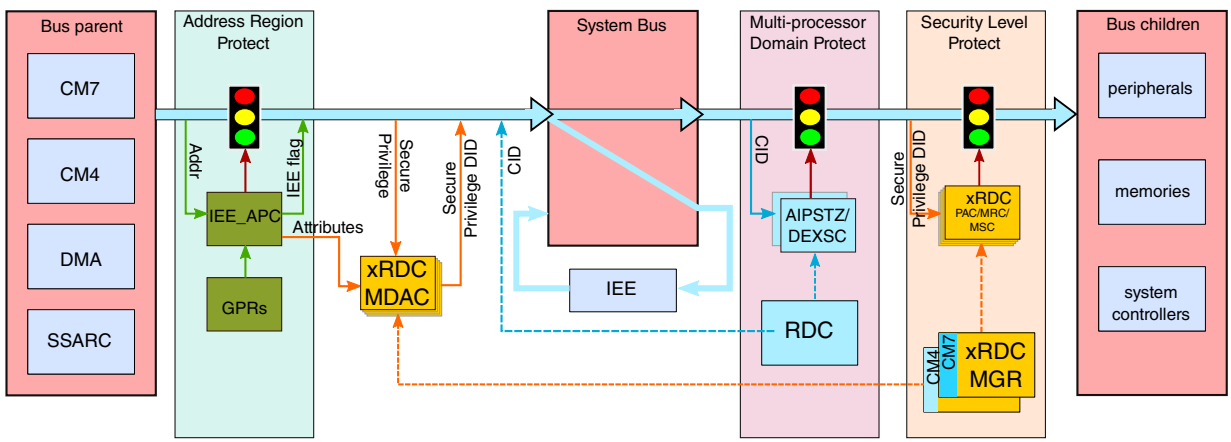


Figure 7-2. Resource Management

- RDC is used as first-level control:
  - Protects the access to resources based on CPU domain assignment in RDC
  - Resources of the chip are assigned to two CPU domains as follows
    - Private resource : Used by either CPUs
    - Shared resource: Used by both CPUs
- XRDC is used as second-level control:
  - Protects the access to resources based on access type (Secure/non-secure access and Privileged/user access)
  - Each CPU has its own XRDC, which supports up to 16 domains

7.4 Secure Key Manager (KEYMGR)

The Key Manager is used to generate, manage, fetch and configure keys for SNVS, OTFADs, and IEE. For better key generation and management, this block integrates UnDocumented Function (UDF) and Physical Unclonable Function (PUF).

- UnDocumented Function (UDF)
  - Provides a deterministic cryptographic function whose characteristics are not standard nor documented.
  - Generates a deterministic, one-way cryptographic function for obfuscation purposes
  - Supports the ability to include cryptographic “salt” to alter UDF calculation
  - Provides mechanisms to lock and reset
- Physical Unclonable Function (PUF)
  - Key generated by silicon entropy, which is silicon-unique and unclonable fingerprint

7.5 High-Assurance Boot (HAB)

The HAB, which is the high-assurance boot feature in the system boot ROM, detects and prevents the execution of unauthorized software (malware) during the boot sequence.

When the unauthorized software is permitted to gain control of the boot sequence, it can be used for a variety of goals, such as exposing stored secrets, circumventing access controls to sensitive data, services, or networks, or for re-purposing the platform. The unauthorized software can enter the platform during upgrades or re-provisioning, or when booting from the USB connections or removable devices.

The HAB protects against unauthorized software by:

- Using digital signatures to recognize the authentic software. This enables you to boot the device to a known initial state and run the software signed by the device manufacturer.
- Using code encryption to protect the confidential software during off-chip storage. When activated, the HAB decrypts the software loaded into the RAM before execution.

7.5.1 HAB process flow

The following figure shows the flow for creating and verifying digital signatures. The top half of this figure shows the signing process, which is performed off-chip. The bottom half shows the verification process performed on-chip during every system boot.

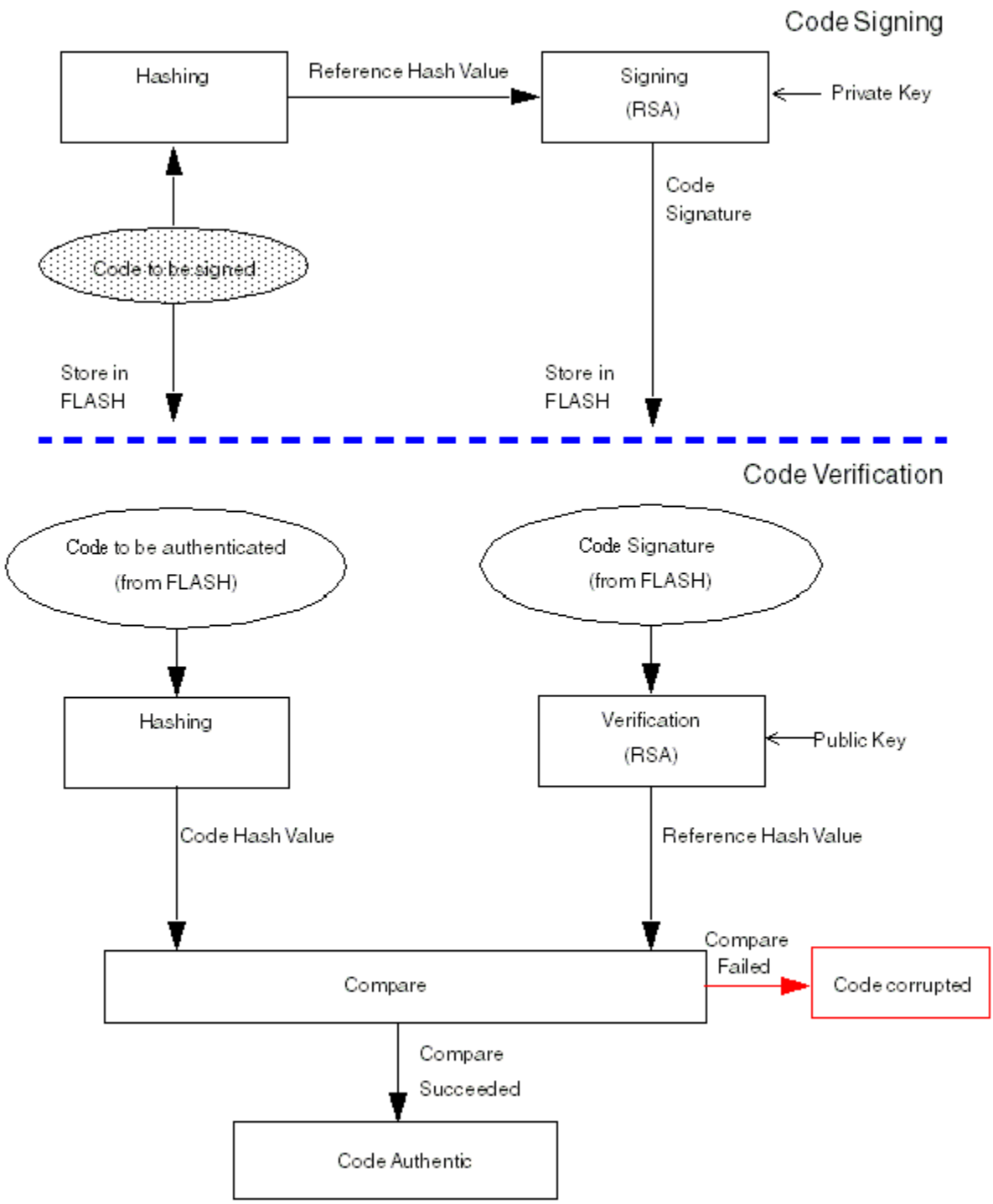


Figure 7-3. Code signing and authentication processes

The original software is programmed into the flash memory (or any other boot device) along with the signature. The HAB uses a public key to recover the reference hash value from the signature; it then compares the reference hash value to the current hash value

calculated from the software in the flash. If the contents of the flash are modified either intentionally or unintentionally, the two hash values do not match and the verification fails.

NOTE

Use OTFAD for providing software confidentiality when the user code executes in place.

7.5.2 HAB feature summary

The HAB features:

- Enforced internal boot via on-chip masked ROM
- Authentication of software loaded from any boot device (including the USB download)
- Authenticated decryption of software loaded from any boot device (including USB download) using the AES keys (128-bit, 256-bit)
- CMS PKCS#1 signature verification using RSA public key (1024, 2048, 3072 or 4096) and the SHA256 hash algorithm
- ECDSA signature verification using ECC public key (P256, P384, P521) and the SHA256 hash algorithm
- Public Key Infrastructure (PKI) support using X.509v3 certificates
- Root public key fingerprint in the manufacturer-programmable on-chip fuses
- Multiple root public keys with revocation by fuses
- Initialization of other security components
- Supports fall-through USB downloader if the primary or recovery boot fails
- Open configuration for development purposes and non-secure platforms
- Closed configuration for shipping secure platforms

On the chip, the HAB is integrated with these security features:

- The HAB initializes the SNVS security monitor state machine. A successful secure boot with the HAB is required for the platform software to gain access to use the CAAM master secret key selected by SNVS.
- The HAB reads the root public key fingerprint, revocation mask, and security configuration from the OCOTP_CTRL.
- Depending on the algorithm, HAB will use the CAAM to accelerate some crypto operations:
  - ECC is accelerated by CAAM
  - RSA is handled by software within HAB

- AES is accelerated by CAAM
- SHA is handled by HAB software or CAAM accelerated depending on the signature command sequence file (CSF)

7.6 Secure Non-Volatile Storage (SNVS) module

- Provides a non-volatile real-time clock maintained by a coin-cell battery during system power down for use in both the secure and non-secure platforms.
- Protects the secure real-time clock against rollback attacks in time-sensitive protocols such as DRM and PKI
- Deters replay attacks in time-independent protocols such as certificate or firmware revocations
- Handles security violation detection and reporting, to defend sensitive data and operations against compromise, both at run-time and during system power-down
- Controls the access to the OTP master secret key used by the CAAM to protect confidential data in the off-chip storage
- Provides non-volatile highly protected storage for an alternative master secret key
- Provides tamper detection
- Provides a 1K bit register (GPRs) protected by tamper
- Supports wakeup source

7.6.1 SNVS architecture

The SNVS is partitioned into two sections: a low-power part (SNVS_LP) and a high-power part (SNVS_HP).

The SNVS_LP block is in the always-powered-up domain. It is isolated from the rest of the logic by isolation cells which are library-instantiated cells that insure that the powered-up logic is not corrupted when the power goes down in the rest of the chip.

The SNVS_LP has these functional units:

- Zeroizable Master Key
- Secure non-rollover real-time counter with alarm
- Non-rollover monotonic counter
- Digital Low-Voltage Detector (LVD)
- General-purpose register
- Control and status registers
- 4KB Secure RAM (SRAM)
- Tamper detection

The SNVS_HP is in the chip power-supply domain. The SNVS_HP provides an interface between the SNVS_LP and the rest of the system. The access to the SNVS_LP registers can be gained through the SNVS_HP only when it is powered up according to the access permission policy.

The SNVS_HP has these functional units:

- IP bus interface
- SNVS_LP interface
- System Security Monitor (SSM)
- Zeroizable Master Key programming mechanism
- Master Key control block
- Non-secure real-time counter with alarm
- Control and status registers
- High Assurance Counter (HAC)

7.6.2 DryICE

The SNVS interface contains mixed logic called DryICE, which provides a 32 kHz system clock, and has voltage, temperature, and clock (VTC) tamper detection monitors. Once a tamper is detected for VTC, a security violation will be sent to the main digital SNVS peripheral. Please see the IOMUXC SNVS GPRs for related DryICE programming.

Table 7-1. DryICE Tampers

Tamper Monitor	Description
Voltage Detector	Checks whether voltages from battery and regulator are normal or not. If the voltage is out-of-range, the out-of-range flag will assert high to SNVS Digital tamper logic.
Temperature Detector	Determins whether the current temperature is appropriate. If the temperature is out-of-range, the out-of-range flag will assert high to SNVS Digital tamper logic.
Clock Detector	Monitors whether the internal 32 kHz crystal oscillator is in it's normal operation. If the XTALOSC 32 kHz clock stops or is out-of-range, the internal 32 kHz source will be switched, and the out-of-range flag will assert high to SNVS Digital tamper logic.

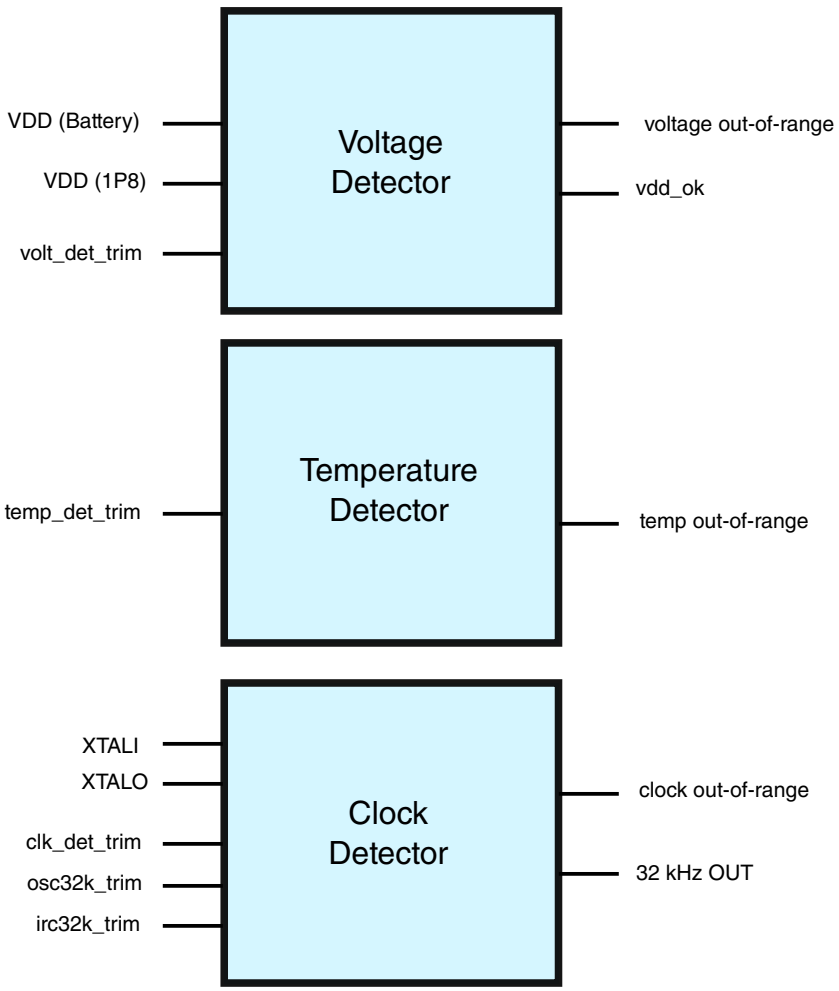


Figure 7-4. DryICE Block Diagram

7.7 Cryptographic Acceleration and Assurance Module (CAAM)

7.7.1 CAAM Overview

CAAM is a cryptographic acceleration module that accelerates block encryption algorithms, stream cipher algorithms, hashing algorithms, and random number generation. It has an integrated DMA engine that allows CAAM to fetch its command programs, read input data and write the resulting output.

CAAM works with the SNVS to provide platform assurance features, including support for High Assurance Boot, detection of and response to potential tamper events, and short term and long-term protection of secret data such as public keypairs, Digital Rights Management keys and proprietary software.

CAAM provides the following features:

- Public Key Cryptography Engine (PKHA)
  - RSA up to 4096 key length; Elliptic Curve (up to P-521)
  - Manufacturing Protection
  - Elliptic supports NIST, Brainpool
  - 64-bit Multiplier for V2X performance requirements (500+ NIST P-256 signatures/sec)
- Symmetric Engines
  - AES 128, 192, 256 with baseline modes
  - AES additional modes include GCM, CMAC
  - 3DES, DES
- Cryptographic Hash Engine
  - SHA-1, SHA-2 224/256/384/512, MD5
  - HMAC
- Random Number Generation (RNG)
  - True random Entropy Source
  - NIST-certified Deterministic Random Bit Generator (Hash-based)

7.8 On-Chip OTP Controller (OCOTP_CTRL)

The OCOTP_CTRL provides the primary user-visible mechanism for interfacing with the on-chip fuses. These fuses' uses include:

- Unique chip identifiers
- Mask revision numbers
- Cryptographic keys
- Security configuration
- Boot characteristics
- Various control signals requiring permanent non-volatility

The OCOTP_CTRL provides:

- Shadow cache of fuse values, loaded at reset, before the system boot
- Ability to read and override the fuse values in the shadow cache (does not affect the fuse element)
- Ability to read the fuses directly (ignoring the shadow cache)
- Ability to write (program) the fuses by software or JTAG
- Fuses and shadow cache bits enforce read-protect, override-protect, and write-protect
- Lock fuses for selected fuse fields
- Scan protection
- Volatile software-accessible signals which can be used for software control of hardware elements (not requiring non-volatility).

7.9 Debug

The debug interface is connected to the DAP and CoreSight debug modules to allow debug through the JTAG interface. The key features of the system debug is listed below. For more information on the CoreSight debug components, please see the System Debug chapter in the Reference Manual.

- 5-pin JTAG and SWD interface (fuse configured)
- Non-intrusive and halt-mode trace / debug options
- Secure Debug with 128-bit protection key
- ARM real-time trace interface (TPIU)

The security levels are selected via the eFuse configuration.

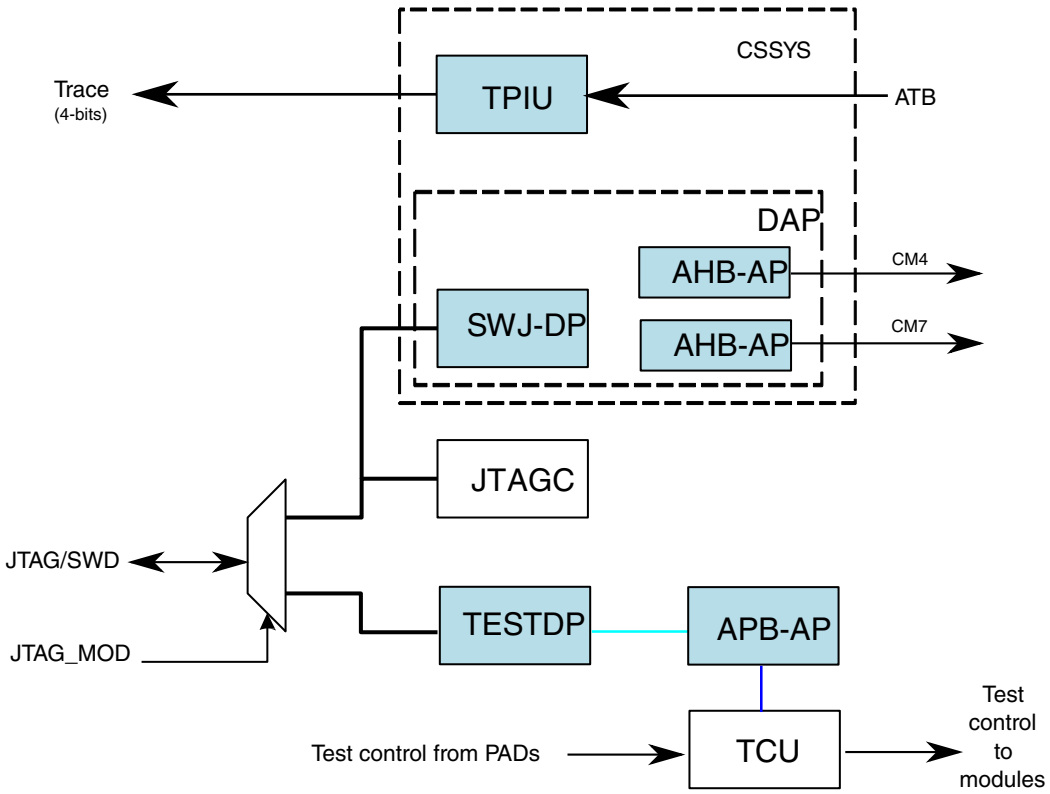


Figure 7-5. System-Level Debug Architecture

7.9.1 JTAG Controller (JTAGC)

There is only one JTAG interface on the chip. Two JTAG modes are supported, which are selected via the JTAG_MOD pin:

- Debug Mode (JTAG_MOD == 0)
  - DAP and JTAGC share the same JTAG by using exclusive instructions.
  - TESTDP will be bypassed.
- Test Mode (JTAG_MOD == 1)
  - TESTDP is the only TAP controller in the daisy chain (IEEE1149.1 compliant).

The JTAGC provides debug access to hardware blocks, including the arm processor and the system bus. This enables program control and manipulation as well as visibility to the chip peripherals and memory.

The JTAG port must be accessible during initial platform development, manufacturing tests, and general troubleshooting. Given its capabilities, the JTAGC provides these security levels:

- JTAG Disabled - JTAG use is permanently blocked.
- No Debug Mode - All security sensitive JTAG features are permanently blocked.

- Secure JTAG Mode - JTAG use is restricted (as in the No-Debug level) unless a secret-key challenge/response protocol is successfully executed.
- JTAG Enabled - JTAG use is unrestricted.

The security levels are selected via the eFuse configuration.

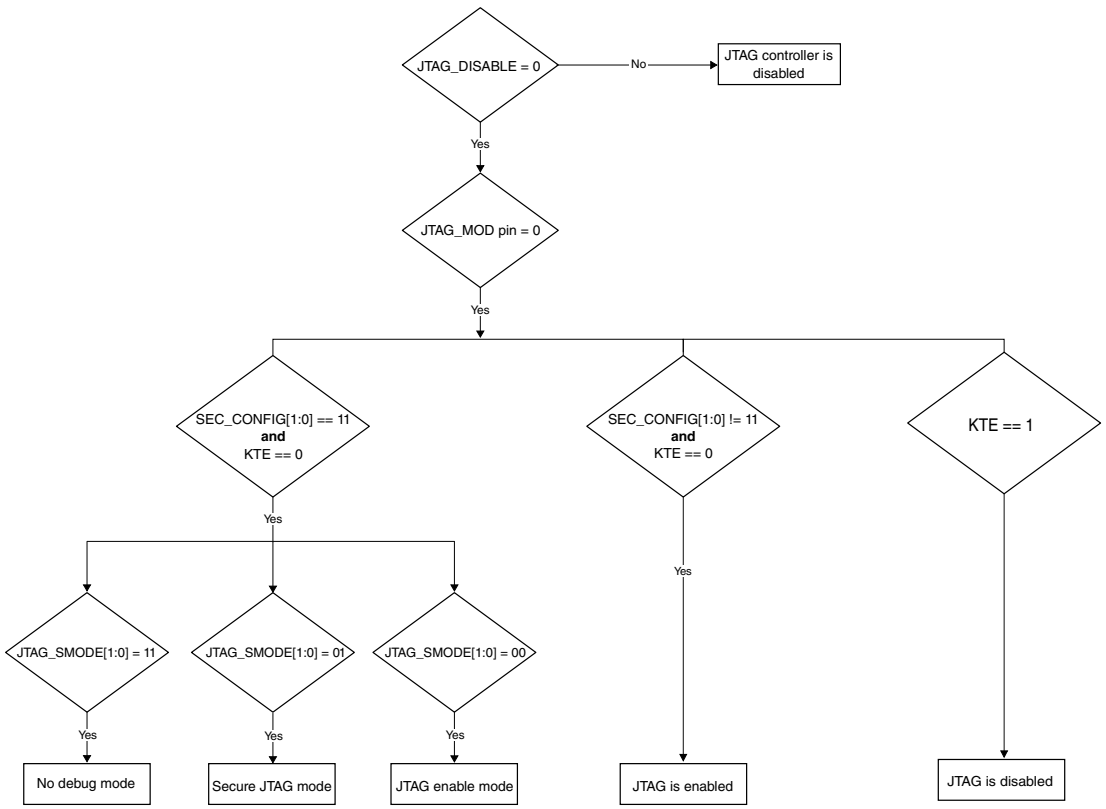


Figure 7-6. Secure JTAGC Configuration

7.10 Inline Encryption Engine (IEE)

IEE provides a means to perform inline encryption and decryption.

It provides the following features:

- SDRAM encryption/decryption
- FlexSPI decryption only
- Secure scan
- Secure on-chip key loading
- I/O direct encrypted storage and retrieval (Stream Support)

7.11 On-the-Fly AES Decryption (OTFAD)

OTFAD works with FlexSPI to provide superior cryptographic decryption capabilities without compromising system performance.

It has the following features:

- AES-128 Counter Mode On-the-Fly Decryption
  - 128-bit key and 128-bit data sizes
  - Receives 64-bit encrypted data from FlexSPI
- Functionally acts as a slave submodule to the FlexSPI
  - Programming model mapped into the upper 1 KB of the FlexSPI's IPS address space

7.12 Memory-Mapped Cryptographic Acceleration Unit (MMCAU)

The Memory-Mapped Cryptographic Acceleration Unit (MMCAU) is a coprocessor that supports the hardware implementation of a set of specialized operations to improve the throughput of software-based security encryption/decryption operations and message digest functions. The MMCAU supports acceleration of the DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms. This module is provided with an optimized C-function library that provides the appropriate software building blocks to implement higher-level security functions. The MMCAU is connected to the CM4 core's private peripheral bus (PPB). This bus is not accessible to the CM7 core, so the MMCAU can only be used by the CM4 core.

7.13 Code Watchdog Timer (CDOG)

The Code Watchdog Timer (CDOG) module helps protect the integrity of software by detecting unexpected changes (faults) in code execution flow. The CDOG provides two primary mechanisms for detecting side-channel attacks and the execution of unexpected instruction sequences:

- Secure Counter (SECURE_COUNTER)
- Instruction Timer (INSTRUCTION_TIMER)

The Secure Counter is a 32-bit accumulating register that holds a dynamically changing value that can periodically be evaluated to determine if a program is executing as expected. If a mismatch is detected, a fault is generated. The Instruction Timer is a 32-bit

count-down timer that an application uses to set the number of instructions that software expects to execute. The Instruction Timer counts off the instructions as they are executed (clocks) and if the number is exhausted before the CDOG is serviced, a fault is generated.

Chapter 8
System Debug

8.1 Overview

This section describes the hardware and software debug and application development features and resources of the chip. It describes the following:

- Core/platform-specific resources
- Resources associated with complex IP blocks
- Chip-wide resources
- Interface to the external debug and development tools

The debug and trace architecture is designed around the following:

- Arm CoreSight architecture, adapted to SoC (for core debug), including a cross-trigger subsystem for cross-domain triggering of debug resources
- JTAG port used to interact with core under the debug by means of JTAGC, the JTAG Controller
- DAP, the debug access port that supports the interface to the Arm RealView Debugging tools and other third-party tools
- TPIU, a trace port interface unit that efficiently accesses the program trace information from the system
- Various chip-wide resources, such as debug features built into the IP blocks and critical signal visibility available through alternate pin functions or observability muxes

8.2 Chip and Arm Platform Debug Architecture

The Arm Debug architecture is based on the CoreSight architecture by Arm. The CoreSight architecture provides a system-wide solution to real-time debug and trace.

The CoreSight architecture is embodied in a set of CoreSight components and compliant processors that form the CoreSight systems. Its architecture maintains the traditional requirements of debug and trace:

- To access the debug functionality without software interaction
- To connect to a running system without performing a reset

Full access to the processor debug capability is available by the Arm debug register map through the Advanced Peripheral Bus (APB) slave port. The core includes a Processor Debug Unit which stops program execution, examines and alters the processor and coprocessor state, examines and alters the memory and input/output peripheral state, and restarts the processor core.

The following diagram shows the system level debug architecture:

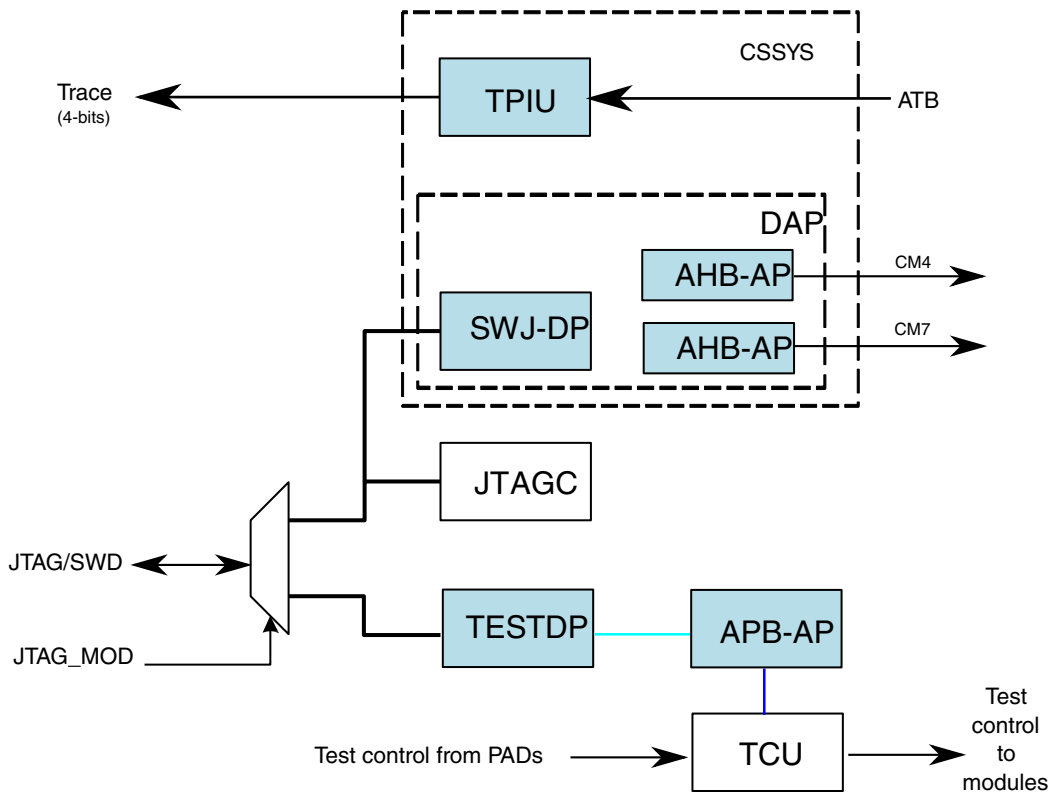


Figure 8-1. System Level Debug Architecture

Table 8-1. List of supported MEM-APs

APSEL	Access Port	IDR
0	CM7 (AHB-AP)	0x84770001
1	CM4 (AHB-AP)	0x24770011
2	APB-AP	0x54770002

8.2.1 Debug Features

- EmbeddedICE-RT logic
  - Support for both the monitor-mode and halt-mode debugging:
  - Core run/halt control, debug status/control
  - Breakpoint/watchpoint control
  - Core-mapped and memory-mapped resource examination/modification
- Data communication channel between the Arm core and the host debugger via JTAG or SWD and the Debug Access Port (DAP) module

The chip includes Arm CoreSight components for debug and trace solutions.

8.2.2 Debug system components

The CoreSight components include:

- ETM (Embedded Trace Macrocell) supporting instruction trace
- ITM (Instrumentation Trace Macrocell)
- TPIU (Trace Port Interface)
- Cross Triggering logic for event routing (CTIs)
- Timestamp Generator (TSGEN)
- Data Watchpoint and Trace (DWT)

Other related IPs and functionality:

- Flash Patch and Breakpoint unit (FPB)

FPB for CM4 cannot be used since the flash address is beyond 0x20000000.

8.2.2.1 AMBA Trace Bus (ATB)

ATB transfers trace data though the chip CoreSight infrastructure. The trace sources are ATB masters and the sinks are ATB slaves. The Arm (via PTM) cores are the data generators. Link components such as the Trace Funnel and Replicator provide both the master and slave interfaces.

The ATB protocol supports:

- Stalling of trace sources to enable the CoreSight components to funnel and combine the sources into a single trace stream

- Association of the trace data with the generating source using trace source IDs. The CoreSight system can trace up to 111 different items at any time
- Capture and transfer of multiple byte bus widths, currently to 32 bits
- A flushing mechanism to force the historic trace to drain from any sources, links, or sinks up to the point that the request is initiated

8.2.2.2 CoreSight Trace Port Interface (TPIU)

TPIU is one of the CoreSight trace sink components. It acts as a bridge between the on-chip trace data and a data stream that is then driven out the trace port.

TPIU uses the ATB interface to accept trace data from a trace source, either directly or by using a trace funnel. TPIU has 4 bit port connected to the chip pad.

The APB interface is the programming interface for the TPIU configuration.

The features of the sub-blocks are as follows:

- Formatter—Inserts source ID signals into the data packet stream so that the trace data can be re-associated with the trace source.
- Asynchronous FIFO—Enables trace data to be driven out at a speed that is not dependent on the on-chip bus clock.
- Register Bank—Contains the management, control and status registers for triggers, flushing behavior and external control.
- Trace out—The Trace out block serializes the formatted data before it goes off-chip.
- Pattern Generator—The Pattern Generator unit provides a simple set of defined bit sequences or patterns that can be output over the Trace Port and be detected by the TPA or other associated Trace Capture Device (TCD). The TCD can use these patterns to indicate if it is possible to increase or decrease the trace port clock speed.

The output of the TPIU is connected via external pins (MPS of TRACEDATA (ARM_TRACEn), which can be 1, 2, or 4 bits). The system may utilize double data rate pins to either use a lower clock speed than that of the 32-bit ATB interface, or use fewer than 4 data pins for the output, based on the ability of the technology used. Given the speed of the ATB, 4 data pins with double data rate is recommended, with the external interface running at half the speed of the ATB. The speed of the external interface is from TRACECLKIN (CSTRACE_CLK_ROOT). TRACECLK (ARM_TRACE_CLK) is equal to TRACECLKIN / 2, and is divided in the TPIU to clock trace data at the trace capture unit.

TPIU used to be part of the Arm platform sub blocks in previous versions of i.MX products, placing the TPIU off platform allows future debug trace sources from the chip level to connect to the TPIU by means of a funnel.

For more information, see Arm Cortex M4 Platform chapter.

For more information, see Arm Cortex M7 Integration and Implementation Manual.

8.2.2.3 Embedded Trace Macrocell v3.5 (ETM)

The Cortex-M4 Embedded Trace Macrocell (ETM-M4) is a debug component that enables a debugger to reconstruct program execution. The CoreSight ETM-M4 supports only instruction trace. You can use it either with the CoreSight Trace Point Interface Unit (TPIU).

The main features of an ETM are:
- Tracing of 16-bit and 32-bit Thumb instructions
- Four EmbeddedICE watchpoint inputs
- A Trace Start/Stop block with EmbeddedICE inputs
- One reduced function counter
- Two external inputs
- A 24-byte FIFO queue
- Global timestamping

8.2.2.4 ITM

The Cortex-M4 ITM is an application-driven trace source that supports printf style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets. There are four sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The four sources in decreasing order of priority are:

1. Software trace—Software can write directly to ITM stimulus registers. This emits packets.
2. Hardware trace—The DWT generates these packets, and the ITM emits them.
3. Time stamping—Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M4 clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.
4. Global system timestamping. Timestamps can optionally be generated using a system-wide 48-bit count value. The same count value can be used to insert timestamps in the ETM trace stream, allowing coarse-grain correlation.

8.2.2.5 Embedded Trace Macrocell (ETM)

Instruction trace, also known as ETM (Embedded Trace Macrocell) trace, is a continuously collected sequence of every executed instruction for a selected portion of the application. ETM generates trace packets and sends them to the trace bus. ETM does not actually output every address or instruction that the processor has reached or executed; it usually generates compressed information about the program flow and outputs full addresses only if needed (for example, if a branch has taken place). Because the debugger knows the application code image, the debugger can then reconstruct the full instruction sequence from the trace data.

For more information about ETM, refer to *Arm® CoreSight™ ETM-M7 Revision r0p1 Technical Reference Manual*.

8.2.2.6 Instrumentation Trace Macrocell

The ITM (Instrumentation Trace Macrocell) generates trace information as packets. There are four sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The four sources in decreasing order of priority are:

- Synchronization: DWT provides periodic requests to make ITM generate synchronization packet. Trace capture hardware uses it to identify the alignment of packet bytes in the bit-stream.
- Software Trace: Application software can write console messages directly to ITM stimulus ports, and output them to the host as trace packets.
- Hardware trace: The DWT generates these packets, and the ITM outputs them.
- Timestamping: ITM can generate timestamp packets that are inserted in to the trace stream, to help the host debugger to find out the timing of events. Timestamps are generated relative to packets. The ITM receives a 64-bit counter to generate the timestamp.

Trace data from ITM will be forwarded to TPIU and streamed out via the trace port.

For more information about ITM, refer to *Armv7-M Architecture Reference Manual*.

8.2.2.7 Cross-Trigger Interface (CTI)

The Cross-Trigger Interface (CTI) component is provided by Arm. A brief description of the CTI is provided below.

This chip contains three CTIs in the following sections:

- CM7 platform
- CM4 platform
- CSSYS

The CTIs in ARM cores have 8 trigger inputs and 8 trigger outputs that connect to logic in the domain to be debugged or profiled. Each CTI includes a 4-channel interface to the CTM. The CSSYS CTI also contains an 8 trigger IN/OUT along with the 4-channel interface to CTM.

The diagram below shows the high-level connections.

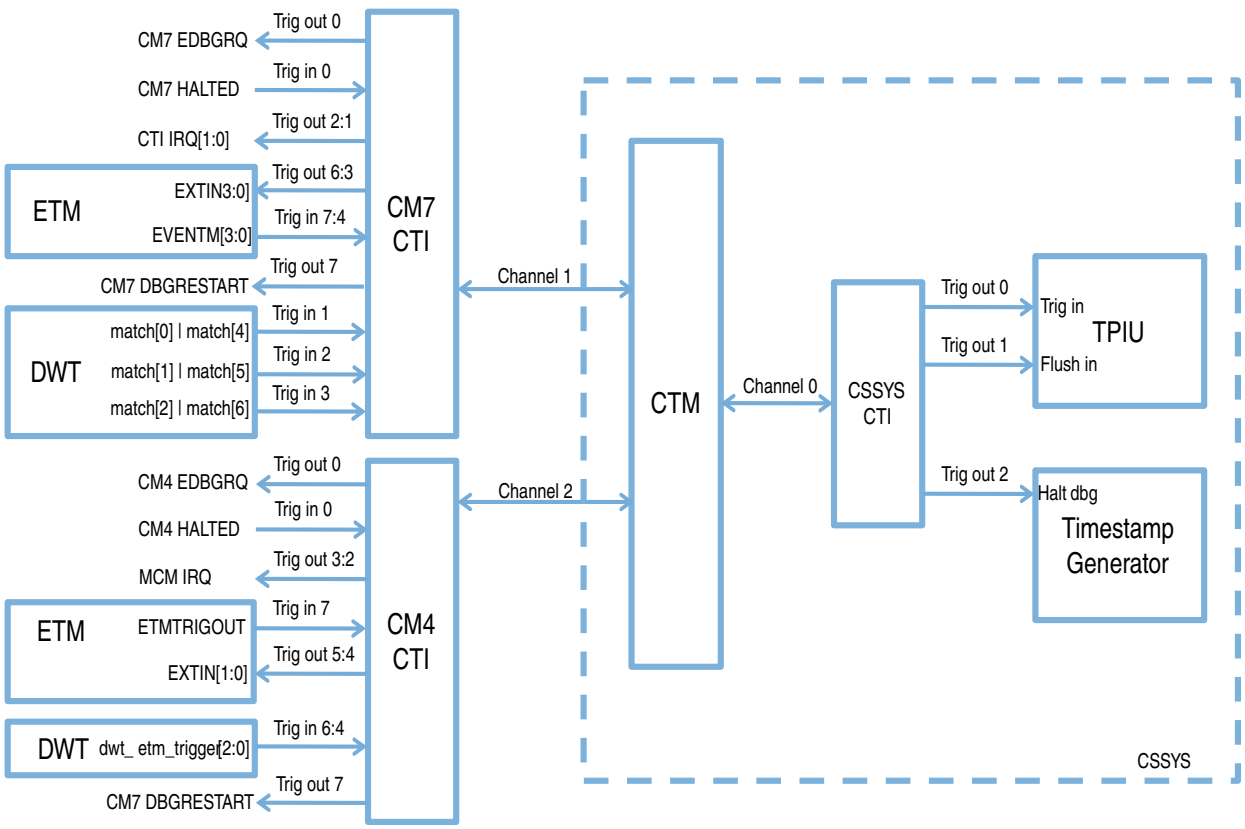


Figure 8-2. High-level connections

8.2.2.8 Cross-Trigger Matrix (CTM)

The CTM (Cross-Trigger Matrix) is provided by ARM. A brief description is provided below. See the ARM documentation for more detail.

The CTM is a relatively simple block with no configuration options. There is one instance of CTM present in the ARM platform. It routes Cross Trigger Interfaces (CTI) from Cortex-M7 and Cortex-M4 to Coresight CTI.

8.2.3 Chip-Specific JTAGC Features

8.2.3.1 JTAG Disable Mode

In addition to three different JTAG security modes that are implemented internally in the JTAG Controller, there is an option to disable the JTAGC functionality by e-fuse configuration. This creates additional JTAG mode "JTAG Disabled" with highest level of JTAG protection. In this mode all JTAG features are disabled.

8.2.3.2 JTAG ID

Table 8-2. i.MX JTAG ID

Device	Silicon revision	JTAG ID
i.MX RT1170	Rev 1.0	088C_601Dh

8.2.3.3 JTAG-to-SWD change sequence

1. Send more than 50 TCK cycles with TMS (SWD_IO) = 1
2. Send the 16-bit sequence on TMS (SWD_IO) = 0111_1001_1110_0111 (MSB transmitted first)
3. Send more than 50 TCK cycles with TMS (SWD_IO) = 1

NOTE

See the ARM documentation for the CoreSight DAP Lite for restrictions.

8.2.4 JTAG controller main features

- IEEE P1149.1 interface to off-chip test and development equipment
- Three levels of security, ranging from no security to no JTAG accessibility to the chip

NOTE

The security levels are detailed in the Security Reference Manual (SRM).

8.2.5 TAP port

The JTAGC supports the following standard JTAG pins:

- TRSTB
- TDI
- TDO
- TCK
- TMS

The figure below shows the TAP structure:

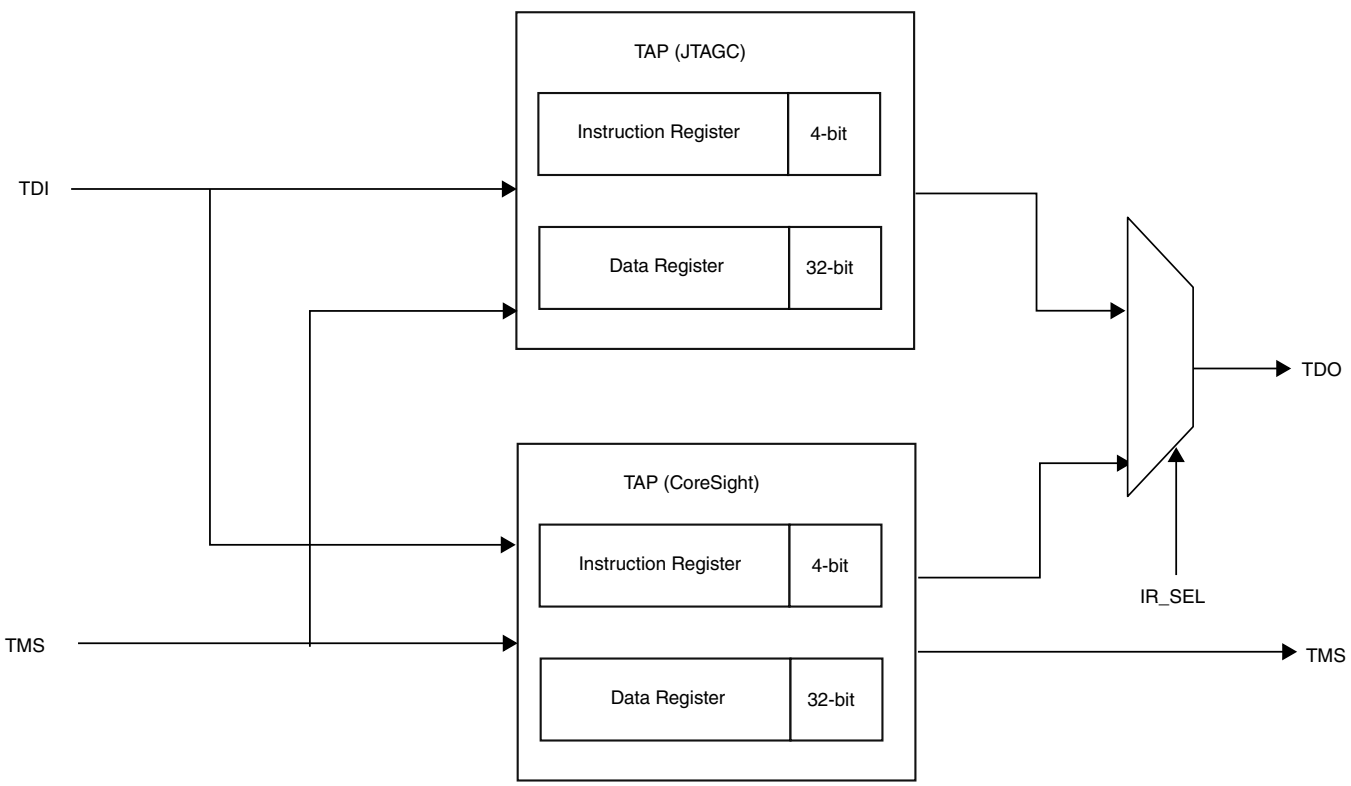


Figure 8-3. Tap structure

Refer to Arm technical documentation for more details.

8.3 Miscellaneous

The Miscellaneous function described in this section provide useful general capabilities.

8.3.1 Clock/Reset/Power

The DAP, ITM, and ETM existing in CM7 power domain will be powered off along with CM7. Similarly, the DAP, ITM, and ETM existing in the CM4 power domain and will be powered off along with CM4. The JTAG controller, TPIU and CoreSight DAP are in WAKEUPMIX power domain.

The debug components can receive resets from the following sources:

- Debug Reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register of the DAP) in the TCLK domain. This allows the debug tools to reset the debug logic.
- System POR reset

Chapter 9
JTAG Controller (JTAGC)

9.1 Chip-specific JTAG information

Table 9-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

9.2 Overview

The JTAGC block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format.

9.2.1 Block diagram

The following is a simplified block diagram of the JTAG Controller (JTAGC) block. See the chip-specific configuration information within this chapter as well as Register description for more information about the JTAGC registers.

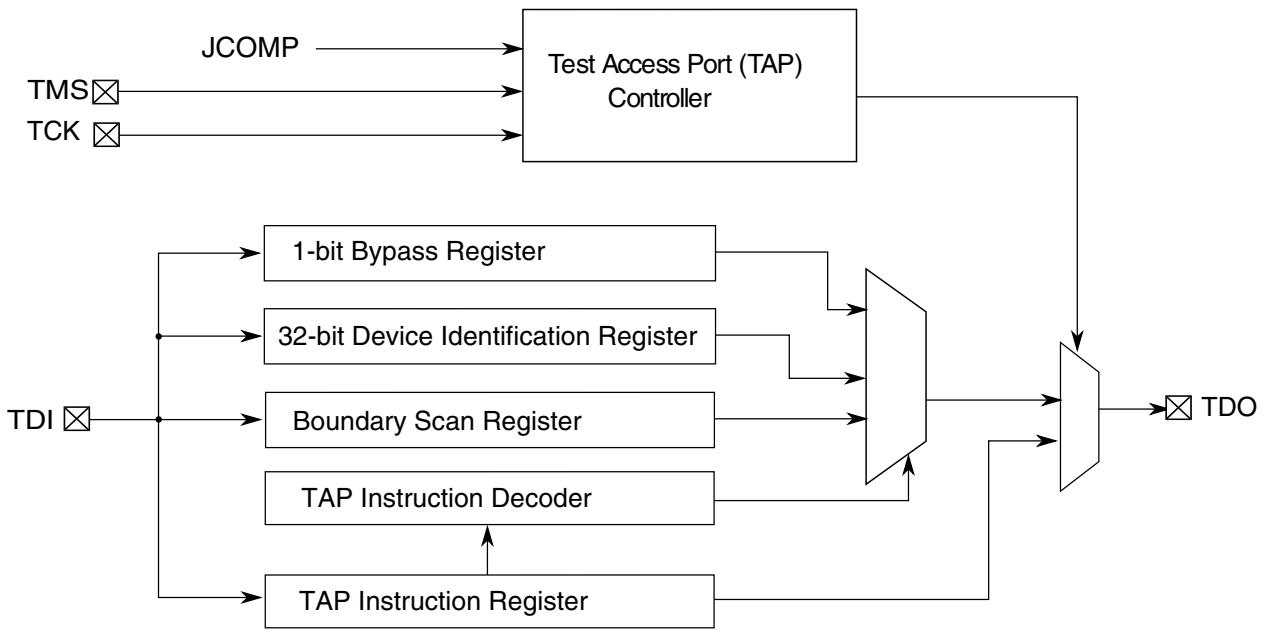


Figure 9-1. JTAG (IEEE 1149.1) block diagram

9.2.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 TAP interface
  - Four pins (TDI, TMS, TCK, and TDO)
- JCOMP input that provides reset control
- Instruction register that supports several IEEE 1149.1–2001 defined instructions as well as several public and private device-specific instructions (see JTAGC block instructions for a list of supported instructions).
- Bypass register, boundary scan register, and device identification register
- TAP controller state machine that controls the operation of the data registers, instruction register, and associated circuitry

9.3 Functional description

This section explains the JTAGC functional description.

9.3.1 Modes of operation

The JTAGC block uses JCOMP and a power-on reset indication as its primary reset signals. Several IEEE 1149.1–2001 defined test modes are supported, as well as a bypass mode.

9.3.1.1 Reset

The JTAGC block is placed in reset when:

- Power-on reset is asserted
- JCOMP is negated
- TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state

Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset or setting JCOMP to a value other than the value required to enable the JTAGC block results in asynchronous entry into the reset state.

When in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the on-chip system logic to continue unhindered.
- The instruction register is loaded with the IDCODE instruction.

9.3.1.2 IEEE 1149.1–2001 defined test modes

The JTAGC block supports several IEEE 1149.1–2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register when the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, SAMPLE, and SAMPLE/PRELOAD.

Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic when the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the following instructions are active:

- BYPASS
- HIGHZ
- CLAMP

The functionality of each test mode is explained in more detail in JTAGC block instructions.

9.3.1.3 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. When in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

9.3.2 JTAGC reset configuration

When in reset, the TAP controller is forced into the Test-Logic-Reset state, thus disabling the test logic and allowing normal operation of the on-chip system logic. In addition, the instruction register is loaded with the IDCODE instruction.

9.3.3 IEEE 1149.1-2001 (JTAG) TAP

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction.

Data is shifted between TDI and TDO through the selected register starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register, test data registers, and the bypass register.

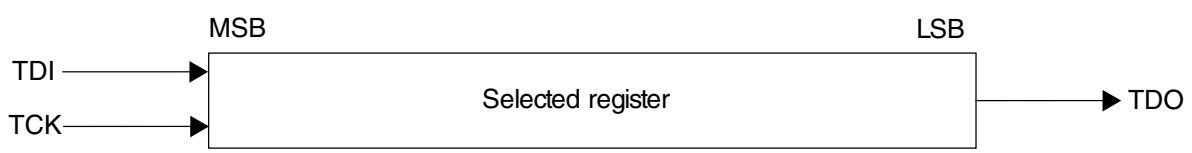
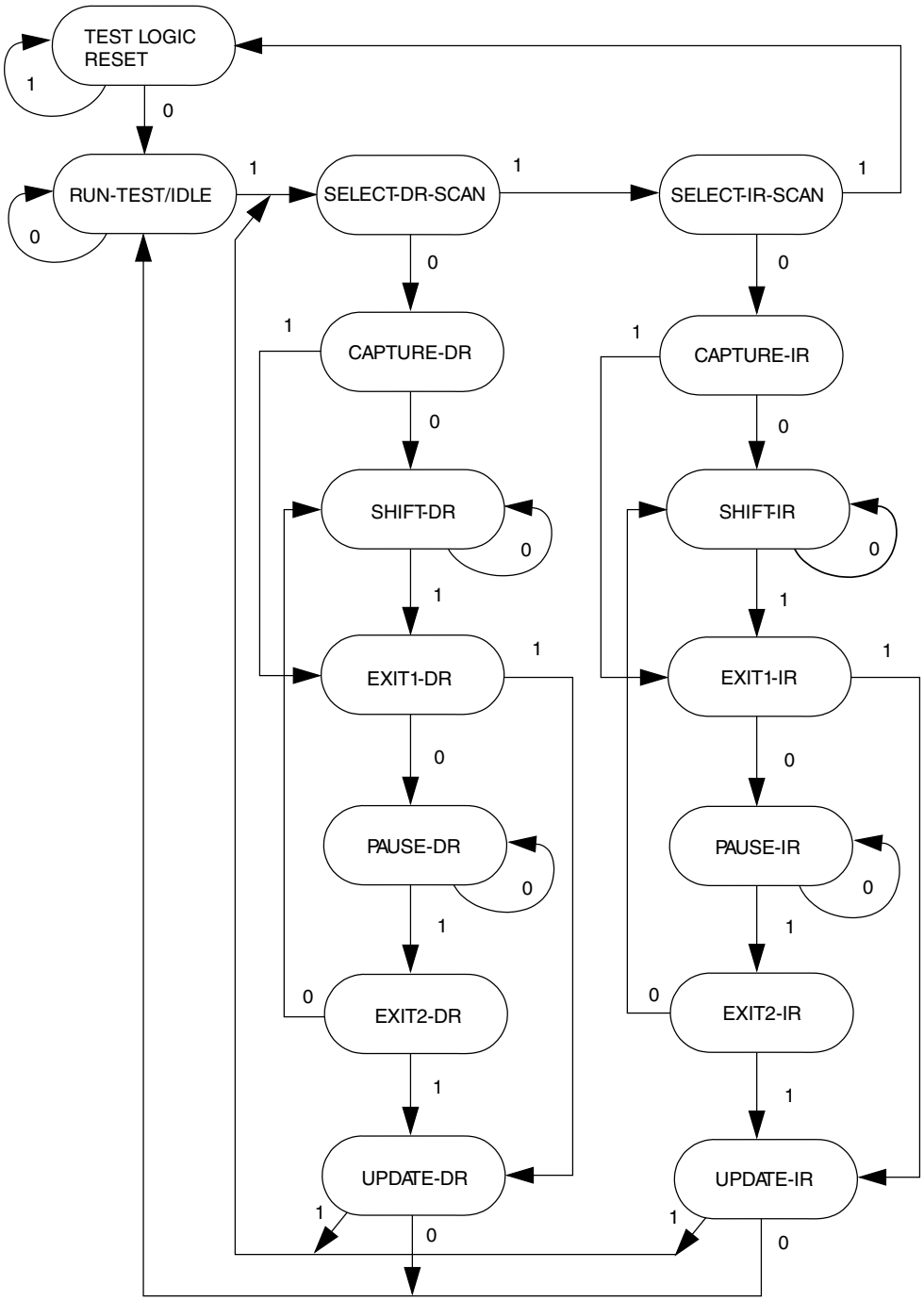


Figure 9-2. Shifting data through a register

9.3.4 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin.

The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the TCK signal. As the figure shows, holding TMS at logic 1 when clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.



The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

Figure 9-3. IEEE 1149.1-2001 TAP controller finite state machine

9.3.4.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting JCOMP to a logic 1 value.

9.3.4.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions when the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated after the required number of bits have been acquired.

9.3.5 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction. See the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

Table 9-2. 4-bit JTAG instructions

Instruction	Code[3:0]	Instruction summary
IDCODE	0000	Selects device identification register for shift
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register and applies preloaded values to output pins. **NOTE:** Execution of this instruction asserts functional reset.
HIGHZ	0101	Selects bypass register and tristates all output pins. **NOTE:** Execution of this instruction asserts functional reset.
ABORT (Arm JTAG-DP Reserved)	1000	This instruction goes the Arm JTAG-DP controller. See the Arm JTAG-DP documentation for more information.
DPACC (Arm JTAG-DP Reserved)	1010	This instruction goes the Arm JTAG-DP controller. See the Arm JTAG-DP documentation for more information.
APACC (Arm JTAG-DP Reserved)	1011	This instruction goes the Arm JTAG-DP controller. See the Arm JTAG-DP documentation for more information.
CLAMP	1101	Selects bypass register and applies preloaded values to output pins.

Table continues on the next page...

Table 9-2. 4-bit JTAG instructions (continued)

Instruction	Code[3:0]	Instruction summary
		NOTE: Execution of this instruction asserts functional reset.
IDCODE (Arm JTAG-DP Reserved)	1110	This instruction goes the Arm JTAG-DP controller. See the Arm JTAG-DP documentation for more information.
BYPASS	1111	Selects bypass register for data operations

9.3.5.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

9.3.5.2 SAMPLE/PRELOAD instruction

The SAMPLE/PRELOAD instruction has two functions:

- The SAMPLE portion of the instruction obtains a sample of the system data and control signals present at the MCU input pins, and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE/PRELOAD instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. Both the data capture and the shift operation are transparent to system operation.
- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

9.3.5.3 SAMPLE instruction

The SAMPLE instruction obtains a sample of the system data and control signals present at the MCU input pins, and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. There is no defined action in the Update-DR state. Both the data capture and the shift operation are transparent to system operation.

9.3.5.4 EXTEST external test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the preloaded data is loaded into the boundary scan register using the SAMPLE/PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state when performing external boundary scan operations.

9.3.5.5 TEST_LEAKAGE instruction

The TEST_LEAKAGE instruction forces the jtag_leakage output signal to high. It is intended to tristate all output pad buffers and disable all of the part's pad input buffers except JCOMP and TEST. The jtag_leakage signal is asserted at the falling edge of TCK following the TAP controller state machine transition from the Update-IR state to the Run-Test-Idle state. When asserted, the part disables TCK, TMS, and TDI inputs and forces them to a logic 1. The TAP controller state machine remains in the Run-Test-Idle state until the JCOMP input is set to a value other than the JTAGC enable encoding. TEST_LEAKAGE also asserts the internal system reset for the MCU to force a predictable internal state.

9.3.5.6 ENABLE_SOC_DATA1 instruction

The ENABLE_SOC_DATA1 instruction captures SoC data and selects the SOC_DATA register for connection as the shift path between TDI and TDO.

9.3.5.7 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. When HIGHZ is active all output drivers are placed in an inactive drive state (for example, high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

9.3.5.8 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register when the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a single bit (the bypass register) when conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

9.3.5.9 BYPASS instruction

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. When the BYPASS instruction is active the system logic operates normally.

9.3.6 Security

The JTAGC provides debug access to several hardware blocks including processors and system bus, which may allow contol and visibility into system peripherals and memory. To prevent unauthorized transactions or visibility that may be used to compromise the system, the JTAGC provides ways of regulating JTAG access.

The JTAG security modes are configured using eFuse configuration. Fuse burning is an irreversible process, and can't be reverted to the un-burned state once burned. The following security mode configurations are supported:

Table 9-3. JTAGC Security Modes

Security Mode	Description
JTAG Disabled Mode	JTAG Disabled Mode is the highest protection mode. JTAG use is permanently blocked.

Table continues on the next page...

Table 9-3. JTAGC Security Modes (continued)

Security Mode	Description
No Debug Mode	No Debug Mode is the next highest protection mode after JTAG Disabled Mode. All security sensitive JTAG features are permanently blocked. JTAG features that aren't disabled in this mode allow internal tests and board connectivity checks that don't reduce the security level of the chip.
Secure JTAG Mode	Secure JTAG Mode offers high security protection, while allowing JTAG functionality. JTAG use is restricted unless a secret-key challenge/response protocol is successfully executed. The intent of this mode is to allow field return testing. When a secured JTAG device is returned for debugging, this mode allows authorized re-activiation of the JTAG.
JTAG Enabled Mode	JTAG Enabled Mode offers low security protection. JTAG use is unrestricted and all JTAG features are enabled.

9.3.6.1 Challenge/Response Mechanism in System JTAG Mode

When JTAGC is in Secure JTAG Mode, the authentication process is as follows:

1. Shift ENABLE_SOC_DATA instruction to IR.
2. Passing through Capture-DR state of the JTAGC and by performing Shift-DR operations Challenge code can be accessed from TDO.
3. Shift ENABLE_CENSOR_CTRL instruction to IR. By performing Shift-DR, operations enter Response code value through TDI. As Update-DR state is entered, Response code is compared with the correct one.

In Fixed Challenge-Response Pair Mode, each part has its individual Challenge-Response pair, which is determined at manufacturing time, and does not change later on. The JTAGC compares the user's response to the expected response.

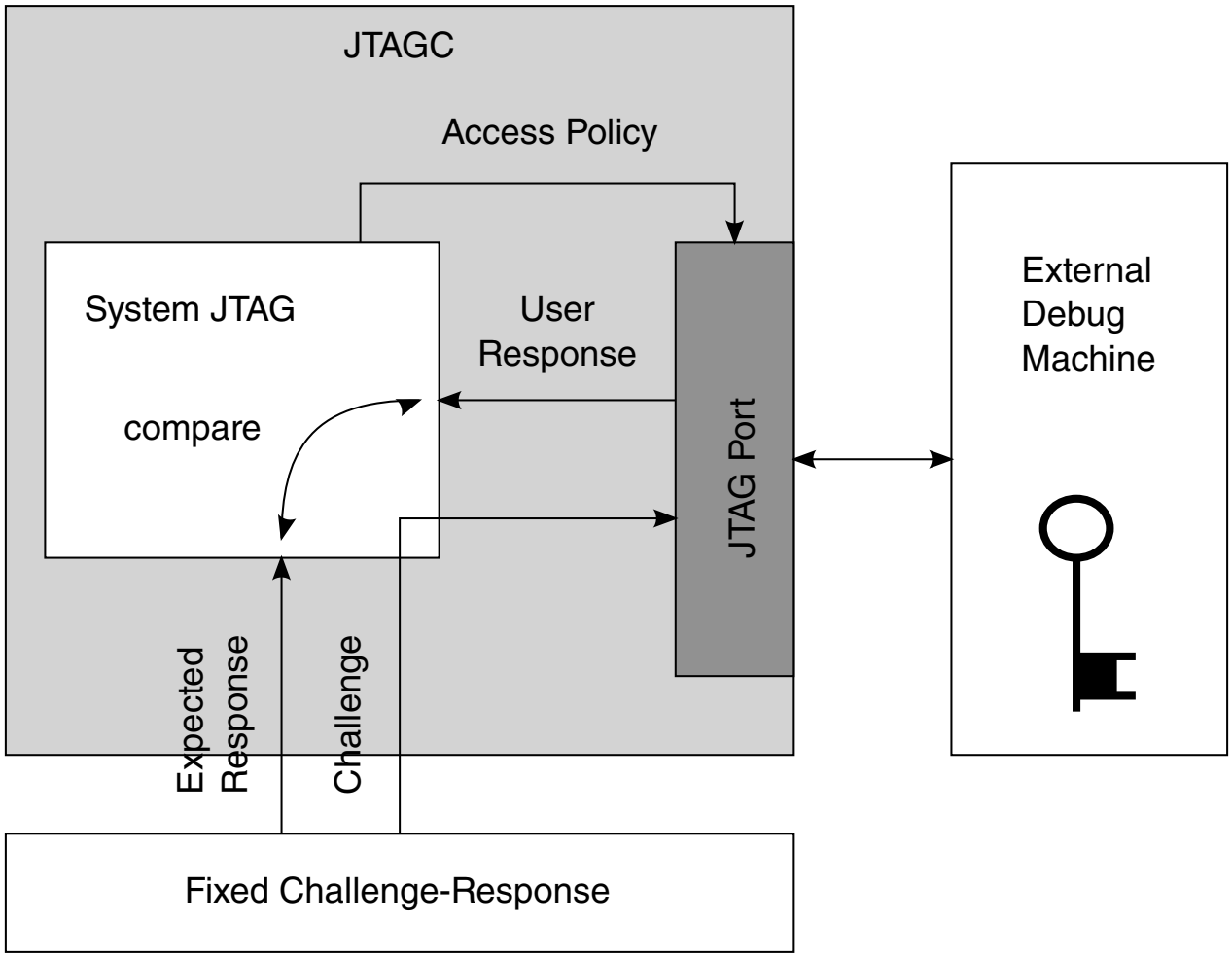


Figure 9-4. Secure JTAG with Fixed Challenge-response Pair

9.3.6.2 Software Enabled JTAG

To increase the flexibility of the JTAGC, an option to enable the JTAG via software is added and is available only in the Secure JTAG Mode. By writing '1' to HAB_JDE (HAB JTAG DEBUG ENABLE) bit in the eFuse controller, the JTAG is opened, regardless of its security mode. It is the responsibility of software to assert or negate this bit. Additionally, a corresponding lock bit is available to ensure that only trusted software is able to set the JDE bit. When the LOCK bit is set, no future change of JDE is possible, until the next POR (power-on-reset) cycle. The platform initialization software should set the LOCK bit for the JDE bit before transferring control to the application code.

The Software JTAG enable allows JTAG enabling without activating the Challenge-Respone mechanism. The JTAG software enable does not allow debug in case of boot or memory fault, as it requires reset before entering debug. This feature can be permanently blocked by burning the dedicated eFuse.

NOTE

The Software enabled JTAG feature reduces the overall security level of the system as it relies on software protections. If this feature is not required, it is strongly recommended to burn the JTAG_HEO eFuse which disables this feature.

9.3.6.3 Kill Trace

The kill trace signal disables any output of the ETM block. The ETM can be accessed either via JTAG port and/or by direct software code. Blocking the JTAG port also yields assertion of the kill trace signal. This resulted in blocking of trace port. The intention of this action is to block any attempt to break into the system via software manipulation of the debug modules. The kill trace, when active, prevents trace output, even in cases where it can be activated via chip pin. The kill trace feature needs to be activated by burning a dedicated eFuse. If the fuse is left intact, kill trace is never activated, as illustrated in the figure below.

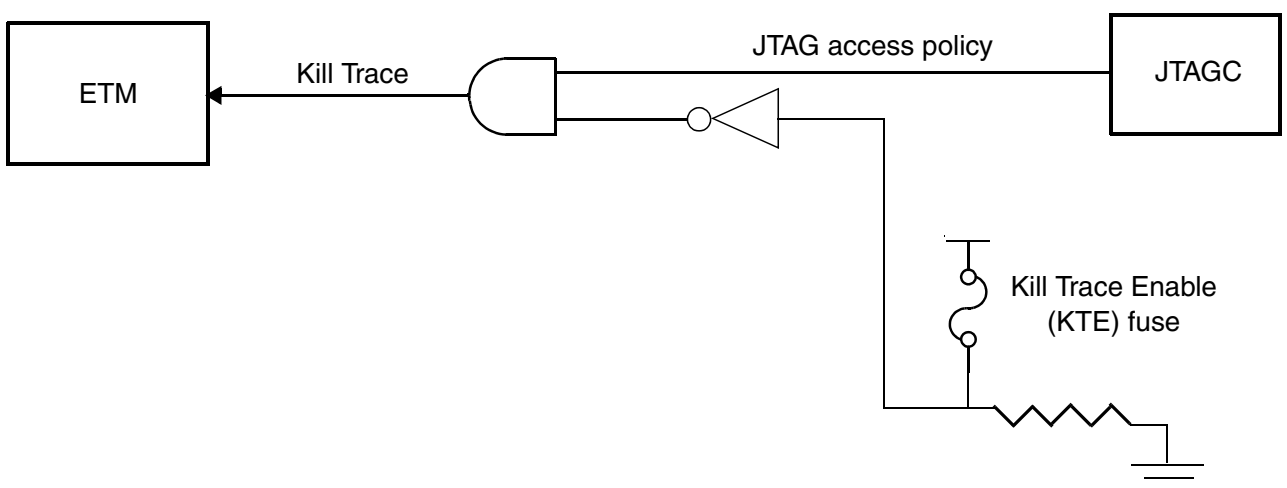


Figure 9-5. Kill Trace eFuse

The kill trace is asserted when the Kill Trace Enable (KTE) fuse is burned and an internal signal in JTAGC is asserted. The internal signal occurs when at least one of the following is true:

- Secure JTAG Mode and no code has been entered

- Secure JTAG Mode with burned Bypass and Re-enable fuses
- Secure JTAG Mode with incorrect response entered
- No Debug Mode
- TRST_B signal is active
- POR hasn't been asserted

9.3.7 Boundary scan

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

9.4 External signal description

The JTAGC consists of a set of signals that connect to off-chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

Table 9-4. JTAG signal properties

Name	I/O	Function	Reset state
TCK	Input	Test clock	Weak pulldown
TDI	Input	Test data in	Weak pullup
TDO	Output	Test data out	High Z[1]
TMS	Input	Test mode select	Weak pullup
JCOMP	Input	JTAG compliancy	Weak pulldown

1. TDO output buffer enable is negated when the JTAGC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented at the TDO pad for use when JTAGC is inactive.

9.4.1 Test clock input (TCK)

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

9.4.2 Test data input (TDI)

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

9.4.3 Test data output (TDO)

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is tristateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in TAP controller state machine.

9.4.4 Test mode select (TMS)

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

9.4.5 JCOMP JTAG compliancy

The JCOMP signal provides IEEE 1149.1-2001 compatibility and provides the ability to share the TAP. The JTAGC TAP controller is enabled when JCOMP is set to the JTAGC enable encoding—otherwise the JTAGC TAP controller remains in reset.

9.5 Initialization/application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Set the JCOMP signal to the JTAGC enable value, thereby enabling the JTAGC TAP controller.
2. Load the appropriate instruction for the test or action to be performed.

9.6 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

9.6.1 Instruction register

The JTAGC block uses a 4-bit instruction register as shown in the following figure.

The instruction register allows instructions to be loaded into the block to select the test to be performed, or the test data register to be accessed, or both. Instructions are shifted in through TDI when the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states.

Synchronous entry into the Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 0b1, making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

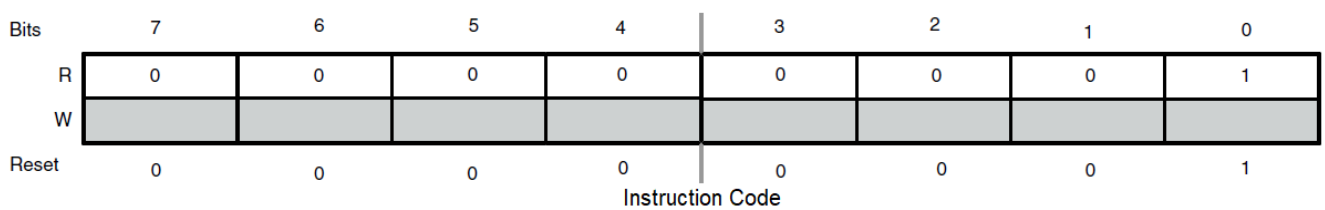


Figure 9-6. Instruction register

9.6.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the following instructions are active:

- BYPASS
- HIGHZ
- CLAMP

After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

9.6.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP.

The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state when the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

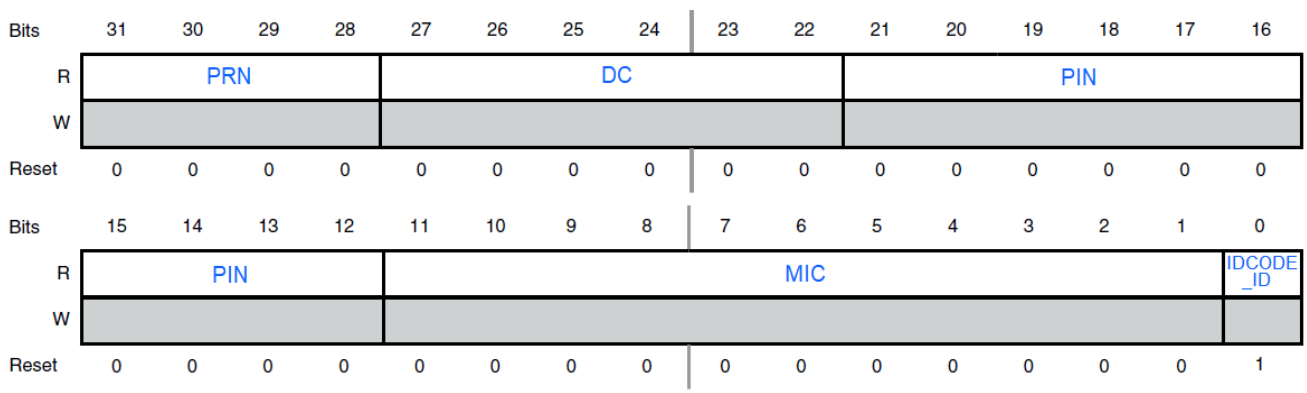


Figure 9-7. Device identification register

The following table describes the device identification register functions. The device identification register values are described in the chip-specific JTAGC information.

Table 9-5. Device identification register field descriptions

Field	Function
PRN	Part revision number Contains the revision number of the part.
DC	Design center Indicates the design center.
PIN	Part identification number Contains the part number of the device.
MIC	Manufacturer identity code Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID .
IDCODE ID	IDCODE register ID Identifies this register as the device identification register and not the bypass register. Always set to 1.

9.6.4 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are active. It is used to:

- Capture input pin data
- Force fixed values on output pins
- Select a logic value and direction for bidirectional pins

Each bit of the boundary scan register represents a separate boundary scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in Boundary scan. The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

Chapter 10
System Boot

10.1 Chip-specific Boot Information

This device has various peripherals supported by the ROM bootloader.

Table 10-1. ROM Bootloader Peripheral PinMux

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
LPUART	1	LPUART1_TX	GPIO_AD_24	ALT0	Can be used for serial downloader mode. Refer to Serial Boot (Serial Downloader) for more information.
		LPUART1_RX	GPIO_AD_25	ALT0	
LPSPI	1	LPSPI1_SCK	GPIO_AD_28	ALT0	Serial NOR/EEPROM connected to one of the LPSPI ports can be used as a recovery device. Refer to Recovery devices for more information. **NOTE:** Recovery device boot is disabled by default. The RECOVERY_BOOT_EN fuse must be blown in order to enable this option. The SPI INSTANCE fuse selects the SPI instance.
		LPSPI1_PCS0	GPIO_AD_29	ALT0	
		LPSPI1_SDO	GPIO_AD_30	ALT0	
		LPSPI1_SDI	GPIO_AD_31	ALT0	
	2	LPSPI2_SCK	GPIO_SD_B2_07	ALT6	
		LPSPI2_PCS0	GPIO_SD_B2_08	ALT6	
		LPSPI2_SDO	GPIO_SD_B2_09	ALT6	
		LPSPI2_SDI	GPIO_SD_B2_10	ALT6	
	3	LPSPI3_SCK	GPIO_DISP_B1_04	ALT9	
		LPSPI3_PCS0	GPIO_DISP_B1_07	ALT9	
		LPSPI3_SDO	GPIO_DISP_B1_06	ALT9	
		LPSPI3_SDI	GPIO_DISP_B1_05	ALT9	
	4	LPSPI4_SCK	GPIO_DISP_B2_12	ALT9	
		LPSPI4_PCS0	GPIO_DISP_B2_15	ALT9	
		LPSPI4_SDO	GPIO_DISP_B2_14	ALT9	
		LPSPI4_SDI	GPIO_DISP_B2_13	ALT9	
SEMC NAND	N/A	**SEMC_DATA00**	**GPIO_EMC_B1_00**	ALT0	Parallel NAND flash connected to the SEMC is a primary boot option. Refer to Parallel NAND flash Boot over SEMC for more information. **NOTE:** By default ROM reads from the 8-bit NAND device via the **bold** pins.
		SEMC_DATA01	**GPIO_EMC_B1_01**	ALT0	
		SEMC_DATA02	**GPIO_EMC_B1_02**	ALT0	
		SEMC_DATA03	**GPIO_EMC_B1_03**	ALT0	
		SEMC_DATA04	**GPIO_EMC_B1_04**	ALT0	
		SEMC_DATA05	**GPIO_EMC_B1_05**	ALT0	

Table continues on the next page...

Table 10-1. ROM Bootloader Peripheral PinMux (continued)

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
		SEMC_DATA06	**GPIO_EMC_B1_06**	ALT0	The "I/O Port Size" fuse must be blown to enable 16-bit NAND device support. The PCS_Selection fuse must be blown to select other PCS configurations.
		SEMC_DATA07	**GPIO_EMC_B1_07**	ALT0	
		SEMC_DATA08	GPIO_EMC_B1_30	ALT0	
		SEMC_DATA09	GPIO_EMC_B1_31	ALT0	
		SEMC_DATA10	GPIO_EMC_B1_32	ALT0	
		SEMC_DATA11	GPIO_EMC_B1_33	ALT0	
		SEMC_DATA12	GPIO_EMC_B1_34	ALT0	
		SEMC_DATA13	GPIO_EMC_B1_35	ALT0	
		SEMC_DATA14	GPIO_EMC_B1_36	ALT0	
		SEMC_DATA15	GPIO_EMC_B1_37	ALT0	
		SEMC_ADDR09	**GPIO_EMC_B1_18**	ALT0	
		SEMC_ADDR11	**GPIO_EMC_B1_19**	ALT0	
		SEMC_ADDR12	**GPIO_EMC_B1_20**	ALT0	
		SEMC_BA1	**GPIO_EMC_B1_22**	ALT0	
		SEMC_CSX0	**GPIO_EMC_B1_41**	ALT0	
		SEMC_CSX1	GPIO_AD_26	ALT2	
		SEMC_CSX2	GPIO_AD_27	ALT2	
		SEMC_CSX3	GPIO_AD_28	ALT2	
uSDHC	1	USDHC1_CD_B	GPIO_AD_32	ALT4	eMMC/MMC or SD/eSD connected to one of the USDHC ports is a primary boot option. Refer to Expansion device for more information on USDHC boot. For uSDHC2, the data lines are configured based on the bus width selection.
		USDHC1_WP	GPIO_AD_33	ALT4	
		USDHC1_VSELECT	GPIO_AD_34	ALT4	
		USDHC1_RESET_B	GPIO_AD_35	ALT4	
		USDHC1_CMD	GPIO_SD_B1_00	ALT0	
		USDHC1_CLK	GPIO_SD_B1_01	ALT0	
		USDHC1_DATA0	GPIO_SD_B1_02	ALT0	
		USDHC1_DATA1	GPIO_SD_B1_03	ALT0	
		USDHC1_DATA2	GPIO_SD_B1_04	ALT0	
		USDHC1_DATA3	GPIO_SD_B1_05	ALT0	
	2	USDHC2_CD_B	GPIO_AD_26	ALT11	
		USDHC2_WP	GPIO_AD_27	ALT11	
		USDHC2_VSELECT	GPIO_AD_28	ALT11	
		USDHC2_DATA3	GPIO_SD_B2_00	ALT0	
		USDHC2_DATA2	GPIO_SD_B2_01	ALT0	
		USDHC2_DATA1	GPIO_SD_B2_02	ALT0	
		USDHC2_DATA0	GPIO_SD_B2_03	ALT0	
		USDHC2_CLK	GPIO_SD_B2_04	ALT0	
		USDHC2_CMD	GPIO_SD_B2_05	ALT0	
		USDHC2_RESET_B	GPIO_SD_B2_06	ALT0	
		USDHC2_DATA4	GPIO_SD_B2_08	ALT0	

Table continues on the next page...

Table 10-1. ROM Bootloader Peripheral PinMux (continued)

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
		USDHC2_DATA5	GPIO_SD_B2_09	ALT0	
		USDHC2_DATA6	GPIO_SD_B2_10	ALT0	
		USDHC2_DATA7	GPIO_SD_B2_11	ALT0	
FlexSPI1 NOR/ NAND (Primary Pin Group)	1	FLEXSPI1_A_DQS (Primary DQS pin)	GPIO_SD_B2_05	ALT1	This pin group is used for FlexSPI NOR/NAND boot if: 1. The FLEXSPI_INSTANCE fuse/ boot pin is 0. 2. The FLEXSPI_PIN_GROUP_SEL fuse bit is 0. By default, ROM accesses the QSPI NOR/NAND via the **bold** pins. This default behavior can be overridden by setting the FLASH_CONNECTION_SEL fuse field. If QSPI, Hyperflash, or Octal memory attached to FlexSPI is a primary boot option. Refer to Serial NOR Flash Boot via FlexSPI for more information. The ROM will read the 512-byte FlexSPI NOR configuration parameters described in FlexSPI Serial NOR Flash Boot Operation. If Serial NAND memory attached to FlexSPI is a primary boot option. Refer to Serial NAND Flash Boot over FlexSPI for more information.
		FLEXSPI1_A_SS0_B	**GPIO_SD_B2_06**	ALT1	
		FLEXSPI1_A_SS1_B	GPIO_SD_B1_02	ALT9	
		FLEXSPI1_A_SCLK	**GPIO_SD_B2_07**	ALT1	
		FLEXSPI1_A_DATA0	**GPIO_SD_B2_08**	ALT1	
		FLEXSPI1_A_DATA1	**GPIO_SD_B2_09**	ALT1	
		FLEXSPI1_A_DATA2	GPIO_SD_B2_10	ALT1	
		FLEXSPI1_A_DATA3	GPIO_SD_B2_11	ALT1	
		FLEXSPI1_B_DATA3	GPIO_SD_B2_00	ALT1	
		fLEXSPI1_B_DATA2	GPIO_SD_B2_01	ALT1	
		FLEXSPI1_B_DATA1	GPIO_SD_B2_02	ALT1	
		FLEXSPI1_B_DATA0	GPIO_SD_B2_03	ALT1	
		FLEXSPI1_B_SCLK	GPIO_SD_B2_04	ALT1	
		FLEXSPI1_B_DQS	GPIO_SD_B1_05	ALT8	
		FLEXSPI1_B_SS0_B	GPIO_SD_B1_04	ALT8	
		FLEXSPI1_B_SS1_B	GPIO_SD_B1_03	ALT9	
FlexSPI1 NOR/ NAND (Secondary Pin Group)	1	FLEXSPI1_A_DQS	GPIO_AD_17	ALT3	This pin group is used for FlexSPI NOR/NAND boot if: 1. FLEXSPI_INSTANCE fuse/boot pin is 0. 2. FLEXSPI_PIN_GROUP_SEL fuse bit is 1. **NOTE:** 1. Only PORTA is supported via the 2nd Pin group. 2. The PortB chip select and combination mode are not allowed. Octal devices using the PortA and PortB data lines in combination mode are supported. The memory/memories must be connected to the
		FLEXSPI1_A_SS0_B	**GPIO_AD_18**	ALT3	
		FLEXSPI1_A_SCLK	**GPIO_AD_19**	ALT3	
		FLEXSPI1_A_DATA0	**GPIO_AD_20**	ALT3	
		FLEXSPI1_A_DATA1	**GPIO_AD_21**	ALT3	
		FLEXSPI1_A_DATA2	GPIO_AD_22	ALT3	
		FLEXSPI1_A_DATA3	GPIO_AD_23	ALT3	
		FLEXSPI1_B_SCLK	GPIO_AD_16	ALT3	
		FLEXSPI1_B_DATA0	GPIO_AD_15	ALT3	

Table continues on the next page...

Table 10-1. ROM Bootloader Peripheral PinMux (continued)

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
		FLEXSPI1_B_DATA1	GPIO_AD_14	ALT3	FlexSPI1_A_SS0_B chip select. 3. The Maximum Flash frequency supported on this port is 100MHz. By default, ROM accesses the QSPI NOR/NAND via the **bold** pins. This default behavior can be overridden by setting the FLASH_CONNECTION_SEL fuse field. If QSPI, Hyperflash, or Octal memory attached to FlexSPI is a primary boot option. Refer to Serial NOR Flash Boot via FlexSPI for more information. The ROM will read the 512-byte FlexSPI NOR configuration parameters described in FlexSPI Serial NOR Flash Boot Operation. If Serial NAND memory attached to FlexSPI is a primary boot option. Refer to Serial NAND Flash Boot over FlexSPI for more information.
		FLEXSPI1_B_DATA2	GPIO_AD_13	ALT3	
		FLEXSPI1_B_DATA3	GPIO_AD_12	ALT3	
FlexSPI2 NOR/NAND (Primary Pin Group)	2	FLEXSPI2_B_DATA7	GPIO_EMC_B1_41	ALT4	This pin group is used for FlexSPI NOR/NAND boot if: 1. FLEXSPI_INSTANCE fuse/boot pin is 1. 2. FLEXSPI_PIN_GROUP_SEL fuse bit is 0. By default, ROM accesses the QSPI NOR/NAND via the **bold** pins. This default behavior can be overridden by setting the FLASH_CONNECTION_SEL fuse field. If QSPI, Hyperflash, or Octal memory attached to FlexSPI is a primary boot option. Refer to Serial NOR Flash Boot via FlexSPI for more information. The ROM will read the 512-byte FlexSPI NOR configuration parameters described in FlexSPI Serial NOR Flash Boot Operation. If Serial NAND memory attached to FlexSPI is a primary boot option. Refer to Serial NAND Flash Boot over FlexSPI for more information.
		FLEXSPI2_B_DATA6	GPIO_EMC_B2_00	ALT4	
		FLEXSPI2_B_DATA5	GPIO_EMC_B2_01	ALT4	
		FLEXSPI2_B_DATA4	GPIO_EMC_B2_02	ALT4	
		FLEXSPI2_B_DATA3	GPIO_EMC_B2_03	ALT4	
		FLEXSPI2_B_DATA2	GPIO_EMC_B2_04	ALT4	
		FLEXSPI2_B_DATA1	GPIO_EMC_B2_05	ALT4	
		FLEXSPI2_B_DATA0	GPIO_EMC_B2_06	ALT4	
		FLEXSPI2_B_DQS	GPIO_EMC_B2_07	ALT4	
		FLEXSPI2_B_SS0_B	GPIO_EMC_B2_08	ALT4	
		FLEXSPI2_B_SCLK	GPIO_EMC_B2_09	ALT4	
		FLEXSPI2_A_SCLK	**GPIO_EMC_B2_10**	ALT4	
		FLEXSPI2_A_SS0_B	**GPIO_EMC_B2_11**	ALT4	
		FLEXSPI2_A_DQS	GPIO_EMC_B2_12	ALT4	
		FLEXSPI2_A_DATA0	**GPIO_EMC_B2_13**	ALT4	

Table continues on the next page...

Table 10-1. ROM Bootloader Peripheral PinMux (continued)

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
		FLEXSPI2_A_DATA1	**GPIO_EMC_B2_14**	ALT4	
		FLEXSPI2_A_DATA2	GPIO_EMC_B2_15	ALT4	
		FLEXSPI2_A_DATA3	GPIO_EMC_B2_16	ALT4	
		FLEXSPI2_A_DATA4	GPIO_EMC_B2_17	ALT4	
		FLEXSPI2_A_DATA5	GPIO_EMC_B2_18	ALT4	
		FLEXSPI2_A_DATA6	GPIO_EMC_B2_19	ALT4	
		FLEXSPI2_A_DATA7	GPIO_EMC_B2_20	ALT4	
FlexSPI2 NOR/ NAND (Secondary Pin Group)	2	**FLEXSPI2_A_SS0_B**	**GPIO_SD_B1_00**	ALT6	This pin group is used for FlexSPI NOR/NAND boot if: 1. FLEXSPI_INSTANCE fuse/boot pin is 1. 2. FLEXSPI_PIN_GROUP_SEL fuse bit is 1. **NOTE:** This pin group does not offer a DQS pad, so only low-speed read is supported when this pin group is chosen. By default, ROM accesses the QSPI NOR/NAND via the **bold** pins. This default behavior can be overridden by setting the FLASH_CONNECTION_SEL fuse field. If QSPI, Hyperflash, or Octal memory attached to FlexSPI is a primary boot option. Refer to Serial NOR Flash Boot via FlexSPI for more information. The ROM will read the 512-byte FlexSPI NOR configuration parameters described in FlexSPI Serial NOR Flash Boot Operation. If Serial NAND memory attached to FlexSPI is a primary boot option. Refer to Serial NAND Flash Boot over FlexSPI for more information.
		FLEXSPI2_A_SCLK	**GPIO_SD_B1_01**	ALT6	
		FLEXSPI2_A_DATA0	**GPIO_SD_B1_02**	ALT6	
		FLEXSPI2_A_DATA1	**GPIO_SD_B1_03**	ALT6	
		FLEXSPI2_A_DATA2	GPIO_SD_B1_04	ALT6	
		FLEXSPI2_A_DATA3	GPIO_SD_B1_05	ALT6	
FlexSPI RESET	-	GPIO4_IO03 (Primary RESET Pin)	GPIO_SD_B1_00	ALT5	The FlexSPI RESET pin is used as the dedicated RESET pin to restore the Serial NOR/NAND device connected to the FlexSPI interface. The FlexSPI RESET pin is enabled if the RESET_PIN_EN fuse bit is blown.
		GPIO_MUX2_IO08 (Secondary RESET Pin)	GPIO_EMC_B1_40	ALT5	

Table 10-1. ROM Bootloader Peripheral PinMux

Peripheral	Instance	Port (IO function)	PAD	Mode	Note
					The pin option is selected by RESET_PIN_SEL fuse bit.

NOTE

1. The boot ROM always tries to access Serial NOR/NAND flash using SS0 as the chip selection signal, as well as PORTA by default. It can access the Serial NOR/NAND device via CS0 and PORTB, if the FLASH_CONNECTION_SEL fuse is blown with a value of 2.
2. For Octal NOR/NAND, connected to FlexSPI1, the upper 4-bit pads must connect to the PORTB_DATA pads.
3. For Octal NOR/NAND, connected to FlexSPI2, the upper 4-bit pads must connect to the same PORT with the upper 4-bit pads.
4. PORTB_SCLK serves as PORTA_SCLK_B for the 1.8V HyperFLASH/HyperRAM.
5. A_SS1/B_SS1 is not supported by default.
6. The FlexSPI RESET pin is not used by default. This feature can be enabled by the RESET_PIN_EN and RESET_PIN_SEL fuses.

10.1.1 ROM settings and considerations for user applications and debugging

The boot ROM configures the clocks of the boot core, bus, and other required controllers to a high frequency of clock roots during boot for high-performance consideration. When the user application needs to re-configure the clock roots, PLLs or the PFDs, the SoC may crash if the clocks are not handled properly. To avoid such potential issues, below are some suggestions:

- Switch the root clock of the boot core, bus, and other affected controllers to a fixed clock root. before configuring the clocks.
- Configure the clock of each peripheral explicitly instead of relying on the default settings listed in the Reference Manual.

The following tables provide the clock frequency configurations for Cortex-M7/Cortex-M4 core boot.

Table 10-2. Normal Frequency Clock Configurations for Cortex-M7 Core Boot

BOOT_FREQ (0x9A0[3])	LPB_BOOT (0x9A0[5:4])	Core Clock Frequency (MHz)
0	0	400
	1	200
	2	100
	3	50
1	0	696
	1	348
	2	174
	3	87

Table 10-3. Normal Frequency Clock Configurations for Cortex-M4 Core Boot

BOOT_FREQ (0x9A0[3])	LPB_BOOT (0x9A0[5:4])	Core Clock Frequency (MHz)
0	0	200
	1	100
	2	50
	3	25
1	0	240
	1	120
	2	60
	3	30

The following table provides the list of registers that are updated by ROM during ROM execution.

Table 10-4. Registers modified by ROM

Module	Register	Value	Note
CCM	CLOCK_ROOT0_CONTROL	The value varies at different frequencies	CM7 clock root
	CLOCK_ROOT1_CONTROL		CM4 clock root
	CLOCK_ROOT2_CONTROL		Bus clock root
	CLOCK_ROOT3_CONTROL		Bus LPSR clock root
	CLOCK_ROOT20_CONTROL		FlexSPI1 clock root
	CLOCK_ROOT25_CONTROL		LPUART1 clock root
	CLOCK_GROUP0_CONTROL	0x00020002	FLEXRAM clock group
PLL	PLL_480_CTRL	0x2020201B	-
	PLL_480_PFD	0x1820110D	
	PLL_528_CTRL	0x20802008	

Table continues on the next page...

Table 10-4. Registers modified by ROM (continued)

Module	Register	Value	Note
	PLL_528_PFD	0x6058505B	
OSC	OSC_400M_CTRL1	0x00000000	-
	OSC_400M_CTRL2	0x00000001	
	OSC_24M_CTRL	0x40000014	
USB-PHY	USBPHYx_CTRL	0x2818C000	ROM sets the USB-PHY registers if the device runs under serial downloader mode.
	USBPHYx_PLL_SIC	0x80F03040	
	USBPHYx_PWD	0x00000000	

The registers in the modules listed below may be updated if they are in use for specific boot modes. ROM does not restore the registers in these modules after they are updated. The user application should not rely on the default values of the registers in these modules.

- FlexSPI1 - if the device boots from memory connected to FlexSPI1.
- FlexSPI2 - if the device boots from memory connected to FlexSPI2.
- uSDHC1 - if the device boots from memory connected to uSDHC1.
- uSDHC2 - if the device boots from memory connected to uSDHC2.
- SEMC - if the device boots from memory connected to SEMC.
- MECC - if the MECC_ENABLE Fuse bit is blown.
- FLEXRAM - if the FLEXRAMECC_ENABLE Fuse bit is blown.
- CAAM - HAB library uses some features in CAAM, and they cannot be restored before leaving ROM.
- SNVS - if the SNVS state was changed by ROM in HAB closed mode.
- ANADIG_OSC - The OSC related configurations are not restored.
- ANADIG_MISC - The AI interface related settings are not restored.
- ANADIG_PMU - The ANADIG PMU related settings are not restored.
- CCM - The clock settings for the modules, such as the CM7 core, are not restored.
- ANADIG_PLL - The ANADIG_PLL related settings are not restored (e.g. PLL_480_CTRL, PLL_480_PFD, PLL_528_CTRL, PLL_528_PFD).
- SCB->SHCSR - The UsageFault, MemManageFault, and BusFault enabling flags are not cleared.
- IOMUXC - cannot fully be restored because some settings are still in use before leaving ROM.
- IOMUXC_GPR - used in the IEE related test cases, cannot be cleared before leaving ROM.
- WDOG3/WDOG4 - disabled by ROM.
- CDOG – CDOG was enabled and used as the hardware secure counter during ROM execution, and ROM partially restores before leaving ROM due to the IP restriction.
- SRC GPR - ROM occupies SRC GPR0-GPR4 and GPR9.

10.1.2 Debug behavior

The debug port is disabled on the SoC out of reset. ROM enables the debug port in either of the following conditions:

- ROM is ready to jump to a valid user application.
- A debug request is received from the debug tool in the serial downloader mode or before finishing the image validity check.

10.1.3 Manufacturing protection

The Manufacturing protection (MP) key pair is generated by HAB. ROM will request generation of the (MP) key pair.

10.2 Overview

The boot process begins at any Reset where the hardware reset logic forces the Arm core (boot core is determined by BT_CORE_SEL fuse setting) to begin execution starting from the on-chip boot ROM.

If the BT_CORE_SEL fuse is blown, ROM will be executing from the M4 core instead of the M7 core. Booting from the M4 core can be slower due to the slower ROM execution time with the M4, especially if using HAB.

The boot ROM code uses the state of the internal register BOOT_MODE[1:0] as well as the state of various eFuses and/or GPIO settings to determine the boot flow behavior of the device.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial downloader support (USB-HID and UART)
- Device Configuration Data (DCD)
- Digital signature based High-Assurance Boot (HAB)
- eXternal Memory Configuration Data (XMCD)
- Encrypted execute-in-place (XIP) on Serial NOR via FlexSPI interface powered by:
  - Inline Encryption Engine (IEE)
  - On-the-Fly AES Decryption (OTFAD)

The boot ROM supports boot device as below:

- Serial NOR Flash via FlexSPI

- Serial NAND Flash via FlexSPI
- SLC RAWNAND Flash via SEMC
- SD/MMC via uSDHC
- SPI NOR/EEPROM via LPSPI

The boot ROM uses the state of the BOOT_MODE and eFuses to determine the boot device. For development purposes, the eFuses used to determine the boot device may be overridden using the GPIO pin inputs.

The boot ROM code also allows the downloading of programs to be run on the device. An example is a provisioning program that can make further use of the serial connection to provide a boot device with a new image. Typically, the provisioning program is downloaded to the internal RAM and allows to program the boot devices, such as the FlexSPI NOR flash. The ROM serial downloader uses a high-speed USB in a non-stream mode connection.

The Device Configuration Data (DCD) feature allows the boot ROM code to obtain the SOC configuration data from an external program image residing on the boot device. As an example, the DCD can be used to program the SDRAM controller for optimal settings improving the boot performance. The DCD is restricted to the memory areas and peripheral addresses that are considered essential for the boot purposes (see Write data command).

The External Memory Configuration Data (XMCD) allows the boot ROM code to configure the SDRAM connected to the SEMC controller, or the HyperRAM/APMemory PSRAM via the FlexSPI controller from an external program image residing on the boot device. The XMCD aims to simplify the external RAM enablement, and it is a replacement for the legacy Device Configuration Data (DCD).

A key feature of the boot ROM is the ability to perform a secure boot, also known as a High-Assurance Boot (HAB). This is supported by the HAB security library which is a subcomponent of the ROM code. The HAB uses a combination of hardware and software together with the Public Key Infrastructure (PKI) protocol to protect the system from executing unauthorized program images. Before the HAB allows the user image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then included as a part of the final program image. If configured to do so, the ROM verifies the signatures using the public keys included in the program image. In addition to supporting the digital signature verification to authenticate the program images, encrypted boot is also supported. The encrypted boot can be used to prevent the cloning of the program image directly off the boot device. A secure boot with HAB can be performed on all boot devices supported on the chip in addition to the serial downloader. The HAB library in the boot ROM also provides the API functions, allowing the additional boot chain components (e.g., bootloader, application) to extend the secure boot chain. The out-of-fab setting for the

SEC_CONFIG is the open configuration, in which the ROM/HAB performs the image authentication, but all authentication errors are ignored and the image is still allowed to execute.

10.3 Boot modes

During boot, the ROM's behavior is defined by the boot mode pin settings, as described in Boot mode pin settings.

10.3.1 Boot mode pin settings

The device has four boot modes (one is reserved for NXP use). The boot mode is selected based on the binary value stored in the internal BOOT_MODE register.

The BOOT_MODE register is initialized by sampling the BOOT_MODE0 and BOOT_MODE1 inputs on the rising edge of the POR_B. After these inputs are sampled, their subsequent state does not affect the contents of the internal BOOT_MODE register. The state of the internal BOOT_MODE register may be read from the BMOD[1:0] field of the SRC Boot Mode Register (SRC_SBMR2). The available boot modes are: Boot From Fuses, Serial Downloader, and Internal Boot. See this table for settings:

Table 10-5. Boot MODE pin settings

BOOT_MODE[1:0]	Boot Type
00	Boot From Fuses
01	Serial Downloader
10	Internal Boot
11	Reserved

10.3.2 High-level boot sequence

The figure found here show the high-level boot ROM code flow.

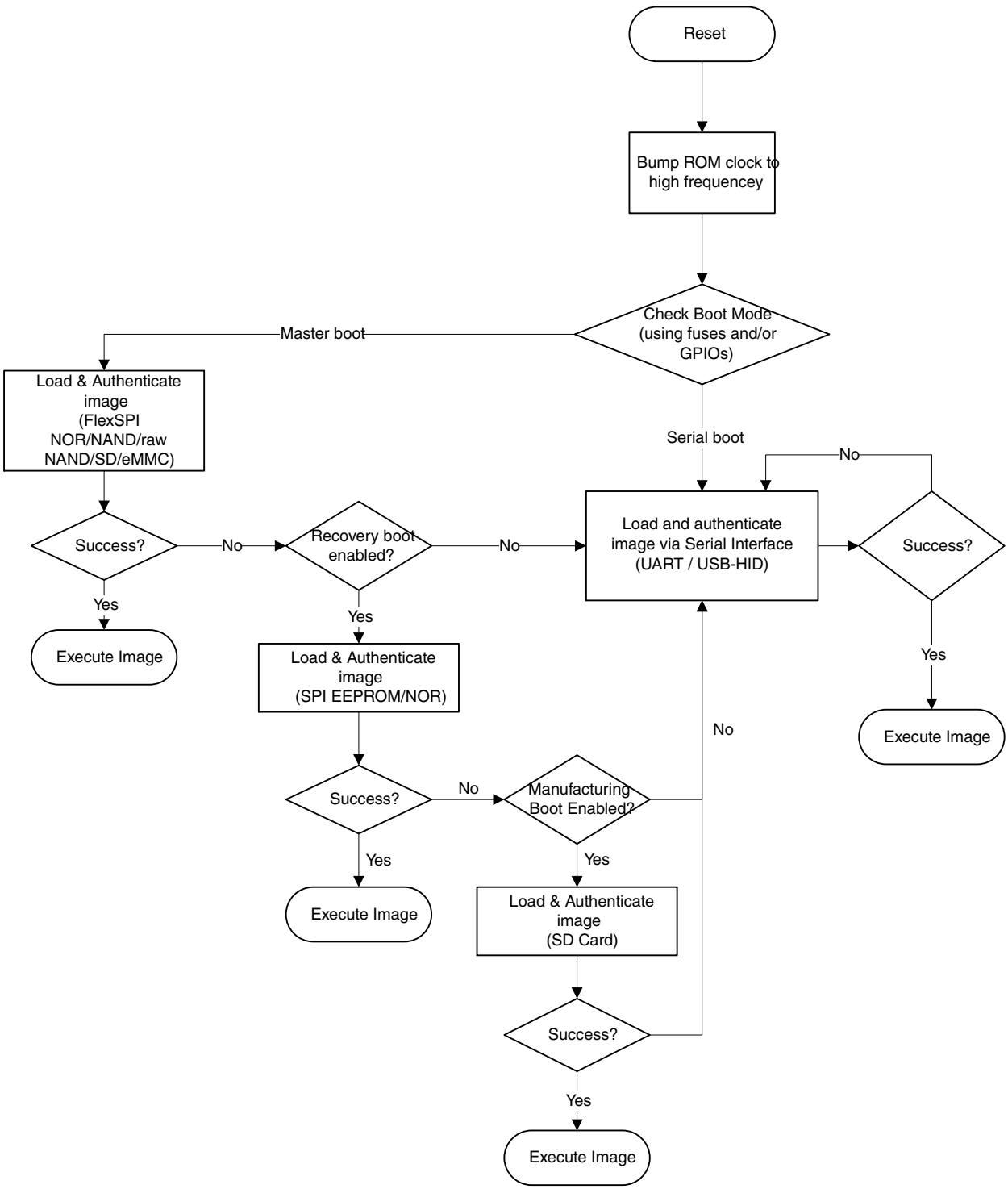


Figure 10-1. Boot flow

10.3.3 Boot From Fuses mode (BOOT_MODE[1:0] = 00b)

A value of 00b in the BOOT_MODE[1:0] register selects the Boot From Fuses mode.

This mode is similar to the Internal Boot mode described in Internal Boot mode (BOOT_MODE[1:0] = 10b) with one difference. In this mode, the override from the GPIO boot pins are ignored. The boot ROM code uses the boot eFuse settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses, the boot flow is controlled by the BT_FUSE_SEL eFuse value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, flash) was not programmed yet, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFuses may be configured incorrectly for the hardware on the platform. In such case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using the Boot From Fuses mode addresses this problem.

Setting the BT_FUSE_SEL=0 forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a program image and blow the BT_FUSE_SEL and the other boot configuration eFuses. After the reset, the boot ROM code determines that the BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs an internal boot according to the new eFUSE settings. This allows the user to set BOOT_MODE[1:0]=00b on a production device and burn the fuses on the same device (by forcing the entry to the Serial Downloader), without changing the value of the BOOT_MODE[1:0] or the pullups/pulldowns on the BOOT_MODE pins.

10.3.4 Serial Downloader (BOOT_MODE[1:0] = 01b)

The Serial Downloader provides a means to download a Program Image to the on-chip RAM over USB or UART serial connection. In this mode, typically a host PC can communicate to the ROM bootloader using serial download protocol. Serial downloader and the protocol are discussed in Serial Downloader.

10.3.5 Internal Boot mode (BOOT_MODE[1:0] = 10b)

A value of 10b in the BOOT_MODE[1:0] register selects the Internal Boot mode. In this mode, the processor continues to execute the boot code from the internal boot ROM.

The boot code performs the hardware initialization, loads the program image from the chosen boot device, performs the image validation using the HAB library (see Boot security settings), and then jumps to an address derived from the program image. If an error occurs during the internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader (BOOT_MODE[1:0] = 01b)). A secure boot using the HAB is possible in all the three boot modes.

When set to the Internal Boot, the boot flow may be controlled by a combination of eFuse settings with an option of overriding the fuse settings using the General Purpose I/O (GPIO) pins. The GPIO Boot Select FUSE (BT_FUSE_SEL) determines whether the ROM uses the GPIO pins for a selected number of configuration parameters or eFuses in this mode.

- If BT_FUSE_SEL = 1, all boot options are controlled by the eFuses described in Boot eFuse descriptions.
- If BT_FUSE_SEL = 0, the specific boot configuration parameters may be set using the GPIO pins rather than eFuses. The fuses that can be overridden when in this mode are indicated in the GPIO column of Boot eFuse descriptions. GPIO boot overrides provides the details of the GPIO pins.

The use of the GPIO overrides is intended for development since these pads are used for other purposes in the deployed products. NXP recommends controlling the boot configuration by the eFuses in the deployed products and reserving the use of the GPIO mode for the development and testing purposes only.

10.3.6 Boot security settings

The internal boot modes use one of three security configurations.

- Closed: This level is intended for use with shipping-secure products. All HAB functions are executed and the security hardware is initialized (the Security Controller or SNVS enters the Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, and the boot flow is aborted with the control being passed to the serial downloader. At this level, the execution does not leave the internal ROM unless the target executable image is authenticated.
- Open: This level is intended for use in non-secure products or during the development phases of a secure product. All HAB functions are executed in the same

way as for a closed system. The security hardware is initialized (except for the SNVS which is left in the Non-Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, but have no influence on the boot flow which continues as if the errors did not occur. This configuration is useful for a secure product development because the program image runs even if the authentication data is missing or incorrect, and the error log can be examined to determine the cause of the authentication failure.

- Field Return: This level is intended for the parts returned from the shipped products.

10.4 Device configuration

This section describes the external inputs that control the behavior of the Boot ROM code.

This includes the boot device selection (FlexSPI NOR FlexSPI NAND,SD, MMC, and so on), boot device configuration (SD bus width, speed, and so on), and other. In general, the source for this configuration comes from the eFuses embedded inside the chip. However, certain configuration parameters can be sourced from the GPIO pins, allowing further flexibility during the development process.

10.4.1 Boot eFuse descriptions

This table is a comprehensive list of the configuration parameters that the ROM uses.

Table 10-6. Boot eFuse descriptions

Fuse	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
BT_FUSE_SEL	OEM	In the Internal Boot mode BOOT_MODE[1:0] = 10, the BT_FUSE_SEL fuse determines whether the boot settings indicated by a Yes in the GPIO column are controlled by the GPIO pins or the eFuse settings in the On-Chip OTP Controller (OCOTP). In the Boot From Fuse mode BOOT_MODE[1:0] = 00, the BT_FUSE_SEL fuse indicates whether the bit configuration eFuses are programmed.	NA	0	If BOOT_MODE[1:0] = 10b: • 0 - The bits of the SBMR are overridden by the GPIO pins. • 1 - The specific bits of the SBMR are controlled by the eFuse settings. If BOOT_MODE[1:0] = 00b: • 0—The BOOT configuration eFuses are not programmed yet. The boot flow jumps to the serial downloader. • 1—The BOOT configuration eFuses are programmed. The regular boot flow is performed.

Table continues on the next page...

Table 10-6. Boot eFuse descriptions (continued)

Fuse	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
UNIQUE_ID[63:0]	NXP	Device Unique ID, 64-bit UID	NA	Unique ID	Settings vary—used by HAB
BOOT_CFG1[7:0]	OEM	Boot configuration 1	Yes	0	Specific to the selected boot mode
BOOT_CFG2[3:0]	OEM	Boot configuration 2	Yes	0	Specific to the selected boot mode
LPB_BOOT	OEM	Low-Power Boot	No	0	Divide (Core / Bus) based on Boot frequencies: 00 - Div by 1 01 - Div by 2 10 - Div by 4 11 - Div by 8
WDOG_ENABLE	OEM	Watchdog reset counter enable	No	0	WDOG1 enable: 0—The watchdog reset counter is disabled during boot. 1—The watchdog reset counter is enabled during boot.
PAD_SETTINGS	OEM	Override values for the SD/MMC and NAND boot modes	No	0	Override these IO PAD settings: • PAD_SETTINGS[0]—Slew Rate • PAD_SETTINGS[3:1]—Drive Strength • PAD_SETTINGS[5:4]—Speed Settings • PAD_SETTINGS[6]— Pull Up / Down Configuration • PAD_SETTINGS[7]—Hysteresis Enable

1. This setting is overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO Boot Overrides for the corresponding GPIO pin.
2. 0 = intact fuse and 1= blown fuse

10.4.2 GPIO boot overrides

This table provides a list of the GPIO boot overrides:

Table 10-7. GPIO Boot Overrides

Package Pin	Direction on reset	eFuse
GPIO_LPSR_02/BOOT_MODE[0]	Input	Boot Mode selection
GPIO_LPSR_03/BOOT_MODE[1]	Input	

Table continues on the next page...

Table 10-7. GPIO Boot Overrides (continued)

Package Pin	Direction on reset	eFuse
GPIO_DISP_B1_06	Input	BOOT_CFG1[0]
GPIO_DISP_B1_07	Input	BOOT_CFG1[1]
GPIO_DISP_B1_08	Input	BOOT_CFG1[2]
GPIO_DISP_B1_09	Input	BOOT_CFG1[3]
GPIO_DISP_B1_10	Input	BOOT_CFG1[4]
GPIO_DISP_B1_11	Input	BOOT_CFG1[5]
GPIO_DISP_B2_00	Input	BOOT_CFG1[6]
GPIO_DISP_B2_01	Input	BOOT_CFG1[7]
GPIO_DISP_B2_02	Input	BOOT_CFG2[0]
GPIO_DISP_B2_03	Input	BOOT_CFG2[1]
GPIO_DISP_B2_04	Input	BOOT_CFG2[2]
GPIO_DISP_B2_05	Input	BOOT_CFG2[3]

NOTE

Refer to the Fusemap chapter for more information on fuses mapped to the GPIO pins.

The input pins above are sampled at boot, and can be used to override the corresponding eFuse values, depending on the setting of the BT_FUSE_SEL fuse.

10.4.3 Device Configuration Data (DCD)

The DCD is the configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various on-chip peripherals. See Device Configuration Data (DCD) for more details on DCD.

10.5 Device initialization

This section describes the details of the ROM and provides the initialization details.

This includes details on:

- The ROM memory map
- The RAM memory map
- On-chip blocks that the ROM must use or change the POR register default values
- Clock initialization
- Enabling the L1 I/D cache
- Exception handling and interrupt handling

10.5.1 Internal ROM/RAM memory map

These figures show the internal ROM and RAM memory map:

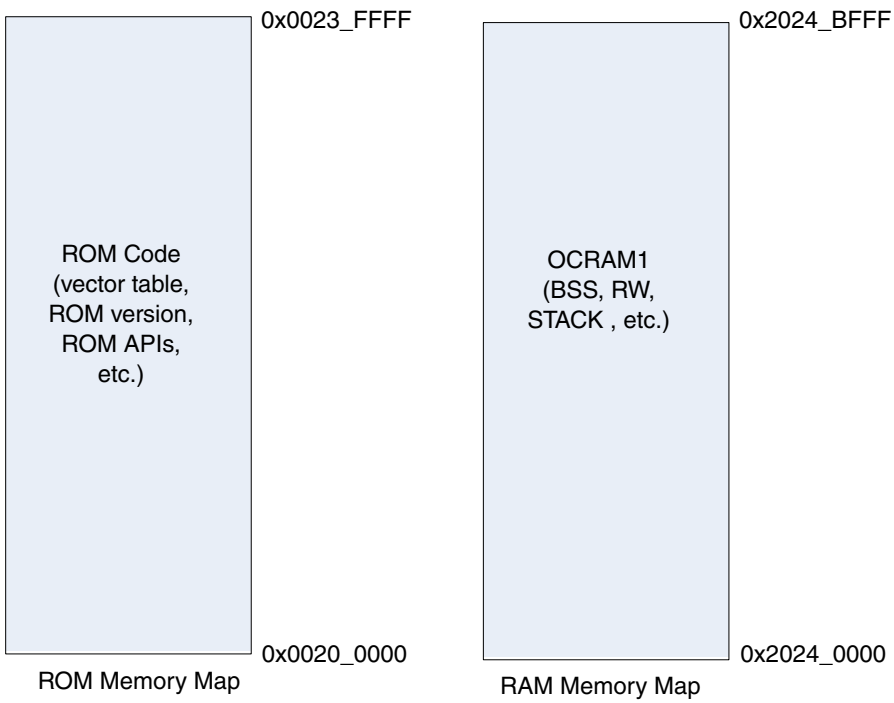


Figure 10-2. Internal ROM and RAM memory map

NOTE

The RAM space occupied by ROM cannot be used as part of the boot image. The entire OCRAM1 region can be used freely after the boot. The above OCRAM1 region must be reserved if the user application needs to call the HAB API/FlexSPI NOR API for image authentication.

10.5.2 Boot block activation

The boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow.

The ROM configures and uses the following blocks (listed in an alphabetical order) during the boot process. Note that the blocks actually used depend on the boot mode and the boot device selection:

Block	Description
CCM	Clock Control Module
FlexSPI	Flexible SPI Interface which supports serial NOR, Serial NAND and serial RAM devices
OCOTP	On Chip One Time Programmable Controller containing the eFuses
IOMUXC	I/O Multiplexer Control which allows the GPIO use to override the eFuse boot settings
IOMUX GPR	I/O Multiplexer control General Purpose Registers
LPSPI	Low Power SPI interface which supports serial NOR/EEPROM devices
SNVS	Secure Non-Volatile Storage
SRC	System Reset Controller
USB	Used for Serial download of a boot device provisioning program
uSDHC	Ultra-Secure Digital Host Controller
WDOG 1	WatchDog Timer
CAAM	Cryptographic Acceleration and Assurance Module
PIT	Periodic Interrupt Timer

10.5.3 Enabling Caches

The boot ROM includes a feature that enables the caches to improve the boot speed.

L1 instruction cache is enabled at the start of image download. The L1 data cache is enabled at the start of image authentication. Both the caches are disabled by ROM before leaving the ROM execution.

By default, the L1-ICache and DCache are enabled by ROM. However, there are fuse bits that can be programmed for ROM not to enable the L1 I/DCache at boot.

The Cache features are controlled by the BT_ICACHE_DISABLE fuse and BT_DCACHE_DISABLE fuses. By default, both fuses are not blown meaning the ROM uses the L1-ICache and L1-DCache of the Arm core. This improves the performance of the HAB signature verification software.

10.5.4 Exception error handling

10.5.4.1 Fault handling

All types of Faults cause the NVIC_SystemReset(), including the following:

- HardFault, BusFault, MemMangeFault, UsageFault
- ROM boot flow mismatch (treated as a side-channel attack)

10.5.4.2 Boot failure

ROM indicates the following failures if the boot failure indicator feature is enabled by the BOOT_FAIL_IND_EN and "Boot Failure Indicator Pin Select" fuse fields.

- The primary/recovery/manufacturing boot failed before entering serial downloader
- The HAB API call failed

ROM resets the device immediately upon detection of an illegal boot flow.

10.5.5 Persistent bits

Some modes of the boot ROM require the registers that keep their values after a warm reset. The SRC General-Purpose registers are used for this purpose.

See this table for persistent bits list and description:

Table 10-8. Persistent bits

Bit name	Bit location	Description
PERSIST_SECONDARY_BOOT	SRC_GPR9[30]	This bit identifies which image must be used—primary and secondary. Used only for eMMC/SD/FlexSPI NOR boot.
PERSIST_REDUNDANT_BOOT	SRC_GPR9[27:26]	This field identifies which image must be used - 0/1/2/3. Used for both SPI NAND and SLC raw NAND devices.
FLASH_STATE_CTX	SRC_GPR2[31:0]	Flash status context. This field logs the flash state during boot, so that the flash driver can retrieve the state of the flash and operate the flash devices properly.
ROM_API_CTX	SRC_GPR3[31:0]	ROM API context. Reserved for ROM API use.

10.6 Boot devices

The chip supports the following boot flash devices:

- Serial NOR flash via FlexSPI Interface
- Serial NAND Flash via FlexSPI Interface
- NAND Flash with SEMC interface, located on CS0, 8-bit/16-bit bus width.
- SD/MMC/eSD/SDXC/eMMC5.0 via uSDHC interface, supporting high capacity cards
- Serial NOR/EEPROM boot via LPSPI

The selection of the external boot device type is controlled by the BOOT_CFG1[7:4] eFUSEs. See this table for more details:

Table 10-9. Boot device selection

BOOT_CFG1[7:4]	Boot device
0000b	Serial NOR boot via FlexSPI
01xxb	SD Boot via uSDHC
10xxb	eMMC/MMC boot via uSDHC
001xb	SLC NAND boot via SEMC
11xxb	Serial NAND boot via FlexSPI

NOTE

A user can use the LPSPI to boot from Serial NOR/EEPROM. Booting from Serial NOR/EEPROM via an LPSPI port is not intended as the primary device, but as a recovery boot device if the primary boot device fails.

In order to enable a recovery boot device, EEPROM_RECOVERY_EN fuse must be set to 1, and other fuses listed in Serial NOR/EEPROM eFUSE configuration must be properly set.

10.6.1 Serial NOR Flash Boot via FlexSPI

10.6.1.1 Serial NOR eFUSE Configuration

Table 10-10. Fuse definition for Serial NOR over FlexSPI

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
BOOT_CFG1[0]	OEM	xSPI FLASH Auto Probe	Yes	0	0 – Disabled 1 – Enabled
BOOT_CFG1[1]	OEM	Encrypted XIP	Yes	0	0 – Disabled 1 – Enabled
BOOT_CFG1[3:2]	OEM	xSPI FLASH Auto Probe Type	Yes	0	0 – QuadSPI NOR 1 – MXIC Octal 2 – Micron Octal 3 – Adesto Octal
BOOT_CFG1[7:4]	OEM	Boot device selection	Yes	0	0 – Serial NOR device is selected as boot device

Table continues on the next page...

Table 10-10. Fuse definition for Serial NOR over FlexSPI (continued)

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
BOOT_CFG2[2:0]	OEM	xSPI Flash Type	Yes	0	000b–Boot with default 0x03 Read Enabled 001b–Reserved 010b–HyperFlash 1V8 011b–HyperFlash 3V0 100b–MXIC Octal Read 101b–Micron Octal Read
BOOT_CFG2[3]	OEM	FLEXSPI instance	Yes	0	0 – FLEXSPI1 1 – FLEXSPI2
0xC80[2:0] (BOOT_CONFIG_ MISC)	OEM	xSPI FLASH Frequency	No	0	0 – 100MHz 1 – 120MHz 2 – 133MHz 3 – 166MHz 5 – 80MHz 6 – 60MHz Others – Reserved
0xC80[4:3] (BOOT_CONFIG_ MISC)	OEM	Hold time before access the Flash device	No	0	0 – 500us 1 – 1ms 2 – 3ms 3 – 10ms
0xC80[5]	OEM	RESET_PIN _SEL Reset pin selection	No	0	0 - Primary reset pin 1 - Secondary reset pin Please refer to Table 10-1 for details.
0xC80[6]	OEM	JEDEC_HW _RESET_EN Enable JEDEC hardware reset sequence	No	0	0 - JEDEC hardware reset sequence is not performed 1 - JEDEC hardware reset sequence is performed before accessing the flash device
0xC80[7]	OEM	RESET_PIN _EN Enable reset pin	No	0	0 - External reset pin is not enabled 1 - External reset pin is enabled
0xC80[11:8] (BOOT_CONFIG_ MISC)	OEM	xSPI FLASH Dummy Cycle	No	0	0 – Dummy cycles is auto-probed Others – User specified dummy cycles for Read command
0xC80[15:12] (BOOT_CONFIG_ MISC)	OEM	xSPI FLASH image size	No	0	0 – Image size equals to Secondary Image offset 1 – 1MB

Table continues on the next page...

Table 10-10. Fuse definition for Serial NOR over FlexSPI (continued)

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
					2 – 2MB ... – ... 12 – 12MB 13 – 256KB 14 – 512KB 15 – 768KB
0xC80[23:16] (BOOT_CONFIG_ MISC)	OEM	xSPI FLASH_SEC_IMAGE_OFFSET	No	0	Offset = 256KB × fuse value
0x9A0[9:8]		FLASH_CONNECTION_SEL Select the FLASH connection options	No		0 - Connected via PORTA CS0 1 - Parallel Mode 2 - Connected Via PORTB CS0
0x9A0[10]		FLEXSPI_PIN_GROUP_SEL FlexSPI Pin Group Selection	No		0 - Primary group 1 - Secondary group

NOTE

If the xSPI FLASH Auto Probe feature is enabled, the following is the logic how this feature works with other fuse combinations:

- Flash Type - If Flash type is 0, the "xSPI FLASH Auto Probe Type" takes effect for the Flash type selection. If Flash Type is greater than 1, the "Flash Type" fuse is used for Flash type selection, ROM will issue specific command to probe the presence of Serial NOR FLASH.
- xSPI FLASH Frequency - This field is used for specifying the Flash working frequency.

10.6.1.2 FlexSPI Serial NOR Flash Boot Operation

The Boot ROM attempts to boot from Serial NOR flash if the BOOT_CFG1 [7:4] fuses are programmed to 0b’0000 as shown in the Serial NOR eFUSE Configuration table, then the ROM will initialize FlexSPI interface. FlexSPI interface initialization is a two-step process.

1. The ROM expects the 512-byte FlexSPI NOR configuration parameters (as explained in the next section) to be present at offset 0x400 in Serial NOR flash attached to FLEXSPI_A_SS0_B. The ROM can probe the presence of Serial NOR Flash and generate these configuration parameters accordingly via the combination of Fuses in the table above, or reads these configuration parameters using the read command specified by BOOT_CFG 2[2:0] with serial clock operating at 30 MHz.
2. ROM configures FlexSPI interface with the parameters provided in configuration block read from Serial NOR flash and starts the boot procedure. Refer to Table 10-18 for details regarding FlexSPI configuration parameters and to the FlexSPI NOR boot flow chart for detailed boot flow chart of FlexSPI NOR.

Both booting an XIP and non XIP image are supported from Serial NOR Flash. For XIP boot, the image has to be built for FlexSPI address space; and for non XIP, the image can be built to execute from internal RAM.

The Boot ROM supports booting from the secondary image in the Serial NOR flash, once the first image is failed to boot. To enable this feature, xSPI_FLASH_SEC_IMAGE_OFFSET must be burned.

NOTE

A non-XIP image's execute address space should NOT be overlapped with the reserved OCRAM region described in the section "Internal ROM/RAM memory map". And it is also highly recommended to avoid using address space near 0x00000000, typically up to 0x00000003.

10.6.1.3 FlexSPI NOR boot flow chart

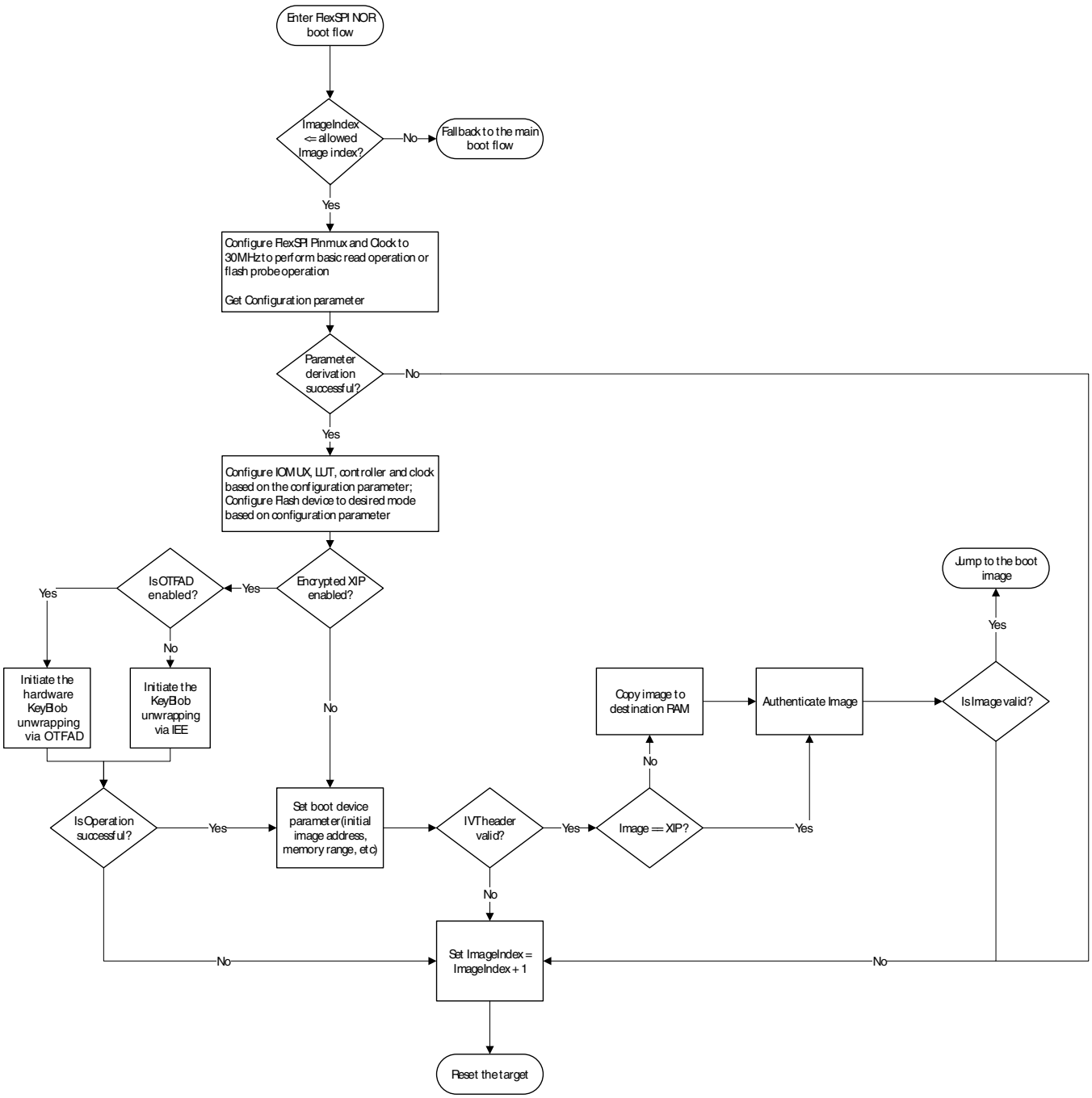


Figure 10-3. FlexSPI NOR boot flow

10.6.2 Serial NAND Flash Boot over FlexSPI

The boot ROM supports a number of Serial NAND Flash devices from different vendors. The Embedded Error Correction and Control (ECC) module in SPI NAND devices are used to detect and correct the errors.

10.6.2.1 Serial NAND eFUSE Configuration

The boot ROM determines the configuration of external Serial NAND flash by parameters, either provided by eFUSE, or sampled on GPIO pins during boot. See below table for parameters details.

Table 10-11. Fuse definition for Serial NAND over FlexSPI

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
BOOT_CFG1[1:0]	OEM	Search Stride for FCB and DBBT Search strides in terms of page	Yes	0	0 - 64 1 - 128 2 - 256 3 - 32
BOOT_CFG1[3:2]	OEM	Hold Time before access to Serial NAND	Yes	0	0 - Hold time determined by Read Status command 1 - 500us 2 - 1ms 3 - 3ms
BOOT_CFG1[4]	OEM	Column address width	Yes	0	0 - 12 bits 1 - 13 bits
BOOT_CFG1[5]	OEM	Default safe communication frequency	Yes	0	0 – High Speed (50MHz) 1 – Low Speed (30MHz)
BOOT_CFG1[7:6]	OEM	Primary boot device selection	Yes	0	00 – Serial NOR 11 – Serial NAND
BOOT_CFG2[0]	OEM	Boot Search Count of FCB and DBBT	Yes	0	0 - 1 1 - 2
BOOT_CFG2[2:1]	OEM	CS de-asserted interval between two commands	Yes	0	0 – 100ns 1 – 200ns 2 – 400ns 3 – 50ns
BOOT_CFG2[3]	OEM	FlexSPI instance	Yes		0 - FlexSPI1 1 - FlexSPI2

Table continues on the next page...

Table 10-11. Fuse definition for Serial NAND over FlexSPI (continued)

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
0x9A0[9:8]	OEM	FLASH_CONNECTION_SEL	No	0	Flash connection section. 0 - PORTA 2 - PORTB
0x9A0[10]	OEM	FLEXSPI_PIN_GROUP_SEL	No	0	FlexSPI Pin group selection. 0 - Primary group 1 - 2nd pin group
0xC80[4:0]	OEM	PAGE_READ_TIME	No	0	ROM waits an interval between page read and read cache command. 0 - 75µs Others - PAGE_READ_TIME * 10µs This fuse takes effect if the BYPASS_READ_STATUS fuse is blown.
0xC80[7]	OEM	SPI NAND BOOT - Override Busy Offset	No	0	0 – Use default busy bit offset 0 1 – Override default busy bit offset using Busy bit offset
0xC80[13:8]	OEM	SPI NAND BOOT - Busy Bit offset	No	0	
0xC80[14]	OEM	BYPASS_ECC_READ	No	0	Bypass ECC read if the default does not support ECC feature.
0xC80[15]	OEM	BYPASS_READ_STATUS	No	0	Bypass the FLASH busy check via read status command. Use PAGE_READ_TIME FUSE field instead.
0xC80[23:16]	OEM	Page Read Command	No	0	Available only if it is not 0
0xC80[31:24]	OEM	Cache Read command	No	0	Available only if it is not 0

NOTE

BOOT_CFGx sampled on GPIO pins depends on BT_FUSE_SEL setting.

NOTE

All the fuse fields on 0xC80 are defined for the NAND FLASH device which does not support the de-facto SPI NAND standard. For all the existing ECC-free SPI NAND FLASH, these fuse fields are not needed.

10.6.2.2 FlexSPI NAND Flash Boot Flow and Boot Control Blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As part of the NAND media initialization, the ROM driver uses safe NAND timings to search for a Firmware Configuration Block (FCB) that contains the optimum NAND timings, page address of Discovered Bad Block Table (DBBT) Search Area and start page address of primary and secondary firmware.

The built-in HW ECC in Serial NAND device is used during data read, the FCB data structure is protected using CRC checksum. Driver reads raw 2048 bytes of first sector and runs through CRC check that determines whether FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments page number to Search Stride number of pages to read for the next BCB until SearchCount pages have been read.

If search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters into serial download mode.

The FCB contains the page address of DBBT Search Area, and the page address for primary and secondary boot images. DBBT is searched in DBBT Search Area just like how FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using starting page address from FCB.

Figure 10-4 shows the state diagram of FCB search.

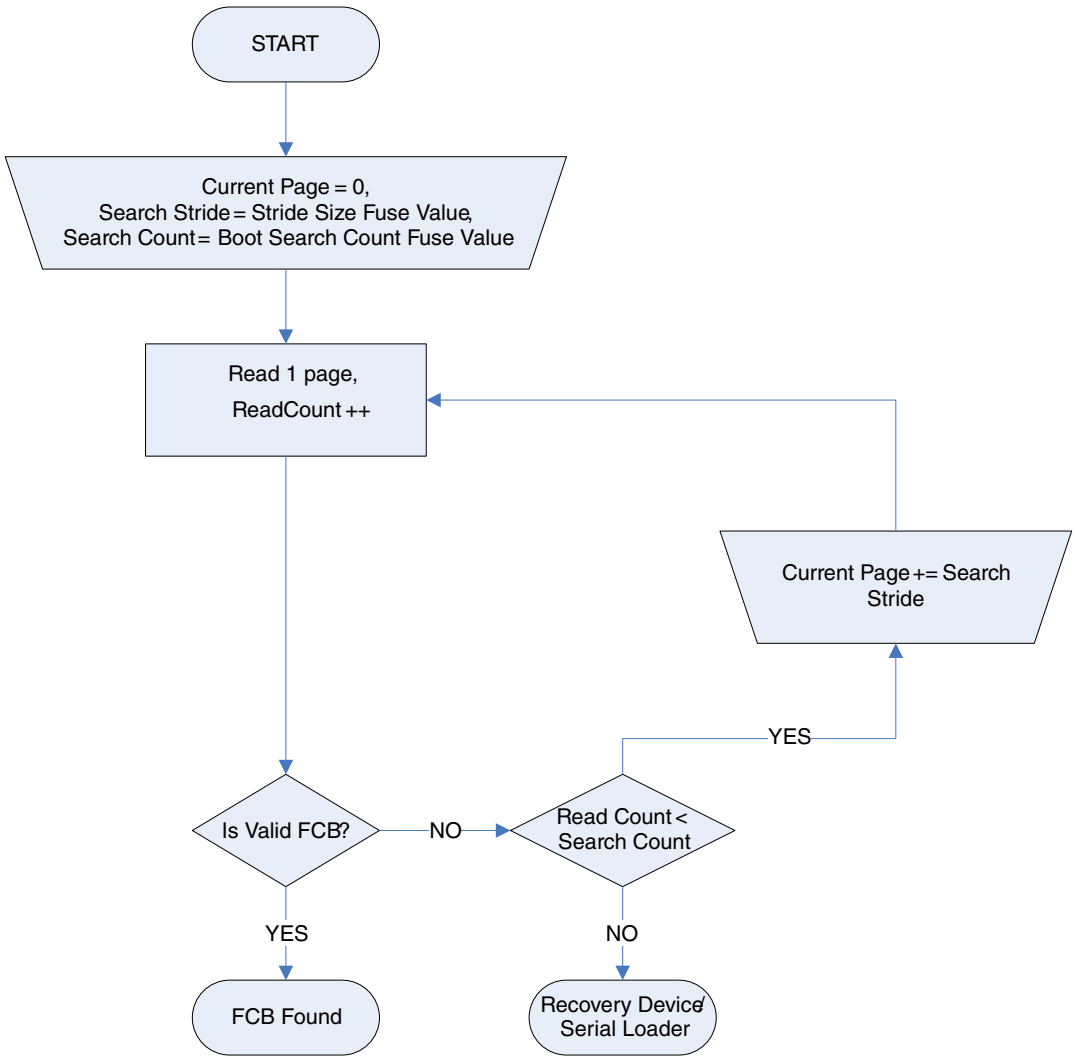


Figure 10-4. FCB Search Flow

After FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If DBBTSearchStartPage is 0 in the FCB, then ROM assumes that there are no bad blocks in the NAND device boot area. See Figure 10-5 for the DBBT search flow.

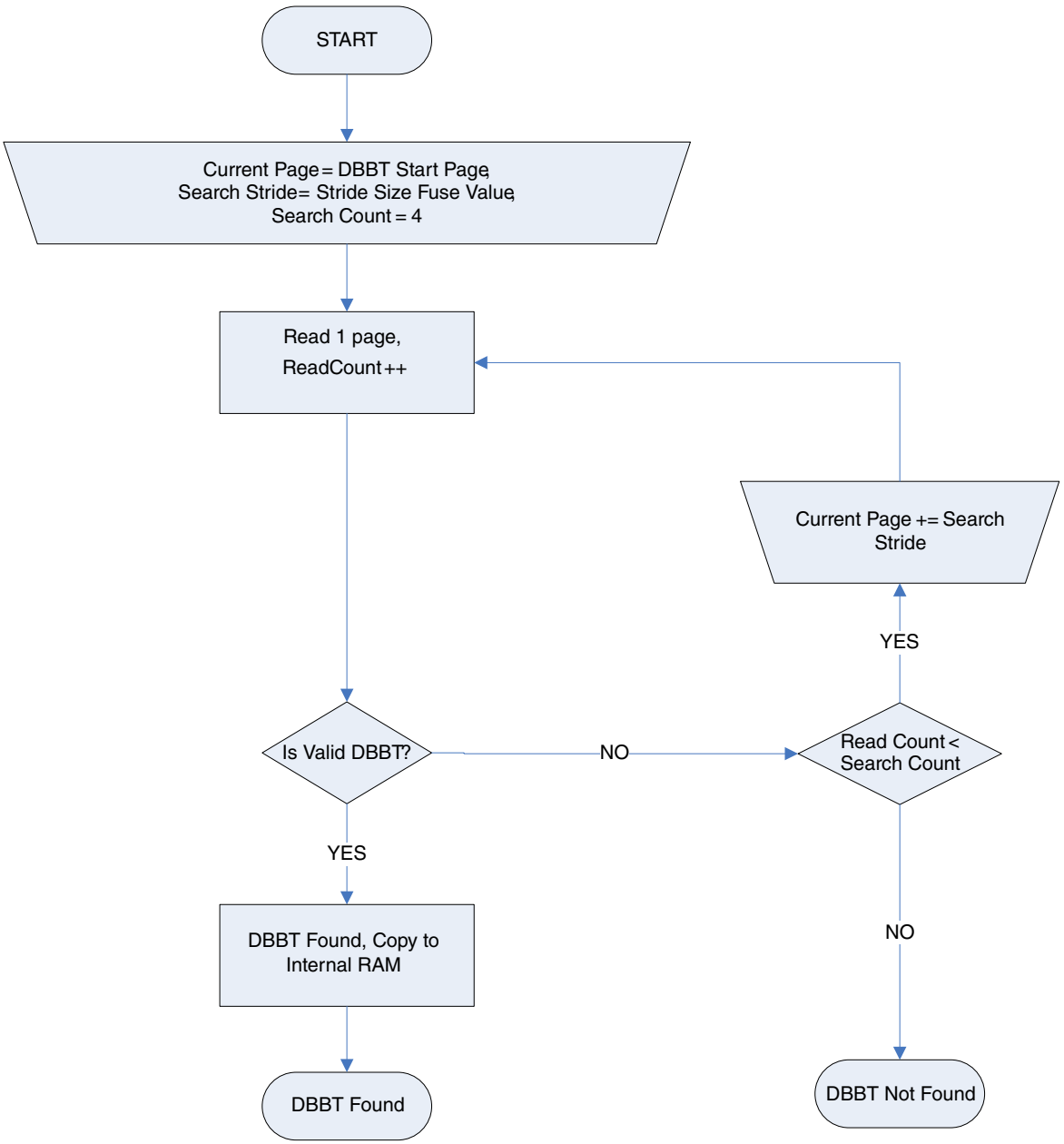


Figure 10-5. DBBT Search Flow

If during primary image read there is a page with a number of errors higher than ECC can correct, the boot ROM will turn on PERSIST_SECONDARY_BOOT bit and perform SW reset (After SW reset, secondary image is used).

If during secondary image read there is a page with number of errors higher than ECC can correct, the boot ROM goes to serial downloader.

10.6.2.3 Firmware Configuration Block

The FCB is at the first page in the first good block. The FCB should be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, in case the first Serial NAND block becomes corrupted, the ROM will find its copy in the next Serial NAND block. The search area should span over at least two Serial NAND blocks. The location information for DBBT search area and images are all specified in the FCB. The following table shows the Flash Control Block Structure.

Table 10-12. Flash Control Block Structure

Name	Offset	Size Bytes	Description
crcChecksum	0x000	4	Checksum
fingerprint	0x004	4	0x4E46_4342 ASCII: “NFCB”
version	0x008	4	0x0000_0001
DBBTSearchStartPage	0x00C	4	Start Page address for bad block table search area
searchStride	0x010	2	Search stride for DBBT and FCB search. Not used by ROM Max value is 8.
searchCount	0x012	2	Copies of DBBT and FCB. Not used by ROM, max value is 8.
firmwareCopies	0x014	4	Firmware copies Valid range 1-4.
Reserved	0x018	40	Reserved for future use Must be set to 0.
firmwareInfoTable	0x40	64	This table consists of (up to 8 entries): Field / Size / Description StartPage / 4 / Start page of this firmware pageCount / 4 / Pages in this firmware

Table continues on the next page...

Table 10-12. Flash Control Block Structure (continued)

Name	Offset	Size Bytes	Description
			NOTE: The StartPage must be the first page of a NAND block.
Reserved	0x080	128	Reserved Must be set to 0
spiNandConfigBlock	0x100	512	Serial NAND configuration block over FlexSPI
Reserved	0x300	256	Must be set to 0

NOTE

1. The “crcChecksum” is calculated with an MPEG2 variant of CRC-32. See Table 10-13 for more details.
2. The “crcChecksum” calculation starts from fingerprint to the end of FCB, 1020 bytes in total.
3. The “spiNandConfigBlock” is FlexSPI NAND configuration block which consists of common FlexSPI memory configuration block and Serial NAND specified configuration parameters.

Table 10-13. CRC-32 variant algorithm

Property	Description
Width	32 bits
Polynomial	0x04C11DB7
Init Value	0xFFFFFFFF
Reflect in	False
Reflect Out	False
XOR Out	0x00000000

10.6.2.4 Discovered Bad Blocks Table (DBBT)

Table 10-14. DBBT Structure

Name	Offset	Size in Bytes	Decription
crcChecksum	0x000	4	Checksum
Fingerprint	0x004	4	32-bit word with a value of 0x4442_4254, in ASCII “DBBT”

Table continues on the next page...

Table 10-14. DBBT Structure (continued)

Name	Offset	Size in Bytes	Decription
Version	0x008	4	32-bit version number, this version of DBBT is 0x00000001
-	0x00C	4	Reserved
badBlockNumber	0x010	4	Number of bad blocks
Reserved	0x014	12	Must be filled with 0x00s
Bad Block entries	0x020	1024 (256*4)	Each bad block entry is a 32-bit value specifying the number of a found bad block. The number of valid bad block entries is specified by the badBlockNumber field, where valid bad block entries are stored sequentially starting at the beginning of the bad block entries segment. Unused bad block entries (those beyond the badBlockNumber) should be filled with 0xFs.

NOTE

1. Maximum badBlockNumber is 256.
2. The "crcChecksum" is calculated with the same algorithm as the one in FCB, from Fingerprint to the end of DBBT, 1052 bytes in total.

10.6.2.5 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

10.6.3 Serial NOR and NAND Configuration based on FlexSPI Interface

The ROM SW supports Serial NOR and Serial NAND based on FlexSPI module, using a 448-bytes common FlexSPI configuration block and several specified parameters for Serial NOR and Serial NAND respectively. See the following sections for more details.

10.6.3.1 FlexSPI Configuration Block

FlexSPI Configuration block consists of parameters regarding specific Flash devices including read command sequence, quad mode enablement sequence (optional), etc.

Table 10-15. FlexSPI Configuration block

Name	Offset	Size(bytes)	Description
Tag	0x000	4	0x42464346, ascii: 'FCFB'
Version	0x004	4	[07:00] bugfix = 0 [15:08] minor [23:16] major = 1 [31:24] ascii 'V'
-	0x008	4	Reserved
readSampleClkSrc	0x00C	1	0 – internal loopback 1 – loopback from DQS pad 3 – Flash provided DQS
csHoldTime	0x00D	1	Serial Flash CS Hold Time Recommend default value is 0x03
csSetupTime	0x00E	1	Serial Flash CS setup time Recommended default value is 0x03
columnAdressWidth	0x00F	1	3 – For HyperFlash 12/13 – For Serial NAND, see the NAND FLASH datasheet to find the correct value 0 – Other devices
deviceModeCfgEnable	0x010	1	Device Mode Configuration Enable feature 0 – Disabled 1 – Enabled
deviceModeType	0x011	1	0 - Generic 1 - Quad Enable 2 - SPI-to-xSPI mode 3 - xSPI-to-SPI mode 4 - SPI-to-NoCmd mode
waitTimeCfgCommands	0x012	2	Wait time for all configuration commands, unit 100us. Available for device that support v1.1.0 FlexSPI configuration block. If it is greater than 0, ROM will wait waitTimeCfgCommands * 100us for all device memory configuration commands instead of using read status to wait until these commands complete.
deviceModeSeq	0x014	4	Sequence parameter for device mode configuration Bit[7:0] - number of LUT sequences for Device mode configuration command

Table continues on the next page...

Table 10-15. FlexSPI Configuration block (continued)

Name	Offset	Size(bytes)	Description
			Bit[15:8] - starting LUT index of Device mode configuration command Bit[31:16] - must be 0
deviceModeArg	0x018	4	Device Mode argument, effective only when deviceModeCfgEnable = 1
configCmdEnable	0x01C	1	Config Command Enable feature 0 – Disabled 1 – Enabled
-	0x01D	3	Reserved
configCmdSeqs	0x020	12	Sequences for Config Command, allow 3 separate configuration command sequences.
-	0x02C	4	Reserved
cfgCmdArgs	0x030	12	Arguments for each separate configuration command sequence.
-	0x03C	4	Reserved
controllerMiscOption	0x040	4	Bit0 – differential clock enable Bit1 – CK2 enable, must set to 0 in this silicon Bit2 – ParallelModeEnable Bit3 – wordAddressableEnable Bit4 – Safe Configuration Frequency enable set to 1 for the devices that support DDR Read instructions Bit5 – Pad Setting Override Enable Bit6 – DDR Mode Enable, set to 1 for device supports DDR read command Bit 7 - Unused Bit 8 - Second Pin group Bit 9 - Second DQS pin group Bit 10 - Write Mask Enable Bit 11 - Write Opt1 Clear
deviceType	0x044	1	1 – Serial NOR 2 – Serial NAND 3 - Serial RAM (HyperRAM/APMemory)
sflashPadType	0x045	1	1 – Single pad 2 – Dual pads 4 – Quad pads 8 – Octal pads
serialClkFreq	0x046	1	Chip specific value, for this silicon 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz

Table continues on the next page...

Table 10-15. FlexSPI Configuration block (continued)

Name	Offset	Size(bytes)	Description
			4 – 80 MHz 5 – 100 MHz 6 – 120 MHz 7 – 133 MHz 8 – 166 MHz Other value: 30 MHz
lutCustomSeqEnable	0x047	1	0 – Use pre-defined LUT sequence index and number 1 - Use LUT sequence parameters provided in this block
-	0x048	8	Reserved
sflashA1Size	0x050	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
sflashA2Size	0x054	4	The same as above
sflashB1Size	0x058	4	The same as above
sflashB2Size	0x05C	4	The same as above
csPadSettingOverride	0x060	4	Set to 0 if it is not supported
sclkPadSettingOverride	0x064	4	Set to 0 if it is not supported
dataPadSettingOverride	0x068	4	Set to 0 if it is not supported
dqsPadSettingOverride	0x06C	4	Set to 0 if it is not supported
timeoutInMs	0x070	4	Maximum wait time during read busy status 0 – Disabled timeout checking feature Other value – Timeout if the wait time exceeds this value
commandInterval	0x074	4	Unit: ns Currently, it is used for SPI NAND only at high frequency
dataValidTime	0x078	4	Time from clock edge to data valid edge, unit ns. This field is used when the FlexSPI Root clock is less than 100 MHz and the read sample clock source is device provided DQS signal without CK2 support. [31:16] data valid time for DLLB in terms of 0.1 ns [15:0] data valid time for DLLA in terms of 0.1 ns
busyOffset	0x07C	2	busy bit offset, valid range :0-31
busyBitPolarity	0x07E	2	0 – busy bit is 1 if device is busy 1 – busy bit is 0 if device is busy
lookupTable	0x080	256	Lookup table
lutCustomSeq	0x180	48	Customized LUT sequence, see below table for details.
	0x1B0	16	Reserved for future use

Note:

1. To customize the LUT sequence for some specific device, users need to enable “lutCustomSeqEnable” and fill in corresponding “lutCustomSeq” field specified by command index below.

2. For Serial (SPI) NOR, the pre-defined LUT index is as follows:

Table 10-16. LUT sequence definition for Serial NOR

Command Index	Name	Index in lookup table	Description
0	Read	0	Read command Sequence
1	ReadStatus	1	Read Status command
2	WriteEnable	3	Write Enable command sequence
3	EraseSector	5	Erase Sector Command
4	PageProgram	9	Page Program Command
5	ChipErase	11	Full Chip Erase
6	Dummy	15	Dummy Command as needed
7-12	Reserved	2,4,6,7,8,10,12,13,14	All reserved indexes can be freely used for other purpose
13	NOR_CMD_LUT_SEQ_IDX_READ_SFDP	13	Read SFDP sequence in lookupTable id stored in config block
14	NOR_CMD_LUT_SEQ_IDX_RESTORE_NOCMD	14	Restore 0-4-4/0-8-8 mode sequence id in lookupTable stored in config block

3. For Serial (SPI) NAND, the pre-defined LUT index is as follows:

Table 10-17. LUT sequence definition for Serial NAND

Command Index	Name	Index in lookup table	Description
0	ReadFromCache	0	Read from cache
1	ReadStatus	1	Read Status
2	WriteEnable	3	Write Enable
3	BlockErase	5	Erase block
4	ProgramLoad	9	Program Load
5	ReadPage	11	Read page to cache
6	ReadEccStatus	13	Read ECC Status
7	ProgramExecute	14	Program Execute
8	ReadFromCacheOdd	4	Read from Cache while page in odd plane
9	ProgramLoadOdd	10	Program Load for pages within odd blocks
	Reserved	2,6,7,8,12,15	All reserved indexes can be freely used for other purposes

NOTE

1. All the pre-defined LUT indexes are only applicable to boot stage. User application can use the whole 16 LUT entries freely based on their requirement.
2. The FlexSPI NOR ROM APIs occupy LUT index 0-5. User application should **NOT** use these LUT indexes if the ROM API is called in their application frequently.

10.6.3.2 Serial NOR configuration block (512 bytes)

Table 10-18. Serial NOR configuration block

Name	Offset	Size (Bytes)	Description
memCfg	0	448	The common memory configuration block, see FlexSPI configuration block for more details
pageSize	0x1C0	4	Page size in terms of bytes, not used by ROM
sectorSize	0x1C4	4	Sector size in terms of bytes, not used by ROM
ipCmdSerialClkFreq	0x1C8	1	Chip specific value, not used by ROM 0 – No change, keep current serial clock unchanged 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 80 MHz 5 – 100 MHz 6 – 120 MHz – 133 MHz
isUniformBlockSize	0x1C9	1	Device has uniform block size 0 - No, block size != sector size 1 - Yes, block size = sector size
isDataOrderSwapped	0x1CA	1	The data order is swapped in OPI DDR mode 0 - Not swapped 1 - Swapped (this is applicable to Macronix MX25UM512/256/128 series only)

Table continues on the next page...

Table 10-18. Serial NOR configuration block (continued)

Name	Offset	Size (Bytes)	Description
Reserved	0x1CB	5	Reserved
blockSize	0x1D0	4	blockSize
flashStateCtx	0x1D4	4	Flash State Context after being configured [7:0] Flash POR Mode: 0x00 - Extended SPI mode 0x42 - QPI DDR mode 0x82 - OPI DDR mode -- [15:8] Flash Current Mode: 0x00 - Extended SPI mode 0x42 - QPI DDR mode 0x82 - OPI DDR mode -- [23:16] Reserved -- [31:24] Flash Restoring Sequence: 0x6 - Send 0x66_0x99 (Micron Octal Flash) 0x7 - Send 0x6699_0x9966 (Macronix Octal Flash) 0x8 - Send 0x06 0xFF (Adesto Octal Flash)
Reserved	0x1D8	24	Reserved

10.6.3.3 Serial NAND configuration block (512 bytes)

Table 10-19. Serial NAND configuration block

Name	Offset	Size (Bytes)	Description
memCfg	0	448	The common memory configuration block, see FlexSPI configuration block for more details
pageDataSize	0x1C0	4	Page size in terms of bytes, usually, it is 2048 or 4096
pageTotalSize	0x1C4	4	It equals to 2 ^ width of column adddress
pagesPerBlock	0x1C8	4	Pages in one block

Table continues on the next page...

Table 10-19. Serial NAND configuration block (continued)

Name	Offset	Size (Bytes)	Description
bypassReadStatus	0x1CC	1	0 – Read Status Register 1 – Bypass Read status register
bypassEccRead	0x1CD	1	0 – Perform ECC read 1 – Bypass ECC read
hasMultiPlanes	0x1CE	1	0 – Only 1 plane 1 – Has two planes
skippOddBlocks	0x1CF	1	0 – Read Odd blocks 1 – Skip Odd blocks
eccCheckCustomEnable	0x1D0	1	0 – Use the common ECC check command and ECC related masks 1 - Use ECC check related masks provided in this configuration block
ipCmdSerialClkFreq	0x1D1	1	Chip specific value, not used by ROM 0 – No change, keep current serial clock unchanged 1 – 30 MHz 2 – 50 MHz 3 – 60 MHz 4 – 75 MHz 5 – 80 MHz 6 – 100 MHz 7 – 133 MHz 8 – 166 MHz
readPageTimeUs	0x1D2	2	Wait time during page read, this field will take effect on if the bypassReadStatus is set to 1. **NOTE:** Only applicable to ROM.
eccStatusMask	0x1D4	4	ECC Status Mask
eccFailureMask	0x1D8	4	ECC Check Failure mask
blocksPerDevice	0x1DC	4	Blocks in a Serial NAND
Reserved	0x1ED	32	Reserved for future use

Below is an example of Serial NAND configuration block for Winbond W25N01GVZEIG:

```
const flexspi_nand_config_t kSerialNandCfgBlk =
{
```

```
    .memConfig =
    {
        .tag = FLEXSPI_CFG_BLK_TAG,
        .version = FLEXSPI_CFG_BLK_VERSION,
        .readSampleClkSrc = kFlexSPIReadSampleClk_LoopbackInternally,
        .dataHoldTime = 3,
        .dataSetupTime = 3,
        .columnAddressWidth = 12,
        .deviceModeCfgEnable = 1,
        .deviceModeSeq = { 1, 2 },
        .deviceType = kFlexSpiDeviceType_SerialNAND,
        .sflashPadType = kSerialFlash_4Pads,
        .serialClkFreq = kFlexSpiSerialClk_50MHz,
        .lutCustomSeqEnable = 0,
        .sflashA1Size = 128 * 1024 * 1024U * 2, // Flash size = 2 * actual data size
(exclude spare space)
        .lookupTable =
        {
            // Read cache 4 I/0
            [4 * NOR_CMD_LUT_SEQ_IDX_READ] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0xEB,
CADDR_SDR, FLEXSPI_4PAD, 0x10),
            [4 * NOR_CMD_LUT_SEQ_IDX_READ + 1] = FLEXSPI_LUT_SEQ(DUMMY_SDR, FLEXSPI_4PAD,
0x04, READ_SDR, FLEXSPI_4PAD, 0x80),
            // Clear Status1 flag
            [4 * 2] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0x1F, CMD_SDR, FLEXSPI_1PAD,
0xA0),
            [4 * 2 + 1] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD, 0x00, STOP, FLEXSPI_1PAD,
0x00),
            // Read Page
            [4 * NAND_CMD_LUT_SEQ_IDX_READPAGE] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x13, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Read Status
            [4 * NAND_CMD_LUT_SEQ_IDX_READSTATUS] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x0F, CMD_SDR, FLEXSPI_1PAD, 0xC0),
            [4 * NAND_CMD_LUT_SEQ_IDX_READSTATUS + 1] = FLEXSPI_LUT_SEQ(READ_SDR,
FLEXSPI_1PAD, 0x01, STOP, FLEXSPI_1PAD, 0),

            // Write Enable
            [4 * NAND_CMD_LUT_SEQ_IDX_WRITEENABLE] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x06, STOP, FLEXSPI_1PAD, 0),

            // Page Program Load 4x
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMLOAD] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x32, CADDR_SDR, FLEXSPI_1PAD, 0x10),
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMLOAD + 1] = FLEXSPI_LUT_SEQ(WRITE_SDR,
FLEXSPI_4PAD, 0x40, STOP, FLEXSPI_1PAD, 0),
            // Page Program Execute
            [4 * NAND_CMD_LUT_SEQ_IDX_PROGRAMEXECUTE] = FLEXSPI_LUT_SEQ(CMD_SDR,
FLEXSPI_1PAD, 0x10, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Erase Sector
            [4 * NAND_CMD_LUT_SEQ_IDX_ERASEBLOCK] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0xD8, RADDR_SDR, FLEXSPI_1PAD, 0x18),
            // Read ECC status
            [4 * NAND_CMD_LUT_SEQ_IDX_READECCSTAT] = FLEXSPI_LUT_SEQ(CMD_SDR, FLEXSPI_1PAD,
0x0F, CMD_SDR, FLEXSPI_1PAD, 0xC0),
            [4 * NAND_CMD_LUT_SEQ_IDX_READECCSTAT + 1] = FLEXSPI_LUT_SEQ(READ_SDR,
FLEXSPI_1PAD, 0x01, STOP, FLEXSPI_1PAD, 0),
        },
    },
    .pageDataSize = 2048,
    .pageTotalSize = 4096,
    .pagesPerBlock = 64,
};
```

10.6.4 Parallel NAND flash Boot over SEMC

The boot ROM supports a number of Parallel SLC NAND flash devices from different vendors. Both the Error Correction and Control (ECC) module in NAND device and software ECC algorithm (SECDED) in boot ROM can be used to detect the errors, based on the fuse settings.

10.6.4.1 Parallel NAND eFuse Configuration

The boot ROM determines the configuration of external Parallel NAND flash by parameters, either provided by eFuse, or sampled on GPIO pins, during boot. See below table for parameters details:

Table 10-20. Fuse definition for Parallel NAND over SEMC

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
BOOT_CFG1[0]	OEM	Boot Search count of FCB and DBBT	Yes	0	0 – 1 1 – 2
BOOT_CFG1[4:1]	OEM	Search stride for FCB and DBBT in terms of pages	Yes	0000	0000 – 64 Others – 2 ^ BOOT_SEARCH_STRIDE
BOOT_CFG1[7:5]	OEM	Primary boot device selection	Yes	000	001 – Parallel NAND
BOOT_CFG>2[0]	OEM	NAND ONFI compliant	Yes	0	0 – ONFI 1.0 1 – Non-ONFI
BOOT_CFG2 [1]	OEM	SEMC EDO Mode	Yes	0	0 – EDO Mode 1 – Non-EDO Mode
BOOT_CFG2 [2]	OEM	SEMC Access Command	Yes	0	0 – IPG 1 – AXI
0x6E0[20:19]	OEM	SEMC BCH mode	No	0	0 - BCH is not enabled 1 - BCH4 2 - BCH8
0xC80[3]	OEM	I/O port size			0 - 8 bit 1 - 16 bit
0xC80[4]	OEM	ECC algorithm selection	Yes	0	0 - NAND flash-built-in ECC 1 - Software ECC (SECDED) **NOTE:** For “ECC selection” option, it can only be set as Device ECC When NAND device has built-in ECC module and the ECC

Table continues on the next page...

Table 10-20. Fuse definition for Parallel NAND over SEMC (continued)

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
					module is enabled by default.
0xC80[5]	OEM	RDY pin polarity	No		0 – Low active 1 – High active
0xC80[6]	OEM	Ready check type	No		0 – Status Register 1 – R/B# pin
0xC80[7]	OEM	SEMC clock frequency			0 - 133MHz (2nd max) 1 - 166MHz (Max)
0xC80[10:8]	OEM	Row Column address mode	No	000	Applicable only for Non-ONFI device 00x – 5 bytes (CA2+RA3) 010 – 4 bytes (CA2+RA2) 011 – 3 bytes (CA2+RA1) 10x – 4 bytes (CA1+RA3) 110 – 3 bytes (CA1+RA2) 111 – 2 bytes (CA1+RA1)
0xC80[13:11]	OEM	Column address width	No	000	Applicable only for Non-ONFI device 000 – 12 bits 001 – 09 bits 010 – 10 bits 011 – 11 bits 100 – 13 bits 101 – 14 bits 110 – 15 bits 111 – 16 bits
0xC80[14]	OEM	Status command type	No	0	Applicable only for Non-ONFI device 0 – Common 1 – Enhanced
0xC80[18:16]	OEM	Pages in block	No	000	Applicable only for Non-ONFI device 000 – 128 pages 001 – 8 pages 010 – 16 pages 011 – 32 pages 100 – 64 pages 101 – 256 pages 110 – 512 pages

Table continues on the next page...

Table 10-20. Fuse definition for Parallel NAND over SEMC (continued)

Fuse	Config	Definitions	GPIO	Shipped Value	Settings
					111 – 1024 pages
0xC80[20:19]	OEM	PCS Selection	No	0	PCS Selection 0 - CS0 1 - CS1 2 - CS2 3 - CS3
0xC80[24]	OEM	Device ECC initial status	No	0	Applicable only for ONFI 1.0 device 0 – Enabled 1 – Disabled
0xC80[27:25]	OEM	ONFI timing mode	No		000 - Mode0 -10MHz 001 - Mode1 -20MHz 010 - Mode2 -28MHz 011 - Mode3 -33MHz 100 - Mode4 -40MHz 101 - Mode5 -50MHz 11x - Fastest Mode

10.6.4.2 Parallel NAND Flash Boot Control Blocks (BCB)

There are two BCB data structures:

- Firmware Configuration Block (FCB)
- Discovered Bad Blocks Table (DBBT)

As part of the Parallel NAND media initialization, the ROM driver uses proper Parallel NAND parameters specified by FUSE to search for an FCB that contains the complete Parallel NAND parameters, page address of DBBT Search Area and Image info, including image copies, start page address and image size in terms of pages for each image.

FCB data structure is protected using Embedded ECC module in Parallel NAND devices or software ECC in ROM. The ROM driver reads 2048 bytes of first sector and checks the ECC check status to determine whether FCB data is valid or not.

If the FCB is found, the complete NAND parameters (Parallel NAND configuration block) are loaded for further reads, if the ECC fails, or the fingerprint does not match, or the CRC checksum does not match, the Block Search state machine increments page number to Search Stride number of pages to read for the next FCB until Search Count pages have been read.

If search fails to find a valid FCB, the Parallel NAND driver responds with an error and the boot ROM enters into Recovery boot mode (Secondary boot, if it is enabled, or Serial download mode).

The FCB contains the page address of DBBT Search Area, and the info for images. DBBT is searched in DBBT Search area just like how FCB is searched. After the FCB is read, the DBBT is loaded, then the boot image is loaded using starting page address from FCB.

10.6.4.3 Firmware Configuration Block (FCB)

The FCB is at the first page in the first good block. The FCB should be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, in case the first Serial NAND block becomes corrupted, the ROM will find its copy in the next Parallel NAND block. The search area should span over at least two Parallel NAND blocks. The location information for DBBT search area and images are all specified in the FCB. Below Table shows the FCB Structure.

Name	offset	Size (Bytes)	Description
bcbHeader	0x000	12	See Table 10-21 for details
DBBTSerachAreaStartPage	0x00c	4	Start Page address for bad block table search area
searchStride	0x010	2	Search stride for DBBT and FCB search. Not used by ROM Max value is 8.
searchCount	0x012	2	Copies of DBBT and FCB. Not used by ROM, max value is 8.
firmwareCopies	0x014	4	Firmware copies Valid range 1-4.
-	0x018	40	Reserved
firmwareTable	0x040	64	For details see Table 10-22
-	0x080	128	Reserved
nandConfig	0x100	256	Parallel NAND configuration block over SEMC
-	0x200	512	Reserved

Table 10-21. Header Description

Field	Size	Description
crcChecksum	4	Checksum
fingerprint	4	0x4E46_4342 ASCII: “NFCB”
version	4	0x0000_0001

Table 10-22. Table Descriptions

Field	Size	Description
startPage	4	Start page of this firmware
pagesInFirmware	4	Pages in this firmware

NOTE

- The “crcChecksum” is calculated with an MPEG2 variant of CRC-32.
- The “crcChecksum” calculation starts from fingerprint to the end of FCB, 1020 bytes in total.
- The “nandConfig” is SEMC NAND configuration block which consists of common SEMC memory configuration block and Parallel NAND specified configuration parameters. See Parallel NAND eFuse Configuration for more details.

10.6.4.4 Discovered Bad Blocks Table (DBBT)

Table 10-23. DBBT Structure

Name	offset	Size (Bytes)	Description
bcbHeader	0x000	12	See Table 10-24 for details
-	0x00C	4	Reserved
badBlockNumber	0x010	4	Number of bad blocks
-	0x014	12	Reserved
badBlockTable	0x020	1024 (256*4)	Each bad block entry is a 32-bit value specifying the number of a found bad block. The number of valid bad block entries is specified by the badBlockNumber field, where valid bad block entries are stored sequentially starting at the beginning of the bad block entries segment. Unused bad block entries (those beyond the badBlockNumber) should be filled with 0xFs.

Table 10-24. Header Description

Field	Size	Description
crcChecksum	4	Checksum
fingerprint	4	0x4442_4254 ASCII: “DBBT”
version	4	0x0000_0001

NOTE

- Maximum bad block number is 256.
- The "crcChecksum" is calculated with the same algorithm as the one in FCB, from Fingerprint to the end of DBBT, 1052 bytes in total.

10.6.4.5 Bad block handling in ROM

During firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block is checked.

10.6.5 Parallel NAND configuration based on SEMC interface

The ROM SW supports Parallel NAND based on SEMC module, using an 80-bytes common SEMC configuration block and several specified parameters for Parallel NAND. See the following sections for more details.

10.6.5.1 SEMC Configuration Block

SEMC Configuration block consists of all parameters related to specific Flash devices.

Table 10-25. SEMC control block structure

Name	offset	Size (Bytes)	Description
tag	0x000	4	0x434D4553, ascii:”SEMC”
version	0x004	4	0x00010000 [07:00] bugfix = 0 [15:08] minor = 0 [31:16] major = 1

Table continues on the next page...

Table 10-25. SEMC control block structure (continued)

Name	offset	Size (Bytes)	Description
deviceMemType	0x008	1	0 – NOR Flash 1 – NAND Flash
accessCommandType	0x009	1	0 – IPG bus command 1 – AXI32 command
-	0x00A	2	Reserved
asyncClkFreq	0x00C	1	0 – 33MHz 1 – 40MHz 2 – 50MHz 3 – 66MHz 4 – 108MHz 5 – 133MHz 6 – 166MHz
busTimeoutCycles	0x00D	1	0 – 255 * 1024 cycles n – n * 1024 cycles
commandExecutionTimeoutCycles	0x00E	1	0 – 256 * 1024 cycles n – n * 1024 cycles
readStrobeMode	0x00F	1	0 – Dummy read strobe loopbacked internally 1 – Dummy read strobe loopbacked from DQS pad
norMemConfig	0x010	64	See detail in Table SEMC NOR control block structure
nandMemConfig	0x050	64	See detail in Table SEMC NAND control block structure

Table 10-26. SEMC NAND control block structure

Name	offset	Size (Bytes)	Description
axiMemBaseAddress	0x00	4	SoC level Base Address for NAND AXI command
axiMemSizeInByte	0x04	4	SoC level Memory size for NAND AXI command
ipgMemBaseAddress	0x08	4	SoC level Base Address for NAND IPG command
ipgMemSizeInByte	0x0c	4	SoC level Memory size for NAND IPG command
edoMode	0x10	1	0 - EDO mode disabled 1 - EDO mode enabled
ioPortWidth	0x11	1	IO Port bit number
arrayAddressOption	0x12	1	0 – 5 bytes (CA2+RA3) 1 – 4 bytes (CA1+RA3)

Table continues on the next page...

Table 10-26. SEMC NAND control block structure (continued)

Name	offset	Size (Bytes)	Description
			2 – 4 bytes (CA2+RA2) 3 – 3 bytes (CA1+RA2) 4 – 3 bytes (CA2+RA1) 7 – 2 bytes (CA1+RA1)
columnAddressWidth	0x13	1	Column address bit number
burstLengthInBytes	0x14	1	Burst Length
columnAddressOption	0x15	1	0 - PageAreaAccess 1 - SpareAreaAccess
-	0x16	10	Reserved
cePortOutputSelection	0x20	1	0 – CSX0 1 – CSX1 2 – CSX2 3 – CSX3 4 – A8
rdyPortPolarity	0x21	1	0 – Low active 1 – High active
-	0x22	14	Reserved
ceSetupTime	0x30	1	value[3:0] + 1 cycles
ceMinHoldTime	0x31	1	value[3:0] + 1 cycles
ceMinIntervalTime	0x32	1	value[3:0] + 1 cycles
weLowTime	0x33	1	value[3:0] + 1 cycles
weHighTime	0x34	1	value[3:0] + 1 cycles
reLowTime	0x35	1	value[3:0] + 1 cycles
reHighTime	0x36	1	value[3:0] + 1 cycles
weHighToReLowTime	0x37	1	value[5:0] + 1 cycles
reHighToWeLowTime	0x38	1	value[5:0] + 1 cycles
aleToDataStartTime	0x39	1	value[5:0] + 1 cycles
readyToReLowTime	0x3a	1	value[5:0] + 1 cycles
weHighToBusyTime	0x3b	1	value[5:0] + 1 cycles
asyncTurnaroundTime	0x3c	1	value[3:0] + 1 cycles
-	0x3d	3	Reserved

10.6.5.2 Parallel NAND Configuration Block (256 bytes)

Table 10-27. Parallel NAND control block structure

Name	offset	Size (Bytes)	Description
memConfig	0x000	80	See SEMC control block structure for more details
vendorType	0x050	1	0 – Micron 1 – Spansion 2 – Samsung 3 – Winbond 4 – Hynix 5 – Toshiba 6 – Macronix
cellTechnology	0x051	1	0 – SLC 1 – MLC
onfiVersion	0x052	1	0 – Non-ONFI 1 – ONFI 1.0 2 – ONFI 2.0 3 – ONFI 3.0 4 – ONFI 4.0
acTimingTableIndex	0x053	1	0 – User Defined 1 – ONFI 1.0 Mode0 10MHz 2 – ONFI 1.0 Mode1 20MHz 3 – ONFI 1.0 Mode2 28MHz 4 – ONFI 1.0 Mode3 33MHz 5 – ONFI 1.0 Mode4 40MHz 6 – ONFI 1.0 Mode5 50MHz 7 – Auto Detection
enableEccCheck	0x054	1	0 – Enabled 1 – Disabled
eccCheckType	0x055	1	0 – Software ECC 1 – Device ECC
deviceEccStatus	0x056	1	0 – Enabled 1 – Disabled
swEccAlgorithm	0x057	1	0 – SEC Hamming Code
swEccBlockBytes	0x058	4	Software ECC block bytes (256, 512)
readyCheckOption	0x05c	1	0 – Via Status Register 1 – Via R/B# signal
statusCommandType	0x05d	1	0 – Common (0x70) 1 – Enhanced (0x78)

Table continues on the next page...

Table 10-27. Parallel NAND control block structure (continued)

Name	offset	Size (Bytes)	Description
readyCheckTimeoutInMs	0x05e	2	Ready Check timeout
readyCheckIntervalInUs	0x060	2	Ready Check interval
userOnfiAcTimingModeCode	0x080	1	userOnfiAcTimingModeCode
-	0x081	31	Reserved
bytesInPageDataArea	0x0a0	4	Page Main data size
bytesInPageSpareArea	0x0a4	4	Page Spare data size
pagesInBlock	0x0a8	4	Page number in one block
blocksInPlane	0x0ac	4	Block number in one plane
planesInDevice	0x0b0	4	Plane number in Device
-	0x0b4	44	Reserved
enableReadbackVerify	0x0e0	1	0 - Enabled 1 - Disabled
-	0x0e1	3	Reserved
readbackPageBufferAddress	0x0e4	4	Read back page buffer address
-	0x0e8	24	Reserved

10.6.6 Expansion device

The ROM supports booting from the MMC/eMMC and SD/eSD compliant devices.

10.6.6.1 Expansion device eFUSE configuration

The SD/MMC/eSD/eMMC/SDXC boot can be performed using the USDHC ports. The port can be configured based on either the setting of the BOOT_CFG1[1] (Port Select) fuse or its corresponding GPIO overrides.

All USDHC ports support the fast boot. See this table for details:

Table 10-28. USDHC eFuse Descriptions

Fuse	Config	Definition	GPIO	Shipped value	Settings
BOOT_CFG1[0]	OEM	Fast boot support	Yes	0	MMC 0 - Normal boot 1 - Fast boot
BOOT_CFG1[1]	OEM	USDHC port selection	Yes	0	0 - USDHC-1 1 - USDHC-2

Table continues on the next page...

Table 10-28. USDHC eFuse Descriptions (continued)

BOOT_CFG1[2]	OEM	SD loopback clock source sel (for SDR50 and SDR104 only)	Yes	0	0 - through the SD pad 1 - direct
BOOT_CFG1[3]	OEM	SD power cycle enable	Yes	0	0 - No power cycle 1 - Power cycle enabled via the SD_RST pad
BOOT_CFG1[5:4]	OEM	SD/MMC speed mode, and eMMC acknowledge enabled selection	Yes	00	MMC: 0x - Normal speed mode 1x - High-speed mode x0 - eMMC fast boot acknowledge disable x1 - eMMC fast boot acknowledge enable SD: 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104
BOOT_CFG1[7:6]	OEM	Boot device selection	Yes	00	01 - SD/eSD/SDXC boot from the USDHC interface 10 - MMC/eMMC boot from the USDHC interface
BOOT_CFG2[0]	OEM	SD2 voltage selection	Yes	0	MMC: 0 - 3.3 V 1 - 1.8 V
BOOT_CFG2[2:1]	OEM	SD MMC bus width selection	Yes	00	SD: x0 - 1-bit x1 - 4-bit **NOTE:** To use 1-bit mode, the user has to disconnect the SD card's DATA3 line from SoC. MMC: 00 - 4-bit 01 - 8-bit 10 - 4-bit DDR (MMC 4.4) 11 - 8-bit DDR (MMC 4.4)
0x970[0]	OEM	eMMC pre-idle enabled selection	Yes	0	0 - Issue pre-idle command 1 - Do not issue
0x970[1]	OEM	USDHC2 reset polarity selection	Yes	0	0 - Reset active low 1 - Reset active high

Table continues on the next page...

Table 10-28. USDHC eFuse Descriptions (continued)

0x970[2]	OEM	USDHC1 reset polarity selection	Yes	0	0 - Reset active low 1 - Reset active high
0x970[3]	OEM	USDHC1 voltage selection	Yes	0	0 - 3.3 V 1 - 1.8 V
0x970[5]	OEM	Power stable cycle selection	Yes	0	0 - 5 ms 1 - 2.5 ms
0x970[7:6]	OEM	Power cycle selection	Yes	00	00 - 20 ms 01 - 10 ms 10 - 5 ms 11 - 2.5 ms
0x970[9:8]	OEM	SD calibration step	Yes	00	SD 00 - 1 delay cell 01 - 2 delay cells 10 - 4 delay cells 11 - 6 delay cells
0x970[10]	OEM	DLL override selection	Yes	0	0 - DLL Slave Mode for SD/eMMC (delay 10, step 2) 1 - DLL Override Mode for SD/ eMMC
0x970[15]	OEM	Disable SDMMC Manufacture mode	Yes	0	0 - Enable 1 - Disable

The boot code supports these standards:

- MMCv4.4 or less
- eMMCv4.4 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT
- SDXCv3.0

The MMC/SD/eSD/SDXC/eMMC can be connected to any of the USDHC blocks and can be booted by copying 4 KB of data from the MMC/SD/eSD/eMMC device to the internal RAM. After checking the Image Vector Table header value (0xD1) from program image, the ROM code performs a DCD check. After a successful DCD extraction, the ROM code extracts from the Boot Data Structure the destination pointer and length of image to be copied to the RAM device from where the code execution occurs.

The maximum image size to load into the SD/MMC boot is 32 MB. This is due to a limited number of uSDHC ADMA Buffer Descriptors allocated by the ROM.

NOTE

The initial 4 KB of the program image must contain the IVT, DCD, and the Boot Data structures.

Table 10-29. SD/MMC frequencies

	SD	MMC	MMC (DDR mode)
Identification (KHz)	347.22		
Normal-speed mode (MHz)	25	20	25
High-speed mode (MHz)	50	40	50
UHSI SDR50 (MHz)	100		
UHSI SDR104 (MHz)	200		

NOTE

The boot ROM code reads the application image length and the application destination pointer from the image.

10.6.6.2 MMC and eMMC boot

This table provides the MMC and eMMC boot details.

Table 10-30. MMC and eMMC boot details

Normal boot mode	During the initialization (normal boot mode), the MMC frequency is set to 347.22 KHz. When the MMC card enters the identification portion of the initialization, the voltage validation is performed, and the ROM boot code checks the high-voltage settings and the card capacity. The ROM boot code supports both the high-capacity and low-capacity MMC/ eMMC cards. After the initialization phase is complete, the ROM boot code switches to a higher frequency (20 MHz in the normal boot mode or 40 MHz in the high-speed mode). The eMMC is also interfaced via the USDHC and follows the same flow as the MMC. The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD[179] to get the boot partition to be set. If there is no boot partition mentioned in the BOOT_PARTITION_ENABLE field or the user partition was mentioned, the ROM boots from the user partition.
eMMC4.3 or eMMC4.4 device supporting special boot mode	If using an eMMC4.3 or eMMC4.4 device that supports the special boot mode, it can be initiated by pulling the CMD line low. If the BOOT ACK is enabled, the eMMC4.3/eMMC4.4 device sends the BOOT ACK via the DATA lines and the ROM can read the BOOT ACK [S010E] to identify the eMMC4.3/eMMC4.4 device. The eMMC4.3/eMMC4.4 device with the "boot mode" feature can only be supported via the ESDHCV3-3 and with or without the BOOT ACK. If the BOOT

Table continues on the next page...

Table 10-30. MMC and eMMC boot details (continued)

	ACK is enabled, the ROM waits 50 ms to get the BOOT ACK and if the BOOT ACK is received by the ROM. If BOOT ACK is disabled ROM waits 1 second for data. If the BOOT ACK or data was received, the eMMC4.3/eMMC4.4 is booted in the "boot mode", otherwise the eMMC4.3/eMMC4.4 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by the BOOT_CFG1[0] (fast boot) fuse. The BOOT ACK is selected by the BOOT_CFG1[4].
eMMC4.4 device	If using the eMMC4.4 device, the Double Data Rate (DDR) mode can be used. This mode can be selected by the BOOT_CFG2[2:1] (bus width) fuse.

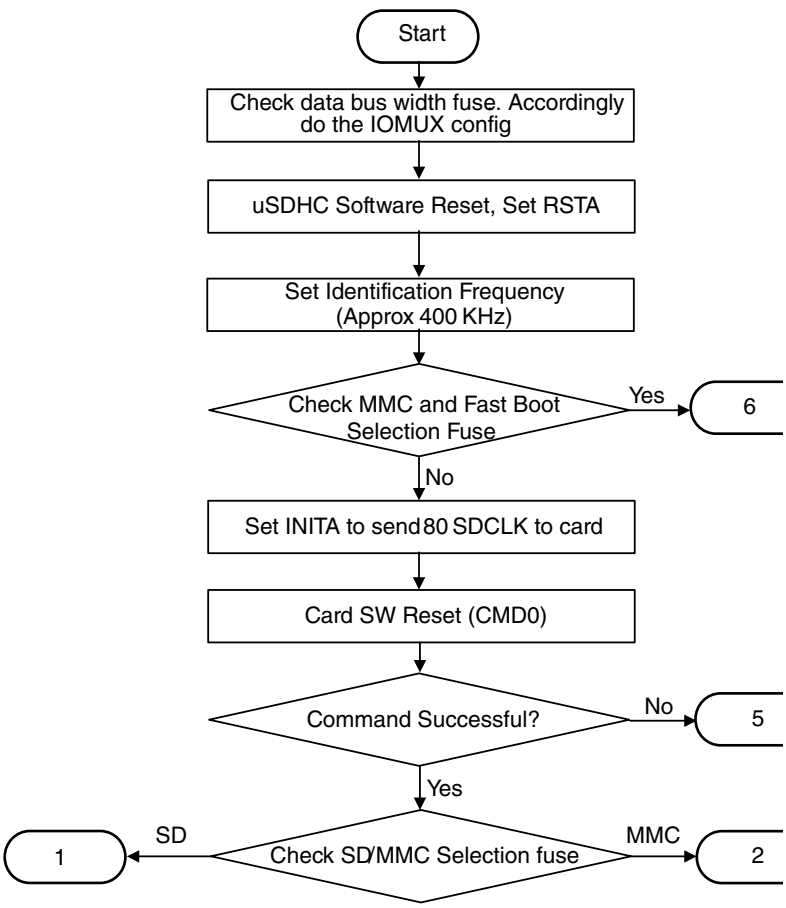


Figure 10-6. Expansion device boot flow (1 of 6)

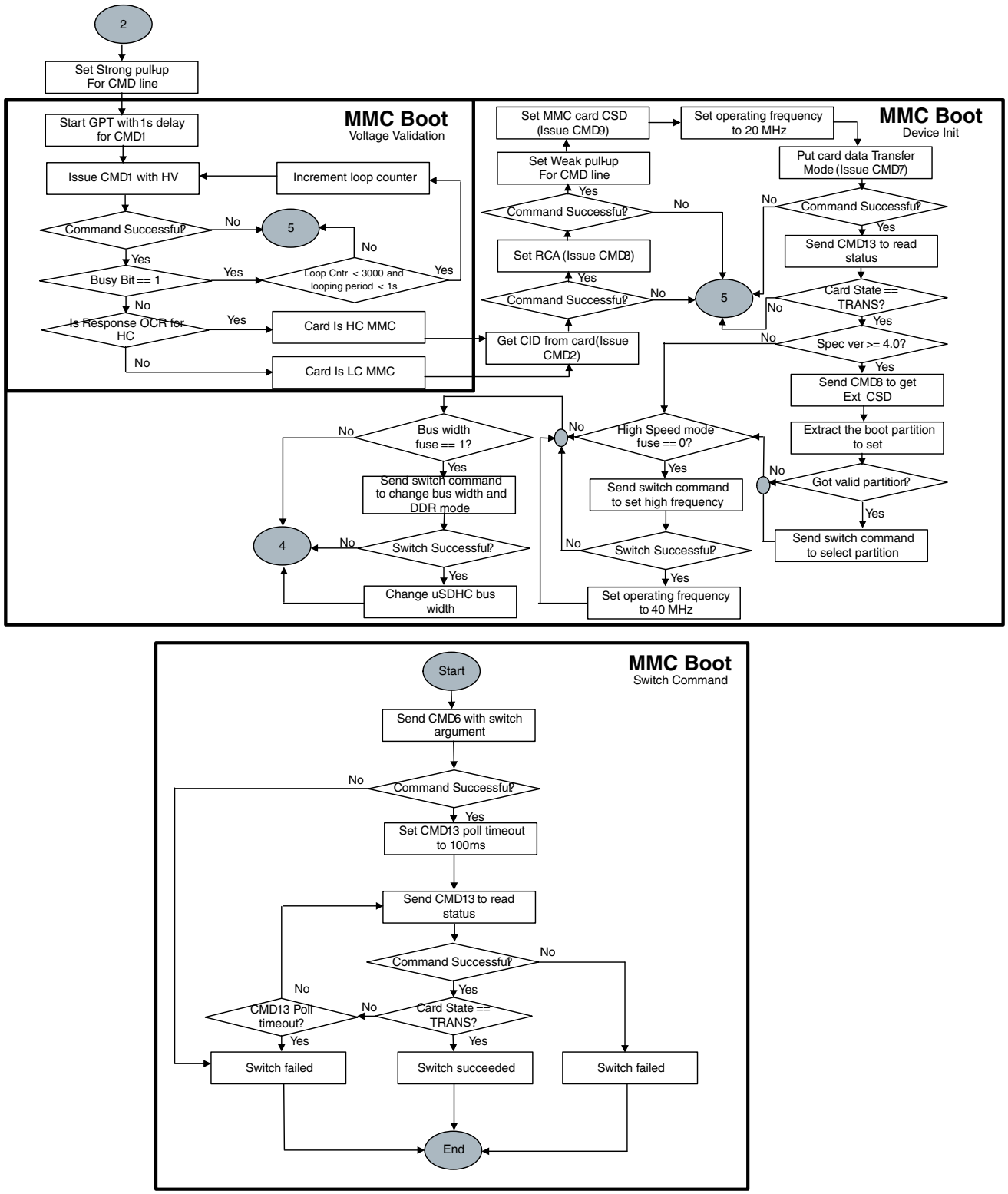


Figure 10-7. Expansion device (MMC) boot flow (2 of 6)

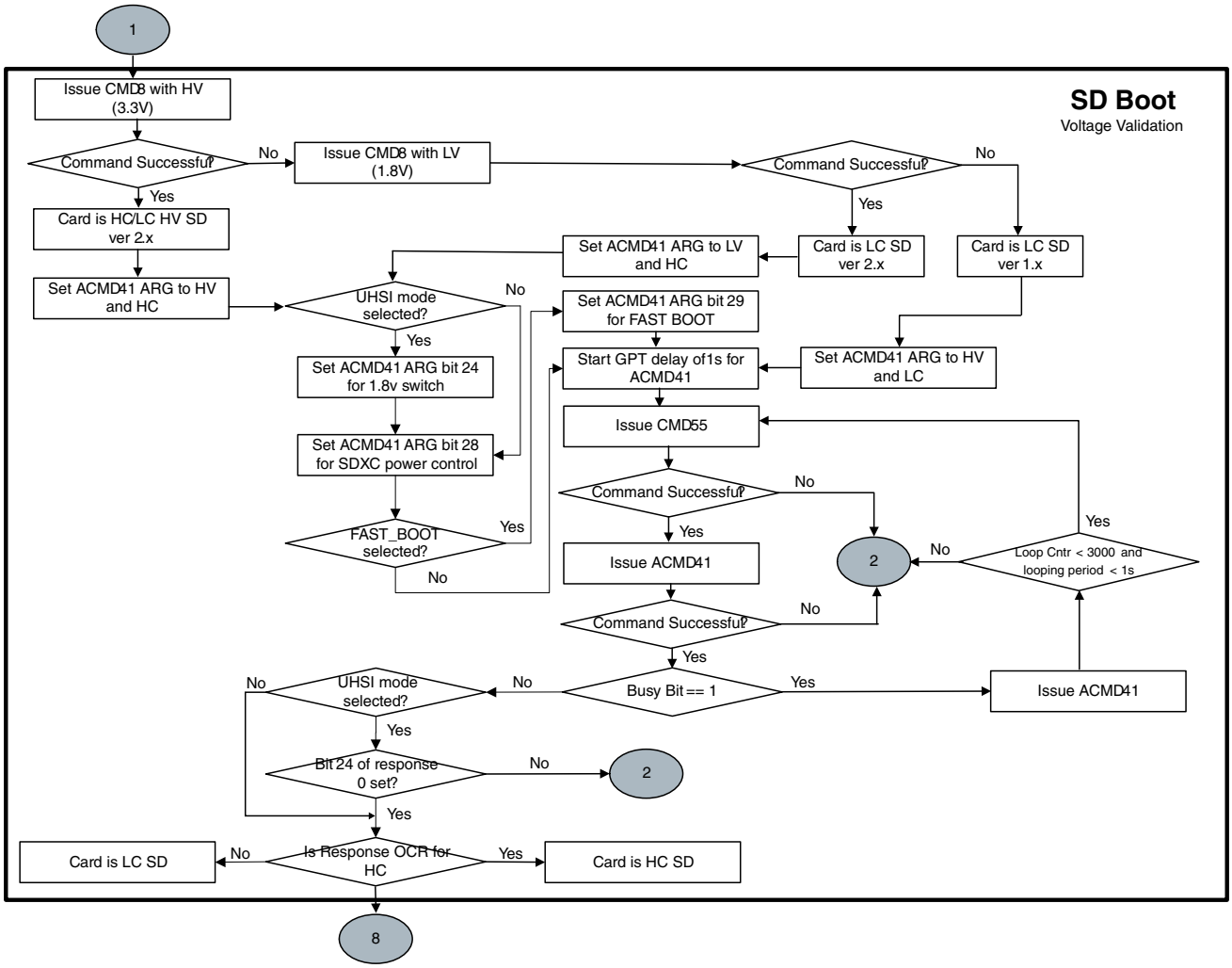


Figure 10-8. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 1

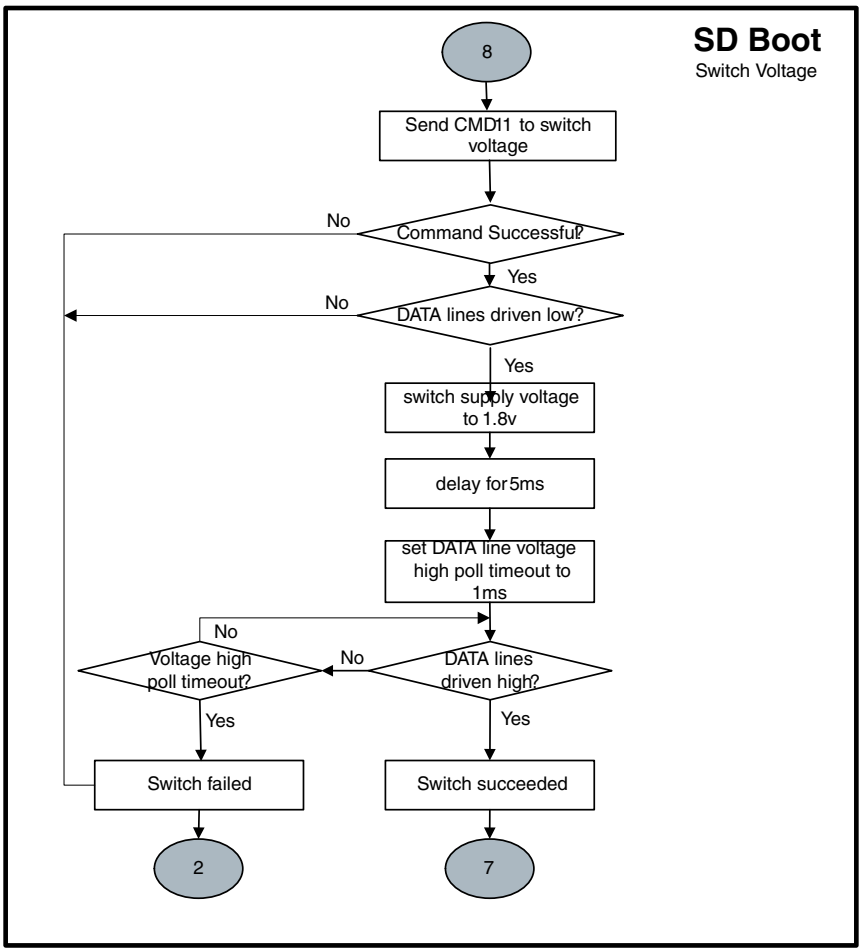


Figure 10-9. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 2

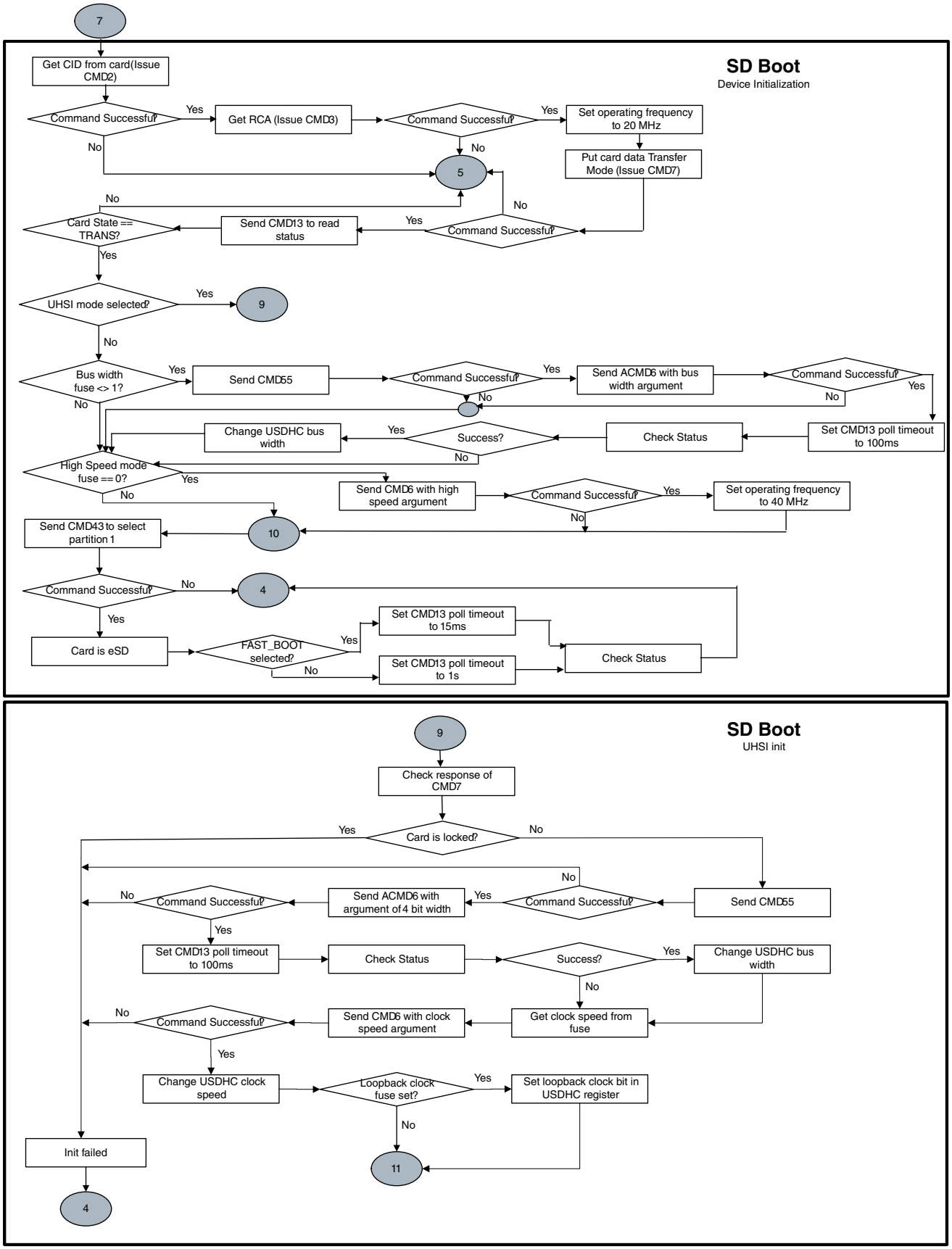


Figure 10-10. Expansion device (MMCSD/eSD/SDXC) boot flow (4 of 6)

i.MX RT1170 Processor Reference Manual, Rev. 4, 05/2025

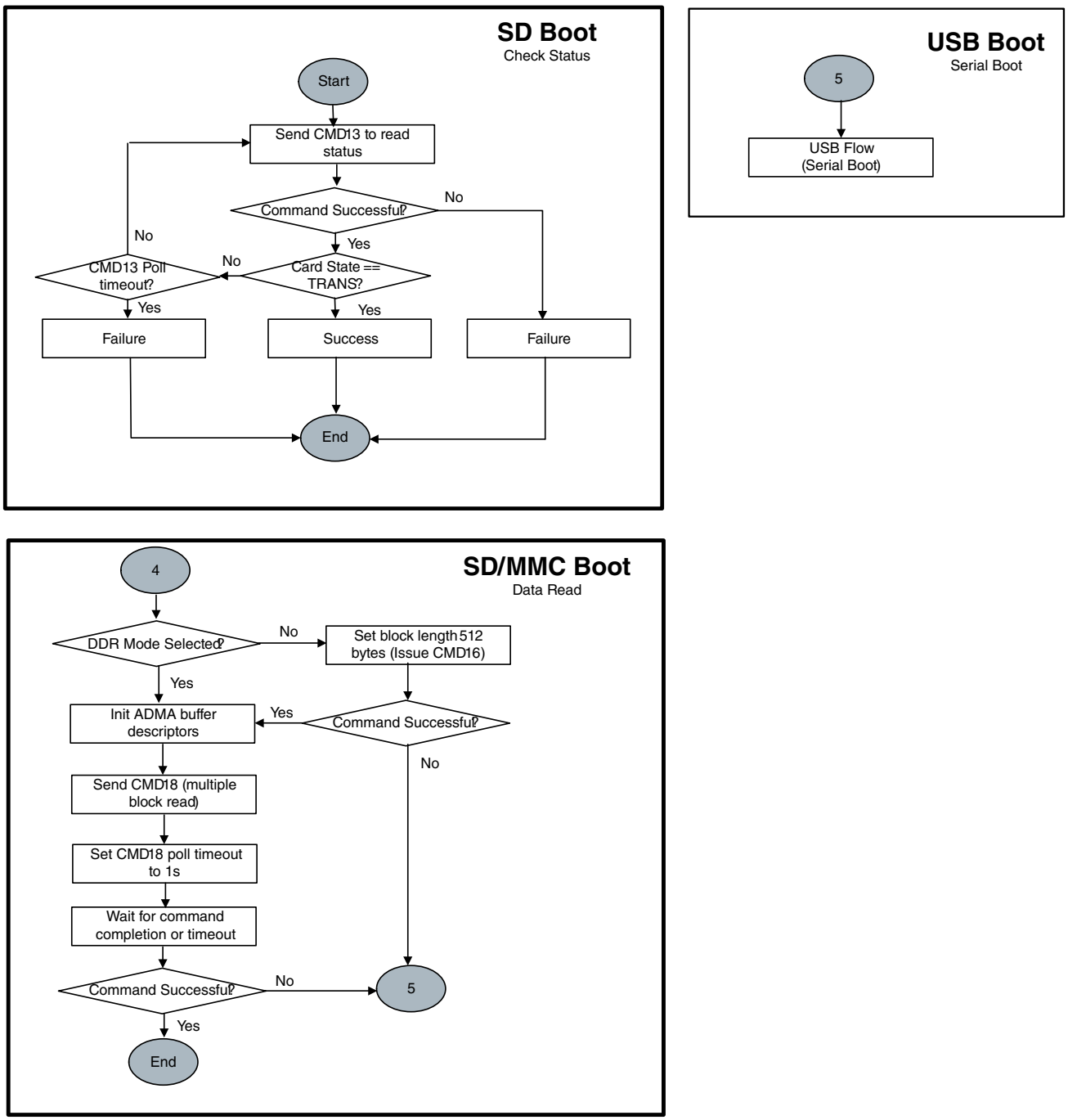
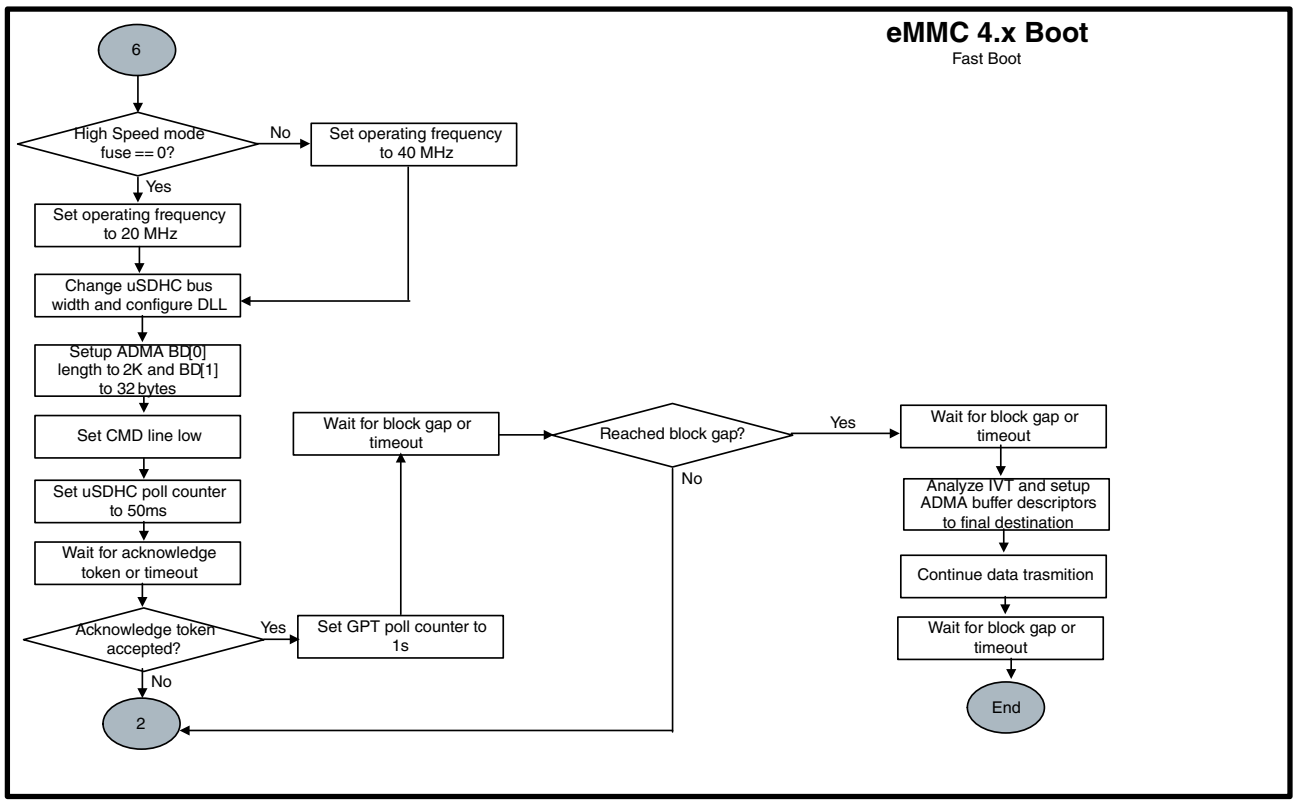


Figure 10-11. Expansion device (SD/eSD) boot flow (5 of 6)



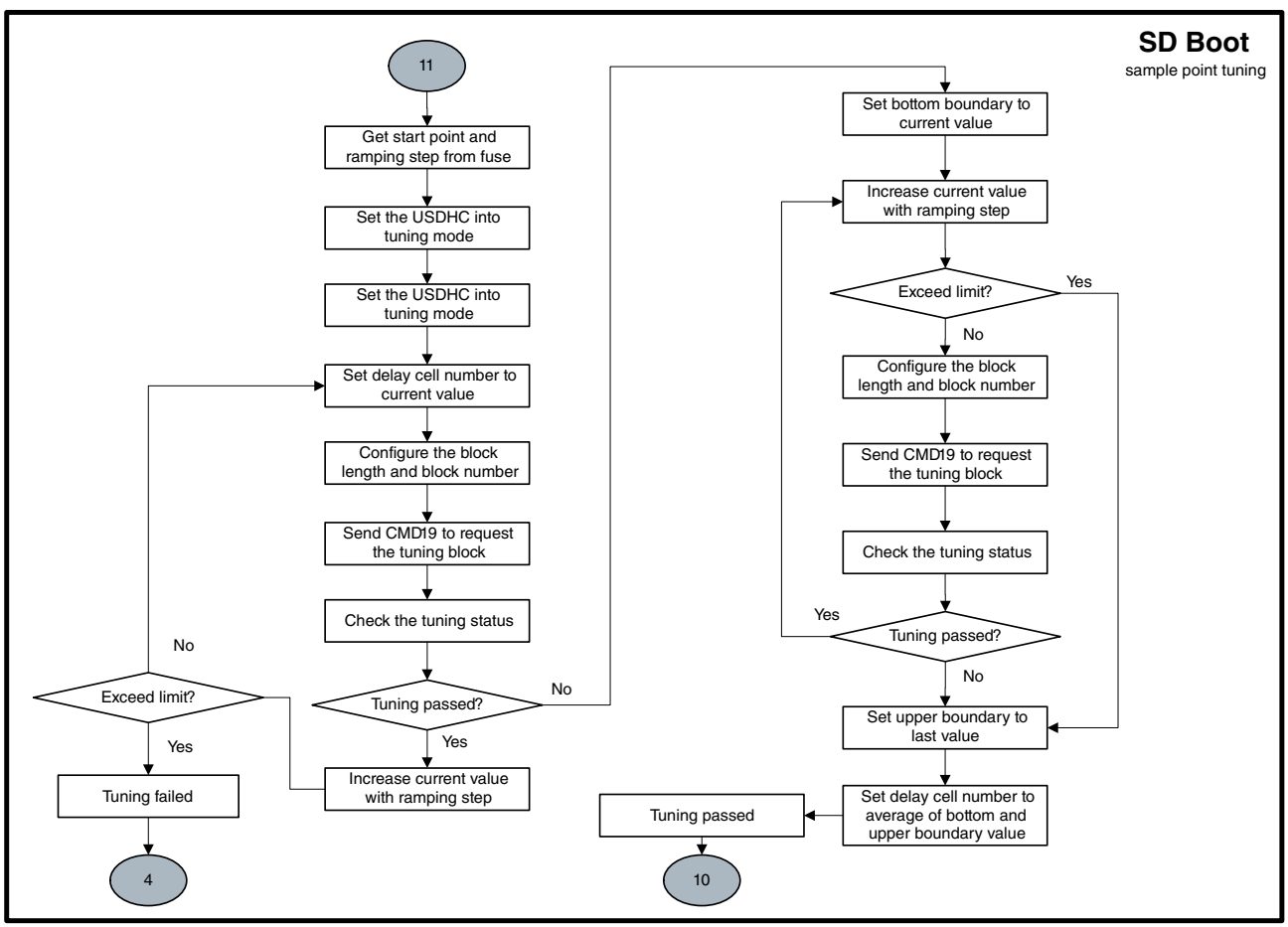


Figure 10-12. Expansion device boot flow (6 of 6)

10.6.6.3 SD, eSD, and SDXC

After the normal boot mode initialization begins, the SD/eSD/SDXC frequency is set to 347.22 kHz. During the identification phase, the SD/eSD/SDXC card voltage validation is performed. During the voltage validation, the boot code first checks with the high-voltage settings; if that fails, it checks with the low-voltage settings.

The capacity of the card is also checked. The boot code supports the high-capacity and low-capacity SD/eSD/SDXC cards after the voltage validation card initialization is done.

During the card initialization, the ROM boot code attempts to set the boot partition for all SD, eSD, and SDXC devices. If this fails, the boot code assumes that the card is a normal SD or SDXC card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, the boot code switches to a higher frequency (25 MHz in the normal-speed mode or 50 MHz in the high-speed mode). The ROM also supports the FAST_BOOT mode booting from the eSD card. This mode can be selected by the BOOT_CFG1[0] (Fast Boot).

For the UHSI cards, the clock speed fuses can be set to SDR50 or SDR104 on USDHC1, USDHC2 ports. This enables the voltage switch process to set the signaling voltage to 1.8 V during the voltage validation. The bus width is fixed at a 4-bit width and a sampling point tuning process is needed to calibrate the number of the delay cells. If the SD Loopback Clock eFuse is set, the feedback clock comes directly from the loopback SD clock, instead of the card clock (by default). The SD clock speed can be selected by the BOOT_CFG1[5:4], and the SD Loopback Clock is selected by the BOOT_CFG1[2].

The UHSI calibration start value (MMC_DLL_DLY[6:0]) and the step value SD_CALIBRATION_STEP[1:0] can be set to optimize the sample point tuning process.

If the SD Power Cycle Enable eFuse is 1, the ROM sets the SD_RST pad low, waits for 5 ms, and then sets the SD_RST pad high. If the SD_RST pad is connected to the SD power supply enable logic on board, it enables the power cycle of the SD card. This may be crucial in case the SD logic is in the 1.8 V states and must be reset to the 3.3 V states.

The SDR50 and SDR104 boots are not supported on the USDHC1 and USDHC2 ports because there are no reset signals for those ports when connected in the IOMUX.

10.6.6.4 Redundant boot support for expansion device

The ROM supports the redundant boot for an expansion device. The primary or secondary image is selected, depending on the PERSIST_SECONDARY_BOOT setting. (see Table 10-8).

If the PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x0 for the primary image.

If the PERSIST_SECONDARY_BOOT is 1, the boot ROM reads the secondary image table from address 0x200 on the boot media and uses the address specified in the table.

Table 10-31. Secondary image table format

Reserved (chipNum)
Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- The tag is used as an indication of the valid secondary image table. It must be 0x00112233.
- The firstSectorNumber is the first 512-byte sector number of the secondary image.

For the secondary image support, the primary image must reserve the space for the secondary image table. See this figure for the typical structures layout on an expansion device.

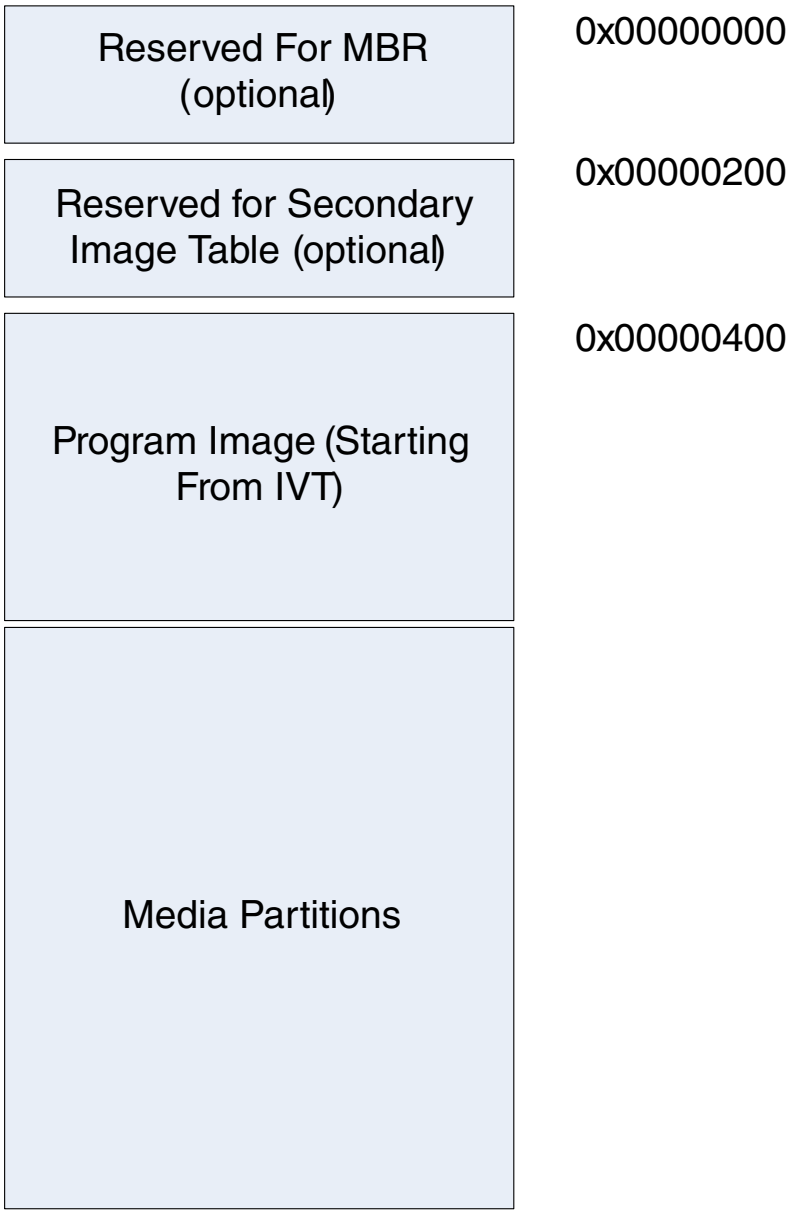


Figure 10-13. Expansion device structures layout

For the Closed mode, if there are failures during primary image authentication, the boot ROM turns on the PERSIST_SECONDARY_BOOT bit (see Table 10-8) and performs the software reset. (After the software reset, the secondary image is used.)

10.6.7 Serial NOR/EEPROM through LPSPI

The chip supports booting from serial memory devices, such as EEPROM and serial flash, using the LPSPI.

These ports are available for serial boot: LPSPI interfaces.

10.6.7.1 Serial NOR/EEPROM eFUSE configuration

The boot ROM code determines the type of device using the following parameters, provided by the eFUSE settings during boot.

Table 10-32. Serial NOR/EEPROM boot eFUSE descriptions

Fuse	Config	Definition	GPIO	Shipped value	Settings
RECOVERY_BOOT_EN (0xC70[24])	OEM	EEPROM recovery enable	No	0	0 - Disabled 1 - Enabled
SPI INSTANCE (0xC70[26:25])	OEM	Port select	No	00	00 - LPSPI1 01 - LPSPI2 10 - LPSPI3 (if applicable in the device) 11 - LPSPI4 (if applicable in the device)
SPI_MEM_SPEED (0xC70[28:27])	OEM	LPSPI Speed select	No	00	00 - 20 MHz 01 - 10 MHz 10 - 5 MHz 11 - 2 MHz

The LPSPI*n* block can be used as a boot device using the LPSPI interface for the serial ROM boot. The SPI interface is configured to operate at speed specified by SPI_MEM_SPEED fuse field.

The boot ROM copies 4 KB of data from the serial ROM device to the internal RAM. After checking the Image Vector Table header value (0xD1) from the program image, the ROM code performs a DCD check. After a successful DCD extraction, the ROM code extracts the destination pointer and length of image from the Boot Data Structure to be copied to the RAM device from where the code execution occurs.

NOTE

The Initial 4 KB of program image must contain the IVT, DCD, and the Boot Data Structures.

10.7 Program image

This section describes the data structures that are required to be included in the user's program image. The program image consists of:

- Image vector table—a list of pointers located at a fixed address that the ROM examines to determine where the other components of the program image are located.
- Boot data—a table that indicates the program image location, program image size in bytes, and the plugin flag.
- Device configuration data—IC configuration data.
- User code and data.

10.7.1 Image Vector Table and Boot Data

The Image Vector Table (IVT) is the data structure that the ROM reads from the boot device supplying the program image containing the required data components to perform a successful boot.

The IVT includes the program image entry point, a pointer to Device Configuration Data (DCD) and other pointers used by the ROM during the boot process. The ROM locates the IVT at a fixed address that is determined by the boot device connected to the Chip. The IVT offset from the base address for each boot device type is defined in the table below. The location of the IVT is the only fixed requirement by the ROM. The remainder or the image memory map is flexible and is determined by the contents of the IVT.

Table 10-33. Image Vector Table Offset and Initial Load Region Size

Boot Device Type	Image Vector Table Offset
FlexSPI NOR	4 Kbyte = 0x1000 bytes
SD/MMC/eSD/eMMC/SDXC	1 Kbyte = 0x400 bytes
SPI NOR/EEPROM SEMC NAND FlexSPI NAND	1 Kbyte = 0x400 bytes
Serial boot	0Kbyte = 0 byte

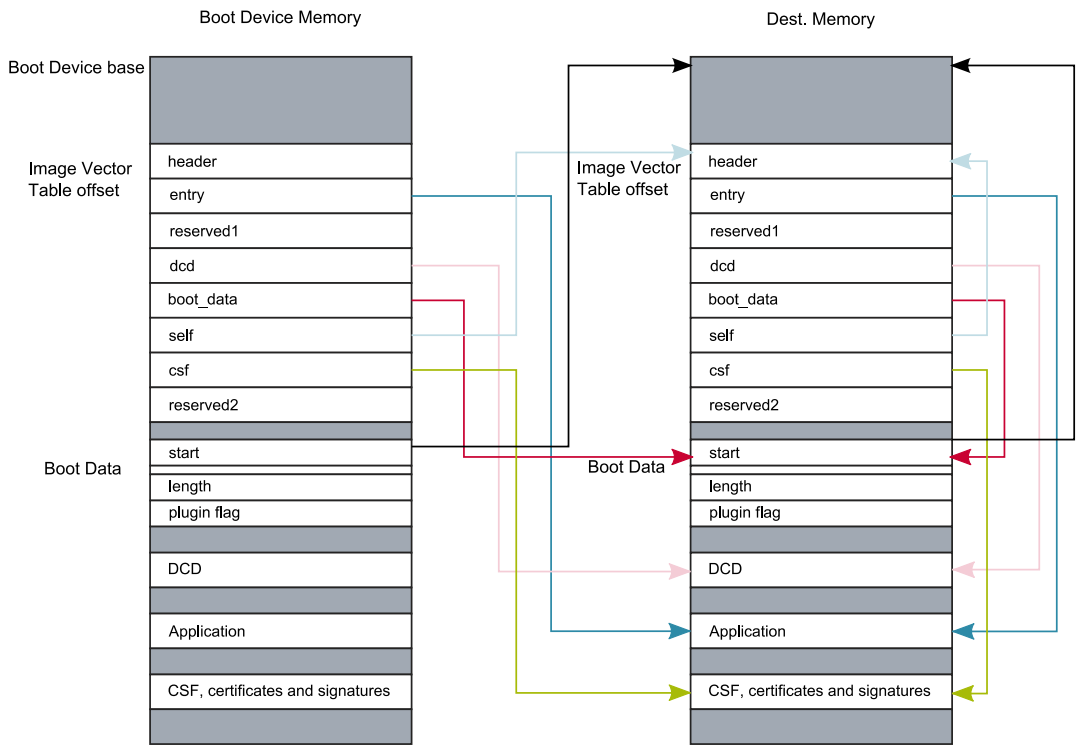


Figure 10-14. Image Vector Table

10.7.1.1 Image vector table structure

The IVT has the following format where each entry is a 32-bit word:

Table 10-34. IVT format

header
entry: Absolute address of the first instruction to execute from the image
reserved1: Reserved and should be zero
dcd: Absolute address of the image DCD. The DCD is optional so this field may be set to NULL if no DCD is required. See Device Configuration Data (DCD) for further details on the DCD.
boot data: Absolute address of the boot data
self: Absolute address of the IVT. Used internally by the ROM
csf: Absolute address of the Command Sequence File (CSF) used by the HAB library. See High-Assurance Boot (HAB) for details on the secure boot using HAB. This field must be set to NULL if a CSF is not provided in the image
reserved2: Reserved and should be zero

Figure 10-15 shows the IVT header format:

Figure 10-15. IVT header format

where:

```
Tag: A single byte field set to 0xD1
Length: a two byte field in big endian format containing the overall length of the IVT,
in bytes, including the header. (the length is fixed and must have a value of 32 bytes)
Version: A single byte field set to 0x40/0x41/0x42/0x43/0x44/0x45
```

10.7.1.2 Boot data structure

The boot data must follow the format defined in the table found here, each entry is a 32-bit word.

Table 10-35. Boot data format

start	Absolute address of the image
length	Size of the program image
plugin	Plugin flag (must be 0, not supported in this device, this field is ignored by the ROM)

10.7.2 Device Configuration Data (DCD)

Upon reset, the chip uses the default register values for all peripherals in the system. However, these settings typically are not ideal for achieving the optimal system performance and there are even some peripherals that must be configured before they can be used.

The DCD is a configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various peripherals on the chip.

For example, some components (such as SDRAM) require some sequence of register programming as a part of the configuration before it is ready to be used. The DCD feature can be used to program the SEMC register to the optimal settings.

The ROM determines the location of the DCD table based on the information located in the Image Vector Table (IVT). See Image Vector Table and Boot Data for more details. The DCD table shown below is a big-endian byte array of the allowable DCD commands. The maximum size of the DCD is limited to 1768 B.

Figure 10-16. DCD data format

The DCD header is 4 B with the following format:

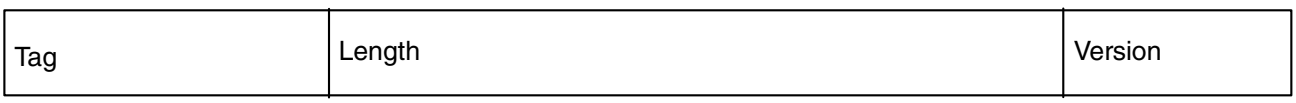


Figure 10-17. DCD header format

where:

```
Tag: A single-byte field set to 0xD2
Length: a two-byte field in the big-endian format containing the overall length of the DCD
(in bytes) including the header
Version: A single-byte field set to 0x41
```

10.7.2.1 Write data command

The write data command is used to write a list of given 1-, 2- or 4-byte values (or bitmasks) to a corresponding list of target addresses.

The Value/Mask fields are always 32-bits. If the parameter field specifies a smaller size, then the extra bytes must be zero.

The format of the write data command (in a big-endian byte array) is shown in this table:

Table 10-36. Write data command format

Tag	Length	Parameter
Address		
Value/Mask		
[Address]		
[Value/Mask]		

Table continues on the next page...

Table 10-36. Write data command format (continued)

...
[Address]
[Value/Mask]

where:

```
Tag: a single-byte field set to 0xCC
Length: a two-byte field in a big-endian format, containing the length of the Write Data
Command (in bytes) including the header
Address: the target address to which the data must be written
Value/Mask: the data value (or bitmask) to be written to the preceding address
```

The parameter field is a single byte divided into bitfields as follows:

Table 10-37. Write data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of the target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only specific bits may be overwritten at the target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

One or more target address and value/bitmask pairs can be specified. The same bytes' and flags' parameters apply to all locations in the command.

When successful, this command writes to each target address in accordance with the flags as follows:

Table 10-38. Interpretation of write data command flags

"Mask"	"Set"	Action	Interpretation
0	0	*address = val_msk	Write value
0	1	*address = val_msk	Write value
1	0	*address &= ~val_msk	Clear bitmask
1	1	*address |= val_msk	Set bitmask

NOTE

If any of the target addresses does not have the same alignment as the data width indicated in the parameter field, none of the values are written.

If any of the values are larger or any of the bitmasks are wider than permitted by the data width indicated in the parameter field, none of the values are written.

If any of the target addresses do not lie within the allowed region, none of the values are written. The list of allowable blocks and target addresses for the chip are provided below.

Table 10-39. Valid DCD address ranges

Address range	Start address	Last address
IOMUX Control (IOMUXC) registers	0x400E_8000	0x400E_8717
ANADIG PLL registers	0x40C8_4000	0x40C8_4393
CCM registers	0x40CC_0000	0x40CC_751F
SEMC registers	0x400D_4000	0x400D_412F
GPT1 registers	0x400E_C000	0x400E_C027
FlexSPI1 registers	0x400C_C000	0x400C_C42B
FlexSPI2 registers	0x400D_0000	0x400D_042B

10.7.2.2 Check data command

The check data command is used to test for a given 1-, 2-, or 4-byte bitmasks from a source address.

The check data command is a big-endian byte array with the format shown in this table:

Table 10-40. Check data command format

Tag	Length	Parameter
Address		
Mask		
[Count]		

where:

```
Tag: a single-byte field set to 0xCF
Length: a two-byte field in the big-endian format containing the length of the check data
command (in bytes) including the header
Address: the source address to test
Mask: the bit mask to test
Count: an optional poll count; If the count is not specified, this command polls
indefinitely
until the exit condition is met. If count = 0, this command behaves as for the NOP.
```

The parameter field is a single byte divided into bitfields, as follows:

Table 10-41. Check data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only the specific bits may be overwritten at a target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

This command polls the source address until either the exit condition is satisfied, or the poll count is reached. The exit condition is determined by the flags as follows:

Table 10-42. Interpretation of check data command flags

"Mask"	"Set"	Action	Interpretation
0	0	(*address & mask) == 0	All bits clear
0	1	(*address & mask) == mask	All bits set
1	0	(*address & mask)!= mask	Any bit clear
1	1	(*address & mask)!= 0	Any bit set

NOTE

If the source address does not have the same alignment as the data width indicated in the parameter field, the value is not read.

If the bitmask is wider than permitted by the data width indicated in the parameter field, the value is not read.

10.7.2.3 NOP command

This command has no effect.

The format of the NOP command is a big-endian four-byte array, as shown in this table:

Table 10-43. NOP command format

Tag	Length	Undefined

where:

```
Tag: a single-byte field set to 0xC0
Length: a two-byte field in big endian containing the length of the NOP command in bytes
(fixed to a
```

```
value of 4)
Undefined: this byte is ignored and can be set to any value.
```

10.8 External Memory Configuration Data (XMCD)

For comprehensive or feature-rich applications, the on-chip RAM may not be enough for the application use. Usually, such an application requires a large capacity of RAM and needs to run on the external RAM for high execution performance. However, enabling the external RAM on the SoC is relatively complicated, and the settings are SoC specific. The Boot ROM on the SoC simplifies the external RAM configuration by introducing the XMCD. The XMCD data structure resides at offset 0x40 starting from IVT header.

The XMCD consists of a XMCD header and a memory specific configuration block. On this SoC, the following types of external RAM devices are supported. Each type of RAM device has a specific configuration block.

- HyperRAM/APMemory via FlexSPI – FlexSPI RAM configuration block
- SDRAM via SEMC – SEMC SDRAM Configuration block

10.8.1 XMCD header

The XMCD header resides at the fixed offset 0x40 from the IVT header. The header definition is as follows.

Table 10-44. XMC Header Definition

[31:28] Tag	[27:24] Version	[23:20] Memory Interface	[19:16] Instance	[15:12] Configuration Block Type	[11:0] Configuration block size
Valid value = 0xC	Fixed value of '0'	0 - FlexSPI 1 - SEMC	SoC defined instances	0 - Simplified Configuration Option Block 1 - Full Configuration Block	Configuration block size (including the XMC Header itself).

NOTE

There is an optional integrity check of the XMC by programming the CRC checksum to the fuse. The ROM can perform the integrity check before adopting the XMC block settings. It calculates the CRC check based on the value of the "Configuration block size" field. ROM determines whether to enable the XMC integrity check by checking the

XMC_CHECK_EN(0x970[14]) fuse field and the fuse word 0xC90. Details of the CRC algorithm used for the integrity check are provided in the following table.

Table 10-45. CRC Algorithm Details

Description	Value
Width	32
Polynomial	0x04C11BD7
Init Value	0xFFFFFFFF
Reflect In	FALSE
Reflect Out	FALSE
XOR Out	0x00000000

The following procedure shows the steps in the CRC calculation:

1. CRC initialization
   - Set the initial CRC as 0xFFFFFFFF, which clears the CRC byte count to 0.
2. CRC calculation
   - Start the calculation from the first byte of the XMC header. Total calculation bytes are "Configuration block size" bytes.
3. CRC finalization
   - Check if the CRC byte count is 4-bytes aligned. If it is not 4-bytes aligned, then pad it with the necessary zeroes to finalize the CRC. Otherwise, return to the current computed CRC.

10.8.2 FlexSPI RAM configuration block

The XMCD offers a simplified FlexSPI RAM configuration option block which can meet the typical usage of the HyperRAM or APMemory devices.

However, the user application may need to enable the advanced features of the external RAM which cannot be configured by the simplified configuration option. In this case, the XMCD also offers the complete 512-bytes FlexSPI RAM configuration block which supports flexible configuration.

10.8.2.1 Simplified FlexSPI RAM configuration option

The simplified FlexSPI RAM configuration option structure is shown in the following tables.

Table 10-46. Simplified FlexSPI RAM Configuration Option 0

[31:28] Tag	[27:24] Option Size	[23:20] device type	[19:16] Reserved	[15:12] Misc.	[11:8] Maximum Frequency	[7:0] Size in MB
Fixed to 0x0C	0 - Option words = 1 1 - Option words = 2 Others-Rereved	0 - HyperRAM 1 - APMemory	Reserved	For HyperRAM 0 - 1.8V 1 - 3V	1 - 30MHz 2 - 50MHz 3 - 60MHz 4 - 80MHz 5 - 100MHz 6 - 120MHz 7 - 133MHz 8 - 166MHz Others – Reserved	0 - Auto detection Others - Size in MB

Table 10-47. Simplified FlexSPI RAM Configuration Option 1

[31:28] RAM_CONNECTION	[27:24] Reserved	[23:20] DQS_PINMUX_GROUP	[19:16] PINMUX ROUP	[15:8] Reserved	[7:4] write dummy cycles	[3:0] read dummy cycles
0 - PORTA 1 - PORTB	Reserved	0 - Default Group 1 - Secondary group	0 - Primary group 1 - Secondary group	Reserved	0 - Auto detection Others - Specified dummy cycles	0 - Auto detection Others - Specified dummy cycles

NOTE

The [23:20] field must be set to 0 because the secondary group of DQS can't be used.

10.8.2.2 Full FlexSPI RAM configuration block

The following table lists the data structure of the full 512-bytes FlexSPI RAM Configuration Block.

Table 10-48. Full HyperRAM/APMemory Configuration Block Structure

Name	Offset	Size(Bytes)	Description
Tag	0x000	4	0x42464346, ascii: "FCFB"
Version	0x004	4	[07:00] bugfix [15:08] minor [23:16] major = 1 [31:24] ascii 'V'
Reserved	0x008	4	Reserved
readSampleClkSrc	0x00c	1	0 – Internal loopback 1 – loopback from DQS pad 2 - Reserved 3 – Flash provided DQS
csHoldTime	0x00d	1	Serial Flash CS Hold Time Recommend default value is 0x03
csSetupTime	0x00e	1	Serial Flash CS setup time Recommended default value is 0x03
columnAdressWidth	0x00f	1	3 – For HyperFlash/HyperRAM 12/13 – For Serial NAND. See datasheet to find correct value. 0 – Other devices
deviceModeCfgEnable	0x010	1	Device Mode Configuration Enable feature 0 – Disabled 1 – Enabled
deviceModeType	0x11	1	Device Mode Type 0 – Generic 1 – Quad Enable 2 – SPI-to-xSPI Mode 3 – xSPI-to-SPI mode
waitTimeCfgCommands	0x12	2	Time in terms of 100us This field is defined for the mode switch from SPI to xSPI mode or vice versa.
deviceModeSeq	0x014	4	Sequence parameter for device mode configuration [7:0] - Number of sequences [15:8] - Sequence Index [31:16] - Reserved, fixed to 0
deviceModeArg	0x018	4	Device Mode argument, effective only when deviceModeCfgEnable = 1
configCmdEnable	0x01c	1	Config Command Enable feature

Table continues on the next page...

Table 10-48. Full HyperRAM/APMemory Configuration Block Structure (continued)

Name	Offset	Size(Bytes)	Description
			0 – Disabled 1 – Enabled
configModeType	0x01d	3	This field has the same definititions as "deviceModeType" byte 0 - configModeType for configCmdSeq[0] byte 1 - configModeType for configCmdSeq[1] byte 2 - configModeType for configCmdSeq[2]
configCmdSeqs	0x020	12	Sequences for Config Command allows 3 separate configuration command sequences. For each configCmdSeq, the definition of the word is: [7:0] - Number of sequences [15:8] - Sequence Index [31:16] - Reserved, fixed to 0
Reserved	0x02c	4	Reserved
cfgCmdArgs	0x030	12	Arguments for each separate configuration command sequence.
Reserved	0x03c	4	Reserved
controllerMiscOption	0x040	4	Bit0 – differential clock enable Bit1 – CK2 enable Bit2 – ParallelModeEnable Bit3 – wordAddressableEnable Bit4 – Half-Speed access enable Bit5 – Pad Setting Override Enable Bit6 – DDR Mode Enable Bit7 - Pad Setting Override Enable Bit 8 - Second Pinmux group Bit 9 - Second DQS pin mux group Bit 10 - Write Mask Enable Bit 11 - Write Opt1 Clear
deviceType	0x044	1	1 – Serial NOR 2 – Serial NAND 3 - Serial RAM
sflashPadType	0x045	1	1 – Single pad 2 – Dual pads 4 – Quad pads 8 – Octal pads
serialClkFreq	0x046	1	Chip specific value
lutCustomSeqEnable	0x047	1	0 – Use pre-defined LUT sequence index and number 1 - Use LUT sequence parameters provided in this block

Table continues on the next page...

Table 10-48. Full HyperRAM/APMemory Configuration Block Structure (continued)

Name	Offset	Size(Bytes)	Description
Reserved	0x048	8	Reserved
sflashA1Size	0x050	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
sflashA2Size	0x054	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
sflashB1Size	0x058	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
sflashB2Size	0x05c	4	For SPI NOR, need to fill with actual size For SPI NAND, need to fill with actual size * 2
csPadSettingOverride	0x060	4	Set to 0 if it is not supported
sclkPadSettingOverride	0x064	4	Set to 0 if it is not supported
dataPadSettingOverride	0x068	4	Set to 0 if it is not supported
dqsPadSettingOverride	0x06c	4	Set to 0 if it is not supported
timeoutInMs	0x070	4	Maximum wait time during read/write
commandInterval	0x074	4	Unit: ns. Currently, it is for SPI NAND at the high working frequency. For the serial NAND and serial RAM device, this field is 0.
dataValidTime	0x078	4	Time from clock edge to data valid edge. Unit: ns. This field takes effect when the FlexSPI Root clock is less than 100MHz, and the read sample clock source is a device provided DQS signal without CK2 support. [31:16] - data valid time for DLLB in terms of 0.1ns [15:0] - data valid time for DLLA in terms of 0.1ns
busyOffset	0x07c	2	Busy bit offset. Valid range :0-31.
busyBitPolarity	0x07e	2	0 – 1 represents busy 1 – 0 represents busy
lookupTable	0x080	16*4*4	Lookup table: The sequence index of read must be set to 0. The sequence index of write must be set to 4.
lutCustomSeq	0x180	4*12	Customized LUT sequence. See the note at the end of the table for details.
Reserved	0x1b0	5*16	Reserved

NOTE

- LUT for read must be placed at LUT entry 0.
- LUT for write must be placed at LUT entry 9.

10.8.2.3 Example of XMCD for HyperRAM support

The following table shows an example of XMCD for HyperRAM support.

Table 10-49. Example XMCD block for HyperRAM

Offset	Field	Description
0	0xC001_0008	Tag = 0x0C Version = 0 Memory Interface: FLEXSPI Instance: 1 - First Instance Configuration block type: Simplified Configuration block size: 8 (4-byte header + 4-byte option block)
1	0xC000_0700	Tag = 0x0C Option_Size = 0 DeviceType: HyperRAM Reserved: 0 Misc: HyperRAM 1V8 Max Freq: 7 Memory Size: Auto-detection

10.8.3 SDRAM support

The XMCD also supports the SDRAM configuration via the SEMC SDRAM Configuration Block, which offers both the simplified configuration option and the fully customizable configuration block.

10.8.3.1 SEMC SDRAM configuration block structure

The SEMC SDRAM configuration block structure is shown in the following table.

Table 10-50. SDRAM Configuration Block structure

Offset	Width (bytes)	Field	Description
0	1	magic_number	Must be 0xA1
1	1	version	Set to 1 for this implementation
2	1	config_option	0x00 - Simplified configuration. Select SDRAM CS0 as default and can only configure clk_MHz, sdram0_size_kB, and port_size. 0xFF - Full configuration. Must configure all fields.
3	1	clk_MHz	Set the working frequency in the unit of MHz

Table continues on the next page...

Table 10-50. SDRAM Configuration Block structure (continued)

Offset	Width (bytes)	Field	Description
4	4	sdram0_size_kB	Set the memory size of SDRAM CS0 in the unit of kilobytes. Range: 4~4*1024*1024
8	1	port_size	Port size of SDRAM 0 - 8bit 1 - 16bit 2 - 32bit Others - Invalid value
9	1	pin_config_pull	Pull config of the SDRAM GPIO pin 0 – forbidden 1 – pull up 2 – pulldown 3 – no pull Others - Invalid value
10	1	pin_config_drive_strength	Driver config of SDRAM GPIO pin 0 – high driver 1 – normal driver Others - Invalid value
11	1	mux_rdy	SDRAM CSn device selection 1 - SDRAM CS1 2 - SDRAM CS2 3 - SDRAM CS3 Others – Invalid for SDRAM, select other external devices **NOTE:** To select CS1/CS2/CS3, you can only use one of the five items: mux_rdy, mux_csx0, mux_csx1,mux_csx2 ,mux_csx3. If using one item, the other items cannot be used, set them to 0.
12	1	mux_csx0	SDRAM CSn device selection 1 - SDRAM CS1 2 - SDRAM CS2 3 - SDRAM CS3 Others – Invalid for SDRAM, select other external devices
13	1	mux_csx1	SDRAM CSn device selection 1 - SDRAM CS1 2 - SDRAM CS2 3 - SDRAM CS3 Others – Invalid for SDRAM, select other external devices
14	1	mux_csx2	SDRAM CSn device selection 1 - SDRAM CS1

Table continues on the next page...

Table 10-50. SDRAM Configuration Block structure (continued)

Offset	Width (bytes)	Field	Description
			2 - SDRAM CS2 3 - SDRAM CS3 Others – Invalid for SDRAM, select other external devices
15	1	mux_csx3	SDRAM CSn device selection 1 - SDRAM CS1 2 - SDRAM CS2 3 - SDRAM CS3 Others – Invalid for SDRAM, select other external devices
16	1	bank	Bank numbers of SDRAM device 0 – 4 banks 1 – 2 banks Others - Invalid value
17	1	burst_len	Burst length 0 – 1 1 – 2 2 – 4 3 – 8 Others - Invalid value
18	1	column_addr_bit_num	Column address bit number 0 – 12bit 1 – 11bit 2 – 10bit 3 – 9bit 4 – 8bit Others - Invalid value
19	1	cas _ latency	CAS Latency 1 – 1 2 – 2 3 – 3 Others - Invalid value
20	1	write_recovery_ns	Write recovery time in unit of nanosecond. This could help to meet tWR timing requirement by the SDRAM device.
21	1	refresh_recovery_ns	Refresh recovery time in unit of nanosecond. This could help to meet tRFC timing requirement by the SDRAM device.
22	1	act2readwrite_ns	Act to read/write wait time in unit of nanosecond. This could help to meet tRCD timing requirement by the SDRAM device.

Table continues on the next page...

Table 10-50. SDRAM Configuration Block structure (continued)

Offset	Width (bytes)	Field	Description
23	1	precharge2act_ns	Precharge to active wait time in unit of nanosecond. This could help to meet tRP timing requirement by SDRAM device.
24	1	act2act_banks_ns	Active to active wait time between two different banks in unit of nanosecond. This could help to meet tRRD timing requirement by the SDRAM device.
25	1	refresh2refresh_ns	Auto refresh to auto refresh wait time in unit of nanosecond. This could help to meet tRFC timing requirement by the SDRAM device.
26	1	selfref_recovery_ns	Self refresh recovery time in unit of nanosecond. This could help to meet tXSR timing requirement by the SDRAM device.
27	1	act2prechage_min_ns	ACT to Precharge minimum time in unit of nanosecond This could help to meet tRAS(min) timing requirement by SDRAM device.
28	4	act2prechage_max_ns	ACT to Precharge maximum time in unit of nanosecond. This could help to meet tRAS(max) timing requirement by the SDRAM device.
32	4	refreshperiod_perrow_ns	Refresh timer period in unit of nanosecond/ Set to (tREF(ms) * 1000000/rows) value.
36	4	mode_register	Define the specific mode of operation of SDRAM. Set to the value required by SDRAM device. **NOTE:** The low bits of the input value must correspond to the low bits of the mode register.
40	4	sdram0_base	Base address of SDRAM CS0 Range: 0x80000000~0xDFFFFFFF **NOTE:** SDRAM CS0~CS3 addresses cannot overlap and when CSn is not being used, set the address to 0.
44	4	sdram1_base	Base address of SDRAM CS1 Range: 0x80000000~0xDFFFFFFF
48	4	sdram2_base	Base address of SDRAM CS2 Range: 0x80000000~0xDFFFFFFF
52	4	sdram3_base	Base address of SDRAM CS3 Range: 0x80000000~0xDFFFFFFF
56	4	sdram1_size_kB	Set the memory size of SDRAM CS1 in unit of kbytes Range: 4~4*1024*1024
60	4	sdram2_size_kB	Set the memory size of SDRAM CS2 in unit of kbytes Range: 4~4*1024*1024
64	4	sdram3_size_kB	Set the memory size of SDRAM CS3 in unit of kbytes Range: 4~4*1024*1024

10.8.3.2 Example for SDRAM support (32-bit)

The following table shows an example for the SDRAM support (32-bit) on this device.

Table 10-51. Example SDRAM Configuration Block

Offset	Field	Description
0	0xC010_000D	Tag = 0xC Version = 0 Memory Interface: SEMC Instance: 0 - ignored Configuration block type: 0 - Ignored (Handled inside the SDRAM configuration structure) Configuration block size: 13 (4-byte header + 9-byte option block)
1	0xA600_01A1	Magic_number = 0xA1 Version = 1 Config_option: Simplified SDRAM clock: 166MHz
2	0x0001_0000	SDRAM CS0 size: 64MBytes
3	0x02	Port_size: 32-bit

10.9 Serial Boot (Serial Downloader)

The Serial boot provides a means to download a boot image to the chip and execute the image over the following serial peripherals:

- LPUART
- USB-HID

The default enabled peripherals for active peripheral detection is determined by the combination of eFUSEs and Boot Mode pins.

The boot ROM can detect the active peripherals and shut down all inactive peripherals before any communications between the host and the device.

The high-level serial boot flow is provided below. Details of communication protocols, sequences, and flows can be found in the following sections.

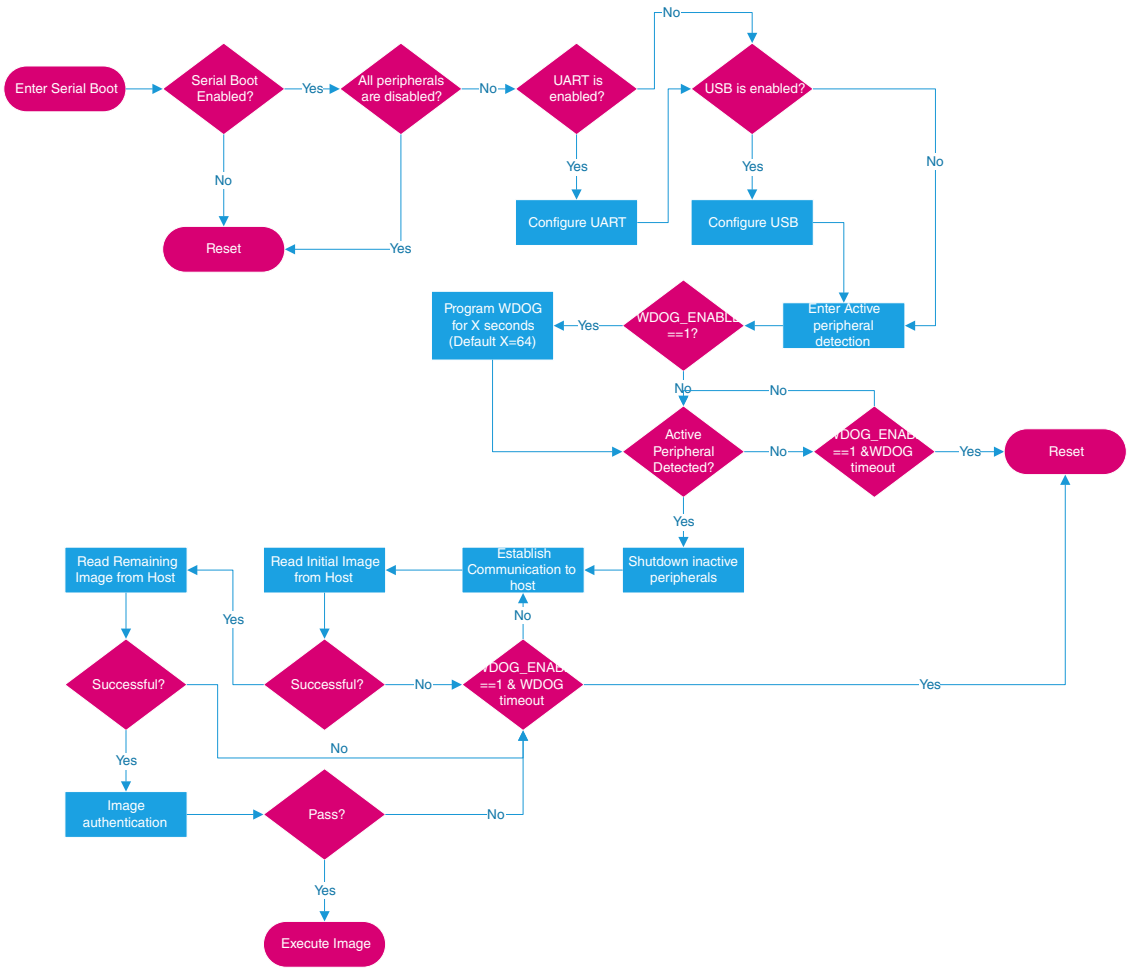


Figure 10-18. Serial Boot Flow

The communication protocol for serial boot mode is compatible with the widely-used MCUBOOT protocol in the NXP MCU products. Using simplified protocol, only two types of phase are supported:

1. **Command-only phase**. “get-property” command is supported, but the supported properties are limited to:
   - Bootloader version
   - Target version
   - Maximum payload size in the packet
   - Security State
   - Last Status

   For all other properties queried by the host, the boot ROM responds with unknown property error code.
2. **Data-only phase**. Boot ROM treats the data from the host as data streaming; no command phase is needed in this phase.

10.9.1 MCU-BOOT protocol

The section demonstrates the general protocol for the packet transfers between the host and the boot ROM. The description includes the transfer of packets for different transactions, such as command with no data phase and data only phase. The next section describes various packet types used in a transaction.

- Each command sent from the host is replied to with a response command.
- Each data sent from the host is replied with an ACK at the packet level.
- In all protocols (described in the following subsections), the Ack sent in response to a Command or Data packet can arrive at any time before, during, or after the Command/Data packet has processed.

Command-only phase

The protocol for command-only phase content:

- Command packet (from the host)
- Response command packet (to host)

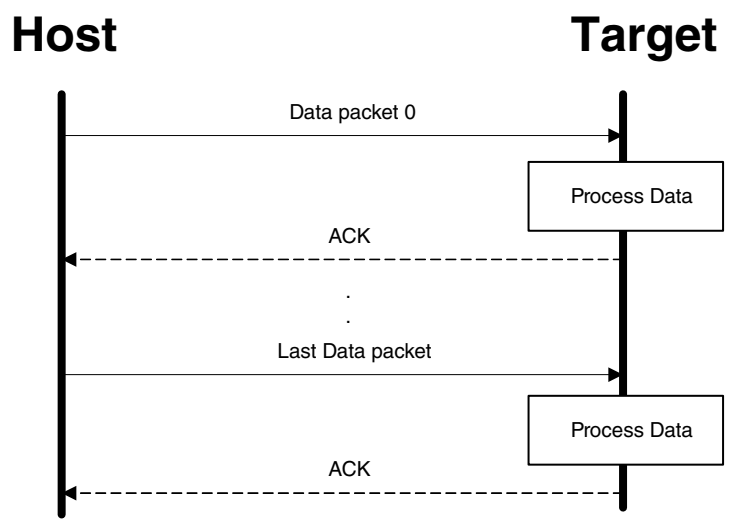


Figure 10-19. Protocol for a command-only phase

Data-only phase

The protocol for data-only phase content:

- A data packet (from the host)

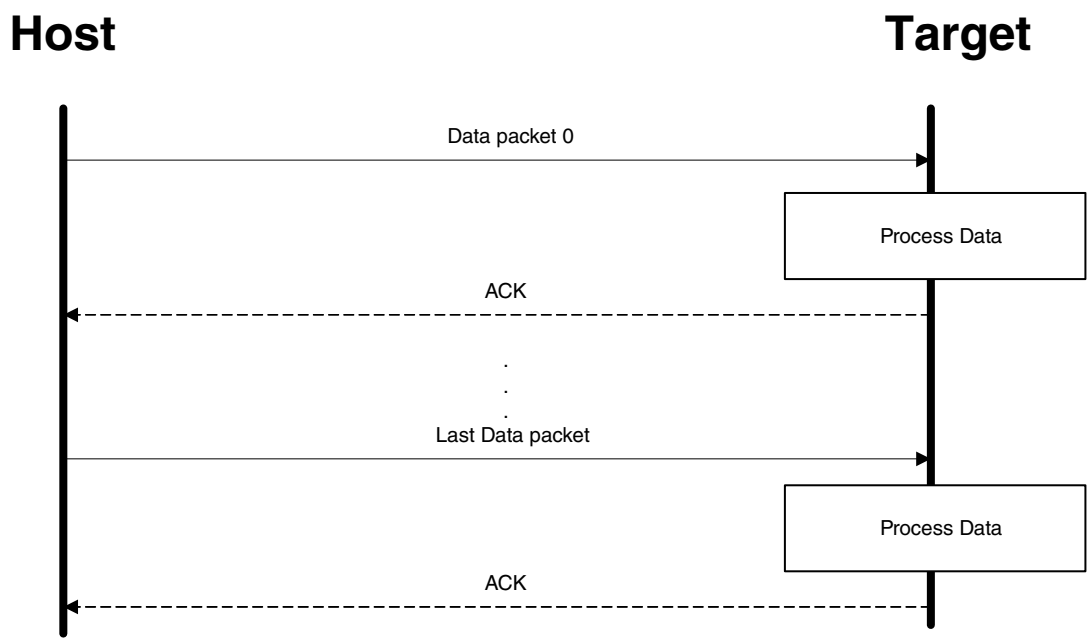


Figure 10-20. Protocol for a data-only phase

There are 2 types of packet protocol defined for the peripherals with or without built-in flow control.

- **Serial packet protocol**. It is defined for UART. A packet level flow control mechanism and error-detection mechanism is designed in this protocol.
- **HID packet protocol**. It is defined for USB-HID, the USB built-in flow control and error-detection is used behind this protocol.

In each protocol, several packet types are defined to let the boot ROM:

- Establish the communication to a host
- Recognize different phases (command-only/ data-only)

10.9.1.1 Packet Types for Serial Packet Protocol

There are 5 types of packets supported by the Serial Packet Protocol:

- Ping packet
- Ping Response packet
- Framing packet
- Command packet
- Data packet

10.9.1.1.1 Ping packet

The ping packet is the first packet sent from a host to target, in order to establish a connection on a UART peripheral for auto-baudrate detection. For other peripherals, it is optional.

Table 10-52. Ping packet format

Byte #	Value	Name
0	0x5A	Start Byte
1	0xA6	Packet Type: Ping packet

10.9.1.1.2 Ping response packet

In response to a Ping packet, the target sends a Ping Response packet.

Table 10-53. Ping response packet format

Byte #	Value	Parameter
0	0x5A	Start Byte
1	0xA7	Packet Type: Ping Response packet
2		Protocol version: bug fix
3		Protocol version: minor
4		Protocol version: major
5		Protocol name: ‘P’ (0x50)
6		Reserved: set to 0
7		Reserved: set to 0
8		crc16_low
9		crc16_high

NOTE

1. Please refer to CRC algorithm for details regarding the CRC algorithm.
2. Byte 0 - 7 are including in the CRC calculation.

Example ping packet:

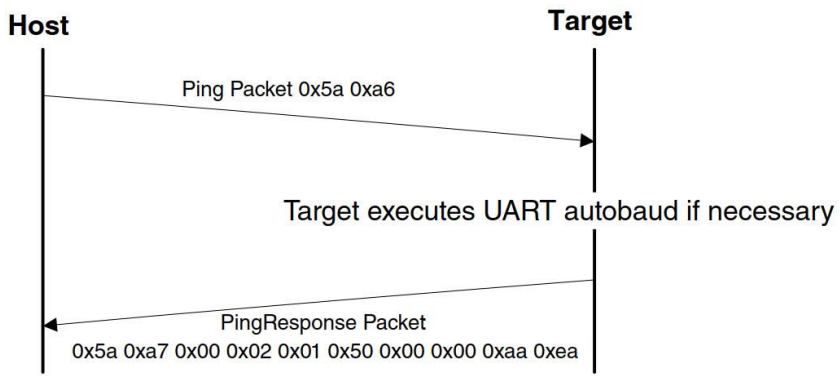


Figure 10-21. Ping Command Protocol

10.9.1.1.3 Framing Packet

The framing packet is for flow control and error detection, for the communications links that do not have such features built-in. The framing packet structure sits between the link layer and the command layer. It wraps command and data packets as well.

NOTE

Command packet is optional in ROM

Every framing packet containing data sent in one direction results in a synchronizing response framing packet in the opposite direction.

The framing packet described in the section is for serial peripherals, including UART

The framing packet format for serial packet is provided below:

Table 10-54. Framing packet format

Byte #	Data type	Value	Field	Notes
0	Framing header	0x5a	Start Byte	Must be 0x5A
1		0xA5/0xA4	Packet Type	0xA5 for Data packet 0xA4 for command packet
2		2	Payload Size	16-bit little-endian, max size is 512 bytes
4		2	CRC-16 checksum	CRC checksum covers the whole packet including start byte, packet type, payload size, and payload data, but does not contain the CRC checksum bytes
6…n	Payload	payload data	The payload of the data	Max size is 512

A particular framing packet that contains only a start byte and a packet type is employed for synchronization between the host and target. The format details are provided below:

Table 10-55. Special Framing Packet Format

Byte #	Value	Parameter
0	0x5A	Start Byte
1	0xAn	Packet Type: 0xA1 - Ack: The previous packet was received successfully; the sending of more data is allowed 0xA2 - NAK: The previous packet was corrupted, and needs to be re-sent. 0xA3 - Abort: Data transfer is aborted.

10.9.1.1.4 CRC algorithm

This section provides details about the CRC-16 algorithm.

The CRC is computed over each byte in the framing packet header, excluding the crc16 field itself, plus all the payload bytes. The CRC algorithm is the XMODEM variant of CRC-16.

The Characteristics of the XMODEM variants are:

Table 10-56. Characteristics of the XMODEM variant

Name	Value
width	16
polynomial	0x1021
init value	0x0000
reflect in	false
reflect out	false
xor out	0x0000
check result	0x31C3

10.9.1.1.5 Command Packet

The command packet carries a 32-bit command header and a list of 32-bit parameters.

Command Packet Format (32 bytes)										
Command Header (4 bytes)				28 bytes for Parameters (Max 7 parameters)						
Tag	Flags	Rsvd	Param Count	Param1 (32-bit)	Param2 (32-bit)	Param3 (32-bit)	Param4 (32-bit)	Param5 (32-bit)	Param6 (32-bit)	Param7 (32-bit)
byte 0	byte 1	byte 2	byte 3							

Command Header Format

Byte #	Command Header Field	
0	Command or Response tag	The command header is 4 bytes long, with these fields.
1	Flags	
2	Reserved. Should be 0x00.	
3	ParameterCount	

Figure 10-22. Command Packet and Command Header Format

The header is followed by 32-bit parameters up to the value of the ParameterCount field specified in the header. Because a command packet is 32 bytes long, only 7 parameters can fit into the command packet.

Command packets are also used by the target to send responses back to the host. Command packets and data packets are embedded into framing packets for all the transfers.

The supported command and response list is provided in the following tables:

Table 10-57. Supported command list

Command	Name
0x07	GetProperty

Table 10-58. Supported response list

Response	Name
0xA0	Generic Response
0xA7	GetProperty Response - used for sending a response to GetProperty command only

10.9.1.1.6 Data Packet

The data packet carries just the data that the host sends to target. The data packet is also wrapped within a framing packet to ensure the correct packet data is received.

The contents of a data packet are simply the data itself. There are no other fields so that the most data per packet can be transferred. Framing packets are responsible for ensuring that the correct packet data is received.

The host must send the data packet with the maximum payload size until the last packet, the packet with smaller payload size is treated as the last packet during the transfer.

10.9.1.1.7 Response packet

The responses are carried using the same command packet format wrapped with framing packet data. Types of responses supported in the boot ROM include:

- GenericResponse
- GetPropertyResponse

GenericResponse: After the boot ROM has processed a command, the boot ROM sends a generic response with status and command tag information to the host. The generic response is the last packet in the command protocol sequence. The generic response packet contains the framing packet data and the command packet data (with generic response tag = 0xA0) and a list of parameters. The parameter count field in the header is always set to 2, for status code and command tag parameters.

The GenericResponse parameters are provided below:

Table 10-59. GenericResponse parameters

Byte #	Parameter	Description
0 - 3	Status code	The status codes are errors encountered during the execution of a command by the target. If a command succeeds, then a success code is returned.
4 - 7	Command tag	The command tag parameter identifies the response to the command sent by the host.

NOTE

GenericResponse in ROM is only used as a response for all unsupported commands, in which:

- Status = 10000 (Unknown command)
- Command tag = command tag in the Command header.

GetPropertyResponse: The target sends the GetPropertyResponse packet in response to the host query that uses the GetProperty command. The GetPropertyResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a GetPropertyResponse tag value (0xA7).

The parameter count field in the header is set to greater than 1, to always include the status code and one or many property values.

The GetPropertyResponse parameters are provided below:

Table 10-60. GetPropertyResponse parameters

Byte #	Value	Parameter
0 - 3		Status code
4 - 7		Property code
. . .		. . .
		Can be up to maximum 6 property values. Limited to the size of the 32-bit command packet and property type.

The possible status are:

- 0 - kStatus_Success
- 10300 - Unknown Property

10.9.1.2 Packet Types for USB-HID protocol

There are 3 types of packets supported in USB-HID protocol:

- Framing packet
- Command packet
- Data packet

10.9.1.2.1 HID reports

There are 4 HID reports defined and used by the boot ROM USB HID protocol. The report ID determines the direction and type of packet sent in the report. Otherwise, the contents of all reports are the same.

The HID reports for the USB-HID protocol are provided below:

Table 10-61. HID reports for the USB-HID protocol

Report ID	Packet Type	Direction
1	Command	OUT (Host -> Device)
2	Data	OUT (Host -> Device)
3	Command	IN (Device -> Host)

10.9.1.2.2 Framing packet

The framing packet has a 4-byte framing header, in which the report ID and the packet length is used for the packet type definition and the real payload size.

Table 10-62. USB-HID Framing Packet

Byte #	Data Type	Value	Note
0	Framing Header	Report ID	
1		Padding byte	Must be 0
2		Packet Length LSB	Max size is limited to 1016 bytes
3		Packet Length MSB	
4	Payload	Packet [0]	
5		Packet [1]	
		…	
N+4-1		Packet [N-1]	

10.9.1.2.3 Command packet

The Report ID needs to be '1' for Command, and it must be '3' for Response. The payload definition is the same with Serial Packet Protocol.

10.9.1.2.4 Data packet

The Report ID must be '2' in data packet. The payload definition is the same with Serial Protocol.

10.9.2 Serial Boot via UART

10.9.2.1 UART Configuration details

Provided below are the details of UART configuration in boot ROM:

- **8-N-1** (8-bit data, No parity bit, 1 stop bit)
- Auto-baud detection is supported during Serial Boot. Maximum baudrate can be up to 6mbps in high speed boot mode.
- The host tool needs to send out the data packet with maximum payload size (512 bytes) set in framing packet by packet, except for the last packet. ROM will end the transfer if the payload size is less than the maximum payload size.
- The first packet on UART must be Ping to perform auto-baud detection.

10.9.2.2 Example Communication sequences

1. **Ping** (for auto-baud detection)
   Host: Ping packet [5a **a6**]
   Device: Ping response < 5a **a7** 00 02 01 50 00 00 aa ea>
2. **Command packet** (Get-Property 1)
   Host: Command [5a **a4** 0c 00 4b 33 07 00 00 02 01 00 00 00 00 00 00 00]
   - 5a - Start Byte
   - **a4** - Packet Type: Command packet
   - 0c 00 - Payload size(0x000c)
   - 4b 33 - CRC checksum(0x334b)
   - 07 00 00 02 01 00 00 00 00 00 00 00 - Payload (Get-property 1)

   Device: Ack <5a **a1**>
   - 5a - Start Byte
   - **a1 -** ACK

   Device: GetPropertyResponse < 5a **a4** 0c 00 64 18 a7 00 00 02 00 00 00 00 00 01 02 4b>
   - 5a - Start Byte
   - **a4** - Packet Type: Command packet
   - 0c 00 - Payload size (0x000c)
   - 64 18 - CRC Checksum (0x1864)
   - a7 00 00 02 00 00 00 00 00 01 02 4b - Payload (GetPropertyResponse, status = 0, property = 0x4b020100)

   Host: Ack [5a **a1**]
   - 5a - Start Byte
   - **a1 -** ACK
3. **Load Image**
   1. First Data packet:
      Host: Data packet [5a **a5** 00 02 55 03 d1 00 20 41 …]
      - 5a - Start Byte
      - **a5** - Packet Type: Data packet
      - 00 02 - Payload size(0x0200)
      - 55 03 - CRC Checksum(0x0355)
      - d1 00 20 41 … - Payload (Data stream)

      Device: Ack <5a **a1**>
      - 5a - Start Byte
      - **a1 -** ACK

   2. Report the remaining data transfer following the same flow as the first data packet.
   3. Last Data packet:
      Host: Data packet [5a **a5** 84 00 37 39 07 46 38 …]
      - 5a - Start Byte
      - **a5** - Packet Type: Data packet
      - 84 00 - Payload size(0x0084)
      - 37 39 - CRC Checksum(0x3937)
      - 07 46 38 … - Payload (Data stream)

      Device: Ack <5a **a1**>

Figure 10-23. Example Communication sequences

10.9.3 Serial Boot via USB-HID

10.9.3.1 USB configuration details

The USB OTG function device driver supports a high speed (HS for UTMI) non-stream mode with a maximal packet size of 512 B and a low-level USB OTG function.

The VID/PID and strings for the USB device driver are listed in the following table.

Table 10-63. VID/PID and strings for USB device driver

Descriptor	Value
VID	0x1FC9 (NXP vendor ID)
PID[1]	0x013D
String Descriptor1 (manufacturer)	NXP Semiconductors
String Descriptor2 (product)	S Blank
String Descriptor4	NXP Flash
String Descriptor5	NXP Flash

1. Allocation based on the BPN (Before Part Number)

10.9.3.2 Endpoints

The HID peripheral uses 3 endpoints:

- Control (0)
- Interrupt IN (1)
- Interrupt OUT (2)

The interrupt OUT endpoint is optional for HID class devices, but the boot ROM uses it as a pipe, where the firmware can NAK send requests from the USB host.

10.9.3.3 Example Communication sequences

1. **Command packet** (Get-Property 1)
   Host: Command [**01** 00 0c 00 07 00 00 02 01 00 00 00 00 00 00 00]
   - **01** - Command report, Direction: OUT, Host to device
   - 00 - Padding byte
   - 0c 00 - Payload size(0x000c)
   - 07 00 00 02 01 00 00 00 00 00 00 00 - payload

   Device: Ack (USB Hardware Ack)
   Device: GetPropertyResponse <**03** 00 0c 00 a7 00 00 02 00 00 00 00 00 01 02 4b 00 00 00 00 00 … 00>
   - **03 -** Command report, Direction: IN, Device to host
   - 00 - Padding byte
   - 0c 00 - Payload size(0x000c)
   - a7 00 00 02 00 00 00 00 00 01 02 4b - Payload (GetPropertyResponse, status=0, property=0x4b020100)
   - 00 00 00 00 00 … 00 - Padding bytes

   Host: Ack (USB Hardware Ack)
2. **Load Image**
   1. First Data packet:
      Host: Data packet [**02** 00 f8 03 d1 00 20 41 01 a4 00 00 00 00 00 00 00 …]
      02 - Data report, Direction: OUT, Host to device
      00 - Padding byte
      f8 03 - Payload size(0x03f8)
      d1 00 20 41 01 a4 00 00 00 00 00 00 00 … - Payload
      Device: Ack (USB Hardware Ack)
   2. Report the remaining data transfer following the same flow as the first data packet.
   3. Last Data packet:
      Host: Data packet [**02** 00 d4 00 20 c5 28 d9 00 f0 0b f8 00 …]
      02 - Data report, Direction: OUT, Host to device
      00 - Padding byte
      d4 00 - Payload size(0x00d4)
      20 c5 28 d9 00 f0 0b f8 00 … - Payload
      Device: Ack (USB Hardware Ack)

Figure 10-24. Example Communication sequences

10.10 Recovery devices

The chip supports recovery devices. If the primary boot device fails, the boot ROM tries to boot from the recovery device using one of the LPSPI ports.

To enable the recovery device, the recovery boot fuse must be set. Additionally, the serial EEPROM fuses must be set as described in Serial NOR/EEPROM through LPSPI.

10.11 SD/MMC manufacture mode

When the internal boot and recover boot (if enabled) failed, the boot goes to the SD/MMC manufacture mode before the serial download mode. In the manufacture mode, one bit bus width is used despite of the fuse setting.

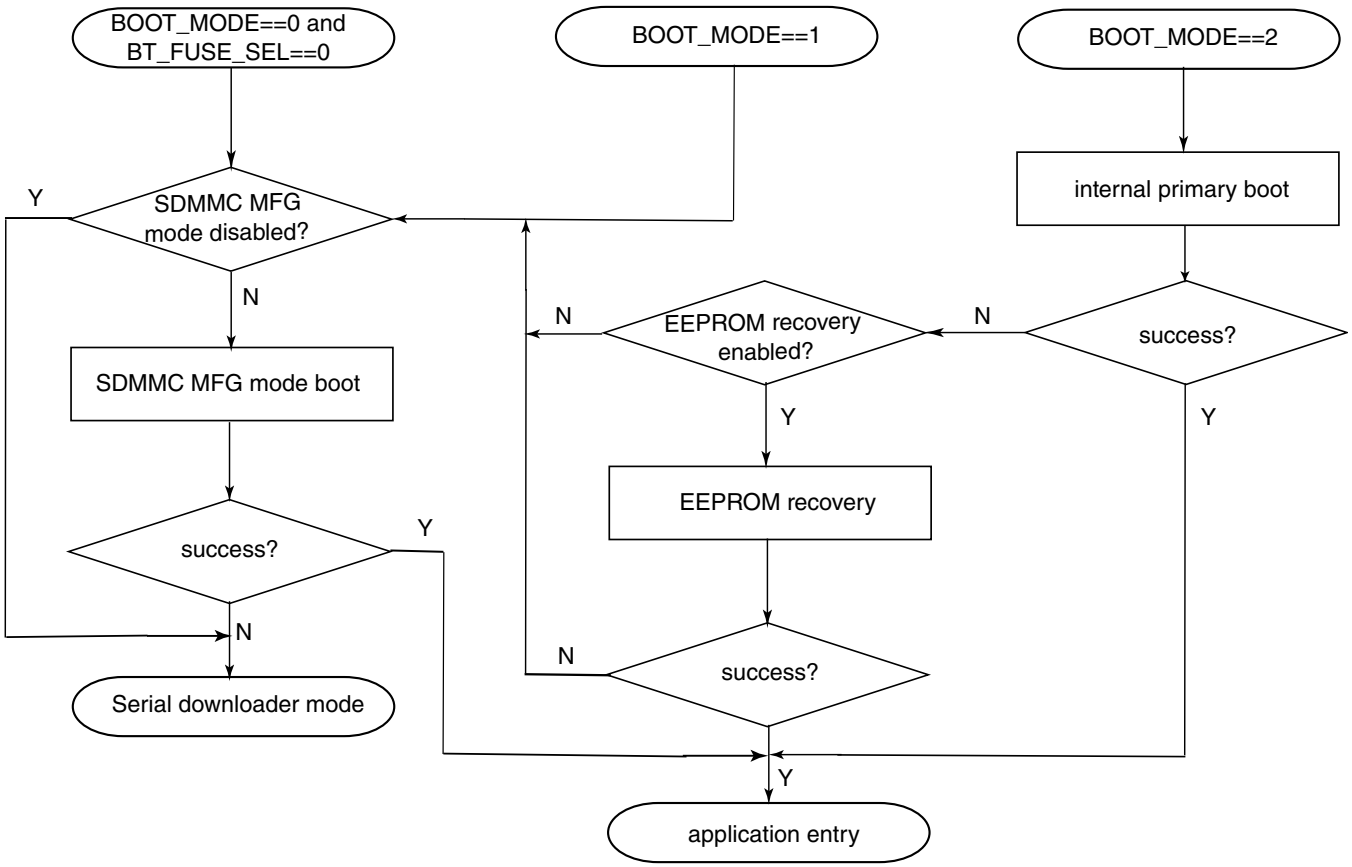


Figure 10-25. SD/MMC manufacture boot flow

10.12 High-Assurance Boot (HAB)

The High Assurance Boot (HAB) component of the ROM protects against the potential threat of attackers modifying the areas of code or data in the programmable memory to make it behave in an incorrect manner. The HAB also prevents the attempts to gain access to features which must not be available.

The integration of the HAB feature with the ROM code ensures that the chip does not enter an operational state if the existing hardware security blocks detected a condition that may be a security threat or if the areas of memory deemed to be important were modified. The HAB uses the RSA/ECDSA digital signatures to enforce these policies.

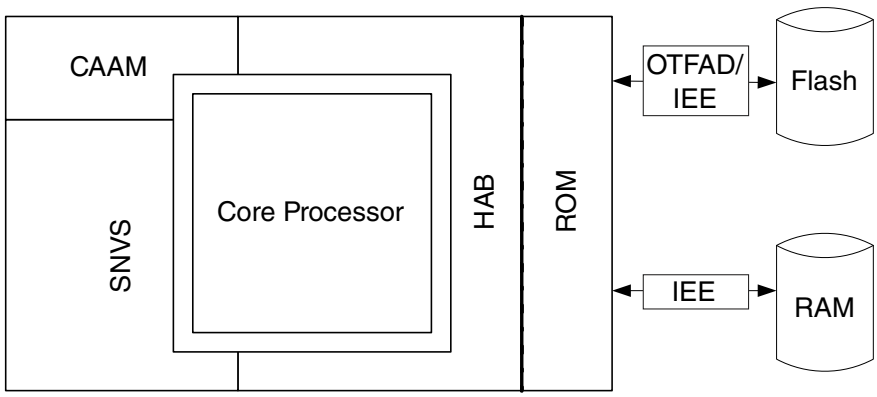


Figure 10-26. Secure boot components

The figure above illustrates the components used during a secure boot using HAB. The HAB interfaces with the SNVS to make sure that the system security state is as expected. The HAB also uses the hardware block to accelerate the SHA-256 message digest operations performed during the signature verifications. The HAB also includes a software implementation of SHA-256 for cases where a hardware accelerator cannot be used. The supported RSA key sizes are 1024, 2048, 3072, and 4096 bits. The supported ECDSA key sizes are P256, P384, and P521. The main features supported by the HAB are:

- X.509 public key certificate support
- CMS signature format support

NOTE

NXP provides the reference Code Signing Tool (CST) for key generation and code signing for use with the HAB library. The CST can be found by searching for "IMX_CST_TOOL" at http://www.nxp.com

10.12.1 HAB API vector table addresses

For devices that perform a secure boot, the HAB library may be called by the boot stages that execute after the ROM code.

The HAB API vector table for this device is at address 0x0021_1C0C .

10.13 ROM APIs

10.13.1 Introduction

The ROM bootloader provides a set of ROM APIs to simplify the In-Application Programming (IAP).

The ROM bootloader supports the following APIs:

- runBootloader API
- FlexSPI NOR FLASH Driver API

The ROM API root pointer is located at address 0x0021_001C. Please see the following figure for details of the ROM API layout.

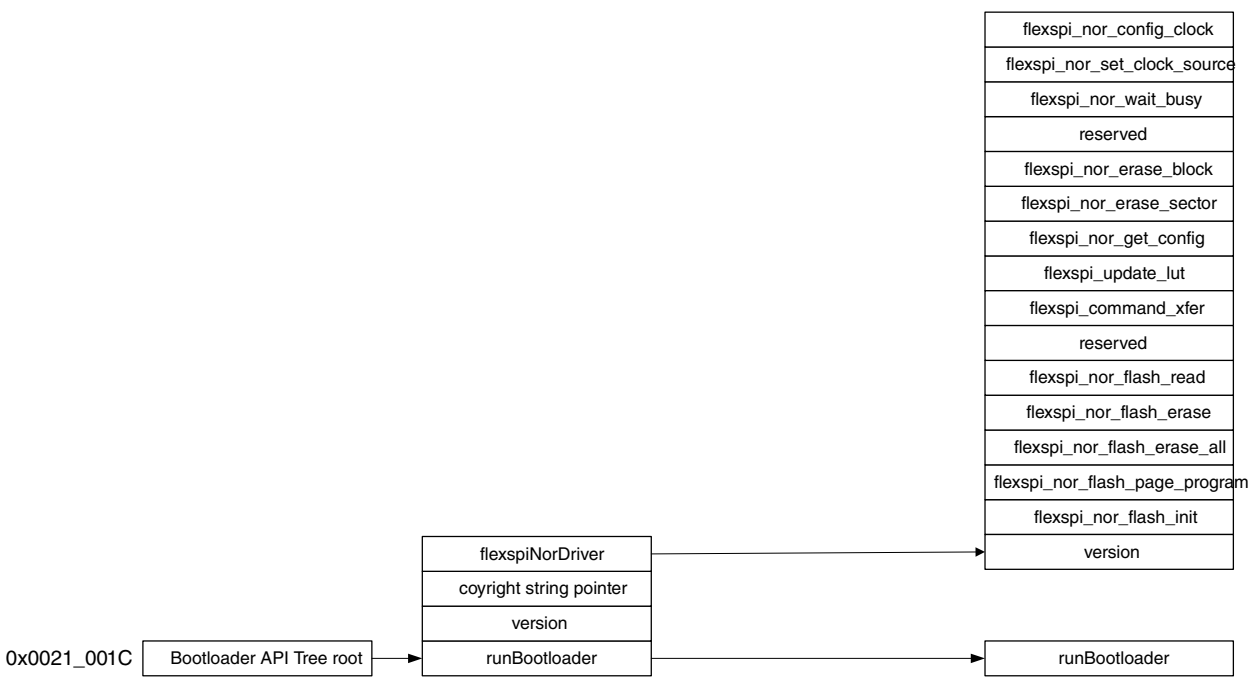


Figure 10-27. ROM API Structure

The ROM API structure definitions are as below:

1. Bootloader API Entry Structure

```
typedef struct BootloaderTree
{
    void (*runBootloader)(void *arg);                  //!< Function to start the bootloader executing.
    standard_version_t version;                        //!< Bootloader version number.
    const char *copyright;                             //!< Copyright string.
    const flexspi_nor_flash_driver_t *flexspiNorDriver; //!< FlexSPI NOR FLASH Driver API.
} bootloader_tree_t;
```

2. FlexSPI NOR Driver API structure

```
typedef struct

{

    uint32_t version;

    status_t (*init)(uint32_t instance, flexspi_nor_config_t *config);

    status_t (*page_program)(uint32_t instance, flexspi_nor_config_t *config, uint32_t dstAddr, const uint32_t *src);

    status_t (*erase_all)(uint32_t instance, flexspi_nor_config_t *config);

    status_t (*erase)(uint32_t instance, flexspi_nor_config_t *config, uint32_t start, uint32_t length);

    status_t (*read)(uint32_t instance, flexspi_nor_config_t *config, uint32_t *dst, uint32_t start, uint32_t bytes);

    uint32_t reserved;

    status_t (*xfer)(uint32_t instance, flexspi_xfer_t *xfer);

    status_t (*update_lut)(uint32_t instance, uint32_t seqIndex, const uint32_t *lutBase, uint32_t numberOfSeq);

    status_t (*get_config)(uint32_t instance, flexspi_nor_config_t *config, serial_nor_config_option_t *option);

    status_t (*erase_sector)(uint32_t instance, flexspi_nor_config_t *config, uint32_t address);

    status_t (*erase_block)(uint32_t instance, flexspi_nor_config_t *config, uint32_t address);

    void (*hw_reset)(uint32_t instance, uint32_t resetLogic);

    status_t (*wait_busy)(uint32_t instance, flexspi_nor_config_t *config, bool isParallelMode, uint32_t address);

    status_t (*set_clock_source)(uint32_t instance, uint32_t clockSrc);

    void (*config_clock)(uint32_t instance, uint32_t freqOption, uint32_t sampleClkMode);

} flexspi_nor_flash_driver_t;
```

10.13.2 FlexSPI NOR APIs

The ROM bootloader provides a set of Serial NOR FLASH APIs to simplify the external FLASH enablement on the chip. The version of the FlexSPI NOR API in the chip's ROM bootloader is 1.7.0.

NOTE

1. The RAM space occupied by ROM must be reserved if the user application needs to call the FlexSPI APIs.
2. FlexSPI APIs will re-initialize the FLEXSPI module. If the user application initialized the FLEXSPI module before calling FlexSPI APIs, re-initialization might be required after calling FlexSPI APIs.

10.13.2.1 FlexSPI NOR prototypes

10.13.2.1.1 flexspi_nor_flash_init

Initialize the Serial NOR device via FLEXSPI:

```
status_t flexspi_nor_flash_init(uint32_t instance, flexspi_nor_config_t *config)
{
    return g_bootloaderTree->flexSpiNorDriver->init(instance, config);
}
```

Please see example code in FLASH API Example.

10.13.2.1.2 flexspi_nor_flash_page_program

Program data to specified Flash address:

```
status_t flexspi_nor_flash_page_program(uint32_t instance, flexspi_nor_config_t *config,
uint32_t dstAddr, const uint32_t *src)
{
    return g_bootloaderTree->flexSpiNorDriver->program(instance, config, dstAddr, src);
}
```

Please see example code in FLASH API Example.

10.13.2.1.3 flexspi_nor_flash_erase_all

Erase the whole Flash array via FLEXSPI:

```
status_t flexspi_nor_flash_erase_all(uint32_t instance, flexspi_nor_config_t *config)
{
    return g_bootloaderTree->flexSpiNorDriver->erase_all(instance, config);
}
```

Example code:

```
flexspi_nor_flash_erase_all(1,config);
```

10.13.2.1.4 flexspi_nor_get_config

Get the Flash configuration block via the serial_nor_config_option_t block. Please see serial_nor_config_t definitions for more details.

```
status_t flexspi_nor_get_config(uint32_t instance, flexspi_nor_config_t *config,
serial_nor_config_option_t *option)
{
    return g_bootloaderTree->flexSpiNorDriver->get_config(instance, config, option);
}
```

Please see example code in FLASH API Example.

10.13.2.1.5 flexspi_nor_flash_erase

Erase specified Flash region. The minimum erase unit is one sector.

```
status_t flexspi_nor_flash_erase(uint32_t instance, flexspi_nor_config_t *config, uint32_t
start, uint32_t length)
{
    return g_bootloaderTree->flexSpiNorDriver->erase(instance, config, start, length);
}
```

Please see example code in FLASH API Example.

10.13.2.1.6 flexspi_nor_flash_read

Read the FLASH via FLEXSPI using IP read command:

```
status_t flexspi_nor_flash_read(uint32_t instance, flexspi_nor_config_t *config, uint32_t
*dst, uint32_t start, uint32_t bytes)
{
    return g_bootloaderTree->flexSpiNorDriver->read(instance, config, dst, start, bytes);
}
```

Example code:

```
uint32_t pageBuffer[256/sizeof(uint32_t)];
flexspi_nor_flash_read(1, config, pageBuffer, 0, sizeof(pageBuffer));
```

10.13.2.1.7 flexspi_update_lut

Update the specified LUT entries:

```
status_t flexspi_update_lut(uint32_t instance, uint32_t seqIndex, const uint32_t *lutBase,
uint32_t numberOfSeq)
{
    return g_bootloaderTree->flexSpiNorDriver->update_lut(instance, seqIndex, lutBase,
numberOfSeq);
}
```

Example code:

```
uint32_t chipEraseLUT[4] = {0x0460, 0, 0, 0};
flexspi_update_lut(1, 1, chipEraseLUT, 1);
```

10.13.2.1.8 flexspi_command_xfer

Execute LUT sequence specified by xfer:

```
status_t flexspi_command_xfer(uint32_t instance, flexspi_xfer_t *xfer)
{
    return g_bootloaderTree->flexSpiNorDriver->xfer(instance, xfer);
}
```

flexspi_xfer_t definition is as below:

```
//!@brief FlexSPI Transfer Context
typedef struct _FlexSpiXfer
{
    flexspi_operation_t operation; //!< FlexSPI operation
    uint32_t baseAddress;          //!< FlexSPI operation base address
    uint32_t seqId;                //!< Sequence Id
    uint32_t seqNum;               //!< Sequence Number
    bool isParallelModeEnable;     //!< Is a parallel transfer
    uint32_t *txBuffer;            //!< Tx buffer
    uint32_t txSize;               //!< Tx size in bytes
    uint32_t *rxBuffer;            //!< Rx buffer
    uint32_t rxSize;               //!< Rx size in bytes
} flexspi_xfer_t;
```

flexspi_operation_t definition is as below:

```
typedef enum _FlexSPIOperationType
{
    kFlexSpiOperation_Command,
    kFlexSpiOperation_Config,
    kFlexSpiOperation_Write,
    kFlexSpiOperation_Read,
    kFlexSpiOperation_End = kFlexSpiOperation_Read,
} flexspi_operation_t;
```

Example code, assuming the LUT index 1 is the Flash WriteEnable command.

```
flexspi_xfer_t flashXfer =
{
    kFlexSpiOperation_Command, 0, 1, 1, false, NULL, 0, NULL, 0
};
flexspi_command_xfer(0, &flashXfer);
```

10.13.2.1.9 flexspi_nor_set_clock_source

Set the clock source of the specified FLEXSPI instance:

```
status_t flexspi_nor_set_clock_source(uint32_t instance, uint32_t source)
{
    return g_botloaderTree->flexSpiNorDriver->set_clock_source(instance, source);
}
```

Example code:

```
#define CLOCK_SOURCE_RC400M (2u)
flexspi_nor_set_clock_source(1, CLOCK_SOURCE_RC400M);
```

10.13.2.1.10 flexspi_nor_configure_clock

Configure the FLEXSPI NOR clock to specified frequency:

```
status_t flexspi_nor_configure_clock(uint32_t instance, uint32_t freq, uint32_t
sampleClkMode)
{
    return g_bootloaderTree->flexSpiNorDriver->config_clock(instance, freq, sampleClkMode);
}
```

Example code:

```
enum
{
    kFlexSpiSerialClk_30MHz = 1,
    kFlexSpiSerialClk_50MHz = 2,
    kFlexSpiSerialClk_60MHz = 3,
    kFlexSpiSerialClk_80MHz = 4,
    kFlexSpiSerialClk_100MHz = 5,
    kFlexSpiSerialClk_120MHz = 6,
    kFlexSpiSerialClk_133MHz = 7,
    kFlexSpiSerialClk_166MHz = 8,
};
enum
{
    kFlexSpiClk_SDR, //!< Clock configure for SDR mode
    kFlexSpiClk_DDR, //!< Clock configurat for DDR mode
};
flexspi_nor_configure_clock(1, kFlexSpiSerialClk_100MHz, kFlexSpiClk_DDR);
```

10.13.2.1.11 serial_nor_config_t definitions

serial_nor_config_t is defined as below:

```
typedef struct _serial_nor_config_option
{
  union
  {
    struct
    {
      uint32_t max_freq : 4;         //!< Maximum supported Frequency
      uint32_t misc_mode : 4;        //!< miscellaneous mode
      uint32_t quad_mode_setting : 4; //!< Quad mode setting
      uint32_t cmd_pads : 4;         //!< Command pads
      uint32_t query_pads : 4;       //!< SFDP read pads
      uint32_t device_type : 4;      //!< Device type
      uint32_t option_size : 4;      //!< Option size, in terms of uint32_t, size = (option_size + 1) * 4
      uint32_t tag : 4;              //!< Tag, must be 0x0E
    } B;
    uint32_t U;
  } option0;

  union
  {
    struct
    {
      uint32_t dummy_cycles : 8;     //!< Dummy cycles before read
      uint32_t reserved0 : 8;
      uint32_t pinmux_group : 4;     //!< The pinmux group selection
      uint32_t dqs_pinmux_group : 4; //!< The DQS Pinmux Group Selection
      uint32_t reserved1 : 4;
      uint32_t flash_connection : 4; //!< Flash connection option: 0 -  Single Flash connected to port A, 1 -
      //! Parallel mode, 2 - Single Flash connected to Port B
    } B;
    uint32_t U;
  } option1;

} serial_nor_config_option_t;
```

Detailed information of serial_nor_config_option_t structure is shown in the following table.

Table 10-64. serial_nor_config_option_t definition

Offset	Field	Description
0	Option0	See Table 10-65 for more details
4	Option1	Optional, valid only if the Option Size field in Option0 is non-zero. See Table 10-66 for more details.

Table 10-65. Option0 definition

Field	Bits	Description
tag	31:28	The tag of the config option, fixed to 0x0C
option_size	27:24	Size in bytes = (Option Size + 1) × 4 It is 0 if only option0 is required
device_type	23:20	Device Detection Type 0 - Read SFDP for SDR commands 1 - Read SFDP for DDR Read commands 2 - HyperFLASH 1V8 3 - HyperFLASH 3V 4 - Macronix Octal DDR 6 - Micron Octal DDR 8 - Adesto EcoXiP DDR
query_pad	19:16	Data pads during Query command (read SFDP or read MID) 0 - 1 2 - 4 3 – 8
cmd_pad	15:12	Data pads during Flash access command 0 - 1 2 - 4 3 – 8
quad_mode_sett ing	11:8	Quad Mode Enable Setting 0 - Not configured 1 - Set bit 6 in Status Register 1 2 - Set bit 1 in Status Register 2 3 - Set bit 7 in Status Register 2 4 - Set bit 1 in Status Register 2 vis 0x31 command **NOTE:** This field is valid only if a device is compliant with JESD216 only (9 longword SDFP table).
misc_mode	7:4	Set to 0.

Table continues on the next page...

Table 10-65. Option0 definition (continued)

Field	Bits	Description
max_freq	3:0	Max Flash Operation speed 0 - Do not change FlexSPI clock setting Others - *See System Boot chapter for more details*. **NOTE:** The field has a restriction that the FlexSPI clock source must be PLL480_PFD0. Keep it as 0 and manually configure the FLEXSPI clock if another clock source is selected in user application.

Table 10-66. Option1 definition

Field	Bits	Description
flash_connection	31:28	Flash connection selection 0 - Single FLASH connected to Port A 1 - Parallel mode 2 - Single FLASH connected to Port B
reserved	27:20	Reserved
pin_group	19:16	PinMux group 0 - Primary pin group 1 - Secondary pin group
reserved	15:8	Reserved
dummy_cycles	7:0	Dummy cycles for the read command 0 - Use detected dummy cycle Others - dummy cycles provided in flash datasheet

NOTE

- These APIs only support FLASH devices connected to SS0.
- The APIs always use 30MHz clock for the programming option. Users will need to change the "ipcmdSerialClkFreq" field in flexspi_nor_config_t field after flexspi_nor_get_config option, if a higher programming speed is expected.
- User application needs to set "max_freq" to 0 and manually configure the FLEXSPI clock before calling the flexspi_nor_get_config API, if the expected FLEXSPI clock source is not the default clock source configured by the ROM bootloader.
- The pad drive strength is configured to full-driver mode (See IOMUXC chapter for more details). Users can change the "dataPadOverride" and "sclkPadOverride" setting in

flexspi_nor_config_t structure after calling flexspi_nor_get_config, if necessary.

10.13.2.1.12 Status codes for the FlexSPI NOR API

The following table lists all the error and status codes for the FlexSPI NOR API.

Table 10-67. Status and error codes for the FlexSPI NOR API

Status	Code	Description
kStatus_Success	0	Operation succeeded without error
kStatus_Fail	1	The operation failed with a generic error
kStatus_InvalidArgument	4	The requested argument is invalid
kStatus_Timeout	5	A timeout occurred
kStatus_FLEXSPI_SequenceExecutionT imeout	7000	The command timed out
kStatus_FLEXSPI_InvalidSequence	7001	Invalid LUT sequence
kStatus_FLEXSPI_DeviceTimeout	7002	The busy time exceeded the specified timeout value
kStatus_FLEXSPINOR_ProgramFail	20100	Program Command failed
kStatus_FLEXSPINOR_EraseSectorFail	20101	Erase sector command failed
kStatus_FLEXSPINOR_EraseAllFail	20102	Erase All command failed
kStatus_FLEXSPINOR_WaitTimeout	20103	The wait time exceeded the specified timeout value
kStatus_FlexSPINOR_NotSupported	20104	The operation is not supported
kStatus_FlexSPINOR_WriteAlignmentEr ror	20105	Write address is unaligned to page size
kStatus_FlexSPINOR_CommandFailure	20106	Command failed
kStatus_FlexSPINOR_SFDP_NotFound	20107	SFDP table was not found, used for flexspi_nor_flash_get_config API
kStatus_FLEXSPINOR_Flash_NotFoun d	20109	Cannot detect a FLASH device
kStatus_FLEXSPINOR_DTRRead_Dum myProbeFailed	20110	The dummy cycle for DDR/DTR read cannot be probed.

10.13.2.1.13 Typical options for the Serial NOR devices in the market

The following list provides typical options for the serial NOR flash devices in the market.

- QuadSPI NOR - Quad SDR Read: option0 = 0xc0000007 (133MHz)
- QuadSPI NOR - Quad DDR Read: option0 = 0xc0100003 (60MHz)
- HyperFLASH 1V8: option0 = 0xc0233008 (166MHz)
- HyperFLASH 3V0: option0 = 0xc0333005 (100MHz)
- MXIC OPI DDR (OPI DDR enabled by default): option=0xc0433007(133MHz)
- Micron Octal DDR: option0=0xc0600005 (100MHz)
- Micron OPI DDR: option0=0xc0603007 (133MHz), SPI->OPI DDR
- Micron OPI DDR (DDR read enabled by default): option0 = 0xc0633007 (133MHz)
- Adesto OPI DDR: option0=0xc0803007(133MHz)

10.13.2.1.14 FLASH API Example

The following is a typical use case of the FlexSPI NOR API.

```
flexspi_nor_config_t config;
serial_nor_config_option_t option;
status_t status;

uint32_t address = 0x40000; // 256KB uint32_t
sector_size = 0x1000; // 4KB
uint32_t page_buffer[256/ sizeof(uint32_t)];
uint32_t instance = 1;


option.option0.U = 0xC0000007; // QuadSPI NOR, Frequency: 133MHz


status = flexspi_nor_get_config(instance, &config, &option);
if (status != kStatus_Success)
{
    return status;
}

status = flexspi_nor_flash_init(instance, &config);
if (status != kStatus_Success)
{
    return status;
}
status = flexspi_nor_flash_erase(instance, &config, address , sector_size); // Erase 1
sector
if (status != kStatus_Success)
{
    return status;
}


// Fill data into the page_buffer;
for (uint32_t i=0; i<sizeof(page_buffer)/sizeof(page_buffer[0]); i++)
{
    page_buffer[i] = (i << 24) | (i << 16) | (i << 8) | i;
}
// Program data to destination
status = flexspi_nor_flash_page_program(instance, &config, address, page_buffer); // program
1 page
if (status != kStatus_Success)
{
    return status;
}


// Do cache maintenance here if the D-Cache is enabled
// Use memory mapped access to verify whether data are programmed into Flash correctly
uint32_t mem_address = 0x3000_0000 + address;

if (0 == memcmp((void*)mem_address, page_buffer, sizeof(page_buffer))
{
    // Success

}
else
{
    // Report error
}
```

10.13.3 Enter Bootloader API

The ROM bootloader provides an API for the user application to safely boot from a newly updated application after In-Application Programming (IAP) or re-enter serial downloader mode for an image update.

API prototype

```
void runBootloader(void* arg)
{
    g_bootloaderTree-> runBootloader(arg);
}
```

Table 10-68. ARG definition

Field	Offset	Description
Tag	[31:24]	Fixed value: 0xEB (Enter Boot)
boot mode	[23:20]	0 - Determined by BMODE in SMBR2 or other Fuse combinations 1 - Serial downloader
Serial downloader media	[19:16]	0 - Auto detection 1 - USB 2 - UART
Reserved	[15:04]	Reserved
Boot image selection	[03: 00]	0 - Image 0 1 - Image 1 2 - Image 2 3 - Image 3 **NOTE:** It takes effect only if the boot mode is 0. For FlexSPI NOR boot, the maximum Image index is 1. The address and length of the backup image is specified in the fuse. Image 0 is always the latest image. Image 1 always implies the backup image. If the dual image boot feature is not implemented, the backup image means the 2nd image, otherwise, the backup image means the "old" image. For the NAND FLASH devices, the maximum image index is 3. The Image address is specified in FCB.

Typical use cases:

1. Enter Serial downloader mode and select USB as the communication peripheral:

```
uint32_t arg = 0xeb100000;
runBootloader (&arg);
```

2. Select Image1 as the boot image after reliable update in user application:

```
uint32_t arg = 0xeb000001;
runBootloader (&arg);
```

Chapter 11
External Signals and Pin Multiplexing

11.1 Overview

The chip contains a limited number of pins, most of which have multiple signal options. These signal-to-pin and pin-to-signal options are selected by the input-output multiplexer called IOMUX. The IOMUX is also used to configure other pin characteristics, such as voltage level, drive strength, and hysteresis.

The muxing options table lists the external signals grouped by the module instance, the muxing options for each signal, and the registers used to route the signal to the chosen pad.

11.1.1 Muxing Options

NOTE

For information on external signals for ADC module, refer to Chip-specific LPADC information

Table 11-1. Muxing Options

Instance	Port	Pad	Mode
ACMP1	ACMP1_OUT	GPIO_AD_17	ALT1
	ACMP1_IN1	GPIO_AD_00	No muxing
	ACMP1_IN2	GPIO_AD_01	No muxing
	ACMP1_IN3	GPIO_AD_02	No muxing
	ACMP1_IN4	GPIO_AD_03	No muxing
ACMP2	ACMP2_OUT	GPIO_AD_18	ALT1
	ACMP2_IN1	GPIO_AD_04	No muxing
	ACMP2_IN2	GPIO_AD_05	No muxing
	ACMP2_IN3	GPIO_AD_26	No muxing
	ACMP2_IN4	GPIO_AD_27	No muxing

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
ACMP3	ACMP3_OUT	GPIO_AD_19	ALT1
	ACMP3_IN1	GPIO_AD_28	No muxing
	ACMP3_IN2	GPIO_AD_29	No muxing
	ACMP3_IN3	GPIO_AD_30	No muxing
	ACMP3_IN4	GPIO_AD_31	No muxing
ACMP4	ACMP4_OUT	GPIO_AD_20	ALT1
	ACMP4_IN1	GPIO_AD_32	No muxing
	ACMP4_IN2	GPIO_AD_33	No muxing
	ACMP4_IN3	GPIO_AD_34	No muxing
	ACMP4_IN4	GPIO_AD_35	No muxing
ADC1	ADC1_CH0A	GPIO_AD_06	No muxing
	ADC1_CH0B	GPIO_AD_07	No muxing
	ADC1_CH1A	GPIO_AD_08	No muxing
	ADC1_CH1B	GPIO_AD_09	No muxing
	ADC1_CH2A	GPIO_AD_10	No muxing
	ADC1_CH2B	GPIO_AD_11	No muxing
	ADC1_CH3A	GPIO_AD_12	No muxing
	ADC1_CH3B	GPIO_AD_13	No muxing
	ADC1_CH4A	GPIO_AD_14	No muxing
	ADC1_CH4B	GPIO_AD_15	No muxing
	ADC1_CH5A	GPIO_AD_16	No muxing
	ADC1_CH5B	GPIO_AD_17	No muxing
ADC2	ADC2_CH0A	GPIO_AD_18	No muxing
	ADC2_CH0B	GPIO_AD_19	No muxing
	ADC2_CH1A	GPIO_AD_20	No muxing
	ADC2_CH1B	GPIO_AD_21	No muxing
	ADC2_CH2A	GPIO_AD_22	No muxing
	ADC2_CH2B	GPIO_AD_23	No muxing
	ADC2_CH3A	GPIO_AD_12	No muxing
	ADC2_CH3B	GPIO_AD_13	No muxing
	ADC2_CH4A	GPIO_AD_14	No muxing
	ADC2_CH4B	GPIO_AD_15	No muxing
	ADC2_CH5A	GPIO_AD_16	No muxing
	ADC2_CH5B	GPIO_AD_17	No muxing
	ADC2_CH6A	GPIO_AD_24	No muxing
	ADC2_CH6B	GPIO_AD_25	No muxing
ARM	ARM_TRACE0	GPIO_DISP_B2_02	ALT3
	ARM_TRACE1	GPIO_DISP_B2_03	ALT3
	ARM_TRACE2	GPIO_DISP_B2_04	ALT3

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	ARM_TRACE3	GPIO_DISP_B2_05	ALT3
	ARM_TRACE_CLK	GPIO_DISP_B2_06	ALT3
	ARM_TRACE_SWO	GPIO_DISP_B2_07	ALT3
	ARM_TRACE_SWO	GPIO_LPSR_11	ALT7
CCM	CCM_CLKO1	GPIO_EMC_B1_40	ALT9
	CCM_CLKO2	GPIO_EMC_B1_41	ALT9
	CCM_ENET_REF_CLK_25M	GPIO_EMC_B2_00	ALT1
	CCM_ENET_REF_CLK_25M	GPIO_AD_14	ALT9
	CCM_ENET_REF_CLK_25M	GPIO_DISP_B2_01	ALT9
	CCM_CLK1_N	CCM_CLK1_N	No muxing
	CCM_CLK1_P	CCM_CLK1_P	No muxing
CM4	ARM_CM4_EVENTI	GPIO_LPSR_01	ALT3
	ARM_CM4_EVENTO	GPIO_LPSR_00	ALT3
CM7	ARM_CM7_EVENTI	GPIO_DISP_B2_09	ALT3
	ARM_CM7_EVENTO	GPIO_DISP_B2_08	ALT3
DCDC	DCDC_PSWITCH	DCDC_PSWITCH	No muxing
	DCDC_LP	DCDC_LP	No muxing
	DCDC_MODE	DCDC_MODE	No muxing
	DCDC_LN	DCDC_LN	No muxing
	DCDC_IN	DCDC_IN	No muxing
	DCDC_INQ	DCDC_INQ	No muxing
EMVSIM1	EMVSIM1_CLK	GPIO_AD_01	ALT0
	EMVSIM1_CLK	GPIO_EMC_B2_12	ALT8
	EMVSIM1_PD	GPIO_AD_04	ALT0
	EMVSIM1_PD	GPIO_EMC_B2_15	ALT8
	EMVSIM1_POWER_FAIL	GPIO_AD_05	ALT0
	EMVSIM1_POWER_FAIL	GPIO_EMC_B2_16	ALT8
	EMVSIM1_RST	GPIO_AD_02	ALT0
	EMVSIM1_RST	GPIO_EMC_B2_13	ALT8
	EMVSIM1_SVEN	GPIO_AD_03	ALT0
	EMVSIM1_SVEN	GPIO_EMC_B2_14	ALT8
	EMVSIM1_IO	GPIO_AD_00	ALT0
	EMVSIM1_IO	GPIO_EMC_B2_11	ALT8
EMVSIM2	EMVSIM2_CLK	GPIO_DISP_B2_11	ALT1
	EMVSIM2_CLK	GPIO_AD_07	ALT2
	EMVSIM2_PD	GPIO_DISP_B2_14	ALT1
	EMVSIM2_PD	GPIO_AD_10	ALT2
	EMVSIM2_POWER_FAIL	GPIO_DISP_B2_15	ALT1
	EMVSIM2_POWER_FAIL	GPIO_AD_11	ALT2

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	EMVSIM2_RST	GPIO_DISP_B2_12	ALT1
	EMVSIM2_RST	GPIO_AD_08	ALT2
	EMVSIM2_SVEN	GPIO_DISP_B2_13	ALT1
	EMVSIM2_SVEN	GPIO_AD_09	ALT2
	EMVSIM2_IO	GPIO_DISP_B2_10	ALT1
	EMVSIM2_IO	GPIO_AD_06	ALT2
ENET	ENET_1588_EVENT0_IN	GPIO_AD_34	ALT3
	ENET_1588_EVENT0_OUT	GPIO_AD_35	ALT3
	ENET_1588_EVENT1_IN	GPIO_AD_06	ALT6
	ENET_1588_EVENT1_OUT	GPIO_AD_07	ALT6
	ENET_1588_EVENT2_IN	GPIO_AD_08	ALT6
	ENET_1588_EVENT2_OUT	GPIO_AD_09	ALT6
	ENET_1588_EVENT3_IN	GPIO_AD_10	ALT6
	ENET_1588_EVENT3_OUT	GPIO_AD_11	ALT6
	ENET_COL	GPIO_AD_19	ALT6
	ENET_CRS	GPIO_AD_18	ALT6
	ENET_MDC	GPIO_EMC_B2_19	ALT1
	ENET_MDC	GPIO_AD_32	ALT3
	ENET_MDIO	GPIO_EMC_B2_20	ALT1
	ENET_MDIO	GPIO_AD_33	ALT3
	ENET_RX_DATA0	GPIO_DISP_B2_06	ALT1
	ENET_RX_DATA0	GPIO_AD_26	ALT3
	ENET_RX_DATA1	GPIO_DISP_B2_07	ALT1
	ENET_RX_DATA1	GPIO_AD_27	ALT3
	ENET_RX_DATA2	GPIO_AD_17	ALT6
	ENET_RX_DATA3	GPIO_AD_16	ALT6
	ENET_REF_CLK	GPIO_AD_29	ALT2
	ENET_REF_CLK	GPIO_DISP_B2_05	ALT2
	ENET_REF_CLK	GPIO_DISP_B2_13	ALT4
	ENET_RX_CLK	GPIO_AD_14	ALT6
	ENET_RX_EN	GPIO_DISP_B2_08	ALT1
	ENET_RX_EN	GPIO_AD_24	ALT3
	ENET_RX_ER	GPIO_DISP_B2_09	ALT1
	ENET_RX_ER	GPIO_AD_25	ALT3
	ENET_TX_DATA0	GPIO_DISP_B2_02	ALT1
	ENET_TX_DATA0	GPIO_AD_30	ALT3
	ENET_TX_DATA1	GPIO_DISP_B2_03	ALT1
	ENET_TX_DATA1	GPIO_AD_31	ALT3
	ENET_TX_DATA2	GPIO_AD_13	ALT6

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	ENET_TX_DATA3	GPIO_AD_12	ALT6
	ENET_TX_CLK	GPIO_DISP_B2_05	ALT1
	ENET_TX_CLK	GPIO_AD_29	ALT3
	ENET_TX_EN	GPIO_DISP_B2_04	ALT1
	ENET_TX_EN	GPIO_AD_28	ALT3
	ENET_TX_ER	GPIO_AD_15	ALT6
	ENET_TX_ER	GPIO_SD_B2_07	ALT8
ENET 1G	ENET_1G_1588_EVENT0_IN	GPIO_AD_34	ALT0
	ENET_1G_1588_EVENT0_OUT	GPIO_AD_35	ALT0
	ENET_1G_1588_EVENT1_IN	GPIO_AD_00	ALT2
	ENET_1G_1588_EVENT1_OUT	GPIO_AD_01	ALT2
	ENET_1G_1588_EVENT2_IN	GPIO_AD_02	ALT2
	ENET_1G_1588_EVENT2_OUT	GPIO_AD_03	ALT2
	ENET_1G_1588_EVENT3_IN	GPIO_AD_04	ALT2
	ENET_1G_1588_EVENT3_OUT	GPIO_AD_05	ALT2
	ENET_1G_COL	GPIO_EMC_B2_10	ALT7
	ENET_1G_CRS	GPIO_EMC_B2_09	ALT7
	ENET_1G_MDC	GPIO_EMC_B2_19	ALT2
	ENET_1G_MDC	GPIO_EMC_B1_40	ALT7
	ENET_1G_MDC	GPIO_AD_16	ALT9
	ENET_1G_MDC	GPIO_AD_32	ALT9
	ENET_1G_MDIO	GPIO_EMC_B2_20	ALT2
	ENET_1G_MDIO	GPIO_EMC_B1_41	ALT7
	ENET_1G_MDIO	GPIO_AD_17	ALT9
	ENET_1G_MDIO	GPIO_AD_33	ALT9
	ENET_1G_RX_DATA0	GPIO_DISP_B1_02	ALT1
	ENET_1G_RX_DATA0	GPIO_EMC_B2_15	ALT2
	ENET_1G_RX_DATA0	GPIO_SD_B2_02	ALT2
	ENET_1G_RX_DATA1	GPIO_DISP_B1_03	ALT1
	ENET_1G_RX_DATA1	GPIO_EMC_B2_16	ALT2
	ENET_1G_RX_DATA1	GPIO_SD_B2_03	ALT2
	ENET_1G_RX_DATA2	GPIO_DISP_B1_04	ALT1
	ENET_1G_RX_DATA2	GPIO_SD_B2_04	ALT2
	ENET_1G_RX_DATA2	GPIO_EMC_B2_08	ALT7
	ENET_1G_RX_DATA3	GPIO_DISP_B1_05	ALT1
	ENET_1G_RX_DATA3	GPIO_SD_B2_05	ALT2
	ENET_1G_RX_DATA3	GPIO_EMC_B2_07	ALT7

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	ENET_1G_REF_CLK	GPIO_DISP_B1_11	ALT2
	ENET_1G_REF_CLK	GPIO_EMC_B2_19	ALT3
	ENET_1G_REF_CLK	GPIO_SD_B2_11	ALT3
	ENET_1G_REF_CLK	GPIO_DISP_B2_14	ALT4
	ENET_1G_RX_CLK	GPIO_DISP_B1_01	ALT1
	ENET_1G_RX_CLK	GPIO_SD_B2_01	ALT2
	ENET_1G_RX_CLK	GPIO_EMC_B2_05	ALT7
	ENET_1G_RX_EN	GPIO_DISP_B1_00	ALT1
	ENET_1G_RX_EN	GPIO_EMC_B2_17	ALT2
	ENET_1G_RX_EN	GPIO_SD_B2_00	ALT2
	ENET_1G_RX_ER	GPIO_EMC_B2_18	ALT2
	ENET_1G_RX_ER	GPIO_DISP_B1_01	ALT2
	ENET_1G_TX_DATA0	GPIO_DISP_B1_09	ALT1
	ENET_1G_TX_DATA0	GPIO_EMC_B2_11	ALT2
	ENET_1G_TX_DATA0	GPIO_SD_B2_09	ALT2
	ENET_1G_TX_DATA1	GPIO_DISP_B1_08	ALT1
	ENET_1G_TX_DATA1	GPIO_EMC_B2_12	ALT2
	ENET_1G_TX_DATA1	GPIO_SD_B2_08	ALT2
	ENET_1G_TX_DATA2	GPIO_DISP_B1_07	ALT1
	ENET_1G_TX_DATA2	GPIO_SD_B2_07	ALT2
	ENET_1G_TX_DATA2	GPIO_EMC_B2_04	ALT7
	ENET_1G_TX_DATA3	GPIO_DISP_B1_06	ALT1
	ENET_1G_TX_DATA3	GPIO_SD_B2_06	ALT2
	ENET_1G_TX_DATA3	GPIO_EMC_B2_03	ALT7
	ENET_1G_TX_CLK_IO	GPIO_DISP_B1_11	ALT1
	ENET_1G_TX_CLK_IO	GPIO_EMC_B2_14	ALT2
	ENET_1G_TX_CLK_IO	GPIO_SD_B2_11	ALT2
	ENET_1G_TX_EN	GPIO_DISP_B1_10	ALT1
	ENET_1G_TX_EN	GPIO_EMC_B2_13	ALT2
	ENET_1G_TX_EN	GPIO_SD_B2_10	ALT2
	ENET_1G_TX_ER	GPIO_DISP_B2_00	ALT3
	ENET_1G_TX_ER	GPIO_EMC_B2_06	ALT7
ENET QoS	ENET_QOS_1588_EVENT0_AUX_IN	GPIO_DISP_B2_15	ALT8
	ENET_QOS_1588_EVENT0_IN	GPIO_DISP_B2_14	ALT8
	ENET_QOS_1588_EVENT0_OUT	GPIO_DISP_B2_13	ALT8
	ENET_QOS_1588_EVENT1_AUX_IN	GPIO_EMC_B2_02	ALT7

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	ENET_QOS_1588_EVENT1_IN	GPIO_EMC_B2_01	ALT7
	ENET_QOS_1588_EVENT1_OUT	GPIO_EMC_B2_00	ALT7
	ENET_QOS_1588_EVENT2_AUX_IN	GPIO_SD_B1_04	ALT9
	ENET_QOS_1588_EVENT2_IN	GPIO_AD_21	ALT9
	ENET_QOS_1588_EVENT2_OUT	GPIO_AD_20	ALT9
	ENET_QOS_1588_EVENT3_AUX_IN	GPIO_SD_B1_05	ALT9
	ENET_QOS_1588_EVENT3_IN	GPIO_AD_23	ALT9
	ENET_QOS_1588_EVENT3_OUT	GPIO_AD_22	ALT9
	ENET_QOS_COL	GPIO_DISP_B2_12	ALT8
	ENET_QOS_CRS	GPIO_DISP_B2_11	ALT8
	ENET_QOS_MDC	GPIO_EMC_B2_19	ALT8
	ENET_QOS_MDC	GPIO_AD_26	ALT9
	ENET_QOS_MDIO	GPIO_EMC_B2_20	ALT8
	ENET_QOS_MDIO	GPIO_AD_27	ALT9
	ENET_QOS_RX_DATA0	GPIO_DISP_B1_02	ALT8
	ENET_QOS_RX_DATA0	GPIO_DISP_B2_06	ALT8
	ENET_QOS_RX_DATA1	GPIO_DISP_B1_03	ALT8
	ENET_QOS_RX_DATA1	GPIO_DISP_B2_07	ALT8
	ENET_QOS_RX_DATA2	GPIO_DISP_B1_04	ALT8
	ENET_QOS_RX_DATA3	GPIO_DISP_B1_05	ALT8
	ENET_QOS_REF_CLK	GPIO_EMC_B2_20	ALT3
	ENET_QOS_REF_CLK	GPIO_SD_B2_07	ALT9
	ENET_QOS_REF_CLK	GPIO_DISP_B1_11	ALT9
	ENET_QOS_RX_CLK	GPIO_DISP_B1_01	ALT8
	ENET_QOS_RX_EN	GPIO_DISP_B1_00	ALT8
	ENET_QOS_RX_EN	GPIO_DISP_B2_08	ALT8
	ENET_QOS_RX_ER	GPIO_DISP_B2_09	ALT8
	ENET_QOS_RX_ER	GPIO_DISP_B2_10	ALT8
	ENET_QOS_RX_ER	GPIO_DISP_B1_01	ALT9
	ENET_QOS_TX_DATA0	GPIO_DISP_B1_09	ALT8
	ENET_QOS_TX_DATA0	GPIO_DISP_B2_02	ALT8
	ENET_QOS_TX_DATA1	GPIO_DISP_B1_08	ALT8
	ENET_QOS_TX_DATA1	GPIO_DISP_B2_03	ALT8
	ENET_QOS_TX_DATA2	GPIO_DISP_B1_07	ALT8

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	ENET_QOS_TX_DATA3	GPIO_DISP_B1_06	ALT8
	ENET_QOS_TX_CLK	GPIO_DISP_B1_11	ALT8
	ENET_QOS_TX_CLK	GPIO_DISP_B2_05	ALT8
	ENET_QOS_TX_EN	GPIO_DISP_B1_10	ALT8
	ENET_QOS_TX_EN	GPIO_DISP_B2_04	ALT8
	ENET_QOS_TX_ER	GPIO_DISP_B2_00	ALT8
EWM	EWM_OUT_B	GPIO_EMC_B2_18	ALT3
	EWM_OUT_B	GPIO_DISP_B2_01	ALT8
	EWM_OUT_B	GPIO_AD_12	ALT9
FLEXCAN1	FLEXCAN1_RX	GPIO_AD_07	ALT1
	FLEXCAN1_RX	GPIO_DISP_B2_13	ALT2
	FLEXCAN1_RX	GPIO_DISP_B2_15	ALT6
	FLEXCAN1_TX	GPIO_AD_06	ALT1
	FLEXCAN1_TX	GPIO_DISP_B2_12	ALT2
	FLEXCAN1_TX	GPIO_DISP_B2_14	ALT6
FLEXCAN2	FLEXCAN2_RX	GPIO_AD_01	ALT1
	FLEXCAN2_RX	GPIO_AD_31	ALT2
	FLEXCAN2_TX	GPIO_AD_00	ALT1
	FLEXCAN2_TX	GPIO_AD_30	ALT2
FLEXCAN3	FLEXCAN3_RX	GPIO_LPSR_01	ALT0
	FLEXCAN3_RX	GPIO_LPSR_09	ALT1
	FLEXCAN3_RX	GPIO_LPSR_07	ALT6
	FLEXCAN3_TX	GPIO_LPSR_00	ALT0
	FLEXCAN3_TX	GPIO_LPSR_08	ALT1
	FLEXCAN3_TX	GPIO_LPSR_06	ALT6
FLEXIO1	FLEXIO1_D00	GPIO_EMC_B1_00	ALT8
	FLEXIO1_D01	GPIO_EMC_B1_01	ALT8
	FLEXIO1_D10	GPIO_EMC_B1_10	ALT8
	FLEXIO1_D11	GPIO_EMC_B1_11	ALT8
	FLEXIO1_D12	GPIO_EMC_B1_12	ALT8
	FLEXIO1_D13	GPIO_EMC_B1_13	ALT8
	FLEXIO1_D14	GPIO_EMC_B1_14	ALT8
	FLEXIO1_D15	GPIO_EMC_B1_15	ALT8
	FLEXIO1_D16	GPIO_EMC_B1_16	ALT8
	FLEXIO1_D17	GPIO_EMC_B1_17	ALT8
	FLEXIO1_D18	GPIO_EMC_B1_18	ALT8
	FLEXIO1_D19	GPIO_EMC_B1_19	ALT8
	FLEXIO1_D02	GPIO_EMC_B1_02	ALT8
	FLEXIO1_D20	GPIO_EMC_B1_20	ALT8

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	FLEXIO1_D21	GPIO_EMC_B1_21	ALT8
	FLEXIO1_D22	GPIO_EMC_B1_22	ALT8
	FLEXIO1_D23	GPIO_EMC_B1_23	ALT8
	FLEXIO1_D24	GPIO_EMC_B1_24	ALT8
	FLEXIO1_D25	GPIO_EMC_B1_25	ALT8
	FLEXIO1_D26	GPIO_EMC_B1_26	ALT8
	FLEXIO1_D27	GPIO_EMC_B1_27	ALT8
	FLEXIO1_D28	GPIO_EMC_B1_28	ALT8
	FLEXIO1_D29	GPIO_EMC_B1_29	ALT8
	FLEXIO1_D03	GPIO_EMC_B1_03	ALT8
	FLEXIO1_D30	GPIO_EMC_B1_30	ALT8
	FLEXIO1_D31	GPIO_EMC_B1_31	ALT8
	FLEXIO1_D04	GPIO_EMC_B1_04	ALT8
	FLEXIO1_D05	GPIO_EMC_B1_05	ALT8
	FLEXIO1_D06	GPIO_EMC_B1_06	ALT8
	FLEXIO1_D07	GPIO_EMC_B1_07	ALT8
	FLEXIO1_D08	GPIO_EMC_B1_08	ALT8
	FLEXIO1_D09	GPIO_EMC_B1_09	ALT8
FLEXIO 2	FLEXIO2_D00	GPIO_AD_00	ALT8
	FLEXIO2_D01	GPIO_AD_01	ALT8
	FLEXIO2_D10	GPIO_AD_10	ALT8
	FLEXIO2_D11	GPIO_AD_11	ALT8
	FLEXIO2_D12	GPIO_AD_12	ALT8
	FLEXIO2_D13	GPIO_AD_13	ALT8
	FLEXIO2_D14	GPIO_AD_14	ALT8
	FLEXIO2_D15	GPIO_AD_15	ALT8
	FLEXIO2_D16	GPIO_AD_16	ALT8
	FLEXIO2_D17	GPIO_AD_17	ALT8
	FLEXIO2_D18	GPIO_AD_18	ALT8
	FLEXIO2_D19	GPIO_AD_19	ALT8
	FLEXIO2_D02	GPIO_AD_02	ALT8
	FLEXIO2_D20	GPIO_AD_20	ALT8
	FLEXIO2_D21	GPIO_AD_21	ALT8
	FLEXIO2_D22	GPIO_AD_22	ALT8
	FLEXIO2_D23	GPIO_AD_23	ALT8
	FLEXIO2_D24	GPIO_AD_24	ALT8
	FLEXIO2_D25	GPIO_AD_25	ALT8
	FLEXIO2_D26	GPIO_AD_26	ALT8
	FLEXIO2_D27	GPIO_AD_27	ALT8

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	FLEXIO2_D28	GPIO_AD_28	ALT8
	FLEXIO2_D29	GPIO_AD_29	ALT8
	FLEXIO2_D3	GPIO_AD_03	ALT8
	FLEXIO2_D30	GPIO_AD_30	ALT8
	FLEXIO2_D31	GPIO_AD_31	ALT8
	FLEXIO2_D04	GPIO_AD_04	ALT8
	FLEXIO2_D05	GPIO_AD_05	ALT8
	FLEXIO2_D06	GPIO_AD_06	ALT8
	FLEXIO2_D07	GPIO_AD_07	ALT8
	FLEXIO2_D08	GPIO_AD_08	ALT8
	FLEXIO2_D09	GPIO_AD_09	ALT8
FLEXPWM1	FLEXPWM1_PWM0_A	GPIO_EMC_B1_23	ALT1
	FLEXPWM1_PWM0_A	GPIO_AD_00	ALT4
	FLEXPWM1_PWM1_A	GPIO_EMC_B1_25	ALT1
	FLEXPWM1_PWM1_A	GPIO_AD_02	ALT4
	FLEXPWM1_PWM2_A	GPIO_EMC_B1_27	ALT1
	FLEXPWM1_PWM2_A	GPIO_AD_04	ALT4
	FLEXPWM1_PWM3_A	GPIO_EMC_B1_38	ALT1
	FLEXPWM1_PWM0_B	GPIO_EMC_B1_24	ALT1
	FLEXPWM1_PWM0_B	GPIO_AD_01	ALT4
	FLEXPWM1_PWM1_B	GPIO_EMC_B1_26	ALT1
	FLEXPWM1_PWM1_B	GPIO_AD_03	ALT4
	FLEXPWM1_PWM2_B	GPIO_EMC_B1_28	ALT1
	FLEXPWM1_PWM2_B	GPIO_AD_05	ALT4
	FLEXPWM1_PWM3_B	GPIO_EMC_B1_39	ALT1
	FLEXPWM1_PWM0_X	GPIO_AD_06	ALT11
	FLEXPWM1_PWM1_X	GPIO_AD_07	ALT11
	FLEXPWM1_PWM2_X	GPIO_AD_08	ALT11
	FLEXPWM1_PWM3_X	GPIO_AD_09	ALT11
FLEXPWM2	FLEXPWM2_PWM0_A	GPIO_EMC_B1_06	ALT1
	FLEXPWM2_PWM0_A	GPIO_AD_24	ALT4
	FLEXPWM2_PWM1_A	GPIO_EMC_B1_08	ALT1
	FLEXPWM2_PWM1_A	GPIO_AD_26	ALT4
	FLEXPWM2_PWM2_A	GPIO_EMC_B1_10	ALT1
	FLEXPWM2_PWM2_A	GPIO_AD_28	ALT4
	FLEXPWM2_PWM3_A	GPIO_EMC_B1_19	ALT1
	FLEXPWM2_PWM0_B	GPIO_EMC_B1_07	ALT1
	FLEXPWM2_PWM0_B	GPIO_AD_25	ALT4
	FLEXPWM2_PWM1_B	GPIO_EMC_B1_09	ALT1

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	FLEXPWM2_PWM1_B	GPIO_AD_27	ALT4
	FLEXPWM2_PWM2_B	GPIO_EMC_B1_11	ALT1
	FLEXPWM2_PWM2_B	GPIO_AD_29	ALT4
	FLEXPWM2_PWM3_B	GPIO_EMC_B1_20	ALT1
	FLEXPWM2_PWM0_X	GPIO_AD_10	ALT11
	FLEXPWM2_PWM1_X	GPIO_AD_11	ALT11
	FLEXPWM2_PWM2_X	GPIO_AD_12	ALT11
	FLEXPWM2_PWM3_X	GPIO_AD_13	ALT11
FLEXPWM3	FLEXPWM3_PWM0_A	GPIO_EMC_B1_29	ALT1
	FLEXPWM3_PWM0_A	GPIO_EMC_B2_00	ALT11
	FLEXPWM3_PWM1_A	GPIO_EMC_B1_31	ALT1
	FLEXPWM3_PWM1_A	GPIO_EMC_B2_02	ALT11
	FLEXPWM3_PWM2_A	GPIO_EMC_B1_33	ALT1
	FLEXPWM3_PWM2_A	GPIO_EMC_B2_04	ALT11
	FLEXPWM3_PWM3_A	GPIO_EMC_B1_21	ALT1
	FLEXPWM3_PWM3_A	GPIO_EMC_B2_06	ALT11
	FLEXPWM3_PWM0_B	GPIO_EMC_B1_30	ALT1
	FLEXPWM3_PWM0_B	GPIO_EMC_B2_01	ALT11
	FLEXPWM3_PWM1_B	GPIO_EMC_B1_32	ALT1
	FLEXPWM3_PWM1_B	GPIO_EMC_B2_03	ALT11
	FLEXPWM3_PWM2_B	GPIO_EMC_B1_34	ALT1
	FLEXPWM3_PWM2_B	GPIO_EMC_B2_05	ALT11
	FLEXPWM3_PWM3_B	GPIO_EMC_B1_22	ALT1
	FLEXPWM3_PWM3_B	GPIO_EMC_B2_07	ALT11
	FLEXPWM3_PWM0_X	GPIO_AD_14	ALT11
	FLEXPWM3_PWM1_X	GPIO_AD_15	ALT11
	FLEXPWM3_PWM2_X	GPIO_AD_16	ALT11
	FLEXPWM3_PWM3_X	GPIO_AD_17	ALT11
FLEXPWM4	FLEXPWM4_PWM0_A	GPIO_EMC_B1_00	ALT1
	FLEXPWM4_PWM1_A	GPIO_EMC_B1_02	ALT1
	FLEXPWM4_PWM2_A	GPIO_EMC_B1_04	ALT1
	FLEXPWM4_PWM3_A	GPIO_EMC_B1_17	ALT1
	FLEXPWM4_PWM0_B	GPIO_EMC_B1_01	ALT1
	FLEXPWM4_PWM1_B	GPIO_EMC_B1_03	ALT1
	FLEXPWM4_PWM2_B	GPIO_EMC_B1_05	ALT1
	FLEXPWM4_PWM3_B	GPIO_EMC_B1_18	ALT1
	FLEXPWM4_PWM0_X	GPIO_AD_18	ALT11
	FLEXPWM4_PWM1_X	GPIO_AD_19	ALT11
	FLEXPWM4_PWM2_X	GPIO_AD_20	ALT11

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	FLEXPWM4_PWM3_X	GPIO_AD_21	ALT11
FLEXSPI1 A	FLEXSPI1_A_DATA0	GPIO_SD_B2_08	ALT1
	FLEXSPI1_A_DATA0	GPIO_AD_20	ALT3
	FLEXSPI1_A_DATA1	GPIO_SD_B2_09	ALT1
	FLEXSPI1_A_DATA1	GPIO_AD_21	ALT3
	FLEXSPI1_A_DATA2	GPIO_SD_B2_10	ALT1
	FLEXSPI1_A_DATA2	GPIO_AD_22	ALT3
	FLEXSPI1_A_DATA3	GPIO_SD_B2_11	ALT1
	FLEXSPI1_A_DATA3	GPIO_AD_23	ALT3
	FLEXSPI1_A_DQS	GPIO_SD_B2_05	ALT1
	FLEXSPI1_A_DQS	GPIO_AD_17	ALT3
	FLEXSPI1_A_DQS	GPIO_EMC_B2_18	ALT6
	FLEXSPI1_A_SCLK	GPIO_SD_B2_07	ALT1
	FLEXSPI1_A_SCLK	GPIO_AD_19	ALT3
	FLEXSPI1_A_SS0_B	GPIO_SD_B2_06	ALT1
	FLEXSPI1_A_SS0_B	GPIO_AD_18	ALT3
	FLEXSPI1_A_SS1_B	GPIO_SD_B2_04	ALT3
	FLEXSPI1_A_SS1_B	GPIO_SD_B1_02	ALT9
FLEXSPI1_B	FLEXSPI1_B_DATA0	GPIO_SD_B2_03	ALT1
	FLEXSPI1_B_DATA0	GPIO_AD_15	ALT3
	FLEXSPI1_B_DATA1	GPIO_SD_B2_02	ALT1
	FLEXSPI1_B_DATA1	GPIO_AD_14	ALT3
	FLEXSPI1_B_DATA2	GPIO_SD_B2_01	ALT1
	FLEXSPI1_B_DATA2	GPIO_AD_13	ALT3
	FLEXSPI1_B_DATA3	GPIO_SD_B2_00	ALT1
	FLEXSPI1_B_DATA3	GPIO_AD_12	ALT3
	FLEXSPI1_B_DQS	GPIO_SD_B1_05	ALT8
	FLEXSPI1_B_SCLK	GPIO_SD_B2_04	ALT1
	FLEXSPI1_B_SCLK	GPIO_AD_16	ALT3
	FLEXSPI1_B_SS0_B	GPIO_SD_B2_05	ALT3
	FLEXSPI1_B_SS0_B	GPIO_SD_B1_04	ALT8
	FLEXSPI1_B_SS1_B	GPIO_AD_35	ALT9
	FLEXSPI1_B_SS1_B	GPIO_SD_B1_03	ALT9
FLEXSPI2_A	FLEXSPI2_A_DATA0	GPIO_EMC_B2_13	ALT4
	FLEXSPI2_A_DATA0	GPIO_SD_B1_02	ALT6
	FLEXSPI2_A_DATA1	GPIO_EMC_B2_14	ALT4
	FLEXSPI2_A_DATA1	GPIO_SD_B1_03	ALT6
	FLEXSPI2_A_DATA2	GPIO_EMC_B2_15	ALT4
	FLEXSPI2_A_DATA2	GPIO_SD_B1_04	ALT6

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	FLEXSPI2_A_DATA3	GPIO_EMC_B2_16	ALT4
	FLEXSPI2_A_DATA3	GPIO_SD_B1_05	ALT6
	FLEXSPI2_A_DATA4	GPIO_EMC_B2_17	ALT4
	FLEXSPI2_A_DATA5	GPIO_EMC_B2_18	ALT4
	FLEXSPI2_A_DATA6	GPIO_EMC_B2_19	ALT4
	FLEXSPI2_A_DATA7	GPIO_EMC_B2_20	ALT4
	FLEXSPI2_A_DQS	GPIO_EMC_B2_12	ALT4
	FLEXSPI2_A_SCLK	GPIO_EMC_B2_10	ALT4
	FLEXSPI2_A_SCLK	GPIO_SD_B1_01	ALT6
	FLEXSPI2_A_SS0_B	GPIO_EMC_B2_11	ALT4
	FLEXSPI2_A_SS0_B	GPIO_SD_B1_00	ALT6
	FLEXSPI2_A_SS1_B	GPIO_AD_01	ALT9
FLEXSPI2_B	FLEXSPI2_B_DATA0	GPIO_EMC_B2_06	ALT4
	FLEXSPI2_B_DATA1	GPIO_EMC_B2_05	ALT4
	FLEXSPI2_B_DATA2	GPIO_EMC_B2_04	ALT4
	FLEXSPI2_B_DATA3	GPIO_EMC_B2_03	ALT4
	FLEXSPI2_B_DATA4	GPIO_EMC_B2_02	ALT4
	FLEXSPI2_B_DATA5	GPIO_EMC_B2_01	ALT4
	FLEXSPI2_B_DATA6	GPIO_EMC_B2_00	ALT4
	FLEXSPI2_B_DATA7	GPIO_EMC_B1_41	ALT4
	FLEXSPI2_B_DQS	GPIO_EMC_B2_07	ALT4
	FLEXSPI2_B_SCLK	GPIO_EMC_B2_09	ALT4
	FLEXSPI2_B_SS0_B	GPIO_EMC_B2_08	ALT4
	FLEXSPI2_B_SS1_B	GPIO_AD_00	ALT9
GPIO1	GPIO1_IO00	GPIO_EMC_B1_00	ALT5
	GPIO1_IO01	GPIO_EMC_B1_01	ALT5
	GPIO1_IO02	GPIO_EMC_B1_02	ALT5
	GPIO1_IO03	GPIO_EMC_B1_03	ALT5
	GPIO1_IO04	GPIO_EMC_B1_04	ALT5
	GPIO1_IO05	GPIO_EMC_B1_05	ALT5
	GPIO1_IO06	GPIO_EMC_B1_06	ALT5
	GPIO1_IO07	GPIO_EMC_B1_07	ALT5
	GPIO1_IO08	GPIO_EMC_B1_08	ALT5
	GPIO1_IO09	GPIO_EMC_B1_09	ALT5
	GPIO1_IO10	GPIO_EMC_B1_10	ALT5
	GPIO1_IO11	GPIO_EMC_B1_11	ALT5
	GPIO1_IO12	GPIO_EMC_B1_12	ALT5
	GPIO1_IO13	GPIO_EMC_B1_13	ALT5
	GPIO1_IO14	GPIO_EMC_B1_14	ALT5

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO1_IO15	GPIO_EMC_B1_15	ALT5
	GPIO1_IO16	GPIO_EMC_B1_16	ALT5
	GPIO1_IO17	GPIO_EMC_B1_17	ALT5
	GPIO1_IO18	GPIO_EMC_B1_18	ALT5
	GPIO1_IO19	GPIO_EMC_B1_19	ALT5
	GPIO1_IO20	GPIO_EMC_B1_20	ALT5
	GPIO1_IO21	GPIO_EMC_B1_21	ALT5
	GPIO1_IO22	GPIO_EMC_B1_22	ALT5
	GPIO1_IO23	GPIO_EMC_B1_23	ALT5
	GPIO1_IO24	GPIO_EMC_B1_24	ALT5
	GPIO1_IO25	GPIO_EMC_B1_25	ALT5
	GPIO1_IO26	GPIO_EMC_B1_26	ALT5
	GPIO1_IO27	GPIO_EMC_B1_27	ALT5
	GPIO1_IO28	GPIO_EMC_B1_28	ALT5
	GPIO1_IO29	GPIO_EMC_B1_29	ALT5
	GPIO1_IO30	GPIO_EMC_B1_30	ALT5
	GPIO1_IO31	GPIO_EMC_B1_31	ALT5
GPIO_MUX2[1]	GPIO_MUX2_IO00	GPIO_EMC_B1_32	ALT5
	GPIO_MUX2_IO01	GPIO_EMC_B1_33	ALT5
	GPIO_MUX2_IO02	GPIO_EMC_B1_34	ALT5
	GPIO_MUX2_IO03	GPIO_EMC_B1_35	ALT5
	GPIO_MUX2_IO04	GPIO_EMC_B1_36	ALT5
	GPIO_MUX2_IO05	GPIO_EMC_B1_37	ALT5
	GPIO_MUX2_IO06	GPIO_EMC_B1_38	ALT5
	GPIO_MUX2_IO07	GPIO_EMC_B1_39	ALT5
	GPIO_MUX2_IO08	GPIO_EMC_B1_40	ALT5
	GPIO_MUX2_IO09	GPIO_EMC_B1_41	ALT5
	GPIO_MUX2_IO10	GPIO_EMC_B2_00	ALT5
	GPIO_MUX2_IO11	GPIO_EMC_B2_01	ALT5
	GPIO_MUX2_IO12	GPIO_EMC_B2_02	ALT5
	GPIO_MUX2_IO13	GPIO_EMC_B2_03	ALT5
	GPIO_MUX2_IO14	GPIO_EMC_B2_04	ALT5
	GPIO_MUX2_IO15	GPIO_EMC_B2_05	ALT5
	GPIO_MUX2_IO16	GPIO_EMC_B2_06	ALT5
	GPIO_MUX2_IO17	GPIO_EMC_B2_07	ALT5
	GPIO_MUX2_IO18	GPIO_EMC_B2_08	ALT5
	GPIO_MUX2_IO19	GPIO_EMC_B2_09	ALT5
	GPIO_MUX2_IO20	GPIO_EMC_B2_10	ALT5
	GPIO_MUX2_IO21	GPIO_EMC_B2_11	ALT5

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO_MUX2_IO22	GPIO_EMC_B2_12	ALT5
	GPIO_MUX2_IO23	GPIO_EMC_B2_13	ALT5
	GPIO_MUX2_IO24	GPIO_EMC_B2_14	ALT5
	GPIO_MUX2_IO25	GPIO_EMC_B2_15	ALT5
	GPIO_MUX2_IO26	GPIO_EMC_B2_16	ALT5
	GPIO_MUX2_IO27	GPIO_EMC_B2_17	ALT5
	GPIO_MUX2_IO28	GPIO_EMC_B2_18	ALT5
	GPIO_MUX2_IO29	GPIO_EMC_B2_19	ALT5
	GPIO_MUX2_IO30	GPIO_EMC_B2_20	ALT5
	GPIO_MUX2_IO31	GPIO_AD_00	ALT5
GPIO_MUX3[2]	GPIO_MUX3_IO00	GPIO_AD_01	ALT5
	GPIO_MUX3_IO01	GPIO_AD_02	ALT5
	GPIO_MUX3_IO02	GPIO_AD_03	ALT5
	GPIO_MUX3_IO03	GPIO_AD_04	ALT5
	GPIO_MUX3_IO04	GPIO_AD_05	ALT5
	GPIO_MUX3_IO05	GPIO_AD_06	ALT5
	GPIO_MUX3_IO06	GPIO_AD_07	ALT5
	GPIO_MUX3_IO07	GPIO_AD_08	ALT5
	GPIO_MUX3_IO08	GPIO_AD_09	ALT5
	GPIO_MUX3_IO09	GPIO_AD_10	ALT5
	GPIO_MUX3_IO10	GPIO_AD_11	ALT5
	GPIO_MUX3_IO11	GPIO_AD_12	ALT5
	GPIO_MUX3_IO12	GPIO_AD_13	ALT5
	GPIO_MUX3_IO13	GPIO_AD_14	ALT5
	GPIO_MUX3_IO14	GPIO_AD_15	ALT5
	GPIO_MUX3_IO15	GPIO_AD_16	ALT5
	GPIO_MUX3_IO16	GPIO_AD_17	ALT5
	GPIO_MUX3_IO17	GPIO_AD_18	ALT5
	GPIO_MUX3_IO18	GPIO_AD_19	ALT5
	GPIO_MUX3_IO19	GPIO_AD_20	ALT5
	GPIO_MUX3_IO20	GPIO_AD_21	ALT5
	GPIO_MUX3_IO21	GPIO_AD_22	ALT5
	GPIO_MUX3_IO22	GPIO_AD_23	ALT5
	GPIO_MUX3_IO23	GPIO_AD_24	ALT5
	GPIO_MUX3_IO24	GPIO_AD_25	ALT5
	GPIO_MUX3_IO25	GPIO_AD_26	ALT5
	GPIO_MUX3_IO26	GPIO_AD_27	ALT5
	GPIO_MUX3_IO27	GPIO_AD_28	ALT5
	GPIO_MUX3_IO28	GPIO_AD_29	ALT5

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO_MUX3_IO29	GPIO_AD_30	ALT5
	GPIO_MUX3_IO30	GPIO_AD_31	ALT5
	GPIO_MUX3_IO31	GPIO_AD_32	ALT5
GPIO4	GPIO4_IO00	GPIO_AD_33	ALT5
	GPIO4_IO01	GPIO_AD_34	ALT5
	GPIO4_IO02	GPIO_AD_35	ALT5
	GPIO4_IO03	GPIO_SD_B1_00	ALT5
	GPIO4_IO04	GPIO_SD_B1_01	ALT5
	GPIO4_IO05	GPIO_SD_B1_02	ALT5
	GPIO4_IO06	GPIO_SD_B1_03	ALT5
	GPIO4_IO07	GPIO_SD_B1_04	ALT5
	GPIO4_IO08	GPIO_SD_B1_05	ALT5
	GPIO4_IO09	GPIO_SD_B2_00	ALT5
	GPIO4_IO10	GPIO_SD_B2_01	ALT5
	GPIO4_IO11	GPIO_SD_B2_02	ALT5
	GPIO4_IO12	GPIO_SD_B2_03	ALT5
	GPIO4_IO13	GPIO_SD_B2_04	ALT5
	GPIO4_IO14	GPIO_SD_B2_05	ALT5
	GPIO4_IO15	GPIO_SD_B2_06	ALT5
	GPIO4_IO16	GPIO_SD_B2_07	ALT5
	GPIO4_IO17	GPIO_SD_B2_08	ALT5
	GPIO4_IO18	GPIO_SD_B2_09	ALT5
	GPIO4_IO19	GPIO_SD_B2_10	ALT5
	GPIO4_IO20	GPIO_SD_B2_11	ALT5
	GPIO4_IO21	GPIO_DISP_B1_00	ALT5
	GPIO4_IO22	GPIO_DISP_B1_01	ALT5
	GPIO4_IO23	GPIO_DISP_B1_02	ALT5
	GPIO4_IO24	GPIO_DISP_B1_03	ALT5
	GPIO4_IO25	GPIO_DISP_B1_04	ALT5
	GPIO4_IO26	GPIO_DISP_B1_05	ALT5
	GPIO4_IO27	GPIO_DISP_B1_06	ALT5
	GPIO4_IO28	GPIO_DISP_B1_07	ALT5
	GPIO4_IO29	GPIO_DISP_B1_08	ALT5
	GPIO4_IO30	GPIO_DISP_B1_09	ALT5
	GPIO4_IO31	GPIO_DISP_B1_10	ALT5
GPIO5	GPIO5_IO00	GPIO_DISP_B1_11	ALT5
	GPIO5_IO01	GPIO_DISP_B2_00	ALT5
	GPIO5_IO02	GPIO_DISP_B2_01	ALT5
	GPIO5_IO03	GPIO_DISP_B2_02	ALT5

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO5_IO04	GPIO_DISP_B2_03	ALT5
	GPIO5_IO05	GPIO_DISP_B2_04	ALT5
	GPIO5_IO06	GPIO_DISP_B2_05	ALT5
	GPIO5_IO07	GPIO_DISP_B2_06	ALT5
	GPIO5_IO08	GPIO_DISP_B2_07	ALT5
	GPIO5_IO09	GPIO_DISP_B2_08	ALT5
	GPIO5_IO10	GPIO_DISP_B2_09	ALT5
	GPIO5_IO11	GPIO_DISP_B2_10	ALT5
	GPIO5_IO12	GPIO_DISP_B2_11	ALT5
	GPIO5_IO13	GPIO_DISP_B2_12	ALT5
	GPIO5_IO14	GPIO_DISP_B2_13	ALT5
	GPIO5_IO15	GPIO_DISP_B2_14	ALT5
	GPIO5_IO16	GPIO_DISP_B2_15	ALT5
GPIO6	GPIO6_IO00	GPIO_LPSR_00	ALT5
	GPIO6_IO01	GPIO_LPSR_01	ALT5
	GPIO6_IO02	GPIO_LPSR_02	ALT5
	GPIO6_IO03	GPIO_LPSR_03	ALT5
	GPIO6_IO04	GPIO_LPSR_04	ALT5
	GPIO6_IO05	GPIO_LPSR_05	ALT5
	GPIO6_IO06	GPIO_LPSR_06	ALT5
	GPIO6_IO07	GPIO_LPSR_07	ALT5
	GPIO6_IO08	GPIO_LPSR_08	ALT5
	GPIO6_IO09	GPIO_LPSR_09	ALT5
	GPIO6_IO10	GPIO_LPSR_10	ALT5
	GPIO6_IO11	GPIO_LPSR_11	ALT5
	GPIO6_IO12	GPIO_LPSR_12	ALT5
	GPIO6_IO13	GPIO_LPSR_13	ALT5
	GPIO6_IO14	GPIO_LPSR_14	ALT5
	GPIO6_IO15	GPIO_LPSR_15	ALT5
GPIO7	GPIO7_IO00	GPIO_EMC_B1_00	ALT10
	GPIO7_IO01	GPIO_EMC_B1_01	ALT10
	GPIO7_IO02	GPIO_EMC_B1_02	ALT10
	GPIO7_IO03	GPIO_EMC_B1_03	ALT10
	GPIO7_IO04	GPIO_EMC_B1_04	ALT10
	GPIO7_IO05	GPIO_EMC_B1_05	ALT10
	GPIO7_IO06	GPIO_EMC_B1_06	ALT10
	GPIO7_IO07	GPIO_EMC_B1_07	ALT10
	GPIO7_IO08	GPIO_EMC_B1_08	ALT10
	GPIO7_IO09	GPIO_EMC_B1_09	ALT10

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO7_IO10	GPIO_EMC_B1_10	ALT10
	GPIO7_IO11	GPIO_EMC_B1_11	ALT10
	GPIO7_IO12	GPIO_EMC_B1_12	ALT10
	GPIO7_IO13	GPIO_EMC_B1_13	ALT10
	GPIO7_IO14	GPIO_EMC_B1_14	ALT10
	GPIO7_IO15	GPIO_EMC_B1_15	ALT10
	GPIO7_IO16	GPIO_EMC_B1_16	ALT10
	GPIO7_IO17	GPIO_EMC_B1_17	ALT10
	GPIO7_IO18	GPIO_EMC_B1_18	ALT10
	GPIO7_IO19	GPIO_EMC_B1_19	ALT10
	GPIO7_IO20	GPIO_EMC_B1_20	ALT10
	GPIO7_IO21	GPIO_EMC_B1_21	ALT10
	GPIO7_IO22	GPIO_EMC_B1_22	ALT10
	GPIO7_IO23	GPIO_EMC_B1_23	ALT10
	GPIO7_IO24	GPIO_EMC_B1_24	ALT10
	GPIO7_IO25	GPIO_EMC_B1_25	ALT10
	GPIO7_IO26	GPIO_EMC_B1_26	ALT10
	GPIO7_IO27	GPIO_EMC_B1_27	ALT10
	GPIO7_IO28	GPIO_EMC_B1_28	ALT10
	GPIO7_IO29	GPIO_EMC_B1_29	ALT10
	GPIO7_IO30	GPIO_EMC_B1_30	ALT10
	GPIO7_IO31	GPIO_EMC_B1_31	ALT10
GPIO8	GPIO8_IO00	GPIO_EMC_B1_32	ALT10
	GPIO8_IO01	GPIO_EMC_B1_33	ALT10
	GPIO8_IO02	GPIO_EMC_B1_34	ALT10
	GPIO8_IO03	GPIO_EMC_B1_35	ALT10
	GPIO8_IO04	GPIO_EMC_B1_36	ALT10
	GPIO8_IO05	GPIO_EMC_B1_37	ALT10
	GPIO8_IO06	GPIO_EMC_B1_38	ALT10
	GPIO8_IO07	GPIO_EMC_B1_39	ALT10
	GPIO8_IO08	GPIO_EMC_B1_40	ALT10
	GPIO8_IO09	GPIO_EMC_B1_41	ALT10
	GPIO8_IO10	GPIO_EMC_B2_00	ALT10
	GPIO8_IO11	GPIO_EMC_B2_01	ALT10
	GPIO8_IO12	GPIO_EMC_B2_02	ALT10
	GPIO8_IO13	GPIO_EMC_B2_03	ALT10
	GPIO8_IO14	GPIO_EMC_B2_04	ALT10
	GPIO8_IO15	GPIO_EMC_B2_05	ALT10
	GPIO8_IO16	GPIO_EMC_B2_06	ALT10

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO8_IO17	GPIO_EMC_B2_07	ALT10
	GPIO8_IO18	GPIO_EMC_B2_08	ALT10
	GPIO8_IO19	GPIO_EMC_B2_09	ALT10
	GPIO8_IO20	GPIO_EMC_B2_10	ALT10
	GPIO8_IO21	GPIO_EMC_B2_11	ALT10
	GPIO8_IO22	GPIO_EMC_B2_12	ALT10
	GPIO8_IO23	GPIO_EMC_B2_13	ALT10
	GPIO8_IO24	GPIO_EMC_B2_14	ALT10
	GPIO8_IO25	GPIO_EMC_B2_15	ALT10
	GPIO8_IO26	GPIO_EMC_B2_16	ALT10
	GPIO8_IO27	GPIO_EMC_B2_17	ALT10
	GPIO8_IO28	GPIO_EMC_B2_18	ALT10
	GPIO8_IO29	GPIO_EMC_B2_19	ALT10
	GPIO8_IO30	GPIO_EMC_B2_20	ALT10
	GPIO8_IO31	GPIO_AD_00	ALT10
GPIO9	GPIO9_IO00	GPIO_AD_01	ALT10
	GPIO9_IO01	GPIO_AD_02	ALT10
	GPIO9_IO02	GPIO_AD_03	ALT10
	GPIO9_IO03	GPIO_AD_04	ALT10
	GPIO9_IO04	GPIO_AD_05	ALT10
	GPIO9_IO05	GPIO_AD_06	ALT10
	GPIO9_IO06	GPIO_AD_07	ALT10
	GPIO9_IO07	GPIO_AD_08	ALT10
	GPIO9_IO08	GPIO_AD_09	ALT10
	GPIO9_IO09	GPIO_AD_10	ALT10
	GPIO9_IO10	GPIO_AD_11	ALT10
	GPIO9_IO11	GPIO_AD_12	ALT10
	GPIO9_IO12	GPIO_AD_13	ALT10
	GPIO9_IO13	GPIO_AD_14	ALT10
	GPIO9_IO14	GPIO_AD_15	ALT10
	GPIO9_IO15	GPIO_AD_16	ALT10
	GPIO9_IO16	GPIO_AD_17	ALT10
	GPIO9_IO17	GPIO_AD_18	ALT10
	GPIO9_IO18	GPIO_AD_19	ALT10
	GPIO9_IO19	GPIO_AD_20	ALT10
	GPIO9_IO20	GPIO_AD_21	ALT10
	GPIO9_IO21	GPIO_AD_22	ALT10
	GPIO9_IO22	GPIO_AD_23	ALT10
	GPIO9_IO23	GPIO_AD_24	ALT10

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO9_IO24	GPIO_AD_25	ALT10
	GPIO9_IO25	GPIO_AD_26	ALT10
	GPIO9_IO26	GPIO_AD_27	ALT10
	GPIO9_IO27	GPIO_AD_28	ALT10
	GPIO9_IO28	GPIO_AD_29	ALT10
	GPIO9_IO29	GPIO_AD_30	ALT10
	GPIO9_IO30	GPIO_AD_31	ALT10
	GPIO9_IO31	GPIO_AD_32	ALT10
GPIO10	GPIO10_IO00	GPIO_AD_33	ALT10
	GPIO10_IO01	GPIO_AD_34	ALT10
	GPIO10_IO02	GPIO_AD_35	ALT10
	GPIO10_IO03	GPIO_SD_B1_00	ALT10
	GPIO10_IO04	GPIO_SD_B1_01	ALT10
	GPIO10_IO05	GPIO_SD_B1_02	ALT10
	GPIO10_IO06	GPIO_SD_B1_03	ALT10
	GPIO10_IO07	GPIO_SD_B1_04	ALT10
	GPIO10_IO08	GPIO_SD_B1_05	ALT10
	GPIO10_IO09	GPIO_SD_B2_00	ALT10
	GPIO10_IO10	GPIO_SD_B2_01	ALT10
	GPIO10_IO11	GPIO_SD_B2_02	ALT10
	GPIO10_IO12	GPIO_SD_B2_03	ALT10
	GPIO10_IO13	GPIO_SD_B2_04	ALT10
	GPIO10_IO14	GPIO_SD_B2_05	ALT10
	GPIO10_IO15	GPIO_SD_B2_06	ALT10
	GPIO10_IO16	GPIO_SD_B2_07	ALT10
	GPIO10_IO17	GPIO_SD_B2_08	ALT10
	GPIO10_IO18	GPIO_SD_B2_09	ALT10
	GPIO10_IO19	GPIO_SD_B2_10	ALT10
	GPIO10_IO20	GPIO_SD_B2_11	ALT10
	GPIO10_IO21	GPIO_DISP_B1_00	ALT10
	GPIO10_IO22	GPIO_DISP_B1_01	ALT10
	GPIO10_IO23	GPIO_DISP_B1_02	ALT10
	GPIO10_IO24	GPIO_DISP_B1_03	ALT10
	GPIO10_IO25	GPIO_DISP_B1_04	ALT10
	GPIO10_IO26	GPIO_DISP_B1_05	ALT10
	GPIO10_IO27	GPIO_DISP_B1_06	ALT10
	GPIO10_IO28	GPIO_DISP_B1_07	ALT10
	GPIO10_IO29	GPIO_DISP_B1_08	ALT10
	GPIO10_IO30	GPIO_DISP_B1_09	ALT10

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO10_IO31	GPIO_DISP_B1_10	ALT10
GPIO11	GPIO11_IO00	GPIO_DISP_B1_11	ALT10
	GPIO11_IO01	GPIO_DISP_B2_00	ALT10
	GPIO11_IO02	GPIO_DISP_B2_01	ALT10
	GPIO11_IO03	GPIO_DISP_B2_02	ALT10
	GPIO11_IO04	GPIO_DISP_B2_03	ALT10
	GPIO11_IO05	GPIO_DISP_B2_04	ALT10
	GPIO11_IO06	GPIO_DISP_B2_05	ALT10
	GPIO11_IO07	GPIO_DISP_B2_06	ALT10
	GPIO11_IO08	GPIO_DISP_B2_07	ALT10
	GPIO11_IO09	GPIO_DISP_B2_08	ALT10
	GPIO11_IO10	GPIO_DISP_B2_09	ALT10
	GPIO11_IO11	GPIO_DISP_B2_10	ALT10
	GPIO11_IO12	GPIO_DISP_B2_11	ALT10
	GPIO11_IO13	GPIO_DISP_B2_12	ALT10
	GPIO11_IO14	GPIO_DISP_B2_13	ALT10
	GPIO11_IO15	GPIO_DISP_B2_14	ALT10
	GPIO11_IO16	GPIO_DISP_B2_15	ALT10
GPIO12	GPIO12_IO00	GPIO_LPSR_00	ALT10
	GPIO12_IO01	GPIO_LPSR_01	ALT10
	GPIO12_IO02	GPIO_LPSR_02	ALT10
	GPIO12_IO03	GPIO_LPSR_03	ALT10
	GPIO12_IO04	GPIO_LPSR_04	ALT10
	GPIO12_IO05	GPIO_LPSR_05	ALT10
	GPIO12_IO06	GPIO_LPSR_06	ALT10
	GPIO12_IO07	GPIO_LPSR_07	ALT10
	GPIO12_IO08	GPIO_LPSR_08	ALT10
	GPIO12_IO09	GPIO_LPSR_09	ALT10
	GPIO12_IO10	GPIO_LPSR_10	ALT10
	GPIO12_IO11	GPIO_LPSR_11	ALT10
	GPIO12_IO12	GPIO_LPSR_12	ALT10
	GPIO12_IO13	GPIO_LPSR_13	ALT10
	GPIO12_IO14	GPIO_LPSR_14	ALT10
	GPIO12_IO15	GPIO_LPSR_15	ALT10
GPIO13[3]	GPIO13_IO00	WAKEUP	ALT5
	GPIO13_IO01	PMIC_ON_REQ	ALT5
	GPIO13_IO02	PMIC_STBY_REQ	ALT5
	GPIO13_IO03	GPIO_SNVS_00	ALT5
	GPIO13_IO04	GPIO_SNVS_01	ALT5

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPIO13_IO05	GPIO_SNVS_02	ALT5
	GPIO13_IO06	GPIO_SNVS_03	ALT5
	GPIO13_IO07	GPIO_SNVS_04	ALT5
	GPIO13_IO08	GPIO_SNVS_05	ALT5
	GPIO13_IO09	GPIO_SNVS_06	ALT5
	GPIO13_IO10	GPIO_SNVS_07	ALT5
	GPIO13_IO11	GPIO_SNVS_08	ALT5
	GPIO13_IO12	GPIO_SNVS_09	ALT5
GPT1	GPT1_CAPTURE1	GPIO_AD_12	ALT2
	GPT1_CAPTURE2	GPIO_AD_13	ALT2
	GPT1_CLK	GPIO_AD_17	ALT2
	GPT1_COMPARE1	GPIO_AD_14	ALT2
	GPT1_COMPARE2	GPIO_AD_15	ALT2
	GPT1_COMPARE3	GPIO_AD_16	ALT2
GPT2	GPT2_CAPTURE1	GPIO_AD_00	ALT3
	GPT2_CAPTURE2	GPIO_AD_01	ALT3
	GPT2_CLK	GPIO_AD_05	ALT3
	GPT2_COMPARE1	GPIO_AD_02	ALT3
	GPT2_COMPARE2	GPIO_AD_03	ALT3
	GPT2_COMPARE3	GPIO_AD_04	ALT3
GPT3	GPT3_CAPTURE1	GPIO_EMC_B2_06	ALT1
	GPT3_CAPTURE1	GPIO_AD_06	ALT3
	GPT3_CAPTURE2	GPIO_EMC_B2_07	ALT1
	GPT3_CAPTURE2	GPIO_AD_07	ALT3
	GPT3_CLK	GPIO_EMC_B2_05	ALT1
	GPT3_CLK	GPIO_AD_11	ALT3
	GPT3_COMPARE1	GPIO_EMC_B2_08	ALT1
	GPT3_COMPARE1	GPIO_AD_08	ALT3
	GPT3_COMPARE2	GPIO_EMC_B2_09	ALT1
	GPT3_COMPARE2	GPIO_AD_09	ALT3
	GPT3_COMPARE3	GPIO_EMC_B2_10	ALT1
	GPT3_COMPARE3	GPIO_AD_10	ALT3
GPT4	GPT4_CAPTURE1	GPIO_SD_B1_00	ALT3
	GPT4_CAPTURE2	GPIO_SD_B1_01	ALT3
	GPT4_CLK	GPIO_SD_B1_05	ALT3
	GPT4_COMPARE1	GPIO_SD_B1_02	ALT3
	GPT4_COMPARE2	GPIO_SD_B1_03	ALT3
	GPT4_COMPARE3	GPIO_SD_B1_04	ALT3
GPT5	GPT5_CAPTURE1	GPIO_EMC_B1_09	ALT2

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	GPT5_CAPTURE2	GPIO_EMC_B1_10	ALT2
	GPT5_CLK	GPIO_EMC_B1_14	ALT2
	GPT5_COMPARE1	GPIO_EMC_B1_11	ALT2
	GPT5_COMPARE2	GPIO_EMC_B1_12	ALT2
	GPT5_COMPARE3	GPIO_EMC_B1_13	ALT2
GPT6	GPT6_CAPTURE1	GPIO_SD_B2_06	ALT4
	GPT6_CAPTURE2	GPIO_SD_B2_07	ALT4
	GPT6_CLK	GPIO_SD_B2_11	ALT4
	GPT6_COMPARE1	GPIO_SD_B2_08	ALT4
	GPT6_COMPARE2	GPIO_SD_B2_09	ALT4
	GPT6_COMPARE3	GPIO_SD_B2_10	ALT4
JTAG	JTAG_MUX_MOD	GPIO_LPSR_13	ALT0
	JTAG_MUX_TCK (SWD)	GPIO_LPSR_14	ALT0
	JTAG_MUX_TDI	GPIO_LPSR_12	ALT0
	JTAG_MUX_TDO	GPIO_LPSR_11	ALT0
	JTAG_MUX_TMS (SWD)	GPIO_LPSR_15	ALT0
	JTAG_MUX_TRSTB	GPIO_LPSR_10	ALT0
KPP	KPP_COL0	GPIO_AD_35	ALT6
	KPP_COL1	GPIO_AD_33	ALT6
	KPP_COL2	GPIO_AD_31	ALT6
	KPP_COL3	GPIO_AD_29	ALT6
	KPP_COL4	GPIO_AD_27	ALT6
	KPP_COL5	GPIO_AD_25	ALT6
	KPP_COL6	GPIO_AD_23	ALT6
	KPP_COL6	GPIO_SD_B1_03	ALT8
	KPP_COL7	GPIO_AD_21	ALT6
	KPP_COL7	GPIO_SD_B1_01	ALT8
	KPP_ROW0	GPIO_AD_34	ALT6
	KPP_ROW1	GPIO_AD_32	ALT6
	KPP_ROW2	GPIO_AD_30	ALT6
	KPP_ROW3	GPIO_AD_28	ALT6
	KPP_ROW4	GPIO_AD_26	ALT6
	KPP_ROW5	GPIO_AD_24	ALT6
	KPP_ROW6	GPIO_AD_22	ALT6
	KPP_ROW6	GPIO_SD_B1_02	ALT8
	KPP_ROW7	GPIO_AD_20	ALT6
	KPP_ROW7	GPIO_SD_B1_00	ALT8
LPI2C1	LPI2C1_HREQ	GPIO_AD_12	ALT1
	LPI2C1_SCL	GPIO_AD_32	ALT0

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	LPI2C1_SCL	GPIO_AD_08	ALT1
	LPI2C1_SCLS	GPIO_AD_10	ALT1
	LPI2C1_SDA	GPIO_AD_33	ALT0
	LPI2C1_SDA	GPIO_AD_09	ALT1
	LPI2C1_SDAS	GPIO_AD_11	ALT1
LPI2C2	LPI2C2_SCL	GPIO_EMC_B2_00	ALT9
	LPI2C2_SCL	GPIO_AD_18	ALT9
	LPI2C2_SDA	GPIO_EMC_B2_01	ALT9
	LPI2C2_SDA	GPIO_AD_19	ALT9
LPI2C3	LPI2C3_SCL	GPIO_DISP_B1_02	ALT2
	LPI2C3_SCL	GPIO_DISP_B2_10	ALT6
	LPI2C3_SDA	GPIO_DISP_B1_03	ALT2
	LPI2C3_SDA	GPIO_DISP_B2_11	ALT6
LPI2C4	LPI2C4_SCL	GPIO_DISP_B2_12	ALT6
	LPI2C4_SCL	GPIO_AD_24	ALT9
	LPI2C4_SDA	GPIO_DISP_B2_13	ALT6
	LPI2C4_SDA	GPIO_AD_25	ALT9
LPI2C5	LPI2C5_HREQ	GPIO_LPSR_12	ALT6
	LPI2C5_SCL	GPIO_LPSR_05	ALT0
	LPI2C5_SCL	GPIO_LPSR_09	ALT6
	LPI2C5_SCLS	GPIO_LPSR_10	ALT6
	LPI2C5_SDA	GPIO_LPSR_04	ALT0
	LPI2C5_SDA	GPIO_LPSR_08	ALT6
	LPI2C5_SDAS	GPIO_LPSR_11	ALT6
LPI2C6	LPI2C6_SCL	GPIO_LPSR_07	ALT0
	LPI2C6_SCL	GPIO_LPSR_11	ALT2
	LPI2C6_SDA	GPIO_LPSR_06	ALT0
	LPI2C6_SDA	GPIO_LPSR_10	ALT2
LPSPI1	LPSPI1_PCS0	GPIO_AD_29	ALT0
	LPSPI1_PCS0	GPIO_EMC_B2_01	ALT8
	LPSPI1_PCS1	GPIO_AD_18	ALT2
	LPSPI1_PCS2	GPIO_AD_19	ALT2
	LPSPI1_PCS3	GPIO_AD_20	ALT2
	LPSPI1_SCK	GPIO_AD_28	ALT0
	LPSPI1_SCK	GPIO_EMC_B2_00	ALT8
	LPSPI1_SIN	GPIO_AD_31	ALT0
	LPSPI1_SIN	GPIO_EMC_B2_03	ALT8
	LPSPI1_SOUT	GPIO_AD_30	ALT0
	LPSPI1_SOUT	GPIO_EMC_B2_02	ALT8

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
LPSPI2	LPSPI2_PCS0	GPIO_AD_25	ALT1
	LPSPI2_PCS0	GPIO_SD_B2_08	ALT6
	LPSPI2_PCS1	GPIO_AD_21	ALT2
	LPSPI2_PCS1	GPIO_SD_B2_11	ALT6
	LPSPI2_PCS2	GPIO_AD_22	ALT2
	LPSPI2_PCS3	GPIO_AD_23	ALT2
	LPSPI2_SCK	GPIO_AD_24	ALT1
	LPSPI2_SCK	GPIO_SD_B2_07	ALT6
	LPSPI2_SIN	GPIO_AD_27	ALT1
	LPSPI2_SIN	GPIO_SD_B2_10	ALT6
	LPSPI2_SOUT	GPIO_AD_26	ALT1
	LPSPI2_SOUT	GPIO_SD_B2_09	ALT6
LPSPI3	LPSPI3_PCS0	GPIO_EMC_B2_05	ALT8
	LPSPI3_PCS0	GPIO_DISP_B1_07	ALT9
	LPSPI3_PCS1	GPIO_EMC_B2_08	ALT8
	LPSPI3_PCS1	GPIO_DISP_B1_08	ALT9
	LPSPI3_PCS2	GPIO_EMC_B2_09	ALT8
	LPSPI3_PCS2	GPIO_DISP_B1_09	ALT9
	LPSPI3_PCS3	GPIO_EMC_B2_10	ALT8
	LPSPI3_PCS3	GPIO_DISP_B1_10	ALT9
	LPSPI3_SCK	GPIO_EMC_B2_04	ALT8
	LPSPI3_SCK	GPIO_DISP_B1_04	ALT9
	LPSPI3_SIN	GPIO_EMC_B2_07	ALT8
	LPSPI3_SIN	GPIO_DISP_B1_05	ALT9
	LPSPI3_SOUT	GPIO_EMC_B2_06	ALT8
	LPSPI3_SOUT	GPIO_DISP_B1_06	ALT9
LPSPI4	LPSPI4_PCS0	GPIO_SD_B2_01	ALT4
	LPSPI4_PCS0	GPIO_DISP_B2_15	ALT9
	LPSPI4_PCS1	GPIO_SD_B2_04	ALT4
	LPSPI4_PCS2	GPIO_SD_B2_05	ALT4
	LPSPI4_PCS3	GPIO_SD_B2_06	ALT3
	LPSPI4_SCK	GPIO_SD_B2_00	ALT4
	LPSPI4_SCK	GPIO_DISP_B2_12	ALT9
	LPSPI4_SIN	GPIO_SD_B2_03	ALT4
	LPSPI4_SIN	GPIO_DISP_B2_13	ALT9
	LPSPI4_SOUT	GPIO_SD_B2_02	ALT4
	LPSPI4_SOUT	GPIO_DISP_B2_14	ALT9
LPSPI5	LPSPI5_PCS0	GPIO_LPSR_03	ALT1
	LPSPI5_PCS0	GPIO_LPSR_13	ALT8

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	LPSPI5_PCS1	GPIO_LPSR_06	ALT8
	LPSPI5_PCS2	GPIO_LPSR_07	ALT8
	LPSPI5_PCS3	GPIO_LPSR_08	ALT8
	LPSPI5_SCK	GPIO_LPSR_02	ALT1
	LPSPI5_SCK	GPIO_LPSR_12	ALT8
	LPSPI5_SIN	GPIO_LPSR_05	ALT1
	LPSPI5_SIN	GPIO_LPSR_15	ALT8
	LPSPI5_SOUT	GPIO_LPSR_04	ALT1
	LPSPI5_SOUT	GPIO_LPSR_14	ALT8
LPSPI6	LPSPI6_PCS0	GPIO_LPSR_09	ALT4
	LPSPI6_PCS1	GPIO_LPSR_08	ALT4
	LPSPI6_PCS2	GPIO_LPSR_07	ALT4
	LPSPI6_PCS3	GPIO_LPSR_06	ALT4
	LPSPI6_SCK	GPIO_LPSR_10	ALT4
	LPSPI6_SIN	GPIO_LPSR_12	ALT4
	LPSPI6_SOUT	GPIO_LPSR_11	ALT4
LPUART1	LPUART1_CTS_B	GPIO_AD_26	ALT0
	LPUART1_RTS_B	GPIO_AD_27	ALT0
	LPUART1_RXD	GPIO_AD_25	ALT0
	LPUART1_RXD	GPIO_DISP_B1_03	ALT9
	LPUART1_RXD	GPIO_DISP_B2_09	ALT9
	LPUART1_TXD	GPIO_AD_24	ALT0
	LPUART1_TXD	GPIO_DISP_B1_02	ALT9
	LPUART1_TXD	GPIO_DISP_B2_08	ALT9
LPUART2	LPUART2_CTS_B	GPIO_DISP_B2_12	ALT3
	LPUART2_RTS_B	GPIO_DISP_B2_13	ALT3
	LPUART2_RXD	GPIO_DISP_B2_11	ALT2
	LPUART2_TXD	GPIO_DISP_B2_10	ALT2
LPUART3	LPUART3_CTS_B	GPIO_SD_B2_07	ALT3
	LPUART3_RTS_B	GPIO_SD_B2_08	ALT3
	LPUART3_RXD	GPIO_AD_31	ALT4
	LPUART3_TXD	GPIO_AD_30	ALT4
LPUART4	LPUART4_CTS_B	GPIO_DISP_B1_05	ALT2
	LPUART4_RTS_B	GPIO_DISP_B1_07	ALT2
	LPUART4_RXD	GPIO_DISP_B1_04	ALT2
	LPUART4_TXD	GPIO_DISP_B1_06	ALT2
LPUART5	LPUART5_CTS_B	GPIO_SD_B2_09	ALT3
	LPUART5_RTS_B	GPIO_SD_B2_10	ALT3
	LPUART5_RXD	GPIO_AD_29	ALT1

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	LPUART5_TXD	GPIO_AD_28	ALT1
LPUART6	LPUART6_CTS_B	GPIO_EMC_B2_00	ALT3
	LPUART6_RTS_B	GPIO_EMC_B2_01	ALT3
	LPUART6_RXD	GPIO_EMC_B1_41	ALT3
	LPUART6_TXD	GPIO_EMC_B1_40	ALT3
LPUART7	LPUART7_CTS_B	GPIO_AD_02	ALT1
	LPUART7_RTS_B	GPIO_AD_03	ALT1
	LPUART7_RXD	GPIO_DISP_B2_07	ALT2
	LPUART7_RXD	GPIO_AD_01	ALT6
	LPUART7_TXD	GPIO_DISP_B2_06	ALT2
	LPUART7_TXD	GPIO_AD_00	ALT6
LPUART8	LPUART8_CTS_B	GPIO_AD_04	ALT1
	LPUART8_RTS_B	GPIO_AD_05	ALT1
	LPUART8_RXD	GPIO_DISP_B2_09	ALT2
	LPUART8_RXD	GPIO_AD_03	ALT6
	LPUART8_TXD	GPIO_DISP_B2_08	ALT2
	LPUART8_TXD	GPIO_AD_02	ALT6
LPUART9	LPUART9_CTS_B	GPIO_SD_B2_02	ALT3
	LPUART9_RTS_B	GPIO_SD_B2_03	ALT3
	LPUART9_RXD	GPIO_SD_B2_01	ALT3
	LPUART9_TXD	GPIO_SD_B2_00	ALT3
LPUART10	LPUART10_CTS_B	GPIO_AD_34	ALT8
	LPUART10_RTS_B	GPIO_AD_35	ALT8
	LPUART10_RXD	GPIO_AD_16	ALT1
	LPUART10_RXD	GPIO_AD_33	ALT8
	LPUART10_TXD	GPIO_AD_15	ALT1
	LPUART10_TXD	GPIO_AD_32	ALT8
LPUART11	LPUART11_CTS_B	GPIO_LPSR_10	ALT1
	LPUART11_RTS_B	GPIO_LPSR_11	ALT1
	LPUART11_RXD	GPIO_LPSR_09	ALT0
	LPUART11_RXD	GPIO_LPSR_05	ALT6
	LPUART11_TXD	GPIO_LPSR_08	ALT0
	LPUART11_TXD	GPIO_LPSR_04	ALT6
LPUART12	LPUART12_CTS_B	GPIO_LPSR_05	ALT3
	LPUART12_RTS_B	GPIO_LPSR_04	ALT3
	LPUART12_RXD	GPIO_LPSR_07	ALT3
	LPUART12_RXD	GPIO_LPSR_01	ALT6
	LPUART12_RXD	GPIO_LPSR_11	ALT8
	LPUART12_TXD	GPIO_LPSR_06	ALT3

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	LPUART12_TXD	GPIO_LPSR_00	ALT6
	LPUART12_TXD	GPIO_LPSR_10	ALT8
MIPI DSI	MIPI_DSI_DN0	MIPI_DSI_DN0	No muxing
	MIPI_DSI_DP0	MIPI_DSI_DP0	No muxing
	MIPI_DSI_DP1	MIPI_DSI_DP1	No muxing
	MIPI_DSI_DN1	MIPI_DSI_DN1	No muxing
	MIPI_DSI_CLKN	MIPI_DSI_CLKN	No muxing
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	No muxing
MIPI CSI2	MIPI_CSI_DP0	MIPI_CSI_DP0	No muxing
	MIPI_CSI_DN0	MIPI_CSI_DN0	No muxing
	MIPI_CSI_DP1	MIPI_CSI_DP1	No muxing
	MIPI_CSI_DN1	MIPI_CSI_DN1	No muxing
	MIPI_DSI_CLKN	MIPI_DSI_CLKN	No muxing
	MIPI_DSI_CLKP	MIPI_DSI_CLKP	No muxing
MQS	MQS_LEFT	GPIO_LPSR_01	ALT2
	MQS_LEFT	GPIO_EMC_B1_41	ALT2
	MQS_LEFT	GPIO_DISP_B2_01	ALT2
	MQS_LEFT	GPIO_LPSR_03	ALT3
	MQS_RIGHT	GPIO_LPSR_00	ALT2
	MQS_RIGHT	GPIO_EMC_B1_40	ALT2
	MQS_RIGHT	GPIO_DISP_B2_00	ALT2
	MQS_RIGHT	GPIO_LPSR_02	ALT3
PDM	PDM_DATA0	GPIO_LPSR_01	ALT1
	PDM_DATA0	GPIO_LPSR_09	ALT3
	PDM_DATA1	GPIO_LPSR_13	ALT1
	PDM_DATA1	GPIO_LPSR_10	ALT3
	PDM_DATA2	GPIO_LPSR_14	ALT1
	PDM_DATA2	GPIO_LPSR_11	ALT3
	PDM_DATA3	GPIO_LPSR_15	ALT1
	PDM_DATA3	GPIO_LPSR_12	ALT3
	PDM_CLK	GPIO_LPSR_00	ALT1
	PDM_CLK	GPIO_LPSR_08	ALT3
PIT1	PIT1_TRIGGER0	GPIO_AD_13	ALT1
	PIT1_TRIGGER0	GPIO_DISP_B2_15	ALT4
	PIT1_TRIGGER0	GPIO_EMC_B2_05	ALT9
	PIT1_TRIGGER1	GPIO_DISP_B2_04	ALT2
	PIT1_TRIGGER1	GPIO_EMC_B2_06	ALT9
	PIT1_TRIGGER2	GPIO_DISP_B2_03	ALT2
	PIT1_TRIGGER2	GPIO_EMC_B2_07	ALT9

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	PIT1_TRIGGER3	GPIO_DISP_B2_02	ALT2
	PIT1_TRIGGER3	GPIO_EMC_B2_08	ALT9
PIT2	PIT2_TRIGGER0	GPIO_LPSR_12	ALT1
	PIT2_TRIGGER0	GPIO_LPSR_09	ALT2
	PIT2_TRIGGER1	GPIO_LPSR_13	ALT2
	PIT2_TRIGGER1	GPIO_LPSR_08	ALT7
	PIT2_TRIGGER2	GPIO_LPSR_14	ALT2
	PIT2_TRIGGER2	GPIO_LPSR_07	ALT7
	PIT2_TRIGGER3	GPIO_LPSR_15	ALT2
	PIT2_TRIGGER3	GPIO_LPSR_06	ALT7
PGMC	PGMC_PMIC_READY	GPIO_AD_32	ALT2
	PGMC_PMIC_STBY_REQ	PMIC_STBY_REQ	ALT0
TMR1	TMR1_TIMER0	GPIO_EMC_B1_17	ALT2
	TMR1_TIMER0	GPIO_DISP_B1_00	ALT3
	TMR1_TIMER0	GPIO_EMC_B2_09	ALT9
	TMR1_TIMER1	GPIO_EMC_B1_38	ALT2
	TMR1_TIMER1	GPIO_DISP_B1_01	ALT3
	TMR1_TIMER1	GPIO_EMC_B2_10	ALT9
	TMR1_TIMER2	GPIO_DISP_B1_02	ALT3
	TMR1_TIMER2	GPIO_EMC_B2_11	ALT9
	TMR1_TIMER3	GPIO_EMC_B2_12	ALT9
TMR2	TMR2_TIMER0	GPIO_EMC_B1_18	ALT2
	TMR2_TIMER0	GPIO_DISP_B1_03	ALT3
	TMR2_TIMER0	GPIO_EMC_B2_13	ALT9
	TMR2_TIMER1	GPIO_EMC_B1_39	ALT2
	TMR2_TIMER1	GPIO_DISP_B1_04	ALT3
	TMR2_TIMER1	GPIO_EMC_B2_14	ALT9
	TMR2_TIMER2	GPIO_DISP_B1_05	ALT3
	TMR2_TIMER2	GPIO_EMC_B2_15	ALT9
	TMR2_TIMER3	GPIO_EMC_B2_16	ALT9
TMR3	TMR3_TIMER0	GPIO_EMC_B1_19	ALT2
	TMR3_TIMER0	GPIO_DISP_B1_06	ALT3
	TMR3_TIMER0	GPIO_EMC_B2_17	ALT9
	TMR3_TIMER1	GPIO_EMC_B2_00	ALT2
	TMR3_TIMER1	GPIO_DISP_B1_07	ALT3
	TMR3_TIMER1	GPIO_EMC_B2_18	ALT9
	TMR3_TIMER2	GPIO_DISP_B1_08	ALT3
	TMR3_TIMER2	GPIO_EMC_B2_19	ALT9
	TMR3_TIMER3	GPIO_EMC_B2_20	ALT9

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
TMR4	TMR4_TIMER0	GPIO_EMC_B1_20	ALT2
	TMR4_TIMER0	GPIO_DISP_B1_09	ALT3
	TMR4_TIMER0	GPIO_AD_04	ALT9
	TMR4_TIMER1	GPIO_EMC_B2_01	ALT2
	TMR4_TIMER1	GPIO_DISP_B1_10	ALT3
	TMR4_TIMER1	GPIO_AD_05	ALT9
	TMR4_TIMER2	GPIO_DISP_B1_11	ALT3
	TMR4_TIMER2	GPIO_AD_06	ALT9
	TMR4_TIMER3	GPIO_AD_07	ALT9
SAI1	SAI1_MCLK	GPIO_AD_17	ALT0
	SAI1_MCLK	GPIO_DISP_B2_03	ALT4
	SAI1_RX_BCLK	GPIO_AD_19	ALT0
	SAI1_RX_BCLK	GPIO_DISP_B2_05	ALT4
	SAI1_RX_DATA0	GPIO_AD_20	ALT0
	SAI1_RX_DATA0	GPIO_DISP_B2_06	ALT4
	SAI1_RX_SYNC	GPIO_AD_18	ALT0
	SAI1_RX_SYNC	GPIO_DISP_B2_04	ALT4
	SAI1_TX_BCLK	GPIO_AD_22	ALT0
	SAI1_TX_BCLK	GPIO_DISP_B2_08	ALT4
	SAI1_TX_DATA0	GPIO_AD_21	ALT0
	SAI1_TX_DATA0	GPIO_DISP_B2_07	ALT4
	SAI1_TX_DATA1 and SAI1_RX_DATA3	GPIO_DISP_B2_02	ALT4
	SAI1_TX_DATA2 and SAI1_RX_DATA2	GPIO_DISP_B2_01	ALT4
	SAI1_TX_DATA3 and SAI1_RX_DATA1	GPIO_DISP_B2_00	ALT4
	SAI1_TX_SYNC	GPIO_AD_23	ALT0
	SAI1_TX_SYNC	GPIO_DISP_B2_09	ALT4
SAI2	SAI2_MCLK	GPIO_EMC_B2_04	ALT2
	SAI2_RX_BCLK	GPIO_EMC_B2_06	ALT2
	SAI2_RX_DATA	GPIO_EMC_B2_07	ALT2
	SAI2_RX_SYNC	GPIO_EMC_B2_05	ALT2
	SAI2_TX_BCLK	GPIO_EMC_B2_09	ALT2
	SAI2_TX_DATA	GPIO_EMC_B2_08	ALT2
	SAI2_TX_SYNC	GPIO_EMC_B2_10	ALT2
SAI3	SAI3_MCLK	GPIO_EMC_B2_17	ALT3
	SAI3_RX_BCLK	GPIO_EMC_B2_12	ALT3
	SAI3_RX_DATA	GPIO_EMC_B2_13	ALT3
	SAI3_RX_SYNC	GPIO_EMC_B2_11	ALT3

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	SAI3_TX_BCLK	GPIO_EMC_B2_15	ALT3
	SAI3_TX_DATA	GPIO_EMC_B2_14	ALT3
	SAI3_TX_SYNC	GPIO_EMC_B2_16	ALT3
SAI4	SAI4_MCLK	GPIO_LPSR_05	ALT2
	SAI4_MCLK	GPIO_LPSR_00	ALT7
	SAI4_RX_BCLK	GPIO_LPSR_07	ALT2
	SAI4_RX_BCLK	GPIO_LPSR_14	ALT7
	SAI4_RX_DATA	GPIO_LPSR_06	ALT2
	SAI4_RX_DATA	GPIO_LPSR_13	ALT7
	SAI4_RX_SYNC	GPIO_LPSR_08	ALT2
	SAI4_RX_SYNC	GPIO_LPSR_15	ALT7
	SAI4_TX_BCLK	GPIO_LPSR_04	ALT2
	SAI4_TX_BCLK	GPIO_LPSR_12	ALT7
	SAI4_TX_DATA	GPIO_LPSR_02	ALT2
	SAI4_TX_DATA	GPIO_LPSR_09	ALT7
	SAI4_TX_SYNC	GPIO_LPSR_03	ALT2
	SAI4_TX_SYNC	GPIO_LPSR_10	ALT7
SEMC	SEMC_ADDR00	GPIO_EMC_B1_09	ALT0
	SEMC_ADDR01	GPIO_EMC_B1_10	ALT0
	SEMC_ADDR02	GPIO_EMC_B1_11	ALT0
	SEMC_ADDR03	GPIO_EMC_B1_12	ALT0
	SEMC_ADDR04	GPIO_EMC_B1_13	ALT0
	SEMC_ADDR05	GPIO_EMC_B1_14	ALT0
	SEMC_ADDR06	GPIO_EMC_B1_15	ALT0
	SEMC_ADDR07	GPIO_EMC_B1_16	ALT0
	SEMC_ADDR08	GPIO_EMC_B1_17	ALT0
	SEMC_ADDR09	GPIO_EMC_B1_18	ALT0
	SEMC_ADDR10	GPIO_EMC_B1_23	ALT0
	SEMC_ADDR11	GPIO_EMC_B1_19	ALT0
	SEMC_ADDR12	GPIO_EMC_B1_20	ALT0
	SEMC_BA0	GPIO_EMC_B1_21	ALT0
	SEMC_BA1	GPIO_EMC_B1_22	ALT0
	SEMC_CAS	GPIO_EMC_B1_24	ALT0
	SEMC_CKE	GPIO_EMC_B1_27	ALT0
	SEMC_CLK	GPIO_EMC_B1_26	ALT0
	SEMC_CLKX0	GPIO_EMC_B2_19	ALT0
	SEMC_CLKX1	GPIO_EMC_B2_20	ALT0
	SEMC_CS0	GPIO_EMC_B1_29	ALT0
	SEMC_CSX0	GPIO_EMC_B1_41	ALT0

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	SEMC_CSX1	GPIO_AD_26	ALT2
	SEMC_CSX2	GPIO_AD_27	ALT2
	SEMC_CSX3	GPIO_AD_28	ALT2
	SEMC_DATA00	GPIO_EMC_B1_00	ALT0
	SEMC_DATA01	GPIO_EMC_B1_01	ALT0
	SEMC_DATA10	GPIO_EMC_B1_32	ALT0
	SEMC_DATA11	GPIO_EMC_B1_33	ALT0
	SEMC_DATA12	GPIO_EMC_B1_34	ALT0
	SEMC_DATA13	GPIO_EMC_B1_35	ALT0
	SEMC_DATA14	GPIO_EMC_B1_36	ALT0
	SEMC_DATA15	GPIO_EMC_B1_37	ALT0
	SEMC_DATA16	GPIO_EMC_B2_00	ALT0
	SEMC_DATA17	GPIO_EMC_B2_01	ALT0
	SEMC_DATA18	GPIO_EMC_B2_02	ALT0
	SEMC_DATA19	GPIO_EMC_B2_03	ALT0
	SEMC_DATA02	GPIO_EMC_B1_02	ALT0
	SEMC_DATA20	GPIO_EMC_B2_04	ALT0
	SEMC_DATA21	GPIO_EMC_B2_05	ALT0
	SEMC_DATA22	GPIO_EMC_B2_06	ALT0
	SEMC_DATA23	GPIO_EMC_B2_07	ALT0
	SEMC_DATA24	GPIO_EMC_B2_09	ALT0
	SEMC_DATA25	GPIO_EMC_B2_10	ALT0
	SEMC_DATA26	GPIO_EMC_B2_11	ALT0
	SEMC_DATA27	GPIO_EMC_B2_12	ALT0
	SEMC_DATA28	GPIO_EMC_B2_13	ALT0
	SEMC_DATA29	GPIO_EMC_B2_14	ALT0
	SEMC_DATA03	GPIO_EMC_B1_03	ALT0
	SEMC_DATA30	GPIO_EMC_B2_15	ALT0
	SEMC_DATA31	GPIO_EMC_B2_16	ALT0
	SEMC_DATA04	GPIO_EMC_B1_04	ALT0
	SEMC_DATA05	GPIO_EMC_B1_05	ALT0
	SEMC_DATA06	GPIO_EMC_B1_06	ALT0
	SEMC_DATA07	GPIO_EMC_B1_07	ALT0
	SEMC_DATA08	GPIO_EMC_B1_30	ALT0
	SEMC_DATA09	GPIO_EMC_B1_31	ALT0
	SEMC_DM0	GPIO_EMC_B1_08	ALT0
	SEMC_DM1	GPIO_EMC_B1_38	ALT0
	SEMC_DM2	GPIO_EMC_B2_08	ALT0
	SEMC_DM3	GPIO_EMC_B2_17	ALT0

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	SEMC_DQS	GPIO_EMC_B1_39	ALT0
	SEMC_DQS4	GPIO_EMC_B2_18	ALT0
	SEMC_RAS	GPIO_EMC_B1_25	ALT0
	SEMC_RDY	GPIO_EMC_B1_40	ALT0
	SEMC_WE	GPIO_EMC_B1_28	ALT0
SNVS LP[4]	SNVS_PMIC_ON_REQ	PMIC_ON_REQ	ALT0
	SNVS_TAMPER1	GPIO_SNVS_00	ALT0
	SNVS_TAMPER2	GPIO_SNVS_01	ALT0
	SNVS_TAMPER3	GPIO_SNVS_02	ALT0
	SNVS_TAMPER4	GPIO_SNVS_03	ALT0
	SNVS_TAMPER5	GPIO_SNVS_04	ALT0
	SNVS_TAMPER6	GPIO_SNVS_05	ALT0
	SNVS_TAMPER7	GPIO_SNVS_06	ALT0
	SNVS_TAMPER8	GPIO_SNVS_07	ALT0
	SNVS_TAMPER9	GPIO_SNVS_08	ALT0
	SNVS_TAMPER10	GPIO_SNVS_09	ALT0
SPDIF	SPDIF_EXT_CLK	GPIO_AD_14	ALT0
	SPDIF_IN	GPIO_AD_15	ALT0
	SPDIF_IN	GPIO_EMC_B2_11	ALT1
	SPDIF_IN	GPIO_DISP_B2_10	ALT9
	SPDIF_LOCK	GPIO_AD_12	ALT0
	SPDIF_OUT	GPIO_AD_16	ALT0
	SPDIF_OUT	GPIO_EMC_B2_12	ALT1
	SPDIF_OUT	GPIO_DISP_B2_11	ALT9
	SPDIF_SR_CLK	GPIO_AD_13	ALT0
SRC	SRC_BOOT_MODE0	GPIO_LPSR_02	ALT0
	SRC_BOOT_MODE1	GPIO_LPSR_03	ALT0
	SRC_BT_CFG0	GPIO_DISP_B1_06	ALT6
	SRC_BT_CFG1	GPIO_DISP_B1_07	ALT6
	SRC_BT_CFG10	GPIO_DISP_B2_04	ALT6
	SRC_BT_CFG11	GPIO_DISP_B2_05	ALT6
	SRC_BT_CFG2	GPIO_DISP_B1_08	ALT6
	SRC_BT_CFG3	GPIO_DISP_B1_09	ALT6
	SRC_BT_CFG4	GPIO_DISP_B1_10	ALT6
	SRC_BT_CFG5	GPIO_DISP_B1_11	ALT6
	SRC_BT_CFG6	GPIO_DISP_B2_00	ALT6
	SRC_BT_CFG7	GPIO_DISP_B2_01	ALT6
	SRC_BT_CFG8	GPIO_DISP_B2_02	ALT6
	SRC_BT_CFG9	GPIO_DISP_B2_03	ALT6

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	SRC_POR_B	POR_B	ALT0
	SRC_RESET_B	ONOFF	ALT0
USB OTG1	USB_OTG1_OC	GPIO_AD_11	ALT0
		GPIO_AD_35	ALT1
	USB_OTG1_PWR	GPIO_AD_10	ALT0
		GPIO_AD_34	ALT1
	USB_OTG1_ID	GPIO_AD_09	ALT0
		GPIO_AD_33	ALT1
	USB_OTG1_DN	USB1_DN	No muxing
	USB_OTG1_DP	USB1_DP	No muxing
	USB_OTG1_VBUS	USB1_VBUS	No muxing
USB OTG2	USB_OTG2_OC	GPIO_AD_06	ALT0
		GPIO_AD_30	ALT1
	USB_OTG2_PWR	GPIO_AD_07	ALT0
		GPIO_AD_31	ALT1
	USB_OTG2_ID	GPIO_AD_08	ALT0
		GPIO_AD_32	ALT1
	USB_OTG2_DN	USB2_DN	No muxing
	USB_OTG2_VBUS	USB2_VBUS	No muxing
	USB_OTG2_DP	USB2_DP	No muxing
USDHC1	USDHC1_CD_B	GPIO_DISP_B1_08	ALT2
		GPIO_AD_32	ALT4
	USDHC1_CLK	GPIO_SD_B1_01	ALT0
	USDHC1_CMD	GPIO_SD_B1_00	ALT0
	USDHC1_DATA0	GPIO_SD_B1_02	ALT0
	USDHC1_DATA1	GPIO_SD_B1_03	ALT0
	USDHC1_DATA2	GPIO_SD_B1_04	ALT0
	USDHC1_DATA3	GPIO_SD_B1_05	ALT0
	USDHC1_RESET_B	GPIO_DISP_B1_10	ALT2
		GPIO_AD_35	ALT4
	USDHC1_VSELECT	GPIO_DISP_B2_01	ALT1
		GPIO_AD_34	ALT4
	USDHC1_WP	GPIO_DISP_B1_09	ALT2
		GPIO_AD_33	ALT4
USDHC2	USDHC2_CD_B	GPIO_EMC_B2_01	ALT1
		GPIO_AD_26	ALT11
	USDHC2_CLK	GPIO_SD_B2_04	ALT0
	USDHC2_CMD	GPIO_SD_B2_05	ALT0
	USDHC2_DATA0	GPIO_SD_B2_03	ALT0

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	USDHC2_DATA1	GPIO_SD_B2_02	ALT0
	USDHC2_DATA2	GPIO_SD_B2_01	ALT0
	USDHC2_DATA3	GPIO_SD_B2_00	ALT0
	USDHC2_DATA4	GPIO_SD_B2_08	ALT0
	USDHC2_DATA5	GPIO_SD_B2_09	ALT0
	USDHC2_DATA6	GPIO_SD_B2_10	ALT0
	USDHC2_DATA7	GPIO_SD_B2_11	ALT0
	USDHC2_RESET_B	GPIO_SD_B2_06	ALT0
		GPIO_EMC_B2_04	ALT1
		GPIO_AD_29	ALT11
	USDHC2_STROBE	GPIO_SD_B2_07	ALT0
	USDHC2_VSELECT	GPIO_EMC_B2_03	ALT1
		GPIO_AD_28	ALT11
	USDHC2_WP	GPIO_EMC_B2_02	ALT1
		GPIO_AD_27	ALT11
VIDEO MUX (CSI)	VIDEO_MUX_CSI_DATA00	GPIO_AD_24	ALT2
	VIDEO_MUX_CSI_DATA01	GPIO_AD_25	ALT2
	VIDEO_MUX_CSI_DATA10	GPIO_AD_11	ALT4
	VIDEO_MUX_CSI_DATA11	GPIO_AD_10	ALT4
	VIDEO_MUX_CSI_DATA12	GPIO_AD_09	ALT4
	VIDEO_MUX_CSI_DATA13	GPIO_AD_08	ALT4
	VIDEO_MUX_CSI_DATA14	GPIO_AD_07	ALT4
	VIDEO_MUX_CSI_DATA15	GPIO_AD_06	ALT4
	VIDEO_MUX_CSI_DATA16	GPIO_EMC_B2_09	ALT3
	VIDEO_MUX_CSI_DATA17	GPIO_EMC_B2_08	ALT3
	VIDEO_MUX_CSI_DATA18	GPIO_EMC_B2_07	ALT3
	VIDEO_MUX_CSI_DATA19	GPIO_EMC_B2_06	ALT3
	VIDEO_MUX_CSI_DATA02	GPIO_AD_23	ALT4
	VIDEO_MUX_CSI_DATA20	GPIO_EMC_B2_05	ALT3
	VIDEO_MUX_CSI_DATA21	GPIO_EMC_B2_04	ALT3
	VIDEO_MUX_CSI_DATA22	GPIO_EMC_B2_03	ALT3
	VIDEO_MUX_CSI_DATA23	GPIO_EMC_B2_02	ALT3
	VIDEO_MUX_CSI_DATA03	GPIO_AD_22	ALT4
	VIDEO_MUX_CSI_DATA04	GPIO_AD_21	ALT4
	VIDEO_MUX_CSI_DATA05	GPIO_AD_20	ALT4
	VIDEO_MUX_CSI_DATA06	GPIO_AD_19	ALT4
	VIDEO_MUX_CSI_DATA07	GPIO_AD_18	ALT4
	VIDEO_MUX_CSI_DATA08	GPIO_AD_17	ALT4
	VIDEO_MUX_CSI_DATA09	GPIO_AD_16	ALT4

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	VIDEO_MUX_CSI_FIELD	GPIO_EMC_B2_10	ALT3
	VIDEO_MUX_CSI_HSYNC	GPIO_AD_15	ALT4
	VIDEO_MUX_CSI_MCLK	GPIO_AD_13	ALT4
	VIDEO_MUX_CSI_PIXCLK	GPIO_AD_12	ALT4
	VIDEO_MUX_CSI_VSYNC	GPIO_AD_14	ALT4
VIDEO MUX DCIC(MIPI DSI)	VIDEO_MUX_EXT_DCIC1	GPIO_DISP_B2_14	ALT3
		GPIO_AD_02	ALT9
		GPIO_AD_28	ALT9
VIDEO MUX DCIC (Parallel LCDIF)	VIDEO_MUX_EXT_DCIC2	GPIO_DISP_B2_15	ALT3
		GPIO_AD_03	ALT9
		GPIO_AD_29	ALT9
VIDEO MUX (LCDIF)	VIDEO_MUX_LCDIF_DOTCLK	GPIO_DISP_B1_00	ALT0
	VIDEO_MUX_LCDIF_DATA00	GPIO_DISP_B1_04	ALT0
	VIDEO_MUX_LCDIF_DATA01	GPIO_DISP_B1_05	ALT0
	VIDEO_MUX_LCDIF_DATA10	GPIO_DISP_B2_02	ALT0
	VIDEO_MUX_LCDIF_DATA11	GPIO_DISP_B2_03	ALT0
	VIDEO_MUX_LCDIF_DATA12	GPIO_DISP_B2_04	ALT0
	VIDEO_MUX_LCDIF_DATA13	GPIO_DISP_B2_05	ALT0
	VIDEO_MUX_LCDIF_DATA14	GPIO_DISP_B2_06	ALT0
	VIDEO_MUX_LCDIF_DATA15	GPIO_DISP_B2_07	ALT0
	VIDEO_MUX_LCDIF_DATA16	GPIO_DISP_B2_08	ALT0
	VIDEO_MUX_LCDIF_DATA17	GPIO_DISP_B2_09	ALT0
	VIDEO_MUX_LCDIF_DATA18	GPIO_DISP_B2_10	ALT0
	VIDEO_MUX_LCDIF_DATA19	GPIO_DISP_B2_11	ALT0
	VIDEO_MUX_LCDIF_DATA02	GPIO_DISP_B1_06	ALT0
	VIDEO_MUX_LCDIF_DATA20	GPIO_DISP_B2_12	ALT0
	VIDEO_MUX_LCDIF_DATA21	GPIO_DISP_B2_13	ALT0

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	VIDEO_MUX_LCDIF_DATA22	GPIO_DISP_B2_14	ALT0
	VIDEO_MUX_LCDIF_DATA23	GPIO_DISP_B2_15	ALT0
	VIDEO_MUX_LCDIF_DATA03	GPIO_DISP_B1_07	ALT0
	VIDEO_MUX_LCDIF_DATA04	GPIO_DISP_B1_08	ALT0
	VIDEO_MUX_LCDIF_DATA05	GPIO_DISP_B1_09	ALT0
	VIDEO_MUX_LCDIF_DATA06	GPIO_DISP_B1_10	ALT0
	VIDEO_MUX_LCDIF_DATA07	GPIO_DISP_B1_11	ALT0
	VIDEO_MUX_LCDIF_DATA08	GPIO_DISP_B2_00	ALT0
	VIDEO_MUX_LCDIF_DATA09	GPIO_DISP_B2_01	ALT0
	VIDEO_MUX_LCDIF_ENABLE	GPIO_DISP_B1_01	ALT0
	VIDEO_MUX_LCDIF_HSYNC	GPIO_DISP_B1_02	ALT0
	VIDEO_MUX_LCDIF_VSYNC	GPIO_DISP_B1_03	ALT0
WDOG1	WDOG1_ANY	GPIO_EMC_B2_17	ALT8
	WDOG1_ANY	GPIO_AD_34	ALT9
	WDOG1_B	GPIO_DISP_B2_00	ALT1
	WDOG1_B	GPIO_DISP_B2_15	ALT2
	WDOG1_B	GPIO_AD_04	ALT6
	WDOG1_B	GPIO_EMC_B2_18	ALT8
	WDOG1_RESET_B_DEB	GPIO_DISP_B2_11	ALT3
	WDOG1_RESET_B_DEB	GPIO_AD_31	ALT9
WDOG2	WDOG2_B	GPIO_DISP_B2_14	ALT2
	WDOG2_B	GPIO_DISP_B2_01	ALT3
	WDOG2_B	GPIO_AD_05	ALT6
	WDOG2_RESET_B_DEB	GPIO_DISP_B2_10	ALT3
	WDOG2_RESET_B_DEB	GPIO_AD_30	ALT9
XBAR	XBAR_INOUT10	GPIO_EMC_B1_36	ALT1
	XBAR_INOUT11	GPIO_EMC_B1_37	ALT1
	XBAR_INOUT12	GPIO_EMC_B1_40	ALT1
	XBAR_INOUT13	GPIO_EMC_B1_41	ALT1
	XBAR_INOUT14	GPIO_EMC_B2_16	ALT1
	XBAR_INOUT15	GPIO_EMC_B2_17	ALT1
	XBAR_INOUT16	GPIO_EMC_B2_18	ALT1

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	XBAR_INOUT17	GPIO_AD_33	ALT2
	XBAR_INOUT18	GPIO_AD_34	ALT2
	XBAR_INOUT19	GPIO_AD_35	ALT2
	XBAR_INOUT20	GPIO_SD_B1_00	ALT2
	XBAR_INOUT20	GPIO_EMC_B2_00	ALT6
	XBAR_INOUT21	GPIO_SD_B1_01	ALT2
	XBAR_INOUT21	GPIO_EMC_B2_01	ALT6
	XBAR_INOUT22	GPIO_SD_B1_02	ALT2
	XBAR_INOUT22	GPIO_EMC_B2_02	ALT6
	XBAR_INOUT23	GPIO_SD_B1_03	ALT2
	XBAR_INOUT23	GPIO_EMC_B2_03	ALT6
	XBAR_INOUT24	GPIO_SD_B1_04	ALT2
	XBAR_INOUT24	GPIO_EMC_B2_04	ALT6
	XBAR_INOUT25	GPIO_SD_B1_05	ALT2
	XBAR_INOUT25	GPIO_EMC_B2_05	ALT6
	XBAR_INOUT26	GPIO_DISP_B1_00	ALT4
	XBAR_INOUT26	GPIO_EMC_B2_06	ALT6
	XBAR_INOUT27	GPIO_DISP_B1_01	ALT4
	XBAR_INOUT27	GPIO_EMC_B2_07	ALT6
	XBAR_INOUT28	GPIO_DISP_B1_02	ALT4
	XBAR_INOUT28	GPIO_EMC_B2_08	ALT6
	XBAR_INOUT29	GPIO_DISP_B1_03	ALT4
	XBAR_INOUT29	GPIO_EMC_B2_09	ALT6
	XBAR_INOUT30	GPIO_DISP_B1_04	ALT4
	XBAR_INOUT30	GPIO_EMC_B2_10	ALT6
	XBAR_INOUT31	GPIO_DISP_B1_05	ALT4
	XBAR_INOUT31	GPIO_EMC_B2_11	ALT6
	XBAR_INOUT32	GPIO_DISP_B1_06	ALT4
	XBAR_INOUT32	GPIO_EMC_B2_12	ALT6
	XBAR_INOUT33	GPIO_DISP_B1_07	ALT4
	XBAR_INOUT33	GPIO_EMC_B2_13	ALT6
	XBAR_INOUT34	GPIO_DISP_B1_08	ALT4
	XBAR_INOUT34	GPIO_EMC_B2_14	ALT6
	XBAR_INOUT35	GPIO_DISP_B1_09	ALT4
	XBAR_INOUT35	GPIO_EMC_B2_15	ALT6
	XBAR_INOUT36	GPIO_DISP_B1_10	ALT4
	XBAR_INOUT37	GPIO_DISP_B1_11	ALT4
	XBAR_INOUT38	GPIO_DISP_B2_10	ALT4
	XBAR_INOUT39	GPIO_DISP_B2_11	ALT4

Table continues on the next page...

Table 11-1. Muxing Options (continued)

Instance	Port	Pad	Mode
	XBAR_INOUT04	GPIO_EMC_B1_12	ALT1
	XBAR_INOUT40	GPIO_DISP_B2_12	ALT4
	XBAR_INOUT05	GPIO_EMC_B1_13	ALT1
	XBAR_INOUT06	GPIO_EMC_B1_14	ALT1
	XBAR_INOUT07	GPIO_EMC_B1_15	ALT1
	XBAR_INOUT08	GPIO_EMC_B1_16	ALT1
	XBAR_INOUT09	GPIO_EMC_B1_35	ALT1
XTALOSC	XTALI	XTALI	No muxing
	XTALO	XTALO	No muxing
	REF_CLK_32K	GPIO_AD_13	ALT9
	REF_CLK_24M	GPIO_AD_14	ALT1

1. This GPIO instance contains a mux that selects between normal GPIO or CM7 fast GPIO. See IOMUXC_GPR_GPR40-41 for more details.
2. This GPIO instance contains a mux that selects between normal GPIO or CM7 fast GPIO. See IOMUXC_GPR_GPR42-43 for more details.
3. For GPIO13_IO3-GPIO13_IO12, GPIO functions are not available on devices that support tamper.
4. Tamper is only available on select part numbers. Refer to the Datasheet for details.

11.1.2 Pin Assignments

Table 11-2. Pin Assignments

Pin Name	Pin Assignments	Pad Settings
POR_B	ALT0 - SRC_POR_B	PUS - Weak PU PUE - Pull Enable DSE - High drive strength SRE - Fast slew rate
ONOFF	ALT0 - SRC_RESET_B	PUS - Weak PU PUE - Pull Enable DSE - High drive strength SRE - Fast slew rate
WAKEUP	ALT5 - GPIO13_IO0 ALT7 - NMI_GLUE_NMI	ODE - Disabled PUS - Weak PU PUE - Pull Enable DSE - High drive strength SRE - Fast slew rate
PMIC_ON_REQ	ALT0 - SNVS_LP_PMIC_ON_REQ ALT5 - GPIO13_IO1	ODE - Disabled PUS - Weak PU PUE - Pull Disable

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
		DSE - High drive strength SRE - Fast slew rate
PMIC_STBY_REQ	ALT0 - CCM_PMIC_VSTBY_REQ ALT5 - GPIO13_IO2	ODE - Disabled PUS - Weak PU PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_00[1]	ALT0 - SNVS_LP_TAMPER1 ALT5 - GPIO13_IO3	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_01	ALT0 - SNVS_LP_TAMPER2 ALT5 - GPIO13_IO4	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_02	ALT0 - SNVS_LP_TAMPER3 ALT5 - GPIO13_IO5	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_03	ALT0 - SNVS_LP_TAMPER4 ALT5 - GPIO13_IO6	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_04	ALT0 - SNVS_LP_TAMPER5 ALT5 - GPIO13_IO7	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_05	ALT0 - SNVS_LP_TAMPER6 ALT5 - GPIO13_IO8	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_06	ALT0 - SNVS_LP_TAMPER7	ODE - Disabled

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT5 - GPIO13_IO9	PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_07	ALT0 - SNVS_LP_TAMPER8 ALT5 - GPIO13_IO10	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_08	ALT0 - SNVS_LP_TAMPER9 ALT5 - GPIO13_IO11	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_SNVS_09	ALT0 - SNVS_LP_TAMPER10 ALT5 - GPIO13_IO12	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_LPSR_00	ALT0 - CAN3_TX ALT1 - PDM_CLK ALT2 - MQS_RIGHT ALT3 - CM4_EVENTO ALT5 - GPIO6_IO0 ALT6 - LPUART12_TX ALT7 - SAI4_MCLK ALT10 - GPIO12_IO0	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_LPSR_01	ALT0 - CAN3_RX ALT1 - PDM_DATA0 ALT2 - MQS_LEFT ALT3 - CM4_EVENTI ALT5 - GPIO6_IO1 ALT6 - LPUART12_RX ALT10 - GPIO12_IO1	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength SRE - Fast slew rate
GPIO_LPSR_02	ALT0 - SRC_BOOT_MODE0 ALT1 - LPSPI5_SCK ALT2 - SAI4_TX_DATA ALT3 - MQS_RIGHT	ODE - Disabled PUS - Weak PD PUE - Pull Disable DSE - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT5 - GPIO6_IO2	SRE - Fast slew rate
	ALT10 - GPIO12_IO2	
GPIO_LPSR_03	ALT0 - SRC_BOOT_MODE1	ODE - Disabled
	ALT1 - LPSPI5_PCS0	PUS - Weak PD
	ALT2 - SAI4_TX_SYNC	PUE - Pull Disable
	ALT3 - MQS_LEFT	DSE - High drive strength
	ALT5 - GPIO6_IO3	SRE - Fast slew rate
	ALT10 - GPIO12_IO3	
GPIO_LPSR_04	ALT0 - LPI2C5_SDA	ODE - Disabled
	ALT1 - LPSPI5_SDO	PUS - Weak PD
	ALT2 - SAI4_TX_BCLK	PUE - Pull Disable
	ALT3 - LPUART12_RTS_B	DSE - High drive strength
	ALT5 - GPIO6_IO4	SRE - Fast slew rate
	ALT6 - LPUART11_TX	
	ALT10 - GPIO12_IO4	
GPIO_LPSR_05	ALT0 - LPI2C5_SCL	ODE - Disabled
	ALT1 - LPSPI5_SDI	PUS - Weak PD
	ALT2 - SAI4_MCLK	PUE - Pull Disable
	ALT3 - LPUART12_CTS_B	DSE - High drive strength
	ALT5 - GPIO6_IO5	SRE - Fast slew rate
	ALT6 - LPUART11_RX	
	ALT7 - NMI_GLUE_NMI	
	ALT10 - GPIO12_IO5	
GPIO_LPSR_06	ALT0 - LPI2C6_SDA	ODE - Disabled
	ALT2 - SAI4_RX_DATA	PUS - Weak PD
	ALT3 - LPUART12_TX	PUE - Pull Disable
	ALT4 - LPSPI6_PCS3	DSE - High drive strength
	ALT5 - GPIO6_IO6	SRE - Fast slew rate
	ALT6 - CAN3_TX	
	ALT7 - PIT2_TRIGGER3	
	ALT8 - LPSPI5_PCS1	
	ALT10 - GPIO12_IO6	
GPIO_LPSR_07	ALT0 - LPI2C6_SCL	ODE - Disabled
	ALT2 - SAI4_RX_BCLK	PUS - Weak PD
	ALT3 - LPUART12_RX	PUE - Pull Disable
	ALT4 - LPSPI6_PCS2	DSE - High drive strength
	ALT5 - GPIO6_IO7	SRE - Fast slew rate
	ALT6 - CAN3_RX	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT7 - PIT2_TRIGGER2	
	ALT8 - LPSPI5_PCS2	
	ALT10 - GPIO12_IO7	
GPIO_LPSR_08	ALT0 - LPUART11_TX	ODE - Disabled
	ALT1 - CAN3_TX	PUS - Weak PD
	ALT2 - SAI4_RX_SYNC	PUE - Pull Disable
	ALT3 - PDM_CLK	DSE - High drive strength
	ALT4 - LPSPI6_PCS1	SRE - Fast slew rate
	ALT5 - GPIO6_IO8	
	ALT6 - LPI2C5_SDA	
	ALT7 - PIT2_TRIGGER1	
	ALT8 - LPSPI5_PCS3	
	ALT10 - GPIO12_IO8	
GPIO_LPSR_09	ALT0 - LPUART11_RX	ODE - Disabled
	ALT1 - CAN3_RX	PUS - Weak PD
	ALT2 - PIT2_TRIGGER0	PUE - Pull Disable
	ALT3 - PDM_DATA0	DSE - High drive strength
	ALT4 - LPSPI6_PCS0	SRE - Fast slew rate
	ALT5 - GPIO6_IO9	
	ALT6 - LPI2C5_SCL	
	ALT7 - SAI4_TX_DATA	
	ALT10 - GPIO12_IO9	
GPIO_LPSR_10	ALT0 - JTAG_MUX_TRSTB	ODE - Disabled
	ALT1 - LPUART11_CTS_B	PUS - Weak PU
	ALT2 - LPI2C6_SDA	PUE - Pull Enable
	ALT3 - PDM_DATA1	DSE - High drive strength
	ALT4 - LPSPI6_SCK	SRE - Fast slew rate
	ALT5 - GPIO6_IO10	
	ALT6 - LPI2C5_SCLS	
	ALT7 - SAI4_TX_SYNC	
	ALT8 - LPUART12_TX	
	ALT10 - GPIO12_IO10	
GPIO_LPSR_11	ALT0 - JTAG_MUX_TDO	ODE - Disabled
	ALT1 - LPUART11_RTS_B	PUS - Weak PD
	ALT2 - LPI2C6_SCL	PUE - Pull Disable
	ALT3 - PDM_DATA2	DSE - High drive strength
	ALT4 - LPSPI6_SDO	SRE - Fast slew rate
	ALT5 - GPIO6_IO11	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT6 - LPI2C5_SDAS	
	ALT7 - CSSYS_TRACE_SWO	
	ALT8 - LPUART12_RX	
	ALT10 - GPIO12_IO11	
GPIO_LPSR_12	ALT0 - JTAG_MUX_TDI	ODE - Disabled
	ALT1 - PIT2_TRIGGER0	PUS - Weak PU
	ALT3 - PDM_DATA3	PUE - Pull Enable
	ALT4 - LPSPI6_SDI	DSE - High drive strength
	ALT5 - GPIO6_IO12	SRE - Fast slew rate
	ALT6 - LPI2C5_HREQ	
	ALT7 - SAI4_TX_BCLK	
	ALT8 - LPSPI5_SCK	
	ALT10 - GPIO12_IO12	
GPIO_LPSR_13	ALT0 - JTAG_MUX_MOD	ODE - Disabled
	ALT1 - PDM_DATA1	PUS - Weak PD
	ALT2 - PIT2_TRIGGER1	PUE - Pull Disable
	ALT5 - GPIO6_IO13	DSE - High drive strength
	ALT7 - SAI4_RX_DATA	SRE - Fast slew rate
	ALT8 - LPSPI5_PCS0	
	ALT10 - GPIO12_IO13	
GPIO_LPSR_14	ALT0 - JTAG_MUX_TCK	ODE - Disabled
	ALT1 - PDM_DATA2	PUS - Weak PD
	ALT2 - PIT2_TRIGGER2	PUE - Pull Enable
	ALT5 - GPIO6_IO14	DSE - High drive strength
	ALT7 - SAI4_RX_BCLK	SRE - Fast slew rate
	ALT8 - LPSPI5_SDO	
	ALT10 - GPIO12_IO14	
GPIO_LPSR_15	ALT0 - JTAG_MUX_TMS	ODE - Disabled
	ALT1 - PDM_DATA3	PUS - Weak PU
	ALT2 - PIT2_TRIGGER3	PUE - Pull Enable
	ALT5 - GPIO6_IO15	DSE - High drive strength
	ALT7 - SAI4_RX_SYNC	SRE - Fast slew rate
	ALT8 - LPSPI5_SDI	
	ALT10 - GPIO12_IO15	
GPIO_EMC_B1_00	ALT0 - SEMC_DATA0	ODE - Disabled
	ALT1 - FLEXPWM4_PWMA0	PULL - PD
	ALT5 - GPIO1_IO0	PDRV - High drive strength
	ALT7 - USBPHY1_TSTI_TX_LS_MODE	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - FLEXIO1_FLEXIO0	
	ALT10 - GPIO7_IO0	
GPIO_EMC_B1_01	ALT0 - SEMC_DATA1	ODE - Disabled
	ALT1 - FLEXPWM4_PWMB0	PULL - PD
	ALT5 - GPIO1_IO1	PDRV - High drive strength
	ALT7 - USBPHY1_TSTI_TX_HS_MODE	
	ALT8 - FLEXIO1_FLEXIO1	
	ALT10 - GPIO7_IO1	
GPIO_EMC_B1_02	ALT0 - SEMC_DATA2	ODE - Disabled
	ALT1 - FLEXPWM4_PWMA1	PULL - PD
	ALT5 - GPIO1_IO2	PDRV - High drive strength
	ALT7 - USBPHY1_TSTI_TX_DN	
	ALT8 - FLEXIO1_FLEXIO2	
	ALT10 - GPIO7_IO2	
GPIO_EMC_B1_03	ALT0 - SEMC_DATA3	ODE - Disabled
	ALT1 - FLEXPWM4_PWMB1	PULL - PD
	ALT5 - GPIO1_IO3	PDRV - High drive strength
	ALT7 - USBPHY1_TSTO_RX_SQUELCH	
	ALT8 - FLEXIO1_FLEXIO3	
	ALT10 - GPIO7_IO3	
GPIO_EMC_B1_04	ALT0 - SEMC_DATA4	ODE - Disabled
	ALT1 - FLEXPWM4_PWMA2	PULL - PD
	ALT5 - GPIO1_IO4	PDRV - High drive strength
	ALT7 - USBPHY1_TSTO_RX_DISCON_DET	
	ALT8 - FLEXIO1_FLEXIO4	
	ALT10 - GPIO7_IO4	
GPIO_EMC_B1_05	ALT0 - SEMC_DATA5	ODE - Disabled
	ALT1 - FLEXPWM4_PWMB2	PULL - PD
	ALT5 - GPIO1_IO5	PDRV - High drive strength
	ALT7 - USBPHY1_TSTO_RX_HS_RXD	
	ALT8 - FLEXIO1_FLEXIO5	
	ALT10 - GPIO7_IO5	
GPIO_EMC_B1_06	ALT0 - SEMC_DATA6	ODE - Disabled
	ALT1 - FLEXPWM2_PWMA0	PULL - PD
	ALT5 - GPIO1_IO6	PDRV - High drive strength
	ALT7 - USBPHY2_TSTO_RX_FS_RXD	
	ALT8 - FLEXIO1_FLEXIO6	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT10 - GPIO7_IO6	
GPIO_EMC_B1_07	ALT0 - SEMC_DATA7 ALT1 - FLEXPWM2_PWMB0 ALT5 - GPIO1_IO7 ALT7 - USBPHY1_TSTO_RX_FS_RXD ALT8 - FLEXIO1_FLEXIO7 ALT10 - GPIO7_IO7	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_08	ALT0 - SEMC_DM0 ALT1 - FLEXPWM2_PWMA1 ALT5 - GPIO1_IO8 ALT7 - USBPHY1_TSTI_TX_DP ALT8 - FLEXIO1_FLEXIO8 ALT10 - GPIO7_IO8	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_09	ALT0 - SEMC_ADDR0 ALT1 - FLEXPWM2_PWMB1 ALT2 - GPT5_CAPTURE1 ALT5 - GPIO1_IO9 ALT7 - USBPHY1_TSTI_TX_EN ALT8 - FLEXIO1_FLEXIO9 ALT10 - GPIO7_IO9	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_10	ALT0 - SEMC_ADDR1 ALT1 - FLEXPWM2_PWMA2 ALT2 - GPT5_CAPTURE2 ALT5 - GPIO1_IO10 ALT7 - USBPHY1_TSTI_TX_HIZ ALT8 - FLEXIO1_FLEXIO10 ALT10 - GPIO7_IO10	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_11	ALT0 - SEMC_ADDR2 ALT1 - FLEXPWM2_PWMB2 ALT2 - GPT5_COMPARE1 ALT5 - GPIO1_IO11 ALT7 - USBPHY2_TSTO_RX_HS_RXD ALT8 - FLEXIO1_FLEXIO11 ALT10 - GPIO7_IO11	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_12	ALT0 - SEMC_ADDR3 ALT1 - XBAR1_XBAR_INOUT4 ALT2 - GPT5_COMPARE2 ALT5 - GPIO1_IO12	ODE - Disabled PULL - PD PDRV - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT7 - USBPHY1_TSTO_PLL_CLK20DIV ALT8 - FLEXIO1_FLEXIO12 ALT10 - GPIO7_IO12	
GPIO_EMC_B1_13	ALT0 - SEMC_ADDR4 ALT1 - XBAR1_XBAR_INOUT5 ALT2 - GPT5_COMPARE3 ALT5 - GPIO1_IO13 ALT7 - USBPHY2_TSTO_PLL_CLK20DIV ALT8 - FLEXIO1_FLEXIO13 ALT10 - GPIO7_IO13	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_14	ALT0 - SEMC_ADDR5 ALT1 - XBAR1_XBAR_INOUT6 ALT2 - GPT5_CLK ALT5 - GPIO1_IO14 ALT7 - USBPHY2_TSTO_RX_SQUELCH ALT8 - FLEXIO1_FLEXIO14 ALT10 - GPIO7_IO14	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_15	ALT0 - SEMC_ADDR6 ALT1 - XBAR1_XBAR_INOUT7 ALT5 - GPIO1_IO15 ALT7 - USBPHY2_TSTO_RX_DISCON_DET ALT8 - FLEXIO1_FLEXIO15 ALT10 - GPIO7_IO15	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_16	ALT0 - SEMC_ADDR7 ALT1 - XBAR1_XBAR_INOUT8 ALT5 - GPIO1_IO16 ALT7 - ANATOP_WB_TSTO ALT8 - FLEXIO1_FLEXIO16 ALT10 - GPIO7_IO16	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_17	ALT0 - SEMC_ADDR8 ALT1 - FLEXPWM4_PWMA3 ALT2 - QTIMER1_TIMER0 ALT5 - GPIO1_IO17 ALT7 - ANATOP_WB_OK ALT8 - FLEXIO1_FLEXIO17 ALT10 - GPIO7_IO17	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_18	ALT0 - SEMC_ADDR9	ODE - Disabled

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT1 - FLEXPWM4_PWMB3	PULL - PD
	ALT2 - QTIMER2_TIMER0	PDRV - High drive strength
	ALT5 - GPIO1_IO18	
	ALT8 - FLEXIO1_FLEXIO18	
	ALT10 - GPIO7_IO18	
GPIO_EMC_B1_19	ALT0 - SEMC_ADDR11	ODE - Disabled
	ALT1 - FLEXPWM2_PWMA3	PULL - PD
	ALT2 - QTIMER3_TIMER0	PDRV - High drive strength
	ALT5 - GPIO1_IO19	
	ALT8 - FLEXIO1_FLEXIO19	
	ALT10 - GPIO7_IO19	
GPIO_EMC_B1_20	ALT0 - SEMC_ADDR12	ODE - Disabled
	ALT1 - FLEXPWM2_PWMB3	PULL - PD
	ALT2 - QTIMER4_TIMER0	PDRV - High drive strength
	ALT5 - GPIO1_IO20	
	ALT8 - FLEXIO1_FLEXIO20	
	ALT10 - GPIO7_IO20	
GPIO_EMC_B1_21	ALT0 - SEMC_BA0	ODE - Disabled
	ALT1 - FLEXPWM3_PWMA3	PULL - PD
	ALT5 - GPIO1_IO21	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO21	
	ALT10 - GPIO7_IO21	
GPIO_EMC_B1_22	ALT0 - SEMC_BA1	ODE - Disabled
	ALT1 - FLEXPWM3_PWMB3	PULL - PD
	ALT5 - GPIO1_IO22	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO22	
	ALT10 - GPIO7_IO22	
GPIO_EMC_B1_23	ALT0 - SEMC_ADDR10	ODE - Disabled
	ALT1 - FLEXPWM1_PWMA0	PULL - PD
	ALT5 - GPIO1_IO23	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO23	
	ALT10 - GPIO7_IO23	
GPIO_EMC_B1_24	ALT0 - SEMC_CAS	ODE - Disabled
	ALT1 - FLEXPWM1_PWMB0	PULL - PD
	ALT5 - GPIO1_IO24	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO24	
	ALT10 - GPIO7_IO24	
GPIO_EMC_B1_25	ALT0 - SEMC_RAS	ODE - Disabled

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT1 - FLEXPWM1_PWMA1	PULL - PD
	ALT5 - GPIO1_IO25	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO25	
	ALT10 - GPIO7_IO25	
GPIO_EMC_B1_26	ALT0 - SEMC_CLK	ODE - Disabled
	ALT1 - FLEXPWM1_PWMB1	PULL - PD
	ALT5 - GPIO1_IO26	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO26	
	ALT10 - GPIO7_IO26	
GPIO_EMC_B1_27	ALT0 - SEMC_CKE	ODE - Disabled
	ALT1 - FLEXPWM1_PWMA2	PULL - PD
	ALT5 - GPIO1_IO27	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO27	
	ALT10 - GPIO7_IO27	
GPIO_EMC_B1_28	ALT0 - SEMC_WE	ODE - Disabled
	ALT1 - FLEXPWM1_PWMB2	PULL - PD
	ALT5 - GPIO1_IO28	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO28	
	ALT10 - GPIO7_IO28	
GPIO_EMC_B1_29	ALT0 - SEMC_CS0	ODE - Disabled
	ALT1 - FLEXPWM3_PWMA0	PULL - PD
	ALT5 - GPIO1_IO29	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO29	
	ALT10 - GPIO7_IO29	
GPIO_EMC_B1_30	ALT0 - SEMC_DATA8	ODE - Disabled
	ALT1 - FLEXPWM3_PWMB0	PULL - PD
	ALT5 - GPIO1_IO30	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO30	
	ALT10 - GPIO7_IO30	
GPIO_EMC_B1_31	ALT0 - SEMC_DATA9	ODE - Disabled
	ALT1 - FLEXPWM3_PWMA1	PULL - PD
	ALT5 - GPIO1_IO31	PDRV - High drive strength
	ALT8 - FLEXIO1_FLEXIO31	
	ALT10 - GPIO7_IO31	
GPIO_EMC_B1_32	ALT0 - SEMC_DATA10	ODE - Disabled
	ALT1 - FLEXPWM3_PWMB1	PULL - PD
	ALT5 - GPIO_MUX2_IO0	PDRV - High drive strength
	ALT10 - GPIO8_IO0	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
GPIO_EMC_B1_33	ALT0 - SEMC_DATA11 ALT1 - FLEXPWM3_PWMA2 ALT5 - GPIO_MUX2_IO1 ALT10 - GPIO8_IO1	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_34	ALT0 - SEMC_DATA12 ALT1 - FLEXPWM3_PWMB2 ALT5 - GPIO_MUX2_IO2 ALT10 - GPIO8_IO2	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_35	ALT0 - SEMC_DATA13 ALT1 - XBAR1_XBAR_INOUT9 ALT5 - GPIO_MUX2_IO3 ALT10 - GPIO8_IO3	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_36	ALT0 - SEMC_DATA14 ALT1 - XBAR1_XBAR_INOUT10 ALT5 - GPIO_MUX2_IO4 ALT10 - GPIO8_IO4	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_37	ALT0 - SEMC_DATA15 ALT1 - XBAR1_XBAR_INOUT11 ALT5 - GPIO_MUX2_IO5 ALT7 - SRC_SYSTEM_RESET ALT10 - GPIO8_IO5	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_38	ALT0 - SEMC_DM1 ALT1 - FLEXPWM1_PWMA3 ALT2 - QTIMER1_TIMER1 ALT5 - GPIO_MUX2_IO6 ALT7 - SRC_EARLY_RESET ALT10 - GPIO8_IO6	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_39	ALT0 - SEMC_DQS ALT1 - FLEXPWM1_PWMB3 ALT2 - QTIMER2_TIMER1 ALT5 - GPIO_MUX2_IO7 ALT7 - SRC_WAKEUPMIX_RESET ALT10 - GPIO8_IO7	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B1_40	ALT0 - SEMC_RDY ALT1 - XBAR1_XBAR_INOUT12 ALT2 - MQS_RIGHT ALT3 - LPUART6_TX ALT5 - GPIO_MUX2_IO8	ODE - Disabled PULL - PD PDRV - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT7 - ENET_1G_MDC	
	ALT9 - CCM_CLKO1	
	ALT10 - GPIO8_IO8	
GPIO_EMC_B1_41	ALT0 - SEMC_CSX0	ODE - Disabled
	ALT1 - XBAR1_XBAR_INOUT13	PULL - PD
	ALT2 - MQS_LEFT	PDRV - High drive strength
	ALT3 - LPUART6_RX	
	ALT4 - FLEXSPI2_B_DATA7	
	ALT5 - GPIO_MUX2_IO9	
	ALT7 - ENET_1G_MDIO	
	ALT9 - CCM_CLKO2	
	ALT10 - GPIO8_IO9	
GPIO_EMC_B2_00	ALT0 - SEMC_DATA16	ODE - Disabled
	ALT1 - CCM_ENET_REF_CLK_25M	PULL - PD
	ALT2 - QTIMER3_TIMER1	PDRV - High drive strength
	ALT3 - LPUART6_CTS_B	
	ALT4 - FLEXSPI2_B_DATA6	
	ALT5 - GPIO_MUX2_IO10	
	ALT6 - XBAR1_XBAR_INOUT20	
	ALT7 - ENET_QOS_1588_EVENT1_OUT	
	ALT8 - LPSPI1_SCK	
	ALT9 - LPI2C2_SCL	
	ALT10 - GPIO8_IO10	
	ALT11 - FLEXPWM3_PWMA0	
GPIO_EMC_B2_01	ALT0 - SEMC_DATA17	ODE - Disabled
	ALT1 - USDHC2_CD_B	PULL - PD
	ALT2 - QTIMER4_TIMER1	PDRV - High drive strength
	ALT3 - LPUART6_RTS_B	
	ALT4 - FLEXSPI2_B_DATA5	
	ALT5 - GPIO_MUX2_IO11	
	ALT6 - XBAR1_XBAR_INOUT21	
	ALT7 - ENET_QOS_1588_EVENT1_IN	
	ALT8 - LPSPI1_PCS0	
	ALT9 - LPI2C2_SDA	
	ALT10 - GPIO8_IO11	
	ALT11 - FLEXPWM3_PWMB0	
GPIO_EMC_B2_02	ALT0 - SEMC_DATA18	ODE - Disabled
	ALT1 - USDHC2_WP	PULL - PD

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT3 - VIDEO_MUX_CSI_DATA23	PDRV - High drive strength
	ALT4 - FLEXSPI2_B_DATA4	
	ALT5 - GPIO_MUX2_IO12	
	ALT6 - XBAR1_XBAR_INOUT22	
	ALT7 - ENET_QOS_1588_EVENT1_AUX_IN	
	ALT8 - LPSPI1_SDO	
	ALT10 - GPIO8_IO12	
	ALT11 - FLEXPWM3_PWMA1	
GPIO_EMC_B2_03	ALT0 - SEMC_DATA19	ODE - Disabled
	ALT1 - USDHC2_VSELECT	PULL - PD
	ALT3 - VIDEO_MUX_CSI_DATA22	PDRV - High drive strength
	ALT4 - FLEXSPI2_B_DATA3	
	ALT5 - GPIO_MUX2_IO13	
	ALT6 - XBAR1_XBAR_INOUT23	
	ALT7 - ENET_1G_TDATA3	
	ALT8 - LPSPI1_SDI	
	ALT10 - GPIO8_IO13	
	ALT11 - FLEXPWM3_PWMB1	
GPIO_EMC_B2_04	ALT0 - SEMC_DATA20	ODE - Disabled
	ALT1 - USDHC2_RESET_B	PULL - PD
	ALT2 - SAI2_MCLK	PDRV - High drive strength
	ALT3 - VIDEO_MUX_CSI_DATA21	
	ALT4 - FLEXSPI2_B_DATA2	
	ALT5 - GPIO_MUX2_IO14	
	ALT6 - XBAR1_XBAR_INOUT24	
	ALT7 - ENET_1G_TDATA2	
	ALT8 - LPSPI3_SCK	
	ALT10 - GPIO8_IO14	
	ALT11 - FLEXPWM3_PWMA2	
GPIO_EMC_B2_05	ALT0 - SEMC_DATA21	ODE - Disabled
	ALT1 - GPT3_CLK	PULL - PD
	ALT2 - SAI2_RX_SYNC	PDRV - High drive strength
	ALT3 - VIDEO_MUX_CSI_DATA20	
	ALT4 - FLEXSPI2_B_DATA1	
	ALT5 - GPIO_MUX2_IO15	
	ALT6 - XBAR1_XBAR_INOUT25	
	ALT7 - ENET_1G_RX_CLK	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - LPSPI3_PCS0	
	ALT9 - PIT1_TRIGGER0	
	ALT10 - GPIO8_IO15	
	ALT11 - FLEXPWM3_PWMB2	
GPIO_EMC_B2_06	ALT0 - SEMC_DATA22	ODE - Disabled
	ALT1 - GPT3_CAPTURE1	PULL - PD
	ALT2 - SAI2_RX_BCLK	PDRV - High drive strength
	ALT3 - VIDEO_MUX_CSI_DATA19	
	ALT4 - FLEXSPI2_B_DATA0	
	ALT5 - GPIO_MUX2_IO16	
	ALT6 - XBAR1_XBAR_INOUT26	
	ALT7 - ENET_1G_TX_ER	
	ALT8 - LPSPI3_SDO	
	ALT9 - PIT1_TRIGGER1	
	ALT10 - GPIO8_IO16	
	ALT11 - FLEXPWM3_PWMA3	
GPIO_EMC_B2_07	ALT0 - SEMC_DATA23	ODE - Disabled
	ALT1 - GPT3_CAPTURE2	PULL - PD
	ALT2 - SAI2_RX_DATA	PDRV - High drive strength
	ALT3 - VIDEO_MUX_CSI_DATA18	
	ALT4 - FLEXSPI2_B_DQS	
	ALT5 - GPIO_MUX2_IO17	
	ALT6 - XBAR1_XBAR_INOUT27	
	ALT7 - ENET_1G_RDATA3	
	ALT8 - LPSPI3_SDI	
	ALT9 - PIT1_TRIGGER2	
	ALT10 - GPIO8_IO17	
	ALT11 - FLEXPWM3_PWMB3	
GPIO_EMC_B2_08	ALT0 - SEMC_DM2	ODE - Disabled
	ALT1 - GPT3_COMPARE1	PULL - PU
	ALT2 - SAI2_TX_DATA	PDRV - High drive strength
	ALT3 - VIDEO_MUX_CSI_DATA17	
	ALT4 - FLEXSPI2_B_SS0_B	
	ALT5 - GPIO_MUX2_IO18	
	ALT6 - XBAR1_XBAR_INOUT28	
	ALT7 - ENET_1G_RDATA2	
	ALT8 - LPSPI3_PCS1	
	ALT9 - PIT1_TRIGGER3	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT10 - GPIO8_IO18	
GPIO_EMC_B2_09	ALT0 - SEMC_DATA24 ALT1 - GPT3_COMPARE2 ALT2 - SAI2_TX_BCLK ALT3 - VIDEO_MUX_CSI_DATA16 ALT4 - FLEXSPI2_B_SCLK ALT5 - GPIO_MUX2_IO19 ALT6 - XBAR1_XBAR_INOUT29 ALT7 - ENET_1G_CRS ALT8 - LPSPI3_PCS2 ALT9 - QTIMER1_TIMER0 ALT10 - GPIO8_IO19	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B2_10	ALT0 - SEMC_DATA25 ALT1 - GPT3_COMPARE3 ALT2 - SAI2_TX_SYNC ALT3 - VIDEO_MUX_CSI_FIELD ALT4 - FLEXSPI2_A_SCLK ALT5 - GPIO_MUX2_IO20 ALT6 - XBAR1_XBAR_INOUT30 ALT7 - ENET_1G_COL ALT8 - LPSPI3_PCS3 ALT9 - QTIMER1_TIMER1 ALT10 - GPIO8_IO20	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_EMC_B2_11	ALT0 - SEMC_DATA26 ALT1 - SPDIF_IN ALT2 - ENET_1G_TDATA0 ALT3 - SAI3_RX_SYNC ALT4 - FLEXSPI2_A_SS0_B ALT5 - GPIO_MUX2_IO21 ALT6 - XBAR1_XBAR_INOUT31 ALT8 - SIM1_TRXD ALT9 - QTIMER1_TIMER2 ALT10 - GPIO8_IO21	ODE - Disabled PULL - PU PDRV - High drive strength
GPIO_EMC_B2_12	ALT0 - SEMC_DATA27 ALT1 - SPDIF_OUT ALT2 - ENET_1G_TDATA1 ALT3 - SAI3_RX_BCLK ALT4 - FLEXSPI2_A_DQS	ODE - Disabled PULL - PD PDRV - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT5 - GPIO_MUX2_IO22	
	ALT6 - XBAR1_XBAR_INOUT32	
	ALT8 - SIM1_CLK	
	ALT9 - QTIMER1_TIMER3	
	ALT10 - GPIO8_IO22	
GPIO_EMC_B2_13	ALT0 - SEMC_DATA28	ODE - Disabled
	ALT2 - ENET_1G_TX_EN	PULL - PD
	ALT3 - SAI3_RX_DATA	PDRV - High drive strength
	ALT4 - FLEXSPI2_A_DATA0	
	ALT5 - GPIO_MUX2_IO23	
	ALT6 - XBAR1_XBAR_INOUT33	
	ALT8 - SIM1_RST_B	
	ALT9 - QTIMER2_TIMER0	
	ALT10 - GPIO8_IO23	
GPIO_EMC_B2_14	ALT0 - SEMC_DATA29	ODE - Disabled
	ALT2 - ENET_1G_TX_CLK_IO	PULL - PD
	ALT3 - SAI3_TX_DATA	PDRV - High drive strength
	ALT4 - FLEXSPI2_A_DATA1	
	ALT5 - GPIO_MUX2_IO24	
	ALT6 - XBAR1_XBAR_INOUT34	
	ALT8 - SIM1_SVEN	
	ALT9 - QTIMER2_TIMER1	
	ALT10 - GPIO8_IO24	
GPIO_EMC_B2_15	ALT0 - SEMC_DATA30	ODE - Disabled
	ALT2 - ENET_1G_RDATA0	PULL - PD
	ALT3 - SAI3_TX_BCLK	PDRV - High drive strength
	ALT4 - FLEXSPI2_A_DATA2	
	ALT5 - GPIO_MUX2_IO25	
	ALT6 - XBAR1_XBAR_INOUT35	
	ALT8 - SIM1_PD	
	ALT9 - QTIMER2_TIMER2	
	ALT10 - GPIO8_IO25	
GPIO_EMC_B2_16	ALT0 - SEMC_DATA31	ODE - Disabled
	ALT1 - XBAR1_XBAR_INOUT14	PULL - PD
	ALT2 - ENET_1G_RDATA1	PDRV - High drive strength
	ALT3 - SAI3_TX_SYNC	
	ALT4 - FLEXSPI2_A_DATA3	
	ALT5 - GPIO_MUX2_IO26	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - SIM1_POWER_FAIL	
	ALT9 - QTIMER2_TIMER3	
	ALT10 - GPIO8_IO26	
GPIO_EMC_B2_17	ALT0 - SEMC_DM3	ODE - Disabled
	ALT1 - XBAR1_XBAR_INOUT15	PULL - PD
	ALT2 - ENET_1G_RX_EN	PDRV - High drive strength
	ALT3 - SAI3_MCLK	
	ALT4 - FLEXSPI2_A_DATA4	
	ALT5 - GPIO_MUX2_IO27	
	ALT8 - WDOG1_WDOG_ANY	
	ALT9 - QTIMER3_TIMER0	
	ALT10 - GPIO8_IO27	
GPIO_EMC_B2_18	ALT0 - SEMC_DQS4	ODE - Disabled
	ALT1 - XBAR1_XBAR_INOUT16	PULL - PD
	ALT2 - ENET_1G_RX_ER	PDRV - High drive strength
	ALT3 - EWM_EWM_OUT_B	
	ALT4 - FLEXSPI2_A_DATA5	
	ALT5 - GPIO_MUX2_IO28	
	ALT6 - FLEXSPI1_A_DQS	
	ALT8 - WDOG1_WDOG_B	
	ALT9 - QTIMER3_TIMER1	
	ALT10 - GPIO8_IO28	
GPIO_EMC_B2_19	ALT0 - SEMC_CLKX0	ODE - Disabled
	ALT1 - ENET_MDC	PULL - PD
	ALT2 - ENET_1G_MDC	PDRV - High drive strength
	ALT3 - ENET_1G_REF_CLK1	
	ALT4 - FLEXSPI2_A_DATA6	
	ALT5 - GPIO_MUX2_IO29	
	ALT8 - ENET_QOS_MDC	
	ALT9 - QTIMER3_TIMER2	
	ALT10 - GPIO8_IO29	
GPIO_EMC_B2_20	ALT0 - SEMC_CLKX1	ODE - Disabled
	ALT1 - ENET_MDIO	PULL - PD
	ALT2 - ENET_1G_MDIO	PDRV - High drive strength
	ALT3 - ENET_QOS_REF_CLK1	
	ALT4 - FLEXSPI2_A_DATA7	
	ALT5 - GPIO_MUX2_IO30	
	ALT8 - ENET_QOS_MDIO	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT9 - QTIMER3_TIMER3	
	ALT10 - GPIO8_IO30	
GPIO_AD_00	ALT0 - SIM1_TRXD	ODE - Disabled
	ALT1 - CAN2_TX	PUS - Weak PU
	ALT2 - ENET_1G_1588_EVENT1_IN	PUE - Pull Enable
	ALT3 - GPT2_CAPTURE1	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMA0	SRE - Fast slew rate
	ALT5 - GPIO_MUX2_IO31	
	ALT6 - LPUART7_TX	
	ALT8 - FLEXIO2_FLEXIO0	
	ALT9 - FLEXSPI2_B_SS1_B	
	ALT10 - GPIO8_IO31	
	ANALOG IO - ACMP1_1	
GPIO_AD_01	ALT0 - SIM1_CLK	ODE - Disabled
	ALT1 - CAN2_RX	PUS - Weak PU
	ALT2 - ENET_1G_1588_EVENT1_OUT	PUE - Pull Enable
	ALT3 - GPT2_CAPTURE2	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMB0	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO0	
	ALT6 - LPUART7_RX	
	ALT8 - FLEXIO2_FLEXIO1	
	ALT9 - FLEXSPI2_A_SS1_B	
	ALT10 - GPIO9_IO0	
	ANALOG IO - ACMP1_2	
GPIO_AD_02	ALT0 - SIM1_RST_B	ODE - Disabled
	ALT1 - LPUART7_CTS_B	PUS - Weak PD
	ALT2 - ENET_1G_1588_EVENT2_IN	PUE - Pull Enable
	ALT3 - GPT2_COMPARE1	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMA1	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO1	
	ALT6 - LPUART8_TX	
	ALT8 - FLEXIO2_FLEXIO2	
	ALT9 - VIDEO_MUX_EXT_DCIC1	
	ALT10 - GPIO9_IO1	
	ANALOG IO - ACMP1_3	
GPIO_AD_03	ALT0 - SIM1_SVEN	ODE - Disabled
	ALT1 - LPUART7_RTS_B	PUS - Weak PD
	ALT2 - ENET_1G_1588_EVENT2_OUT	PUE - Pull Enable

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT3 - GPT2_COMPARE2	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMB1	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO2	
	ALT6 - LPUART8_RX	
	ALT8 - FLEXIO2_FLEXIO3	
	ALT9 - VIDEO_MUX_EXT_DCIC2	
	ALT10 - GPIO9_IO2	
	ANALOG IO - ACMP1_4	
GPIO_AD_04	ALT0 - SIM1_PD	ODE - Disabled
	ALT1 - LPUART8_CTS_B	PUS - Weak PD
	ALT2 - ENET_1G_1588_EVENT3_IN	PUE - Pull Enable
	ALT3 - GPT2_COMPARE3	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMA2	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO3	
	ALT6 - WDOG1_WDOG_B	
	ALT8 - FLEXIO2_FLEXIO4	
	ALT9 - QTIMER4_TIMER0	
	ALT10 - GPIO9_IO3	
	ANALOG IO - ACMP2_1	
GPIO_AD_05	ALT0 - SIM1_POWER_FAIL	ODE - Disabled
	ALT1 - LPUART8_RTS_B	PUS - Weak PD
	ALT2 - ENET_1G_1588_EVENT3_OUT	PUE - Pull Enable
	ALT3 - GPT2_CLK	DSE - High drive strength
	ALT4 - FLEXPWM1_PWMB2	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO4	
	ALT6 - WDOG2_WDOG_B	
	ALT8 - FLEXIO2_FLEXIO5	
	ALT9 - QTIMER4_TIMER1	
	ALT10 - GPIO9_IO4	
	ANALOG IO - ACMP2_2	
GPIO_AD_06	ALT0 - USB_OTG2_OC	ODE - Disabled
	ALT1 - CAN1_TX	PUS - Weak PD
	ALT2 - SIM2_TRXD	PUE - Pull Enable
	ALT3 - GPT3_CAPTURE1	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA15	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO5	
	ALT6 - ENET_1588_EVENT1_IN	
	ALT8 - FLEXIO2_FLEXIO6	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT9 - QTIMER4_TIMER2	
	ALT10 - GPIO9_IO5	
	ALT11 - FLEXPWM1_PWMX0	
	ANALOG IO - ADC1_A0	
GPIO_AD_07	ALT0 - USB_OTG2_PWR	ODE - Disabled
	ALT1 - CAN1_RX	PUS - Weak PD
	ALT2 - SIM2_CLK	PUE - Pull Enable
	ALT3 - GPT3_CAPTURE2	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA14	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO6	
	ALT6 - ENET_1588_EVENT1_OUT	
	ALT8 - FLEXIO2_FLEXIO7	
	ALT9 - QTIMER4_TIMER3	
	ALT10 - GPIO9_IO6	
	ALT11 - FLEXPWM1_PWMX1	
	ANALOG IO - ADC1_B0	
GPIO_AD_08	ALT0 - USBPHY2_OTG_ID	ODE - Disabled
	ALT1 - LPI2C1_SCL	PUS - Weak PD
	ALT2 - SIM2_RST_B	PUE - Pull Enable
	ALT3 - GPT3_COMPARE1	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA13	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO7	
	ALT6 - ENET_1588_EVENT2_IN	
	ALT8 - FLEXIO2_FLEXIO8	
	ALT10 - GPIO9_IO7	
	ALT11 - FLEXPWM1_PWMX2	
	ANALOG IO - ADC1_A1	
GPIO_AD_09	ALT0 - USBPHY1_OTG_ID	ODE - Disabled
	ALT1 - LPI2C1_SDA	PUS - Weak PD
	ALT2 - SIM2_SVEN	PUE - Pull Enable
	ALT3 - GPT3_COMPARE2	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA12	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO8	
	ALT6 - ENET_1588_EVENT2_OUT	
	ALT8 - FLEXIO2_FLEXIO9	
	ALT10 - GPIO9_IO8	
	ALT11 - FLEXPWM1_PWMX3	
	ANALOG IO - ADC1_B1	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
GPIO_AD_10	ALT0 - USB_OTG1_PWR	ODE - Disabled
	ALT1 - LPI2C1_SCLS	PUS - Weak PD
	ALT2 - SIM2_PD	PUE - Pull Enable
	ALT3 - GPT3_COMPARE3	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA11	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO9	
	ALT6 - ENET_1588_EVENT3_IN	
	ALT8 - FLEXIO2_FLEXIO10	
	ALT10 - GPIO9_IO9	
	ALT11 - FLEXPWM2_PWMX0	
	ANALOG IO - ADC1_A2	
GPIO_AD_11	ALT0 - USB_OTG1_OC	ODE - Disabled
	ALT1 - LPI2C1_SDAS	PUS - Weak PD
	ALT2 - SIM2_POWER_FAIL	PUE - Pull Enable
	ALT3 - GPT3_CLK	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA10	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO10	
	ALT6 - ENET_1588_EVENT3_OUT	
	ALT8 - FLEXIO2_FLEXIO11	
	ALT10 - GPIO9_IO10	
	ALT11 - FLEXPWM2_PWMX1	
	ANALOG IO - ADC1_B2	
GPIO_AD_12	ALT0 - SPDIF_LOCK	ODE - Disabled
	ALT1 - LPI2C1_HREQ	PUS - Weak PD
	ALT2 - GPT1_CAPTURE1	PUE - Pull Enable
	ALT3 - FLEXSPI1_B_DATA3	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_PIXCLK	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO11	
	ALT6 - ENET_TDATA3	
	ALT8 - FLEXIO2_FLEXIO12	
	ALT9 - EWM_EWM_OUT_B	
	ALT10 - GPIO9_IO11	
	ALT11 - FLEXPWM2_PWMX2	
	ANALOG IO - ADC12_A3	
GPIO_AD_13	ALT0 - SPDIF_SR_CLK	ODE - Disabled
	ALT1 - PIT1_TRIGGER0	PUS - Weak PD
	ALT2 - GPT1_CAPTURE2	PUE - Pull Enable

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT3 - FLEXSPI1_B_DATA2	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_MCLK	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO12	
	ALT6 - ENET_TDATA2	
	ALT8 - FLEXIO2_FLEXIO13	
	ALT9 - ANATOP_32K_OUT	
	ALT10 - GPIO9_IO12	
	ALT11 - FLEXPWM2_PWMX3	
	ANALOG IO - ADC12_B3	
GPIO_AD_14	ALT0 - SPDIF_EXT_CLK	ODE - Disabled
	ALT1 - ANATOP_24M_OUT	PUS - Weak PD
	ALT2 - GPT1_COMPARE1	PUE - Pull Enable
	ALT3 - FLEXSPI1_B_DATA1	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_VSYNC	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO13	
	ALT6 - ENET_RX_CLK	
	ALT8 - FLEXIO2_FLEXIO14	
	ALT9 - CCM_ENET_REF_CLK_25M	
	ALT10 - GPIO9_IO13	
	ALT11 - FLEXPWM3_PWMX0	
	ANALOG IO - ADC12_A4	
GPIO_AD_15	ALT0 - SPDIF_IN	ODE - Disabled
	ALT1 - LPUART10_TX	PUS - Weak PD
	ALT2 - GPT1_COMPARE2	PUE - Pull Enable
	ALT3 - FLEXSPI1_B_DATA0	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_HSYNC	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO14	
	ALT6 - ENET_TX_ER	
	ALT8 - FLEXIO2_FLEXIO15	
	ALT10 - GPIO9_IO14	
	ALT11 - FLEXPWM3_PWMX1	
	ANALOG IO - ADC12_B4	
GPIO_AD_16	ALT0 - SPDIF_OUT	ODE - Disabled
	ALT1 - LPUART10_RX	PUS - Weak PD
	ALT2 - GPT1_COMPARE3	PUE - Pull Enable
	ALT3 - FLEXSPI1_B_SCLK	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA9	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO15	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT6 - ENET_RDATA3	
	ALT8 - FLEXIO2_FLEXIO16	
	ALT9 - ENET_1G_MDC	
	ALT10 - GPIO9_IO15	
	ALT11 - FLEXPWM3_PWMX2	
	ANALOG IO - ADC12_A5	
GPIO_AD_17	ALT0 - SAI1_MCLK	ODE - Disabled
	ALT1 - ACMP1_CMPO	PUS - Weak PD
	ALT2 - GPT1_CLK	PUE - Pull Enable
	ALT3 - FLEXSPI1_A_DQS	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA8	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO16	
	ALT6 - ENET_RDATA2	
	ALT8 - FLEXIO2_FLEXIO17	
	ALT9 - ENET_1G_MDIO	
	ALT10 - GPIO9_IO16	
	ALT11 - FLEXPWM3_PWMX3	
	ANALOG IO - ADC12_B5	
GPIO_AD_18	ALT0 - SAI1_RX_SYNC	ODE - Disabled
	ALT1 - ACMP2_CMPO	PUS - Weak PU
	ALT2 - LPSPI1_PCS1	PUE - Pull Enable
	ALT3 - FLEXSPI1_A_SS0_B	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA7	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO17	
	ALT6 - ENET_CRS	
	ALT8 - FLEXIO2_FLEXIO18	
	ALT9 - LPI2C2_SCL	
	ALT10 - GPIO9_IO17	
	ALT11 - FLEXPWM4_PWMX0	
	ANALOG IO - ADC2_A0	
GPIO_AD_19	ALT0 - SAI1_RX_BCLK	ODE - Disabled
	ALT1 - ACMP3_CMPO	PUS - Weak PD
	ALT2 - LPSPI1_PCS2	PUE - Pull Enable
	ALT3 - FLEXSPI1_A_SCLK	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA6	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO18	
	ALT6 - ENET_COL	
	ALT8 - FLEXIO2_FLEXIO19	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT9 - LPI2C2_SDA	
	ALT10 - GPIO9_IO18	
	ALT11 - FLEXPWM4_PWMX1	
	ANALOG IO - ADC2_B0	
GPIO_AD_20	ALT0 - SAI1_RX_DATA0	ODE - Disabled
	ALT1 - ACMP4_CMPO	PUS - Weak PD
	ALT2 - LPSPI1_PCS3	PUE - Pull Enable
	ALT3 - FLEXSPI1_A_DATA0	DSE - High drive strength
	ALT4 - VIDEO_MUX_CSI_DATA5	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO19	
	ALT6 - KPP_ROW7	
	ALT8 - FLEXIO2_FLEXIO20	
	ALT9 - ENET_QOS_1588_EVENT2_OUT	
	ALT10 - GPIO9_IO19	
	ALT11 - FLEXPWM4_PWMX2	
	ANALOG IO - ADC2_A1	
GPIO_AD_21	ALT0 - SAI1_TX_DATA0	ODE - Disabled
	ALT2 - LPSPI2_PCS1	PUS - Weak PD
	ALT3 - FLEXSPI1_A_DATA1	PUE - Pull Enable
	ALT4 - VIDEO_MUX_CSI_DATA4	DSE - High drive strength
	ALT5 - GPIO_MUX3_IO20	SRE - Fast slew rate
	ALT6 - KPP_COL7	
	ALT8 - FLEXIO2_FLEXIO21	
	ALT9 - ENET_QOS_1588_EVENT2_IN	
	ALT10 - GPIO9_IO20	
	ALT11 - FLEXPWM4_PWMX3	
	ANALOG IO - ADC2_B1	
GPIO_AD_22	ALT0 - SAI1_TX_BCLK	ODE - Disabled
	ALT2 - LPSPI2_PCS2	PUS - Weak PD
	ALT3 - FLEXSPI1_A_DATA2	PUE - Pull Enable
	ALT4 - VIDEO_MUX_CSI_DATA3	DSE - High drive strength
	ALT5 - GPIO_MUX3_IO21	SRE - Fast slew rate
	ALT6 - KPP_ROW6	
	ALT8 - FLEXIO2_FLEXIO22	
	ALT9 - ENET_QOS_1588_EVENT3_OUT	
	ALT10 - GPIO9_IO21	
	ANALOG IO - ADC2_A2	
GPIO_AD_23	ALT0 - SAI1_TX_SYNC	ODE - Disabled

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT2 - LPSPI2_PCS3	PUS - Weak PD
	ALT3 - FLEXSPI1_A_DATA3	PUE - Pull Enable
	ALT4 - VIDEO_MUX_CSI_DATA2	DSE - High drive strength
	ALT5 - GPIO_MUX3_IO22	SRE - Fast slew rate
	ALT6 - KPP_COL6	
	ALT8 - FLEXIO2_FLEXIO23	
	ALT9 - ENET_QOS_1588_EVENT3_IN	
	ALT10 - GPIO9_IO22	
	ANALOG IO - ADC2_B2	
GPIO_AD_24	ALT0 - LPUART1_TX	ODE - Disabled
	ALT1 - LPSPI2_SCK	PUS - Weak PD
	ALT2 - VIDEO_MUX_CSI_DATA0	PUE - Pull Enable
	ALT3 - ENET_RX_EN	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMA0	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO23	
	ALT6 - KPP_ROW5	
	ALT8 - FLEXIO2_FLEXIO24	
	ALT9 - LPI2C4_SCL	
	ALT10 - GPIO9_IO23	
	ANALOG IO - ADC2_A6	
GPIO_AD_25	ALT0 - LPUART1_RX	ODE - Disabled
	ALT1 - LPSPI2_PCS0	PUS - Weak PD
	ALT2 - VIDEO_MUX_CSI_DATA1	PUE - Pull Enable
	ALT3 - ENET_RX_ER	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMB0	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO24	
	ALT6 - KPP_COL5	
	ALT8 - FLEXIO2_FLEXIO25	
	ALT9 - LPI2C4_SDA	
	ALT10 - GPIO9_IO24	
	ANALOG IO - ADC2_B6	
GPIO_AD_26	ALT0 - LPUART1_CTS_B	ODE - Disabled
	ALT1 - LPSPI2_SDO	PUS - Weak PU
	ALT2 - SEMC_CSX1	PUE - Pull Enable
	ALT3 - ENET_RDATA0	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMA1	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO25	
	ALT6 - KPP_ROW4	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - FLEXIO2_FLEXIO26	
	ALT9 - ENET_QOS_MDC	
	ALT10 - GPIO9_IO25	
	ALT11 - USDHC2_CD_B	
	ANALOG IO - ACMP2_3	
GPIO_AD_27	ALT0 - LPUART1_RTS_B	ODE - Disabled
	ALT1 - LPSPI2_SDI	PUS - Weak PU
	ALT2 - SEMC_CSX2	PUE - Pull Enable
	ALT3 - ENET_RDATA1	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMB1	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO26	
	ALT6 - KPP_COL4	
	ALT8 - FLEXIO2_FLEXIO27	
	ALT9 - ENET_QOS_MDIO	
	ALT10 - GPIO9_IO26	
	ALT11 - USDHC2_WP	
	ANALOG IO - ACMP2_4	
GPIO_AD_28	ALT0 - LPSPI1_SCK	ODE - Disabled
	ALT1 - LPUART5_TX	PUS - Weak PD
	ALT2 - SEMC_CSX3	PUE - Pull Enable
	ALT3 - ENET_TX_EN	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMA2	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO27	
	ALT6 - KPP_ROW3	
	ALT8 - FLEXIO2_FLEXIO28	
	ALT9 - VIDEO_MUX_EXT_DCIC1	
	ALT10 - GPIO9_IO27	
	ALT11 - USDHC2_VSELECT	
	ANALOG IO - ACMP3_1	
GPIO_AD_29	ALT0 - LPSPI1_PCS0	ODE - Disabled
	ALT1 - LPUART5_RX	PUS - Weak PD
	ALT2 - ENET_REF_CLK1	PUE - Pull Enable
	ALT3 - ENET_TX_CLK	DSE - High drive strength
	ALT4 - FLEXPWM2_PWMB2	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO28	
	ALT6 - KPP_COL3	
	ALT8 - FLEXIO2_FLEXIO29	
	ALT9 - VIDEO_MUX_EXT_DCIC2	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT10 - GPIO9_IO28	
	ALT11 - USDHC2_RESET_B	
	ANALOG IO - ACMP3_2	
GPIO_AD_30	ALT0 - LPSPI1_SDO	ODE - Disabled
	ALT1 - USB_OTG2_OC	PUS - Weak PD
	ALT2 - CAN2_TX	PUE - Pull Enable
	ALT3 - ENET_TDATA0	DSE - High drive strength
	ALT4 - LPUART3_TX	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO29	
	ALT6 - KPP_ROW2	
	ALT8 - FLEXIO2_FLEXIO30	
	ALT9 - WDOG2_WDOG_RESET_B_DEB	
	ALT10 - GPIO9_IO29	
	ANALOG IO - ACMP3_3	
GPIO_AD_31	ALT0 - LPSPI1_SDI	ODE - Disabled
	ALT1 - USB_OTG2_PWR	PUS - Weak PD
	ALT2 - CAN2_RX	PUE - Pull Enable
	ALT3 - ENET_TDATA1	DSE - High drive strength
	ALT4 - LPUART3_RX	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO30	
	ALT6 - KPP_COL2	
	ALT8 - FLEXIO2_FLEXIO31	
	ALT9 - WDOG1_WDOG_RESET_B_DEB	
	ALT10 - GPIO9_IO30	
	ANALOG IO - ACMP3_4	
GPIO_AD_32	ALT0 - LPI2C1_SCL	ODE - Disabled
	ALT1 - USBPHY2_OTG_ID	PUS - Weak PD
	ALT2 - CCM_PMIC_RDY	PUE - Pull Enable
	ALT3 - ENET_MDC	DSE - High drive strength
	ALT4 - USDHC1_CD_B	SRE - Fast slew rate
	ALT5 - GPIO_MUX3_IO31	
	ALT6 - KPP_ROW1	
	ALT8 - LPUART10_TX	
	ALT9 - ENET_1G_MDC	
	ALT10 - GPIO9_IO31	
	ANALOG IO - ACMP4_1	
GPIO_AD_33	ALT0 - LPI2C1_SDA	ODE - Disabled
	ALT1 - USBPHY1_OTG_ID	PUS - Weak PD

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT2 - XBAR1_XBAR_INOUT17	PUE - Pull Enable
	ALT3 - ENET_MDIO	DSE - High drive strength
	ALT4 - USDHC1_WP	SRE - Fast slew rate
	ALT5 - GPIO4_IO0	
	ALT6 - KPP_COL1	
	ALT8 - LPUART10_RX	
	ALT9 - ENET_1G_MDIO	
	ALT10 - GPIO10_IO0	
	ANALOG IO - ACMP4_2	
GPIO_AD_34	ALT0 - ENET_1G_1588_EVENT0_IN	ODE - Disabled
	ALT1 - USB_OTG1_PWR	PUS - Weak PD
	ALT2 - XBAR1_XBAR_INOUT18	PUE - Pull Enable
	ALT3 - ENET_1588_EVENT0_IN	DSE - High drive strength
	ALT4 - USDHC1_VSELECT	SRE - Fast slew rate
	ALT5 - GPIO4_IO1	
	ALT6 - KPP_ROW0	
	ALT8 - LPUART10_CTS_B	
	ALT9 - WDOG1_WDOG_ANY	
	ALT10 - GPIO10_IO1	
	ANALOG IO - ACMP4_3	
GPIO_AD_35	ALT0 - ENET_1G_1588_EVENT0_OUT	ODE - Disabled
	ALT1 - USB_OTG1_OC	PUS - Weak PU
	ALT2 - XBAR1_XBAR_INOUT19	PUE - Pull Enable
	ALT3 - ENET_1588_EVENT0_OUT	DSE - High drive strength
	ALT4 - USDHC1_RESET_B	SRE - Fast slew rate
	ALT5 - GPIO4_IO2	
	ALT6 - KPP_COL0	
	ALT8 - LPUART10_RTS_B	
	ALT9 - FLEXSPI1_B_SS1_B	
	ALT10 - GPIO10_IO2	
	ANALOG IO - ACMP4_4	
GPIO_SD_B1_00	ALT0 - USDHC1_CMD	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT20	PULL - PU
	ALT3 - GPT4_CAPTURE1	PDRV - High drive strength
	ALT5 - GPIO4_IO3	
	ALT6 - FLEXSPI2_A_SS0_B	
	ALT8 - KPP_ROW7	
	ALT10 - GPIO10_IO3	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
GPIO_SD_B1_01	ALT0 - USDHC1_CLK	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT21	PULL - PD
	ALT3 - GPT4_CAPTURE2	PDRV - High drive strength
	ALT5 - GPIO4_IO4	
	ALT6 - FLEXSPI2_A_SCLK	
	ALT8 - KPP_COL7	
	ALT10 - GPIO10_IO4	
GPIO_SD_B1_02	ALT0 - USDHC1_DATA0	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT22	PULL - PU
	ALT3 - GPT4_COMPARE1	PDRV - High drive strength
	ALT5 - GPIO4_IO5	
	ALT6 - FLEXSPI2_A_DATA0	
	ALT8 - KPP_ROW6	
	ALT9 - FLEXSPI1_A_SS1_B	
	ALT10 - GPIO10_IO5	
GPIO_SD_B1_03	ALT0 - USDHC1_DATA1	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT23	PULL - PU
	ALT3 - GPT4_COMPARE2	PDRV - High drive strength
	ALT5 - GPIO4_IO6	
	ALT6 - FLEXSPI2_A_DATA1	
	ALT8 - KPP_COL6	
	ALT9 - FLEXSPI1_B_SS1_B	
	ALT10 - GPIO10_IO6	
GPIO_SD_B1_04	ALT0 - USDHC1_DATA2	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT24	PULL - PU
	ALT3 - GPT4_COMPARE3	PDRV - High drive strength
	ALT5 - GPIO4_IO7	
	ALT6 - FLEXSPI2_A_DATA2	
	ALT8 - FLEXSPI1_B_SS0_B	
	ALT9 - ENET_QOS_1588_EVENT2_AUX_IN	
	ALT10 - GPIO10_IO7	
GPIO_SD_B1_05	ALT0 - USDHC1_DATA3	ODE - Disabled
	ALT2 - XBAR1_XBAR_INOUT25	PULL - PD
	ALT3 - GPT4_CLK	PDRV - High drive strength
	ALT5 - GPIO4_IO8	
	ALT6 - FLEXSPI2_A_DATA3	
	ALT8 - FLEXSPI1_B_DQS	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT9 - ENET_QOS_1588_EVENT3_AUX_IN ALT10 - GPIO10_IO8	
GPIO_SD_B2_00	ALT0 - USDHC2_DATA3 ALT1 - FLEXSPI1_B_DATA3 ALT2 - ENET_1G_RX_EN ALT3 - LPUART9_TX ALT4 - LPSPI4_SCK ALT5 - GPIO4_IO9 ALT10 - GPIO10_IO9	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_SD_B2_01	ALT0 - USDHC2_DATA2 ALT1 - FLEXSPI1_B_DATA2 ALT2 - ENET_1G_RX_CLK ALT3 - LPUART9_RX ALT4 - LPSPI4_PCS0 ALT5 - GPIO4_IO10 ALT10 - GPIO10_IO10	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_SD_B2_02	ALT0 - USDHC2_DATA1 ALT1 - FLEXSPI1_B_DATA1 ALT2 - ENET_1G_RDATA0 ALT3 - LPUART9_CTS_B ALT4 - LPSPI4_SDO ALT5 - GPIO4_IO11 ALT10 - GPIO10_IO11	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_SD_B2_03	ALT0 - USDHC2_DATA0 ALT1 - FLEXSPI1_B_DATA0 ALT2 - ENET_1G_RDATA1 ALT3 - LPUART9_RTS_B ALT4 - LPSPI4_SDI ALT5 - GPIO4_IO12 ALT10 - GPIO10_IO12	ODE - Disabled PULL - PD PDRV - High drive strength
GPIO_SD_B2_04	ALT0 - USDHC2_CLK ALT1 - FLEXSPI1_B_SCLK ALT2 - ENET_1G_RDATA2 ALT3 - FLEXSPI1_A_SS1_B ALT4 - LPSPI4_PCS1 ALT5 - GPIO4_IO13 ALT10 - GPIO10_IO13	ODE - Disabled PULL - PU PDRV - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
GPIO_SD_B2_05	ALT0 - USDHC2_CMD	ODE - Disabled
	ALT1 - FLEXSPI1_A_DQS	PULL - PU
	ALT2 - ENET_1G_RDATA3	PDRV - High drive strength
	ALT3 - FLEXSPI1_B_SS0_B	
	ALT4 - LPSPI4_PCS2	
	ALT5 - GPIO4_IO14	
	ALT10 - GPIO10_IO14	
GPIO_SD_B2_06	ALT0 - USDHC2_RESET_B	ODE - Disabled
	ALT1 - FLEXSPI1_A_SS0_B	PULL - PU
	ALT2 - ENET_1G_TDATA3	PDRV - High drive strength
	ALT3 - LPSPI4_PCS3	
	ALT4 - GPT6_CAPTURE1	
	ALT5 - GPIO4_IO15	
	ALT10 - GPIO10_IO15	
GPIO_SD_B2_07	ALT0 - USDHC2_STROBE	ODE - Disabled
	ALT1 - FLEXSPI1_A_SCLK	PULL - PD
	ALT2 - ENET_1G_TDATA2	PDRV - High drive strength
	ALT3 - LPUART3_CTS_B	
	ALT4 - GPT6_CAPTURE2	
	ALT5 - GPIO4_IO16	
	ALT6 - LPSPI2_SCK	
	ALT8 - ENET_TX_ER	
	ALT9 - ENET_QOS_REF_CLK1	
	ALT10 - GPIO10_IO16	
GPIO_SD_B2_08	ALT0 - USDHC2_DATA4	ODE - Disabled
	ALT1 - FLEXSPI1_A_DATA0	PULL - PD
	ALT2 - ENET_1G_TDATA1	PDRV - High drive strength
	ALT3 - LPUART3_RTS_B	
	ALT4 - GPT6_COMPARE1	
	ALT5 - GPIO4_IO17	
	ALT6 - LPSPI2_PCS0	
	ALT10 - GPIO10_IO17	
GPIO_SD_B2_09	ALT0 - USDHC2_DATA5	ODE - Disabled
	ALT1 - FLEXSPI1_A_DATA1	PULL - PD
	ALT2 - ENET_1G_TDATA0	PDRV - High drive strength
	ALT3 - LPUART5_CTS_B	
	ALT4 - GPT6_COMPARE2	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT5 - GPIO4_IO18	
	ALT6 - LPSPI2_SDO	
	ALT10 - GPIO10_IO18	
GPIO_SD_B2_10	ALT0 - USDHC2_DATA6	ODE - Disabled
	ALT1 - FLEXSPI1_A_DATA2	PULL - PD
	ALT2 - ENET_1G_TX_EN	PDRV - High drive strength
	ALT3 - LPUART5_RTS_B	
	ALT4 - GPT6_COMPARE3	
	ALT5 - GPIO4_IO19	
	ALT6 - LPSPI2_SDI	
	ALT10 - GPIO10_IO19	
GPIO_SD_B2_11	ALT0 - USDHC2_DATA7	ODE - Disabled
	ALT1 - FLEXSPI1_A_DATA3	PULL - PD
	ALT2 - ENET_1G_TX_CLK_IO	PDRV - High drive strength
	ALT3 - ENET_1G_REF_CLK1	
	ALT4 - GPT6_CLK	
	ALT5 - GPIO4_IO20	
	ALT6 - LPSPI2_PCS1	
	ALT10 - GPIO10_IO20	
GPIO_DISP_B1_00	ALT0 - VIDEO_MUX_LCDIF_CLK	ODE - Disabled
	ALT1 - ENET_1G_RX_EN	PULL - PD
	ALT3 - QTIMER1_TIMER0	PDRV - High drive strength
	ALT4 - XBAR1_XBAR_INOUT26	
	ALT5 - GPIO4_IO21	
	ALT6 - SDIO_SLV_CMD	
	ALT8 - ENET_QOS_RX_EN	
	ALT10 - GPIO10_IO21	
GPIO_DISP_B1_01	ALT0 - VIDEO_MUX_LCDIF_ENABLE	ODE - Disabled
	ALT1 - ENET_1G_RX_CLK	PULL - PD
	ALT2 - ENET_1G_RX_ER	PDRV - High drive strength
	ALT3 - QTIMER1_TIMER1	
	ALT4 - XBAR1_XBAR_INOUT27	
	ALT5 - GPIO4_IO22	
	ALT6 - SDIO_SLV_CLK	
	ALT8 - ENET_QOS_RX_CLK	
	ALT9 - ENET_QOS_RX_ER	
	ALT10 - GPIO10_IO22	
GPIO_DISP_B1_02	ALT0 - VIDEO_MUX_LCDIF_HSYNC	ODE - Disabled

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT1 - ENET_1G_RDATA0	PULL - PD
	ALT2 - LPI2C3_SCL	PDRV - High drive strength
	ALT3 - QTIMER1_TIMER2	
	ALT4 - XBAR1_XBAR_INOUT28	
	ALT5 - GPIO4_IO23	
	ALT6 - SDIO_SLV_DATA0	
	ALT8 - ENET_QOS_RDATA0	
	ALT9 - LPUART1_TX	
	ALT10 - GPIO10_IO23	
GPIO_DISP_B1_03	ALT0 - VIDEO_MUX_LCDIF_VSYNC	ODE - Disabled
	ALT1 - ENET_1G_RDATA1	PULL - PD
	ALT2 - LPI2C3_SDA	PDRV - High drive strength
	ALT3 - QTIMER2_TIMER0	
	ALT4 - XBAR1_XBAR_INOUT29	
	ALT5 - GPIO4_IO24	
	ALT6 - SDIO_SLV_DATA1	
	ALT8 - ENET_QOS_RDATA1	
	ALT9 - LPUART1_RX	
	ALT10 - GPIO10_IO24	
GPIO_DISP_B1_04	ALT0 - VIDEO_MUX_LCDIF_DATA0	ODE - Disabled
	ALT1 - ENET_1G_RDATA2	PULL - PD
	ALT2 - LPUART4_RX	PDRV - High drive strength
	ALT3 - QTIMER2_TIMER1	
	ALT4 - XBAR1_XBAR_INOUT30	
	ALT5 - GPIO4_IO25	
	ALT6 - SDIO_SLV_DATA2	
	ALT8 - ENET_QOS_RDATA2	
	ALT9 - LPSPI3_SCK	
	ALT10 - GPIO10_IO25	
GPIO_DISP_B1_05	ALT0 - VIDEO_MUX_LCDIF_DATA1	ODE - Disabled
	ALT1 - ENET_1G_RDATA3	PULL - PD
	ALT2 - LPUART4_CTS_B	PDRV - High drive strength
	ALT3 - QTIMER2_TIMER2	
	ALT4 - XBAR1_XBAR_INOUT31	
	ALT5 - GPIO4_IO26	
	ALT6 - SDIO_SLV_DATA3	
	ALT8 - ENET_QOS_RDATA3	
	ALT9 - LPSPI3_SDI	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT10 - GPIO10_IO26	
GPIO_DISP_B1_06	ALT0 - VIDEO_MUX_LCDIF_DATA2 ALT1 - ENET_1G_TDATA3 ALT2 - LPUART4_TX ALT3 - QTIMER3_TIMER0 ALT4 - XBAR1_XBAR_INOUT32 ALT5 - GPIO4_IO27 ALT6 - SRC_BT_CFG0 ALT8 - ENET_QOS_TDATA3 ALT9 - LPSPI3_SDO ALT10 - GPIO10_IO27	ODE - Disabled PULL - No pull PDRV - High drive strength
GPIO_DISP_B1_07	ALT0 - VIDEO_MUX_LCDIF_DATA3 ALT1 - ENET_1G_TDATA2 ALT2 - LPUART4_RTS_B ALT3 - QTIMER3_TIMER1 ALT4 - XBAR1_XBAR_INOUT33 ALT5 - GPIO4_IO28 ALT6 - SRC_BT_CFG1 ALT8 - ENET_QOS_TDATA2 ALT9 - LPSPI3_PCS0 ALT10 - GPIO10_IO28	ODE - Disabled PULL - No pull PDRV - High drive strength
GPIO_DISP_B1_08	ALT0 - VIDEO_MUX_LCDIF_DATA4 ALT1 - ENET_1G_TDATA1 ALT2 - USDHC1_CD_B ALT3 - QTIMER3_TIMER2 ALT4 - XBAR1_XBAR_INOUT34 ALT5 - GPIO4_IO29 ALT6 - SRC_BT_CFG2 ALT8 - ENET_QOS_TDATA1 ALT9 - LPSPI3_PCS1 ALT10 - GPIO10_IO29	ODE - Disabled PULL - No pull PDRV - High drive strength
GPIO_DISP_B1_09	ALT0 - VIDEO_MUX_LCDIF_DATA5 ALT1 - ENET_1G_TDATA0 ALT2 - USDHC1_WP ALT3 - QTIMER4_TIMER0 ALT4 - XBAR1_XBAR_INOUT35 ALT5 - GPIO4_IO30 ALT6 - SRC_BT_CFG3	ODE - Disabled PULL - No pull PDRV - High drive strength

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - ENET_QOS_TDATA0	
	ALT9 - LPSPI3_PCS2	
	ALT10 - GPIO10_IO30	
GPIO_DISP_B1_10	ALT0 - VIDEO_MUX_LCDIF_DATA6	ODE - Disabled
	ALT1 - ENET_1G_TX_EN	PULL - No pull
	ALT2 - USDHC1_RESET_B	PDRV - High drive strength
	ALT3 - QTIMER4_TIMER1	
	ALT4 - XBAR1_XBAR_INOUT36	
	ALT5 - GPIO4_IO31	
	ALT6 - SRC_BT_CFG4	
	ALT8 - ENET_QOS_TX_EN	
	ALT9 - LPSPI3_PCS3	
	ALT10 - GPIO10_IO31	
GPIO_DISP_B1_11	ALT0 - VIDEO_MUX_LCDIF_DATA7	ODE - Disabled
	ALT1 - ENET_1G_TX_CLK_IO	PULL - No pull
	ALT2 - ENET_1G_REF_CLK1	PDRV - High drive strength
	ALT3 - QTIMER4_TIMER2	
	ALT4 - XBAR1_XBAR_INOUT37	
	ALT5 - GPIO5_IO0	
	ALT6 - SRC_BT_CFG5	
	ALT8 - ENET_QOS_TX_CLK	
	ALT9 - ENET_QOS_REF_CLK1	
	ALT10 - GPIO11_IO0	
GPIO_DISP_B2_00	ALT0 - VIDEO_MUX_LCDIF_DATA8	ODE - Disabled
	ALT1 - WDOG1_WDOG_B	PUS - Weak PD
	ALT2 - MQS_RIGHT	PUE - Pull Disable, Highz
	ALT3 - ENET_1G_TX_ER	DSE - High drive strength
	ALT4 - SAI1_TX_DATA3	SRE - Fast slew rate
	ALT5 - GPIO5_IO1	
	ALT6 - SRC_BT_CFG6	
	ALT8 - ENET_QOS_TX_ER	
	ALT10 - GPIO11_IO1	
GPIO_DISP_B2_01	ALT0 - VIDEO_MUX_LCDIF_DATA9	ODE - Disabled
	ALT1 - USDHC1_VSELECT	PUS - Weak PD
	ALT2 - MQS_LEFT	PUE - Pull Disable, Highz
	ALT3 - WDOG2_WDOG_B	DSE - High drive strength
	ALT4 - SAI1_TX_DATA2	SRE - Fast slew rate
	ALT5 - GPIO5_IO2	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT6 - SRC_BT_CFG7	
	ALT8 - EWM_EWM_OUT_B	
	ALT9 - CCM_ENET_REF_CLK_25M	
	ALT10 - GPIO11_IO2	
GPIO_DISP_B2_02	ALT0 - VIDEO_MUX_LCDIF_DATA10	ODE - Disabled
	ALT1 - ENET_TDATA0	PUS - Weak PD
	ALT2 - PIT1_TRIGGER3	PUE - Pull Disable, Highz
	ALT3 - ARM_TRACE0	DSE - High drive strength
	ALT4 - SAI1_TX_DATA1	SRE - Fast slew rate
	ALT5 - GPIO5_IO3	
	ALT6 - SRC_BT_CFG8	
	ALT8 - ENET_QOS_TDATA0	
	ALT10 - GPIO11_IO3	
GPIO_DISP_B2_03	ALT0 - VIDEO_MUX_LCDIF_DATA11	ODE - Disabled
	ALT1 - ENET_TDATA1	PUS - Weak PD
	ALT2 - PIT1_TRIGGER2	PUE - Pull Disable, Highz
	ALT3 - ARM_TRACE1	DSE - High drive strength
	ALT4 - SAI1_MCLK	SRE - Fast slew rate
	ALT5 - GPIO5_IO4	
	ALT6 - SRC_BT_CFG9	
	ALT8 - ENET_QOS_TDATA1	
	ALT10 - GPIO11_IO4	
GPIO_DISP_B2_04	ALT0 - VIDEO_MUX_LCDIF_DATA12	ODE - Disabled
	ALT1 - ENET_TX_EN	PUS - Weak PD
	ALT2 - PIT1_TRIGGER1	PUE - Pull Disable, Highz
	ALT3 - ARM_TRACE2	DSE - High drive strength
	ALT4 - SAI1_RX_SYNC	SRE - Fast slew rate
	ALT5 - GPIO5_IO5	
	ALT6 - SRC_BT_CFG10	
	ALT8 - ENET_QOS_TX_EN	
	ALT10 - GPIO11_IO5	
GPIO_DISP_B2_05	ALT0 - VIDEO_MUX_LCDIF_DATA13	ODE - Disabled
	ALT1 - ENET_TX_CLK	PUS - Weak PD
	ALT2 - ENET_REF_CLK1	PUE - Pull Disable, Highz
	ALT3 - ARM_TRACE3	DSE - High drive strength
	ALT4 - SAI1_RX_BCLK	SRE - Fast slew rate
	ALT5 - GPIO5_IO6	
	ALT6 - SRC_BT_CFG11	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT8 - ENET_QOS_TX_CLK	
	ALT10 - GPIO11_IO6	
GPIO_DISP_B2_06	ALT0 - VIDEO_MUX_LCDIF_DATA14	ODE - Disabled
	ALT1 - ENET_RDATA0	PUS - Weak PD
	ALT2 - LPUART7_TX	PUE - Pull Enable
	ALT3 - ARM_TRACE_CLK	DSE - High drive strength
	ALT4 - SAI1_RX_DATA0	SRE - Fast slew rate
	ALT5 - GPIO5_IO7	
	ALT8 - ENET_QOS_RDATA0	
	ALT10 - GPIO11_IO7	
GPIO_DISP_B2_07	ALT0 - VIDEO_MUX_LCDIF_DATA15	ODE - Disabled
	ALT1 - ENET_RDATA1	PUS - Weak PD
	ALT2 - LPUART7_RX	PUE - Pull Enable
	ALT3 - ARM_TRACE_SWO	DSE - High drive strength
	ALT4 - SAI1_TX_DATA0	SRE - Fast slew rate
	ALT5 - GPIO5_IO8	
	ALT8 - ENET_QOS_RDATA1	
	ALT10 - GPIO11_IO8	
GPIO_DISP_B2_08	ALT0 - VIDEO_MUX_LCDIF_DATA16	ODE - Disabled
	ALT1 - ENET_RX_EN	PUS - Weak PD
	ALT2 - LPUART8_TX	PUE - Pull Enable
	ALT3 - CM7_EVENTO	DSE - High drive strength
	ALT4 - SAI1_TX_BCLK	SRE - Fast slew rate
	ALT5 - GPIO5_IO9	
	ALT8 - ENET_QOS_RX_EN	
	ALT9 - LPUART1_TX	
	ALT10 - GPIO11_IO9	
GPIO_DISP_B2_09	ALT0 - VIDEO_MUX_LCDIF_DATA17	ODE - Disabled
	ALT1 - ENET_RX_ER	PUS - Weak PD
	ALT2 - LPUART8_RX	PUE - Pull Enable
	ALT3 - CM7_EVENTI	DSE - High drive strength
	ALT4 - SAI1_TX_SYNC	SRE - Fast slew rate
	ALT5 - GPIO5_IO10	
	ALT8 - ENET_QOS_RX_ER	
	ALT9 - LPUART1_RX	
	ALT10 - GPIO11_IO10	
GPIO_DISP_B2_10	ALT0 - VIDEO_MUX_LCDIF_DATA18	ODE - Disabled
	ALT1 - SIM2_TRXD	PUS - Weak PD

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
	ALT2 - LPUART2_TX	PUE - Pull Enable
	ALT3 - WDOG2_WDOG_RESET_B_DEB	DSE - High drive strength
	ALT4 - XBAR1_XBAR_INOUT38	SRE - Fast slew rate
	ALT5 - GPIO5_IO11	
	ALT6 - LPI2C3_SCL	
	ALT8 - ENET_QOS_RX_ER	
	ALT9 - SPDIF_IN	
	ALT10 - GPIO11_IO11	
GPIO_DISP_B2_11	ALT0 - VIDEO_MUX_LCDIF_DATA19	ODE - Disabled
	ALT1 - SIM2_CLK	PUS - Weak PD
	ALT2 - LPUART2_RX	PUE - Pull Enable
	ALT3 - WDOG1_WDOG_RESET_B_DEB	DSE - High drive strength
	ALT4 - XBAR1_XBAR_INOUT39	SRE - Fast slew rate
	ALT5 - GPIO5_IO12	
	ALT6 - LPI2C3_SDA	
	ALT8 - ENET_QOS_CRS	
	ALT9 - SPDIF_OUT	
	ALT10 - GPIO11_IO12	
GPIO_DISP_B2_12	ALT0 - VIDEO_MUX_LCDIF_DATA20	ODE - Disabled
	ALT1 - SIM2_RST_B	PUS - Weak PD
	ALT2 - CAN1_TX	PUE - Pull Enable
	ALT3 - LPUART2_CTS_B	DSE - High drive strength
	ALT4 - XBAR1_XBAR_INOUT40	SRE - Fast slew rate
	ALT5 - GPIO5_IO13	
	ALT6 - LPI2C4_SCL	
	ALT8 - ENET_QOS_COL	
	ALT9 - LPSPI4_SCK	
	ALT10 - GPIO11_IO13	
GPIO_DISP_B2_13	ALT0 - VIDEO_MUX_LCDIF_DATA21	ODE - Disabled
	ALT1 - SIM2_SVEN	PUS - Weak PD
	ALT2 - CAN1_RX	PUE - Pull Enable
	ALT3 - LPUART2_RTS_B	DSE - High drive strength
	ALT4 - ENET_REF_CLK1	SRE - Fast slew rate
	ALT5 - GPIO5_IO14	
	ALT6 - LPI2C4_SDA	
	ALT8 - ENET_QOS_1588_EVENT0_OUT	
	ALT9 - LPSPI4_SDI	
	ALT10 - GPIO11_IO14	

Table continues on the next page...

Table 11-2. Pin Assignments (continued)

Pin Name	Pin Assignments	Pad Settings
GPIO_DISP_B2_14	ALT0 - VIDEO_MUX_LCDIF_DATA22 ALT1 - SIM2_PD ALT2 - WDOG2_WDOG_B ALT3 - VIDEO_MUX_EXT_DCIC1 ALT4 - ENET_1G_REF_CLK1 ALT5 - GPIO5_IO15 ALT6 - CAN1_TX ALT8 - ENET_QOS_1588_EVENT0_IN ALT9 - LPSPI4_SDO ALT10 - GPIO11_IO15	ODE - Disabled PUS - Weak PD PUE - Pull Enable DSE - High drive strength SRE - Fast slew rate
GPIO_DISP_B2_15	ALT0 - VIDEO_MUX_LCDIF_DATA23 ALT1 - SIM2_POWER_FAIL ALT2 - WDOG1_WDOG_B ALT3 - VIDEO_MUX_EXT_DCIC2 ALT4 - PIT1_TRIGGER0 ALT5 - GPIO5_IO16 ALT6 - CAN1_RX ALT8 - ENET_QOS_1588_EVENT0_AUX_IN ALT9 - LPSPI4_PCS0 ALT10 - GPIO11_IO16	ODE - Disabled PUS - Weak PU PUE - Pull Enable DSE - High drive strength SRE - Fast slew rate

1. For GPIO_SNVS_00-09, tamper or SNVS_GPIO is determine by selected part numbers. On those devices, the users cannot select between Tamper and SNVS GPIO functionality.

Chapter 12
IOMUX Controller (IOMUXC)

12.1 Chip-specific IOMUXC information

Table 12-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

For GPIO_MUX2 and GPIO_MUX3 instances, the users can select between normal GPIO or CM7 fast GPIO.

12.2 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the IC to share one pad to several functional blocks. This sharing is done by multiplexing the pad's input and output signals.

Every module requires a specific pad setting (such as pull up or down), and for each pad, there are up to 12 muxing options (called ALT modes). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each basic IOMUX cell handles only one pad signal's muxing.

The figure below illustrates the IOMUX/IOMUXC connectivity in the system.

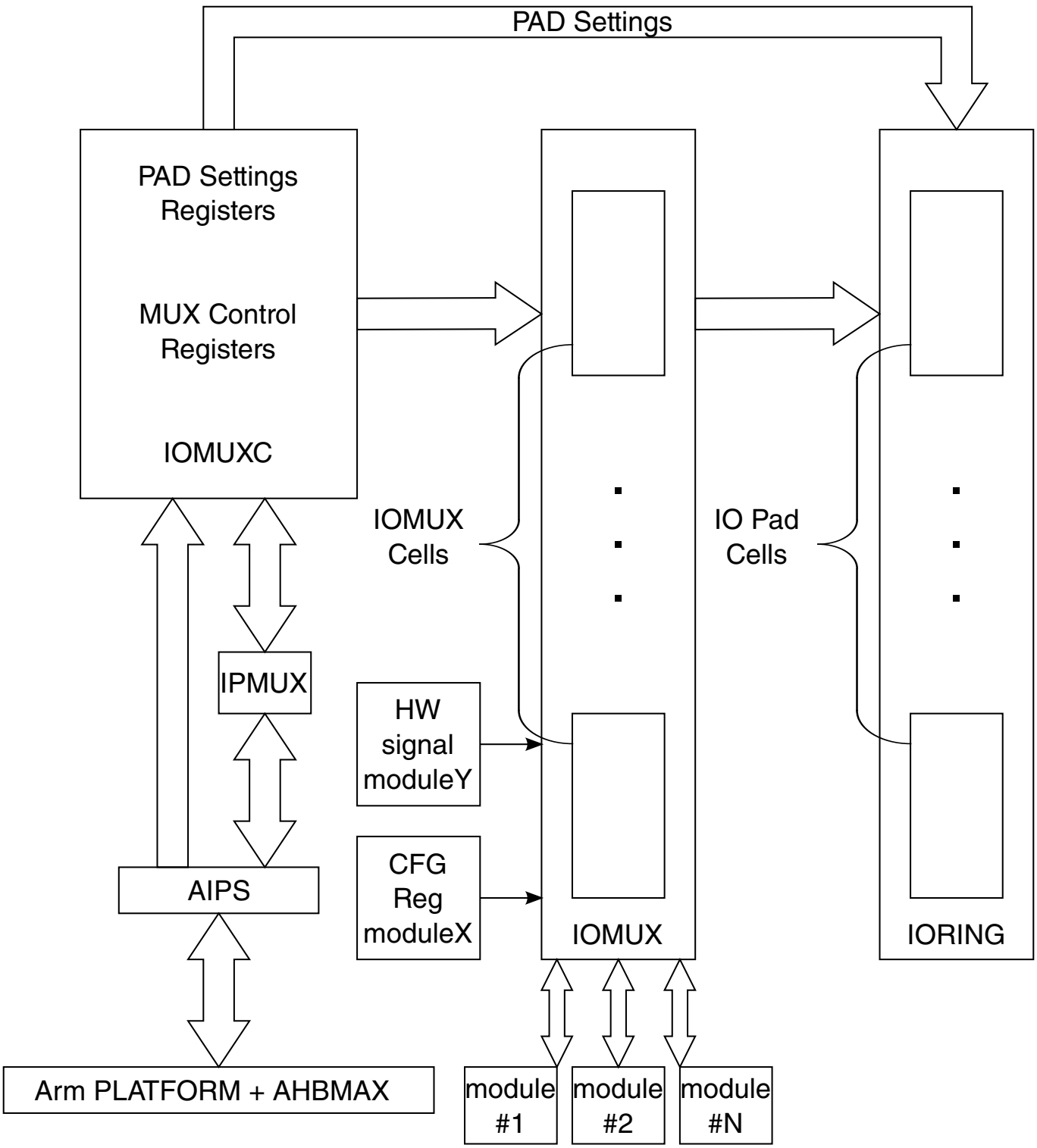


Figure 12-1. IOMUX SoC Level Block Diagram

12.2.1 Block Diagram

The high level illustration of the IO cells is shown in Figure 12-3

12.2.2 Features

The IOMUXC features include:

- 32-bit software mux control registers (IOMUXC_SW_MUX_CTL_PAD_<PAD NAME>) to configure 1 of 12 alternate (ALT) MUX_MODE fields of each pad or a predefined group of pads and to enable the forcing of an input path of the pad(s) (SION bit).
- 32-bit software pad control registers (IOMUXC_SW_PAD_CTL_PAD_<PAD_NAME>) to configure specific pad settings of each pad, or a predefined group of pads.
- 32-bit general purpose registers - GPRxx, 32-bit registers according to SoC requirements for any usage.
- 32-bit input select control registers to control the input path to a module when more than one pad drives this module input.

Each SW MUX/PAD CTL IOMUXC register handles only one pad or one pad's group.

Only the minimum number of registers required by software are implemented by hardware. For example, if only ALT0 and ALT1 modes are used on Pad x then only one bit register will be generated as the MUX_MODE control field in the software mux control register of Pad x.

The software mux control registers may allow the forcing of pads to become input (input path enabled) regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

12.3 Functional description

This section provides a complete functional description of the block.

The IOMUX consists of a number of basic iomux cell units. If only one functional mode is required for a specific pad, would have been no need for IOMUX and the signals can be connected directly from the module to the I/O. The IOMUX cell is required whenever two or more functional modes are required for a specific pad or when one functional mode and the one test mode are required.

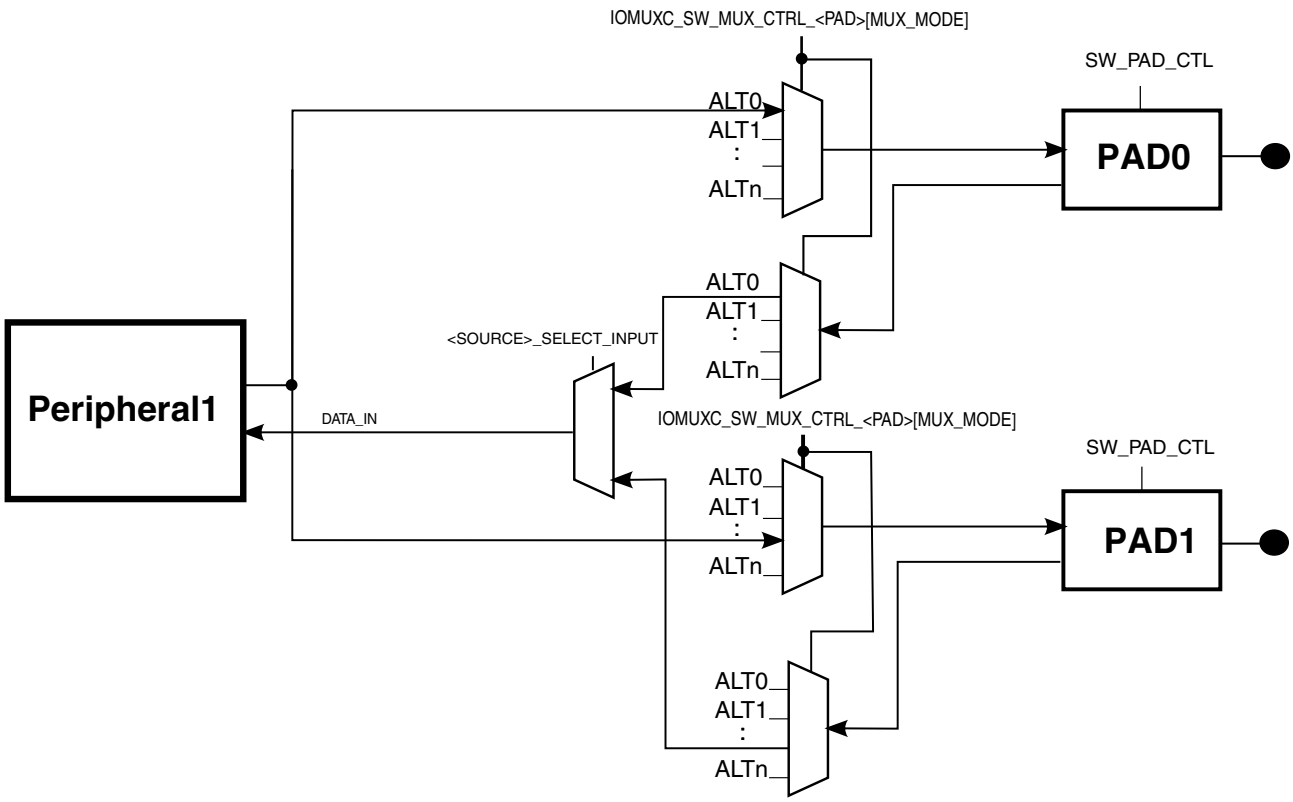


Figure 12-2. IOMUX Cell Block Diagram

12.3.1 ALT6 and ALT7 extended muxing modes

The ALT7 and ALT6 extended muxing modes allow any signal in the system (such as fuse, pad input, JTAG, or software register) to override any software configuration and to force the ALT6/ALT7 muxing mode.

It also allows an IOMUX software register to control a group of pads.

12.3.2 SW Loopback through SION bit

A limited option exists to override the default pad functionality and force the input path to be active regardless of the value driven by the corresponding module. This can be done by setting the SION (Software Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1".

Uses include:

- LoopBack - Module x drives the pad and also receives pad value as an input.

12.3.3 Daisy chain - multi pads driving same module input pin

In some cases, more than one pad may drive a single module input pin. Such cases require the addition of one more level of IOMUXing; all of these input signals are muxed, and a dedicated software controlled register controls the mux in order to select the required input path.

A module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

This means that a module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

The daisy chain is illustrated in the figure below.

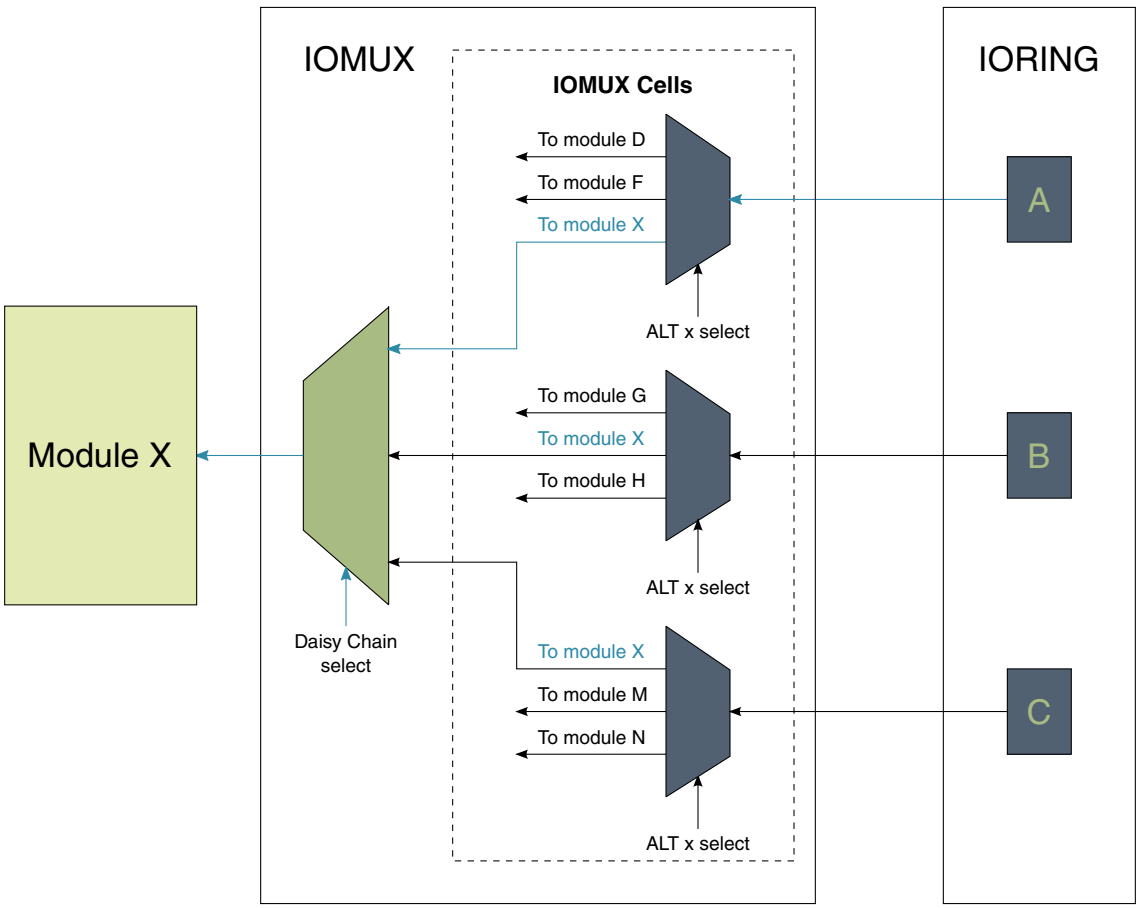


Figure 12-3. Daisy chain illustration

12.3.4 Clocks

The table found here describes the clock sources for IOMUXC. Please see Clock Controller Module (CCM) chapter for clock setting, configuration and gating information.

Table 12-2. IOMUXC Clocks

Clock Name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock

12.4 Memory Map and register definition

This section includes the IOMUXC module memory map and detailed descriptions of all registers.

12.4.1 IOMUXC SNVS GPR register descriptions

12.4.1.1 iomuxc_snvs_gpr memory map

IOMUXC_SNVS_GPR base address: 40C9_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 7Ch	GPR0 General Purpose Register (GPR0 - GPR31)	32	RW	0000_0000h
80h	GPR32 General Purpose Register (GPR32)	32	RW	0000_0000h
84h	GPR33 General Purpose Register (GPR33)	32	RW	0000_0000h

12.4.1.2 GPR0 General Purpose Register (GPR0 - GPR31)

12.4.1.2.1 Offset

For a = 0 to 31:

Register	Offset
GPRa	0h + (a × 4h)

12.4.1.2.2 Function

GPR Register

12.4.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.1.2.4 Fields

Field	Function
31-0 GPR	General purpose bits General purpose secure register. The register will be cleared upon SNVS tamper events. These registers do not have any dedicated functions. They can be set by the user.

12.4.1.3 GPR32 General Purpose Register (GPR32)

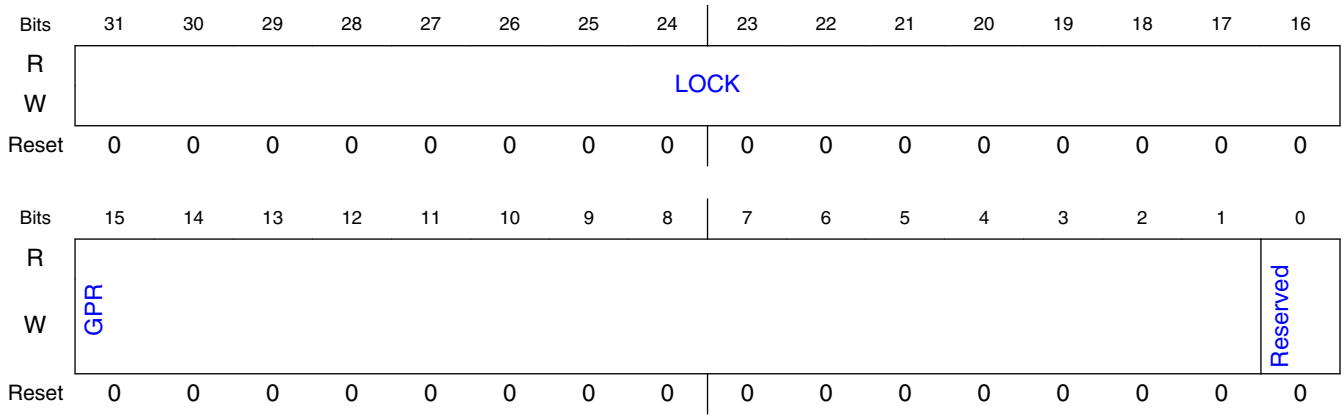
12.4.1.3.1 Offset

Register	Offset
GPR32	80h

12.4.1.3.2 Function

GPR Register

12.4.1.3.3 Diagram



12.4.1.3.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set the write to GPR bit i will be blocked.
15-1 GPR	General purpose bits General lockable purpose bits for SW. These registers do not have any dedicated functions. They can be set by the user.
0 —	Reserved

12.4.1.4 GPR33 General Purpose Register (GPR33)

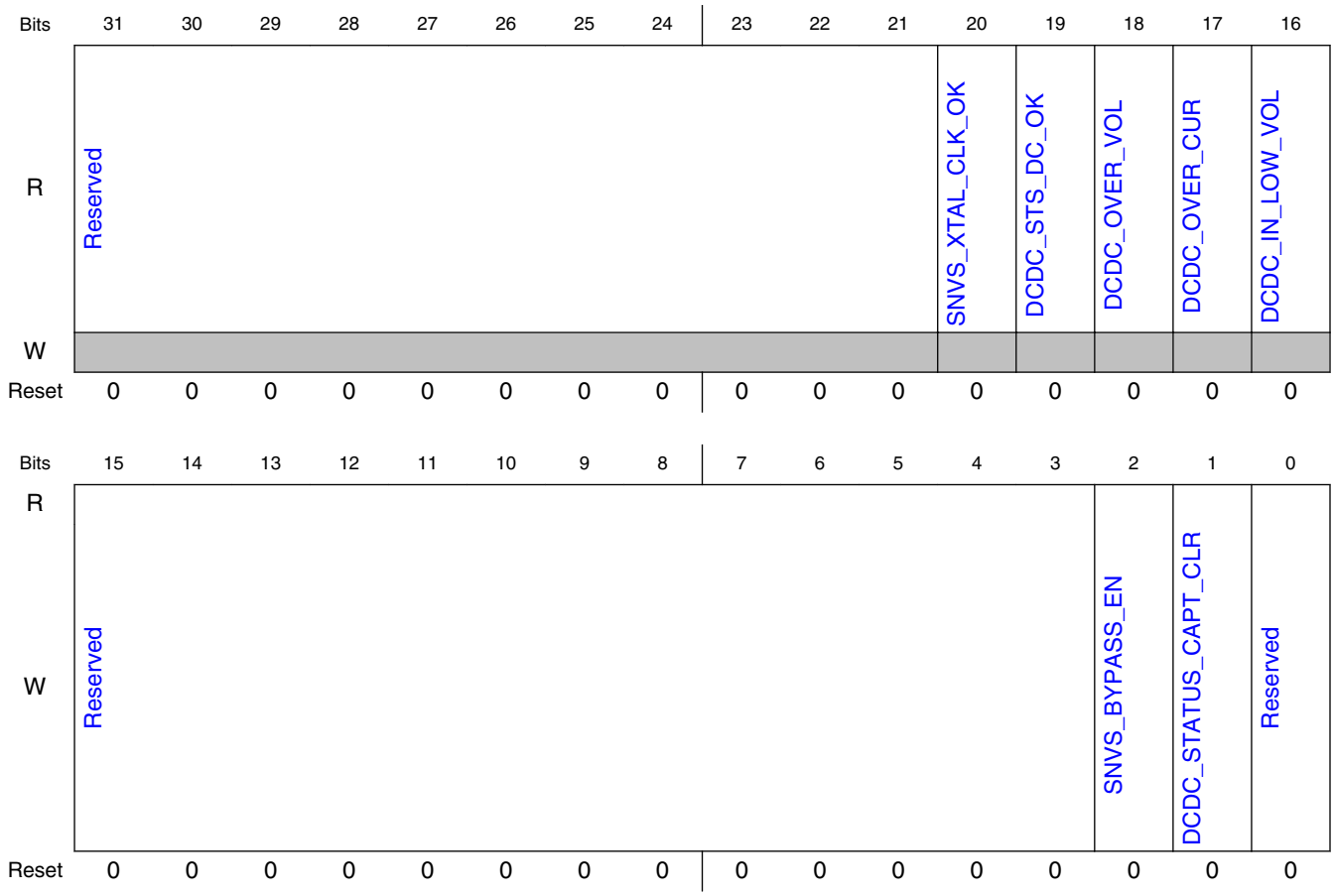
12.4.1.4.1 Offset

Register	Offset
GPR33	84h

12.4.1.4.2 Function

GPR Register

12.4.1.4.3 Diagram



12.4.1.4.4 Fields

Field	Function
31-21 —	Reserved
20 SNVS_XTAL_CLK_OK	32K OSC ok flag This bit high indicates the 32K oscillator is stable into normal operation. 0b - 32K oscillator is NOT stable into normal operation 1b - 32K oscillator is stable into normal operation
19 DCDC_STS_DC_OK	DCDC status OK Status register to indicate DCDC status. 0b - DCDC is settling 1b - DCDC already settled
18 DCDC_OVER_VOL	DCDC output over voltage alert 0b - No Overvoltage on DCDC VDDLP0 or VDDLP8 output 1b - Overvoltage on DCDC VDDLP0 or VDDLP8 output
17	DCDC output over current alert 0b - No Overcurrent on DCDC output

Table continues on the next page...

Field	Function
DCDC_OVER_CUR	1b - Overcurrent on DCDC output
16 DCDC_IN_LOW_VOL	DCDC_IN low voltage detect 0b - Voltage on DCDC_IN is higher than 2.6V 1b - Voltage on DCDC_IN is lower than 2.6V
15-3 —	Reserved
2 SNVS_BYPASS_EN	SNVS LDO_SNVS_ANA bypass enable LDO_SNVS_ANA bypass function enable, when high, cutoff the pull-down current in regulator. 0b - Disable bypass 1b - Enable bypass
1 DCDC_STATUS_CAPT_CLR	DCDC captured status clear Write logic 1 to clear the 3 bits of DCDC captured status: DCDC_OVER_VOL, DCDC_OVER_CUR, and DCDC_IN_LOW_VOL. 0b - No change 1b - Clear the 3 bits of DCDC captured status: DCDC_OVER_VOL, DCDC_OVER_CUR, and DCDC_IN_LOW_VOL
0 —	Reserved

12.4.2 IOMUXC SNVS register descriptions

12.4.2.1 iomuxc_snvs memory map

IOMUXC_SNVS base address: 40C9_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SW_MUX_CTL_PAD_WAKEUP_DIG SW MUX Control Register (SW_MUX_CTL_PAD_WAKEUP_DIG)	32	RW	0000_0005h
4h	SW_MUX_CTL_PAD_PMIC_ON_REQ_DIG SW MUX Control Register (SW_MUX_CTL_PAD_PMIC_ON_REQ_DIG)	32	RW	0000_0000h
8h	SW_MUX_CTL_PAD_PMIC_STBY_REQ_DIG SW MUX Control Register (SW_MUX_CTL_PAD_PMIC_STBY_REQ_DIG)	32	RW	0000_0000h
Ch	SW_MUX_CTL_PAD_GPIO_SNVS_00_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_00_DIG)	32	RW	0000_0005h
10h	SW_MUX_CTL_PAD_GPIO_SNVS_01_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_01_DIG)	32	RW	0000_0005h
14h	SW_MUX_CTL_PAD_GPIO_SNVS_02_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_02_DIG)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
18h	SW_MUX_CTL_PAD_GPIO_SNVS_03_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_03_DIG)	32	RW	0000_0005h
1Ch	SW_MUX_CTL_PAD_GPIO_SNVS_04_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_04_DIG)	32	RW	0000_0005h
20h	SW_MUX_CTL_PAD_GPIO_SNVS_05_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_05_DIG)	32	RW	0000_0005h
24h	SW_MUX_CTL_PAD_GPIO_SNVS_06_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_06_DIG)	32	RW	0000_0005h
28h	SW_MUX_CTL_PAD_GPIO_SNVS_07_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_07_DIG)	32	RW	0000_0005h
2Ch	SW_MUX_CTL_PAD_GPIO_SNVS_08_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_08_DIG)	32	RW	0000_0005h
30h	SW_MUX_CTL_PAD_GPIO_SNVS_09_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_09_DIG)	32	RW	0000_0005h
34h	SW_PAD_CTL_PAD_TEST_MODE_DIG SW PAD Control Register (SW_PAD_CTL_PAD_TEST_MODE_DIG)	32	RW	0000_0006h
38h	SW_PAD_CTL_PAD_POR_B_DIG SW PAD Control Register (SW_PAD_CTL_PAD_POR_B_DIG)	32	RW	0000_000Eh
3Ch	SW_PAD_CTL_PAD_ONOFF_DIG SW PAD Control Register (SW_PAD_CTL_PAD_ONOFF_DIG)	32	RW	0000_000Eh
40h	SW_PAD_CTL_PAD_WAKEUP_DIG SW PAD Control Register (SW_PAD_CTL_PAD_WAKEUP_DIG)	32	RW	0000_000Eh
44h	SW_PAD_CTL_PAD_PMIC_ON_REQ_DIG SW PAD Control Register (SW_PAD_CTL_PAD_PMIC_ON_REQ_DIG)	32	RW	0000_000Ah
48h	SW_PAD_CTL_PAD_PMIC_STBY_REQ_DIG SW PAD Control Register (SW_PAD_CTL_PAD_PMIC_STBY_REQ_DIG)	32	RW	0000_000Ah
4Ch	SW_PAD_CTL_PAD_GPIO_SNVS_00_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_00_DIG)	32	RW	0000_0002h
50h	SW_PAD_CTL_PAD_GPIO_SNVS_01_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_01_DIG)	32	RW	0000_0002h
54h	SW_PAD_CTL_PAD_GPIO_SNVS_02_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_02_DIG)	32	RW	0000_0002h
58h	SW_PAD_CTL_PAD_GPIO_SNVS_03_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_03_DIG)	32	RW	0000_0002h
5Ch	SW_PAD_CTL_PAD_GPIO_SNVS_04_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_04_DIG)	32	RW	0000_0002h
60h	SW_PAD_CTL_PAD_GPIO_SNVS_05_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_05_DIG)	32	RW	0000_0002h
64h	SW_PAD_CTL_PAD_GPIO_SNVS_06_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_06_DIG)	32	RW	0000_0002h
68h	SW_PAD_CTL_PAD_GPIO_SNVS_07_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_07_DIG)	32	RW	0000_0002h
6Ch	SW_PAD_CTL_PAD_GPIO_SNVS_08_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_08_DIG)	32	RW	0000_0002h
70h	SW_PAD_CTL_PAD_GPIO_SNVS_09_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_09_DIG)	32	RW	0000_0002h

12.4.2.2 SW_MUX_CTL_PAD_WAKEUP_DIG SW MUX Control Register (SW_MUX_CTL_PAD_WAKEUP_DIG)

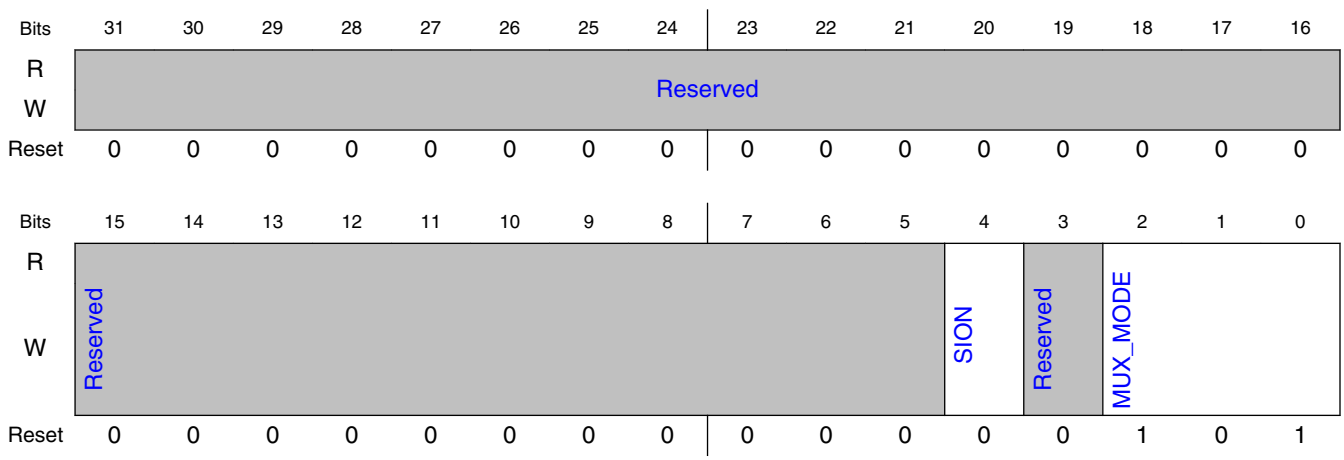
12.4.2.2.1 Offset

Register	Offset
SW_MUX_CTL_PAD_WAKEUP_DIG	0h

12.4.2.2.2 Function

SW_MUX_CTL Register

12.4.2.2.3 Diagram



12.4.2.2.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad WAKEUP_DIG
3 —	- Reserved

Table continues on the next page...

Field	Function
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: WAKEUP_DIG. 101b - Select mux mode: ALT5 mux port: GPIO13_IO00 of instance: GPIO13 111b - Select mux mode: ALT7 mux port: NMI_GLUE_NMI of instance: NMI_GLUE

12.4.2.3 SW_MUX_CTL_PAD_PMIC_ON_REQ_DIG SW MUX Control Register (SW_MUX_CTL_PAD_PMIC_ON_REQ_DIG)

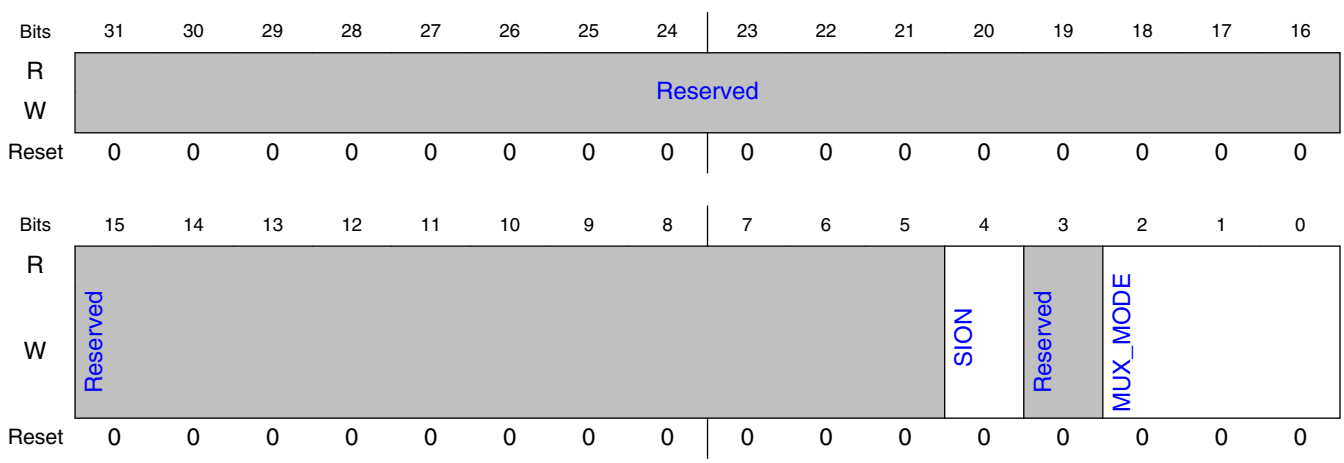
12.4.2.3.1 Offset

Register	Offset
SW_MUX_CTL_PAD_PMIC_ON_REQ_DIG	4h

12.4.2.3.2 Function

SW_MUX_CTL Register

12.4.2.3.3 Diagram



12.4.2.3.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad PMIC_ON_REQ_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: PMIC_ON_REQ_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_LP_PMIC_ON_REQ of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO01 of instance: GPIO13

12.4.2.4 SW_MUX_CTL_PAD_PMIC_STBY_REQ_DIG SW MUX Control Register (SW_MUX_CTL_PAD_PMIC_STBY_REQ_DIG)

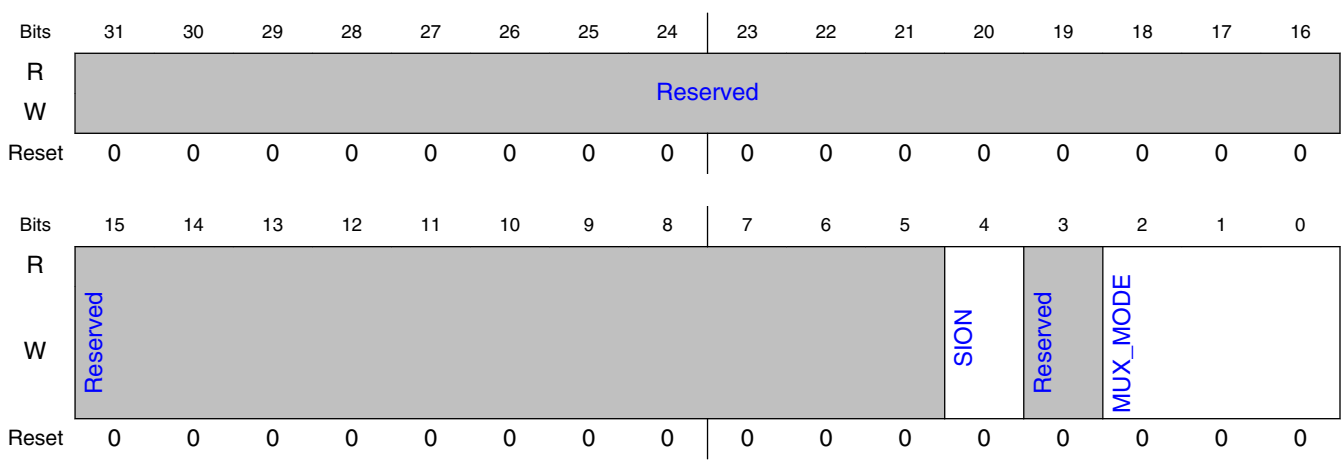
12.4.2.4.1 Offset

Register	Offset
SW_MUX_CTL_PAD_PMIC_STBY_REQ_DIG	8h

12.4.2.4.2 Function

SW_MUX_CTL Register

12.4.2.4.3 Diagram



12.4.2.4.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad PMIC_STBY_REQ_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: PMIC_STBY_REQ_DIG. 000b - Select mux mode: ALT0 mux port: CCM_PMIC_VSTBY_REQ of instance: CCM 101b - Select mux mode: ALT5 mux port: GPIO13_IO02 of instance: GPIO13

12.4.2.5 SW_MUX_CTL_PAD_GPIO_SNVS_00_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_00_DIG)

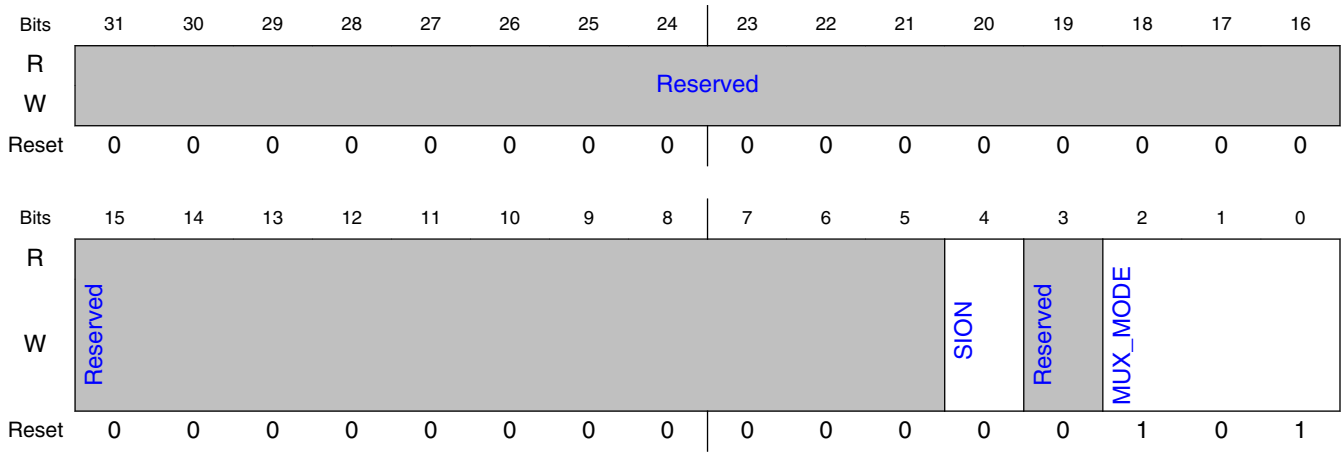
12.4.2.5.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_00_DIG	Ch

12.4.2.5.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.5.3 Diagram



12.4.2.5.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_00_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_00_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER0 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO03 of instance: GPIO13

12.4.2.6 SW_MUX_CTL_PAD_GPIO_SNVS_01_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_01_DIG)

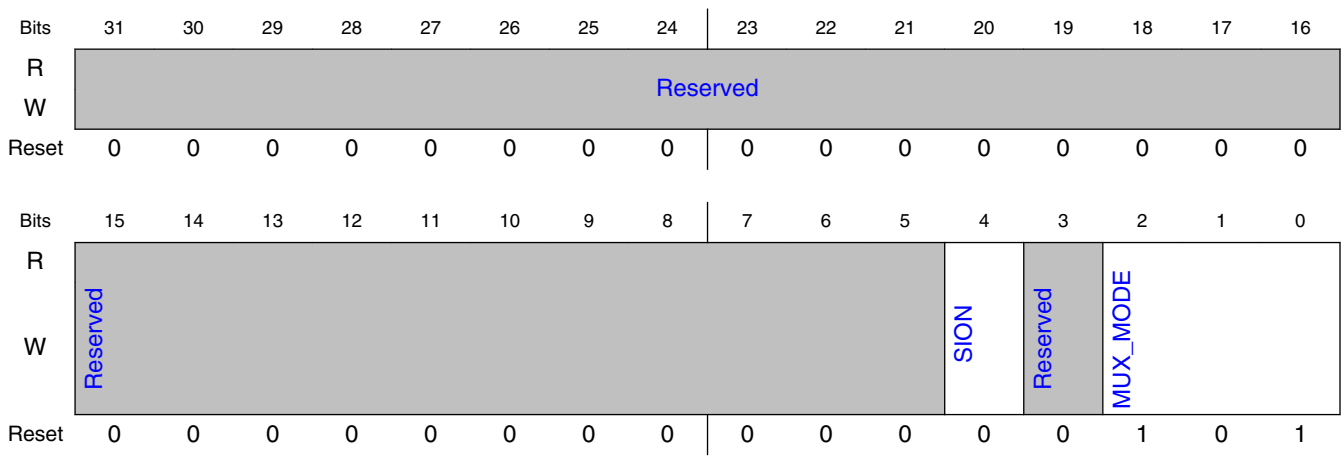
12.4.2.6.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_01_DIG	10h

12.4.2.6.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.6.3 Diagram



12.4.2.6.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_01_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_01_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER1 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO04 of instance: GPIO13

12.4.2.7 SW_MUX_CTL_PAD_GPIO_SNVS_02_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_02_DIG)

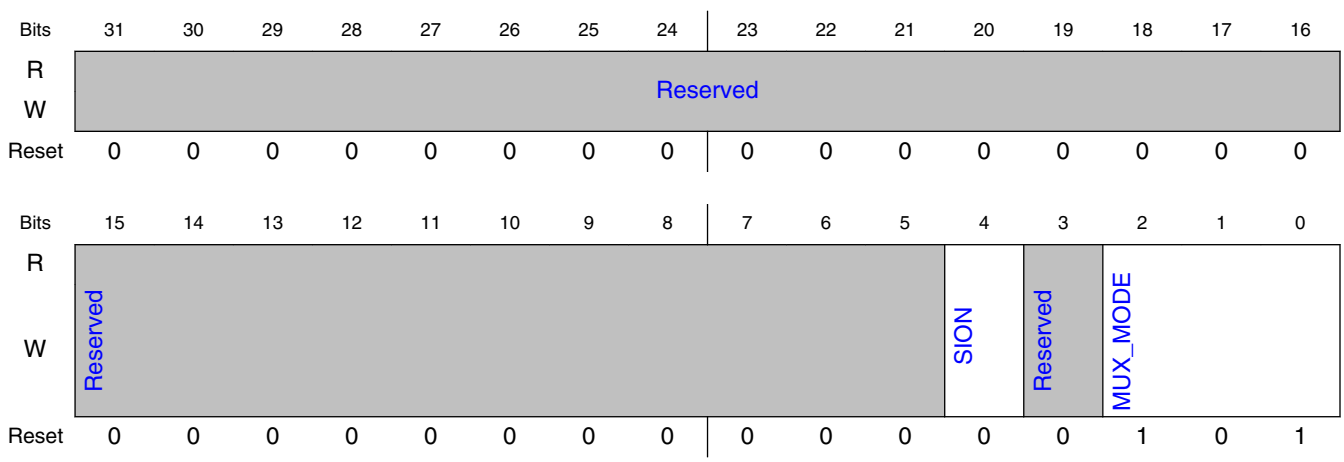
12.4.2.7.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_02_DIG	14h

12.4.2.7.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.7.3 Diagram



12.4.2.7.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_02_DIG
3	-

Table continues on the next page...

Field	Function
—	Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_02_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER2 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO05 of instance: GPIO13

12.4.2.8 SW_MUX_CTL_PAD_GPIO_SNVS_03_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_03_DIG)

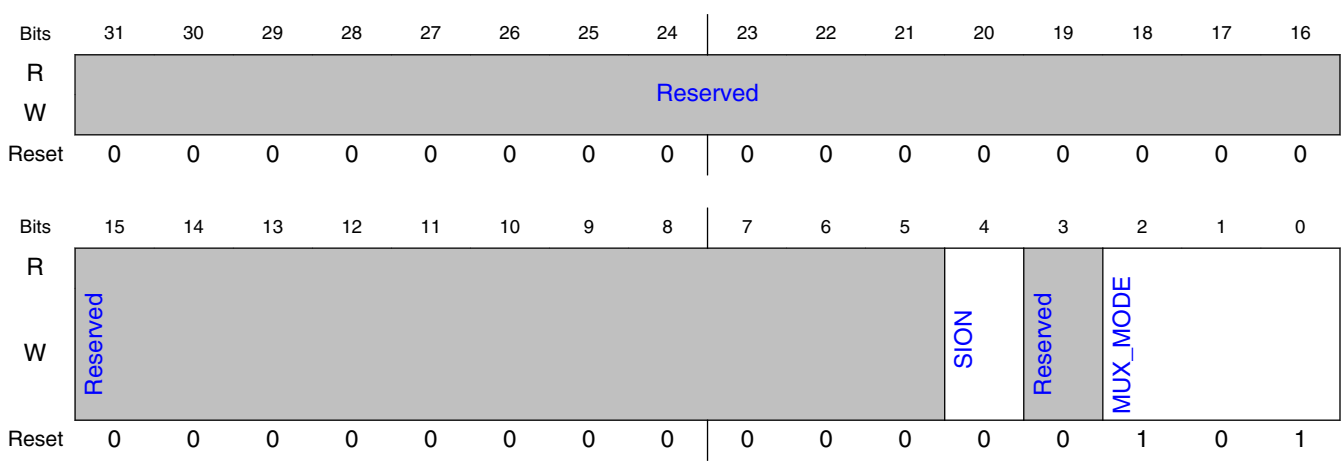
12.4.2.8.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_03_DIG	18h

12.4.2.8.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.8.3 Diagram



12.4.2.8.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_03_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_03_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER3 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO06 of instance: GPIO13

12.4.2.9 SW_MUX_CTL_PAD_GPIO_SNVS_04_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_04_DIG)

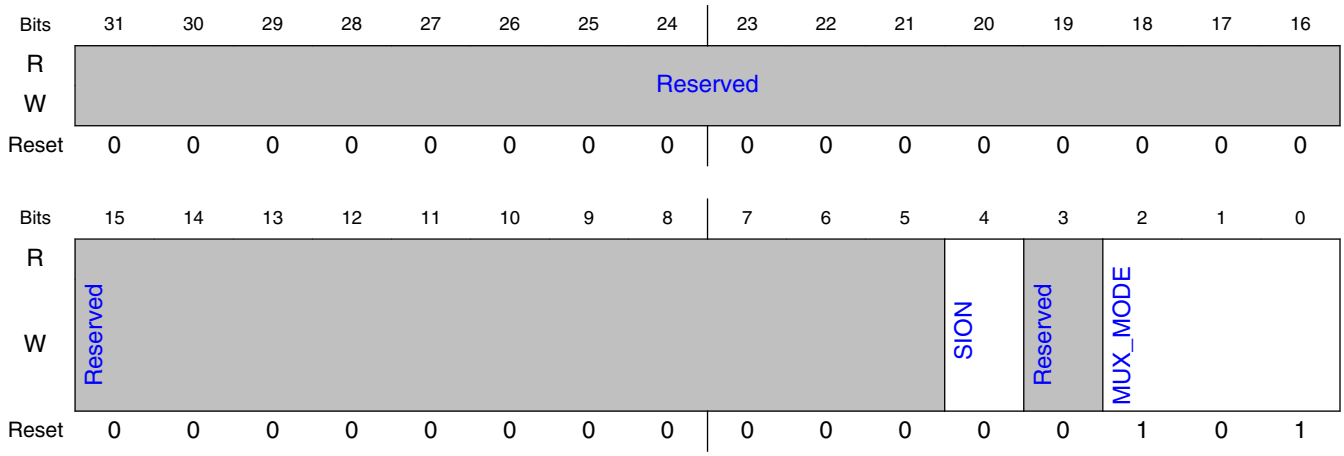
12.4.2.9.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_04_DIG	1Ch

12.4.2.9.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.9.3 Diagram



12.4.2.9.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_04_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_04_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER4 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO07 of instance: GPIO13

12.4.2.10 SW_MUX_CTL_PAD_GPIO_SNVS_05_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_05_DIG)

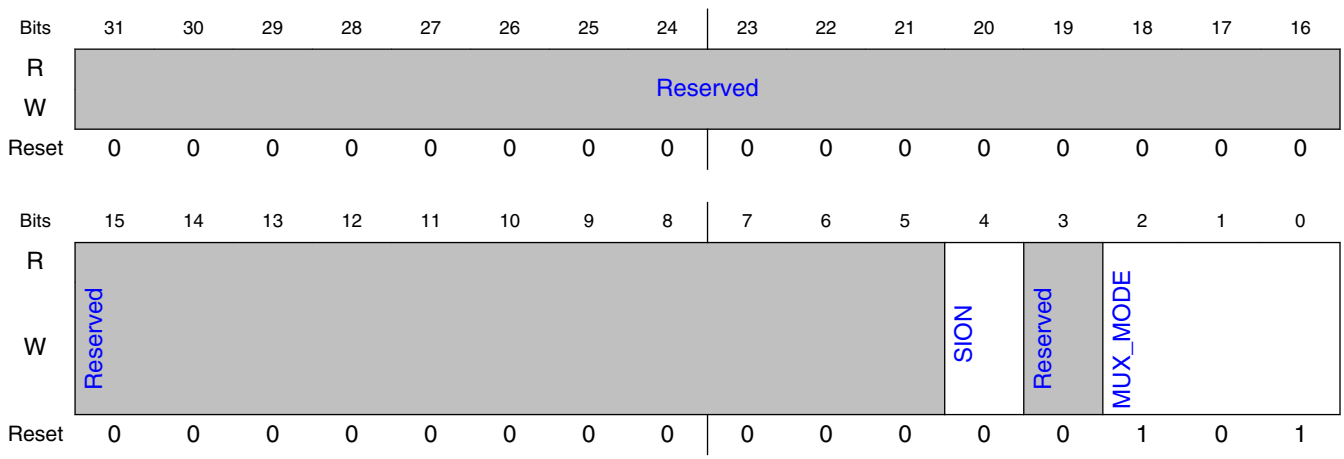
12.4.2.10.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_05_DIG	20h

12.4.2.10.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.10.3 Diagram



12.4.2.10.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_05_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_05_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER5 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO08 of instance: GPIO13

12.4.2.11 SW_MUX_CTL_PAD_GPIO_SNVS_06_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_06_DIG)

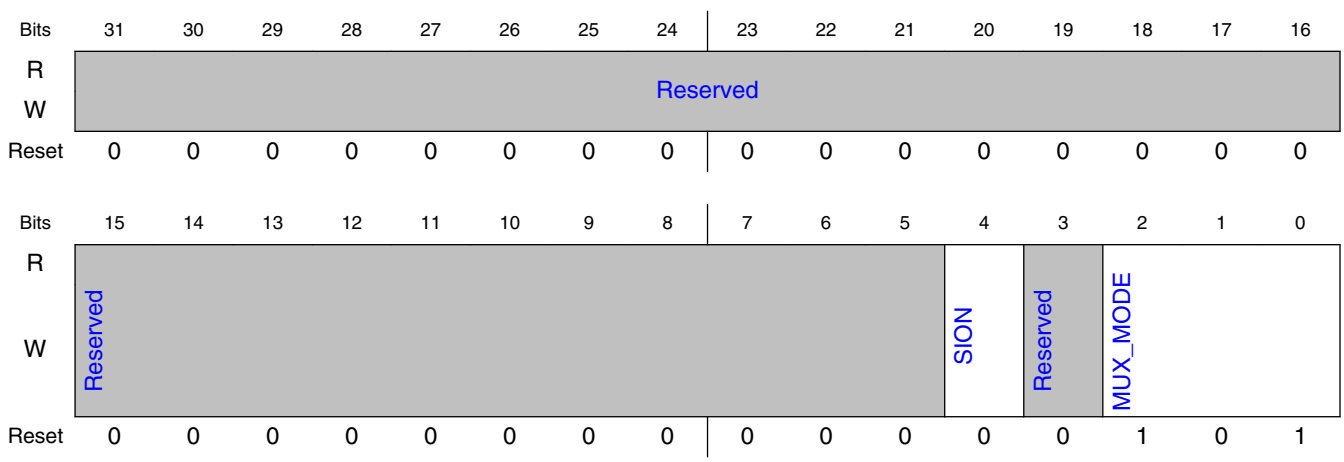
12.4.2.11.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_06_DIG	24h

12.4.2.11.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.11.3 Diagram



12.4.2.11.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_06_DIG
3	-

Table continues on the next page...

Field	Function
—	Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_06_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER6 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO09 of instance: GPIO13

12.4.2.12 SW_MUX_CTL_PAD_GPIO_SNVS_07_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_07_DIG)

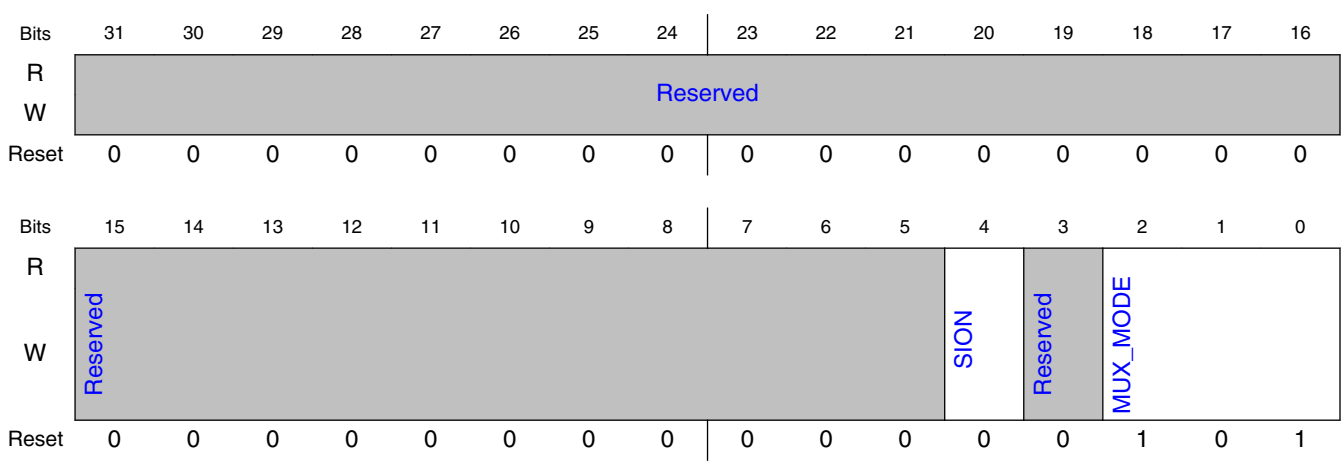
12.4.2.12.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_07_DIG	28h

12.4.2.12.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.12.3 Diagram



12.4.2.12.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_07_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_07_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER7 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO10 of instance: GPIO13

12.4.2.13 SW_MUX_CTL_PAD_GPIO_SNVS_08_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_08_DIG)

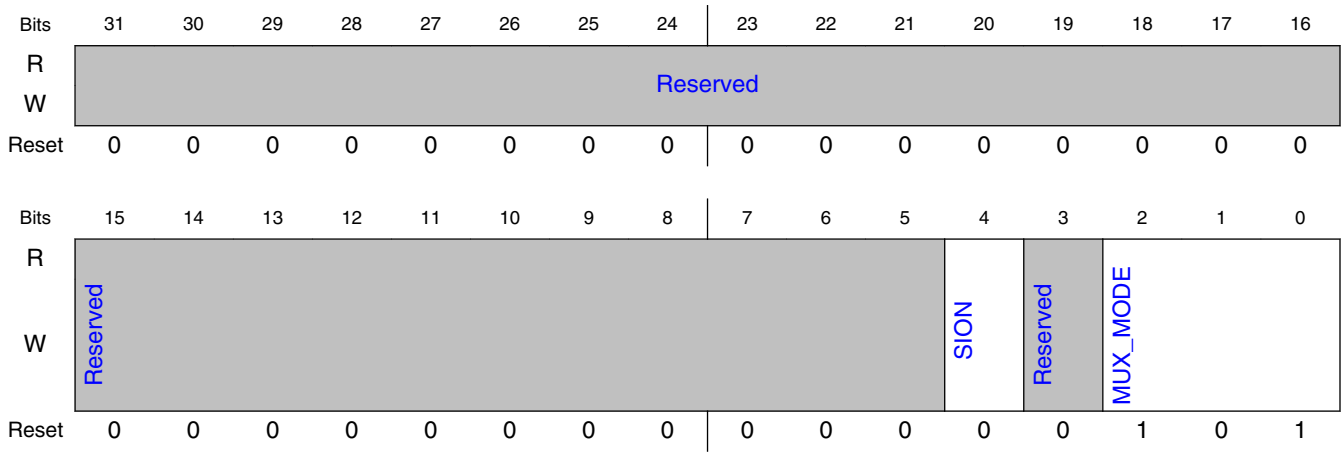
12.4.2.13.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_08_DIG	2Ch

12.4.2.13.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.13.3 Diagram



12.4.2.13.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_08_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_08_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER8 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO11 of instance: GPIO13

12.4.2.14 SW_MUX_CTL_PAD_GPIO_SNVS_09_DIG SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SNVS_09_DIG)

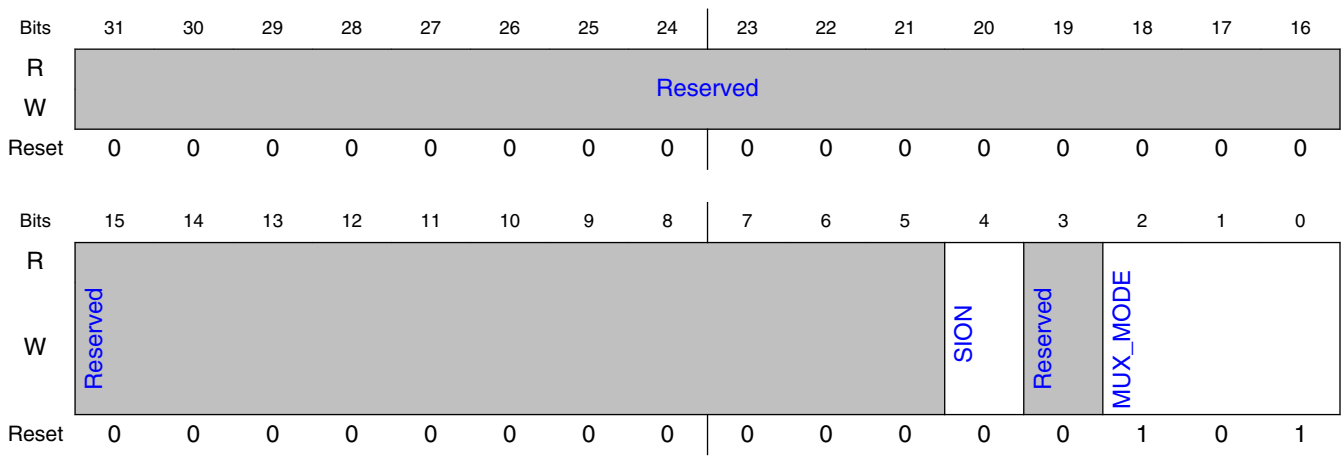
12.4.2.14.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SNVS_09_DIG	30h

12.4.2.14.2 Function

The functionality of this pin is determined by the part number. Tamper function is available only on tamper-enabled parts, and GPIO is the only available function on parts which do not support tamper. The MUX_MODE must be configured to select the function in both cases. Refer to the Datasheet for details on part numbers.

12.4.2.14.3 Diagram



12.4.2.14.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SNVS_09_DIG
3 —	- Reserved
2-0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_SNVS_09_DIG. 000b - Select mux mode: ALT0 mux port: SNVS_TAMPER9 of instance: SNVS_LP 101b - Select mux mode: ALT5 mux port: GPIO13_IO12 of instance: GPIO13

12.4.2.15 SW_PAD_CTL_PAD_TEST_MODE_DIG SW PAD Control Register (SW_PAD_CTL_PAD_TEST_MODE_DIG)

12.4.2.15.1 Offset

Register	Offset
SW_PAD_CTL_PAD_TEST_MODE_DIG	34h

12.4.2.15.2 Function

SW_PAD_CTL Register

12.4.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWP_LOCK		DWP		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									Reserved	Reserved		PUS	PUE	Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

12.4.2.15.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed

Table continues on the next page...

Field	Function
	01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 —	Reserved
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: TEST_MODE_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: TEST_MODE_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.16 SW_PAD_CTL_PAD_POR_B_DIG SW PAD Control Register (SW_PAD_CTL_PAD_POR_B_DIG)

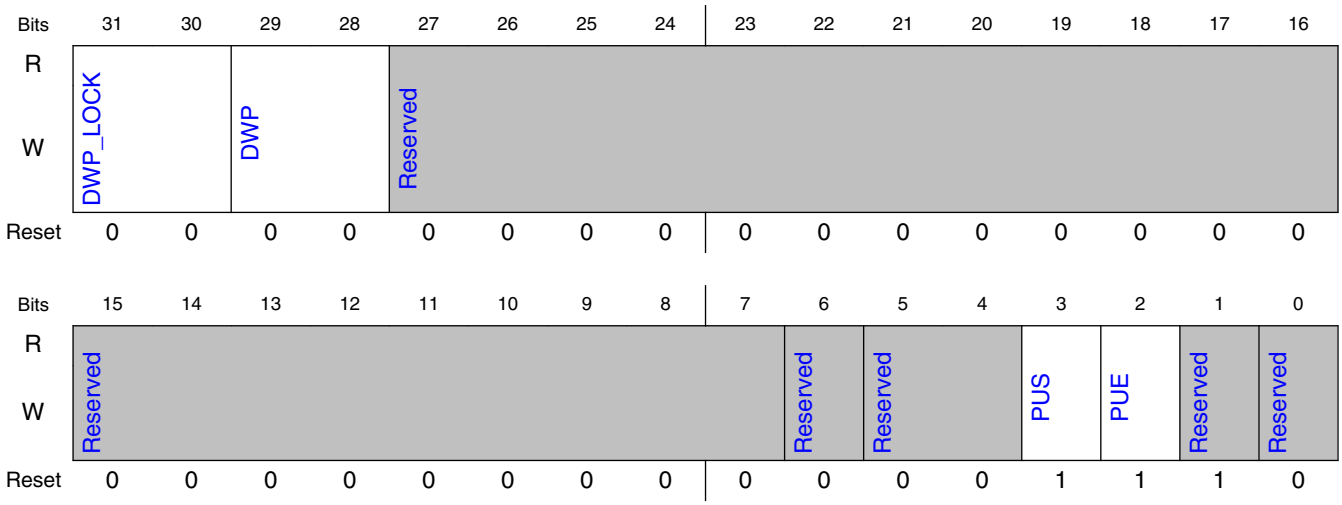
12.4.2.16.1 Offset

Register	Offset
SW_PAD_CTL_PAD_POR_B_DIG	38h

12.4.2.16.2 Function

SW_PAD_CTL Register

12.4.2.16.3 Diagram



12.4.2.16.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 —	Reserved
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: POR_B_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: POR_B_DIG 0b - Pull Disable

Table continues on the next page...

Field	Function
	1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.17 SW_PAD_CTL_PAD_ONOFF_DIG SW PAD Control Register (SW_PAD_CTL_PAD_ONOFF_DIG)

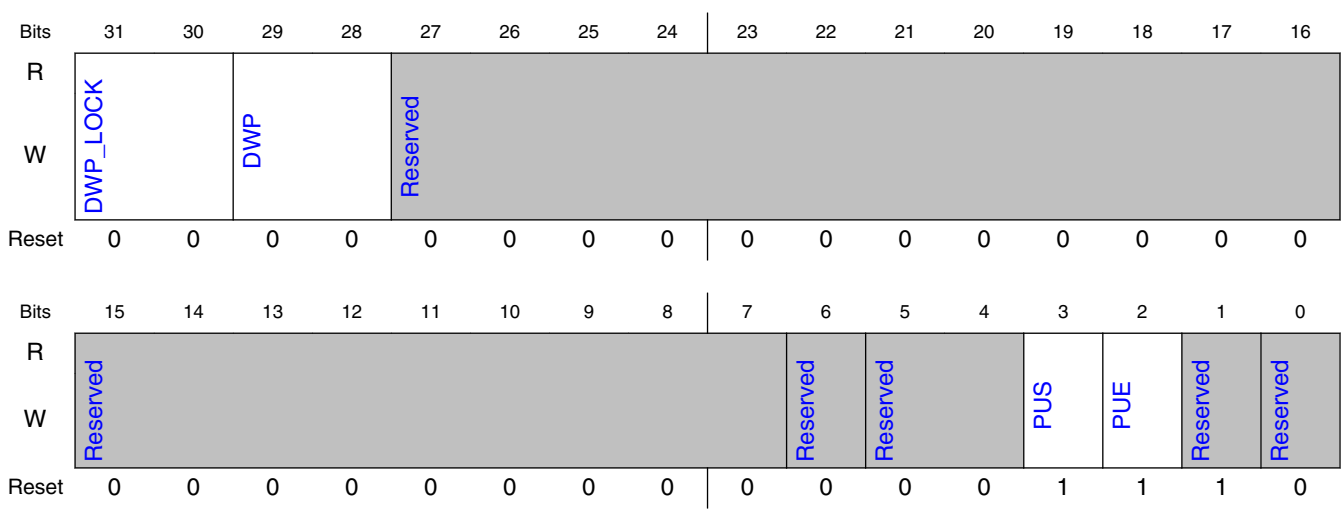
12.4.2.17.1 Offset

Register	Offset
SW_PAD_CTL_PAD_ONOFF_DIG	3Ch

12.4.2.17.2 Function

SW_PAD_CTL Register

12.4.2.17.3 Diagram



12.4.2.17.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 —	Reserved
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: ONOFF_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: ONOFF_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.18 SW_PAD_CTL_PAD_WAKEUP_DIG SW PAD Control Register (SW_PAD_CTL_PAD_WAKEUP_DIG)

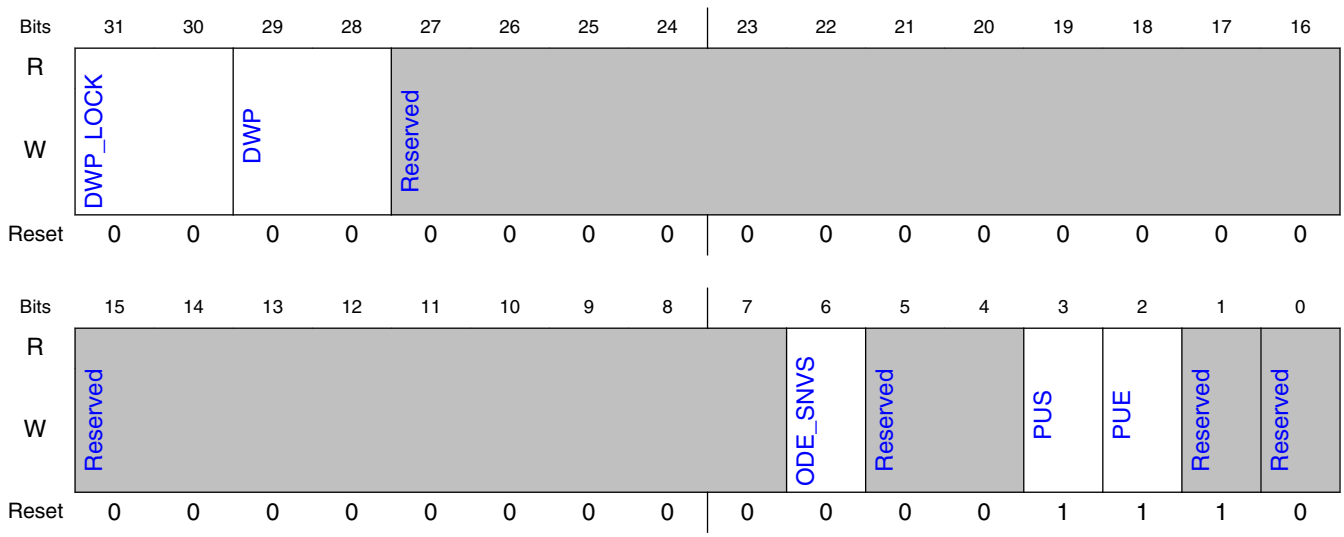
12.4.2.18.1 Offset

Register	Offset
SW_PAD_CTL_PAD_WAKEUP_DIG	40h

12.4.2.18.2 Function

SW_PAD_CTL Register

12.4.2.18.3 Diagram



12.4.2.18.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: WAKEUP_DIG 0b - Disabled 1b - Enabled
5-4	Reserved

Table continues on the next page...

Field	Function
—	
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: WAKEUP_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: WAKEUP_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.19 SW_PAD_CTL_PAD_PMIC_ON_REQ_DIG SW PAD Control Register (SW_PAD_CTL_PAD_PMIC_ON_REQ_DIG)

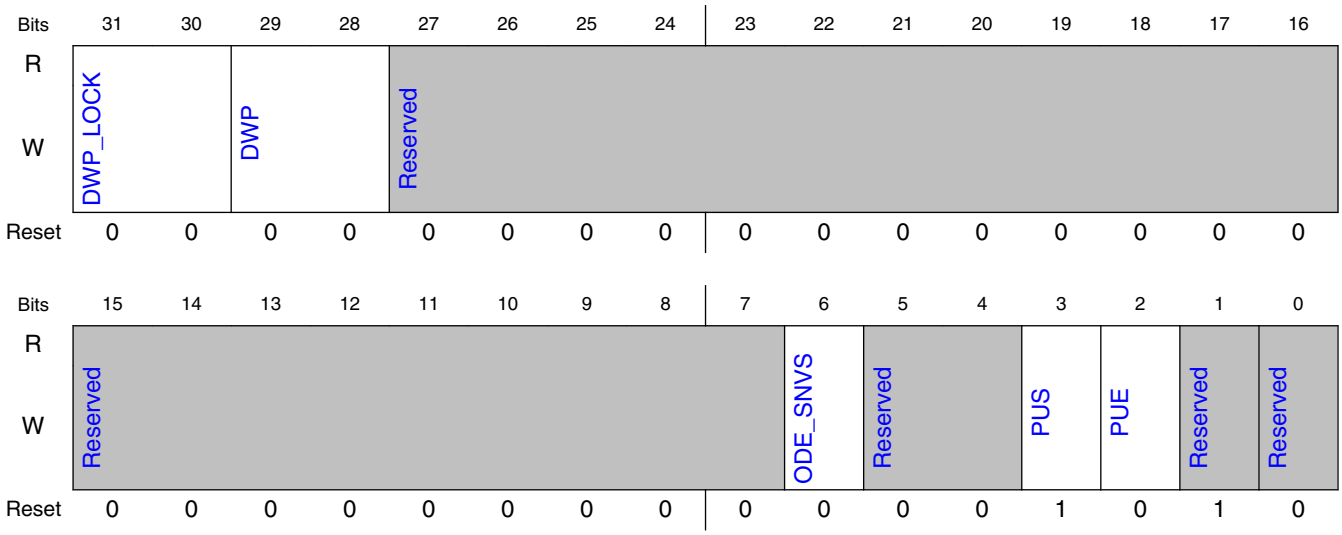
12.4.2.19.1 Offset

Register	Offset
SW_PAD_CTL_PAD_PMIC_ON_REQ_DIG	44h

12.4.2.19.2 Function

SW_PAD_CTL Register

12.4.2.19.3 Diagram



12.4.2.19.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: PMIC_ON_REQ_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PMIC_ON_REQ_DIG 0b - Weak pull down 1b - Weak pull up

Table continues on the next page...

Field	Function
2 PUE	Pull / Keep Select Field Select one out of next values for pad: PMIC_ON_REQ_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.20 SW_PAD_CTL_PAD_PMIC_STBY_REQ_DIG SW PAD Control Register (SW_PAD_CTL_PAD_PMIC_STBY_REQ_DIG)

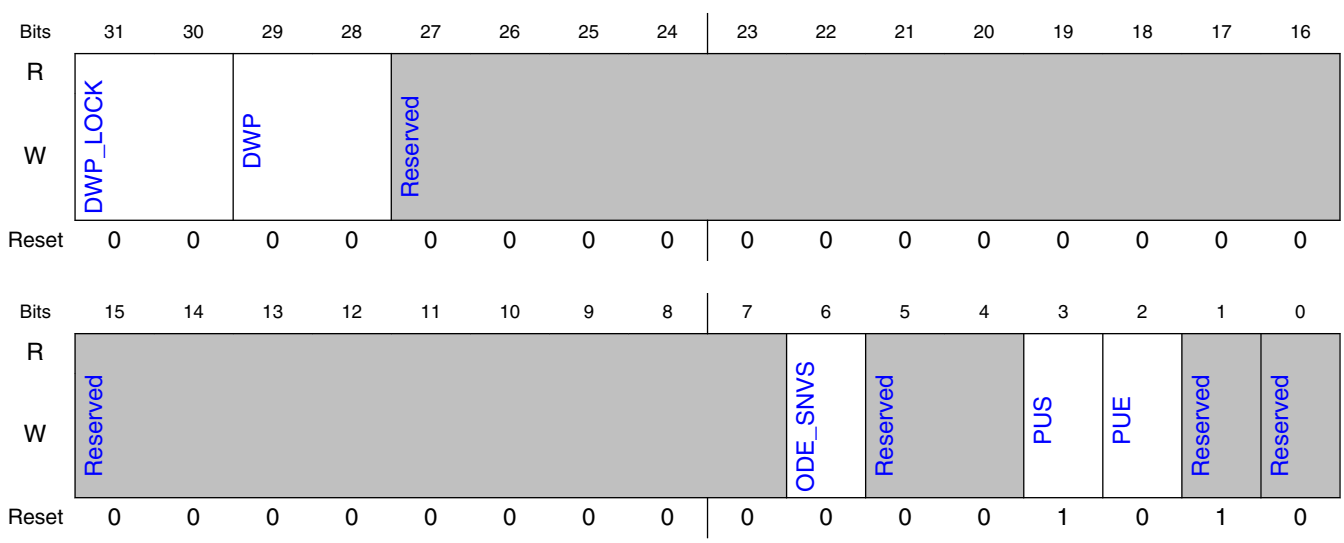
12.4.2.20.1 Offset

Register	Offset
SW_PAD_CTL_PAD_PMIC_STBY_REQ_DIG	48h

12.4.2.20.2 Function

SW_PAD_CTL Register

12.4.2.20.3 Diagram



12.4.2.20.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: PMIC_STBY_REQ_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PMIC_STBY_REQ_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: PMIC_STBY_REQ_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.21 SW_PAD_CTL_PAD_GPIO_SNVS_00_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_00_DIG)

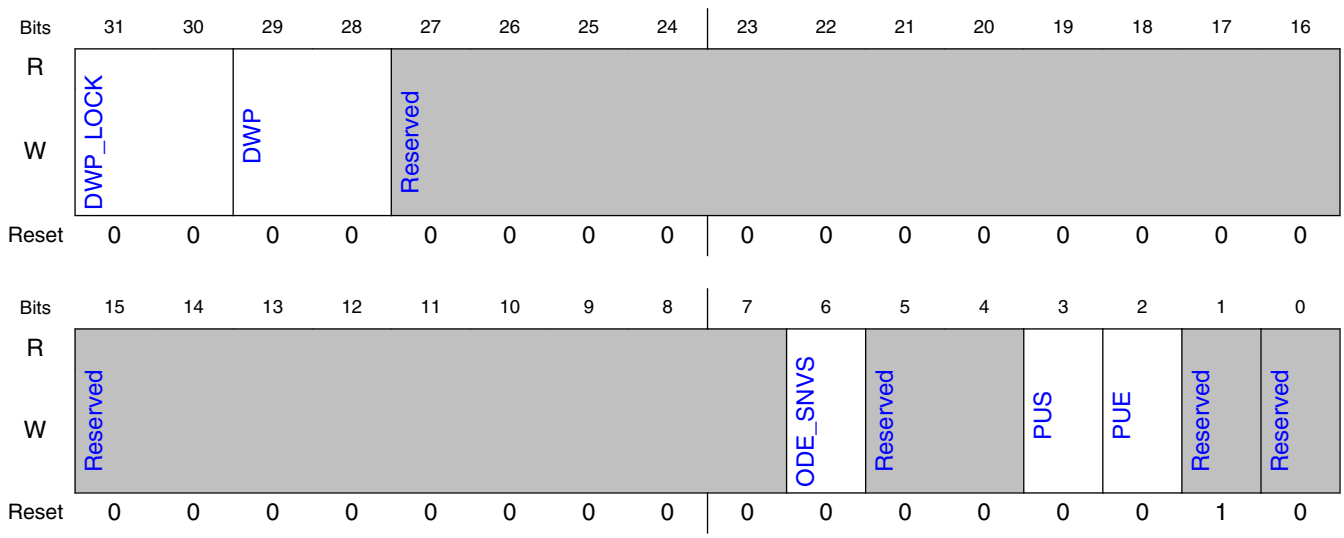
12.4.2.21.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_00_DIG	4Ch

12.4.2.21.2 Function

SW_PAD_CTL Register

12.4.2.21.3 Diagram



12.4.2.21.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7	-

Table continues on the next page...

Field	Function
—	Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_00_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_00_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_00_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.22 SW_PAD_CTL_PAD_GPIO_SNVS_01_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_01_DIG)

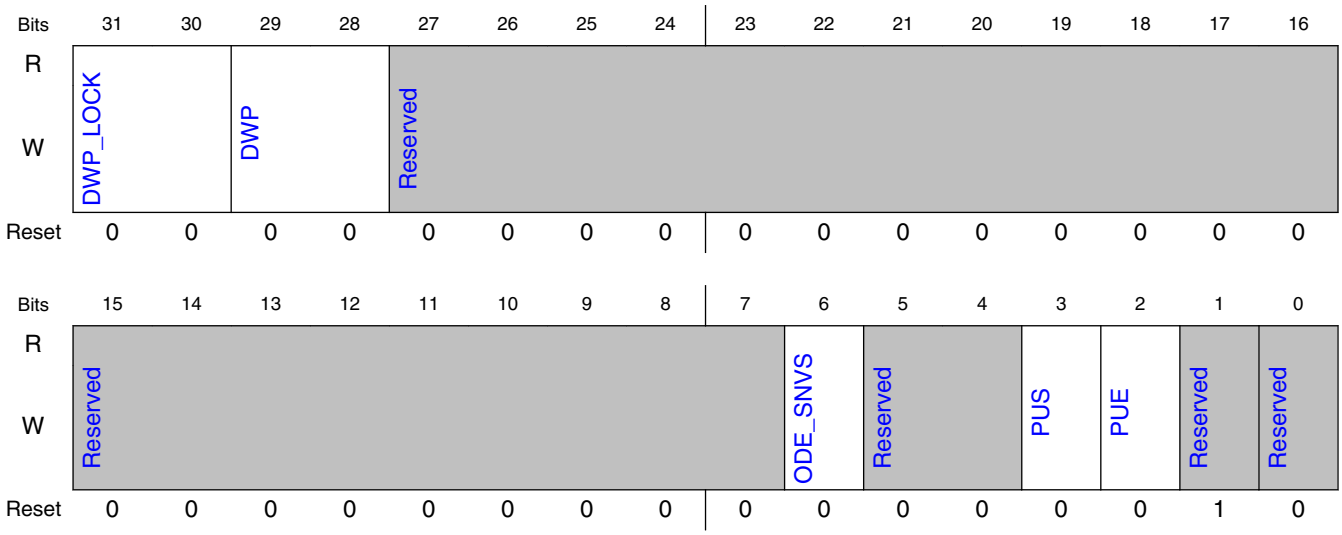
12.4.2.22.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_01_DIG	50h

12.4.2.22.2 Function

SW_PAD_CTL Register

12.4.2.22.3 Diagram



12.4.2.22.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_01_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_01_DIG 0b - Weak pull down 1b - Weak pull up

Table continues on the next page...

Field	Function
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_01_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.23 SW_PAD_CTL_PAD_GPIO_SNVS_02_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_02_DIG)

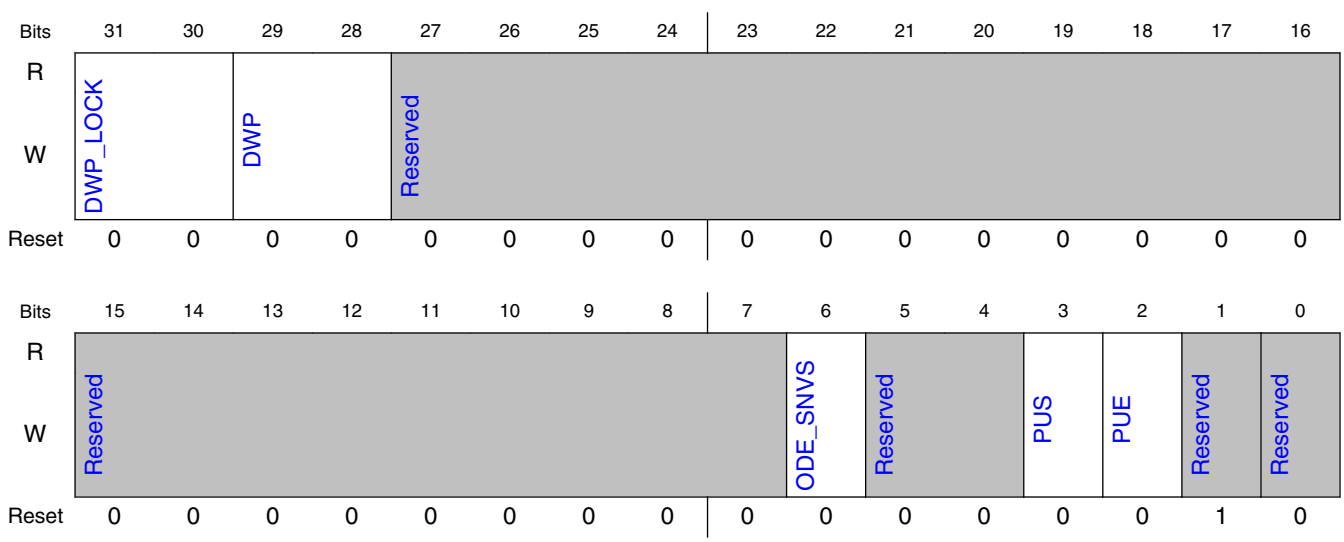
12.4.2.23.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_02_DIG	54h

12.4.2.23.2 Function

SW_PAD_CTL Register

12.4.2.23.3 Diagram



12.4.2.23.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_02_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_02_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_02_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.24 SW_PAD_CTL_PAD_GPIO_SNVS_03_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_03_DIG)

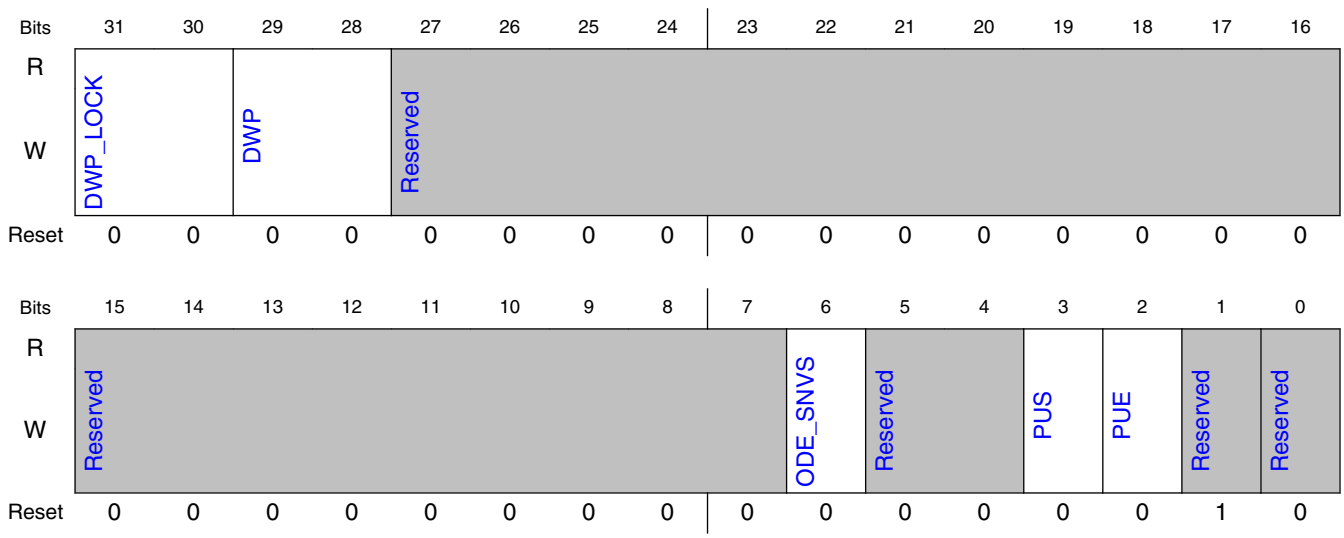
12.4.2.24.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_03_DIG	58h

12.4.2.24.2 Function

SW_PAD_CTL Register

12.4.2.24.3 Diagram



12.4.2.24.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7	-

Table continues on the next page...

Field	Function
—	Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_03_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_03_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_03_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.25 SW_PAD_CTL_PAD_GPIO_SNVS_04_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_04_DIG)

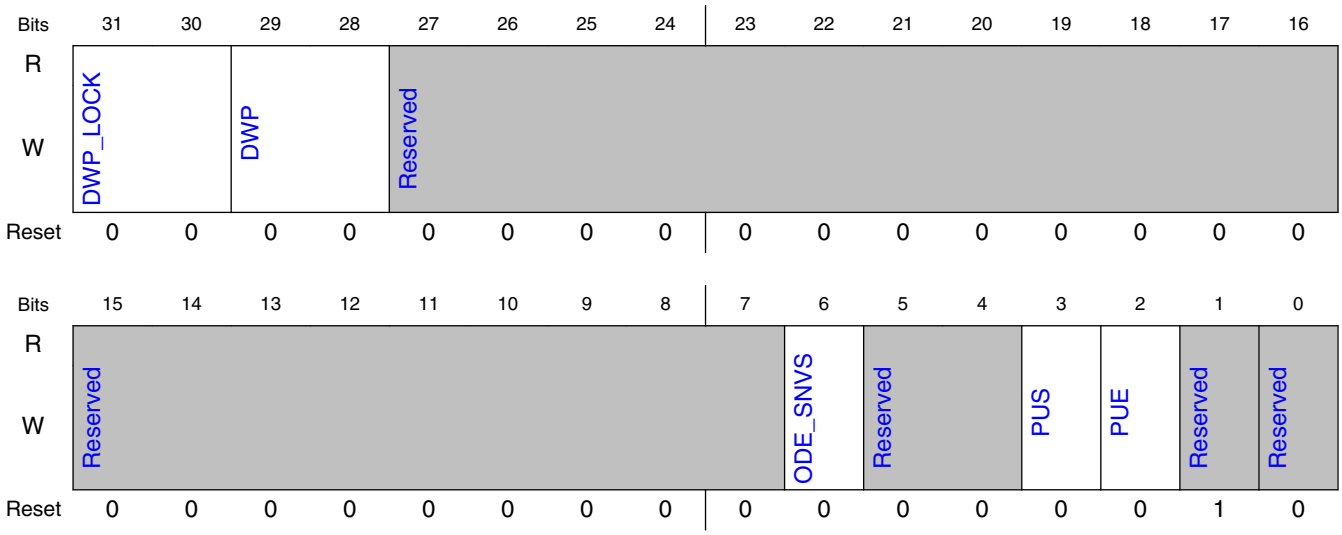
12.4.2.25.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_04_DIG	5Ch

12.4.2.25.2 Function

SW_PAD_CTL Register

12.4.2.25.3 Diagram



12.4.2.25.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_04_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_04_DIG 0b - Weak pull down 1b - Weak pull up

Table continues on the next page...

Field	Function
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_04_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.26 SW_PAD_CTL_PAD_GPIO_SNVS_05_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_05_DIG)

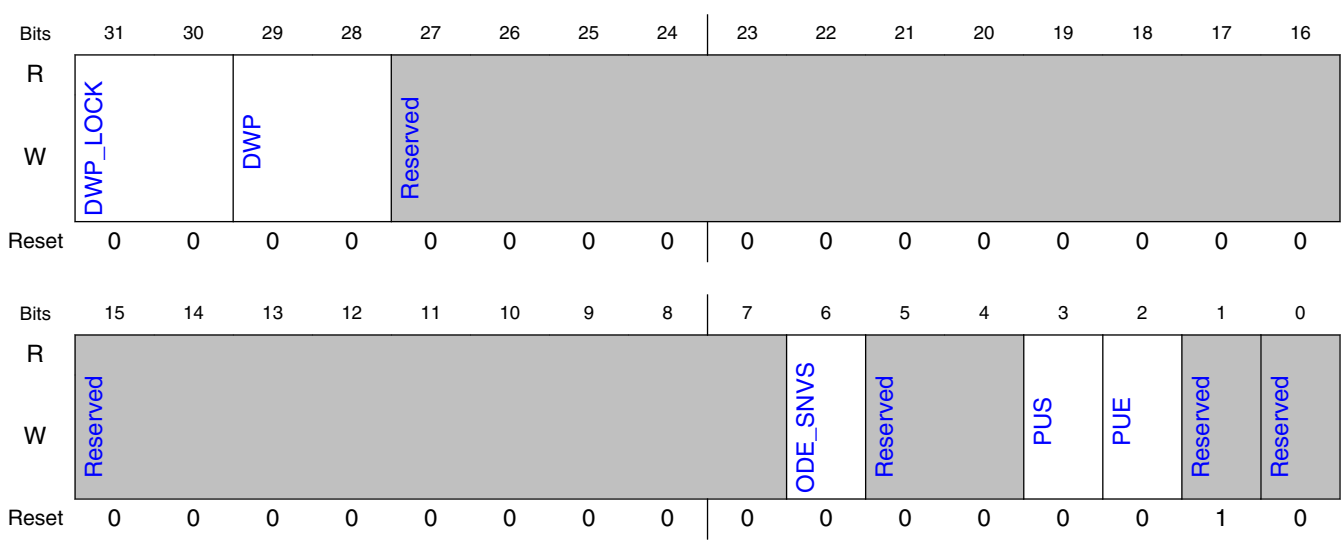
12.4.2.26.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_05_DIG	60h

12.4.2.26.2 Function

SW_PAD_CTL Register

12.4.2.26.3 Diagram



12.4.2.26.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_05_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_05_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_05_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.27 SW_PAD_CTL_PAD_GPIO_SNVS_06_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_06_DIG)

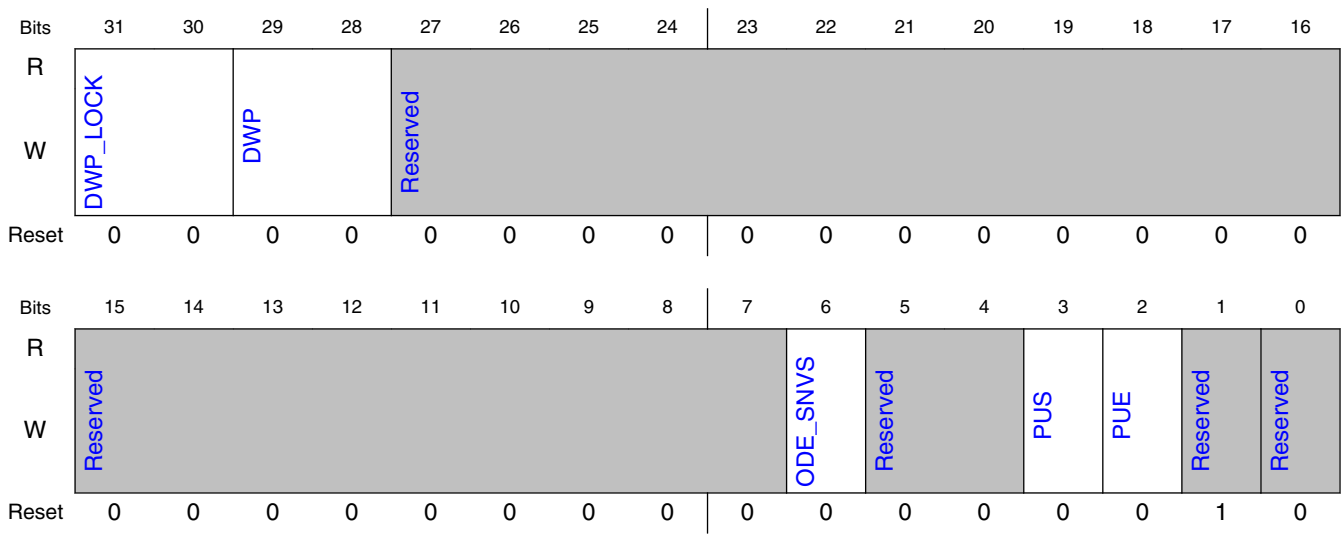
12.4.2.27.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_06_DIG	64h

12.4.2.27.2 Function

SW_PAD_CTL Register

12.4.2.27.3 Diagram



12.4.2.27.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7	-

Table continues on the next page...

Field	Function
—	Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_06_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_06_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_06_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.28 SW_PAD_CTL_PAD_GPIO_SNVS_07_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_07_DIG)

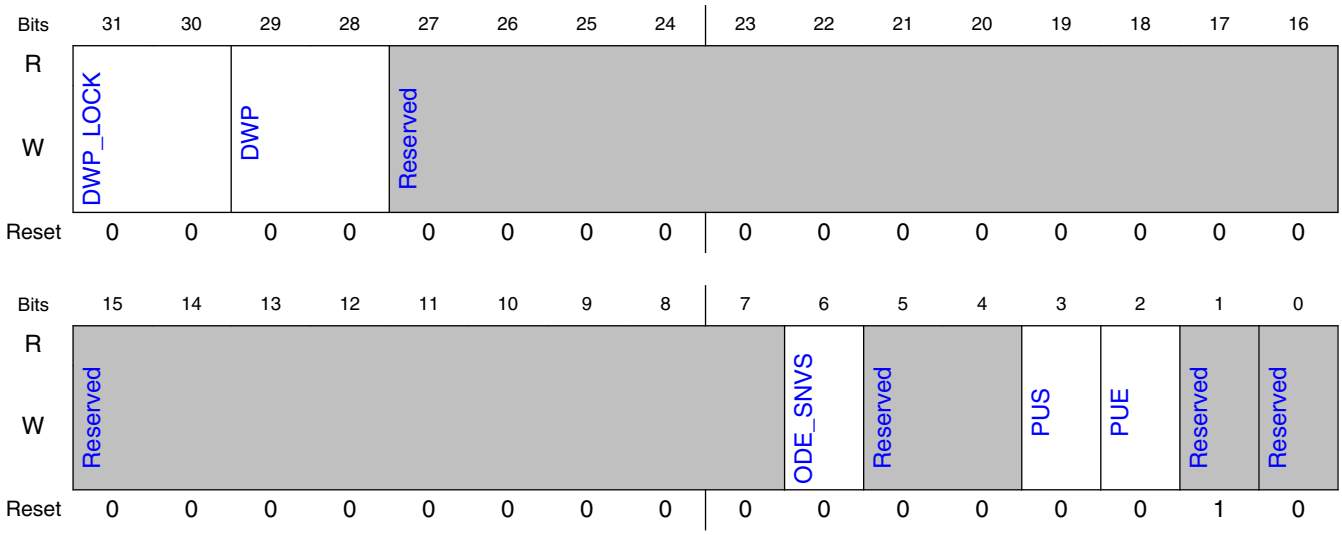
12.4.2.28.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_07_DIG	68h

12.4.2.28.2 Function

SW_PAD_CTL Register

12.4.2.28.3 Diagram



12.4.2.28.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_07_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_07_DIG 0b - Weak pull down 1b - Weak pull up

Table continues on the next page...

Field	Function
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_07_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.29 SW_PAD_CTL_PAD_GPIO_SNVS_08_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_08_DIG)

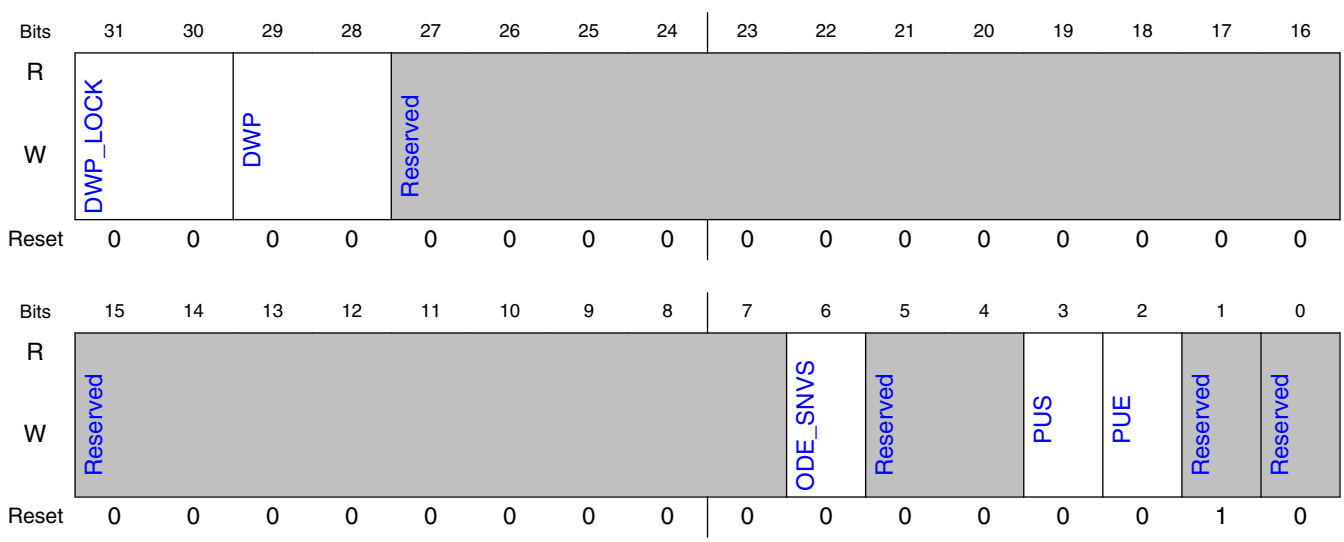
12.4.2.29.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_08_DIG	6Ch

12.4.2.29.2 Function

SW_PAD_CTL Register

12.4.2.29.3 Diagram



12.4.2.29.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7 —	- Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_08_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_08_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_08_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.2.30 SW_PAD_CTL_PAD_GPIO_SNVS_09_DIG SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SNVS_09_DIG)

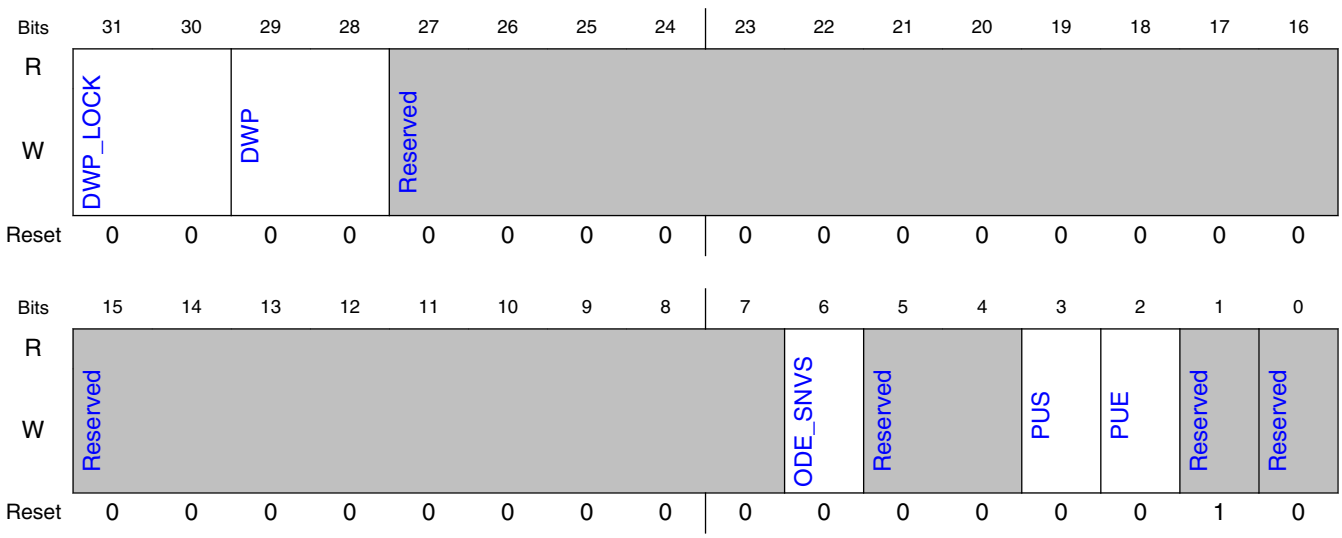
12.4.2.30.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SNVS_09_DIG	70h

12.4.2.30.2 Function

SW_PAD_CTL Register

12.4.2.30.3 Diagram



12.4.2.30.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-7	-

Table continues on the next page...

Field	Function
—	Reserved
6 ODE_SNVS	Open Drain SNVS Field Select one out of next values for pad: GPIO_SNVS_09_DIG 0b - Disabled 1b - Enabled
5-4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_SNVS_09_DIG 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_SNVS_09_DIG 0b - Pull Disable 1b - Pull Enable
1 —	Reserved
0 —	Reserved

12.4.3 IOMUXC LPSR GPR register descriptions

12.4.3.1 iomuxc_lpsr_gpr memory map

IOMUXC_LPSR_GPR base address: 40C0_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPR0 General Purpose Register (GPR0)	32	RW	0000_0400h
4h	GPR1 General Purpose Register (GPR1)	32	RW	0000_0020h
8h	GPR2 General Purpose Register (GPR2)	32	RW	0000_0000h
Ch	GPR3 General Purpose Register (GPR3)	32	RW	0000_0000h
10h	GPR4 General Purpose Register (GPR4)	32	RW	0000_0000h
14h	GPR5 General Purpose Register (GPR5)	32	RW	0000_0000h
18h	GPR6 General Purpose Register (GPR6)	32	RW	0000_0000h
1Ch	GPR7 General Purpose Register (GPR7)	32	RW	0000_0000h
20h	GPR8 General Purpose Register (GPR8)	32	RW	0000_0000h
24h	GPR9 General Purpose Register (GPR9)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
28h	GPR10 General Purpose Register (GPR10)	32	RW	0000_0000h
2Ch	GPR11 General Purpose Register (GPR11)	32	RW	0000_0000h
30h	GPR12 General Purpose Register (GPR12)	32	RW	0000_0000h
34h	GPR13 General Purpose Register (GPR13)	32	RW	0000_0000h
38h	GPR14 General Purpose Register (GPR14)	32	RW	0000_0000h
3Ch	GPR15 General Purpose Register (GPR15)	32	RW	0000_0000h
40h	GPR16 General Purpose Register (GPR16)	32	RW	0000_0000h
44h	GPR17 General Purpose Register (GPR17)	32	RW	0000_0000h
48h	GPR18 General Purpose Register (GPR18)	32	RW	0000_0000h
4Ch	GPR19 General Purpose Register (GPR19)	32	RW	0000_0000h
50h	GPR20 General Purpose Register (GPR20)	32	RW	0000_0000h
54h	GPR21 General Purpose Register (GPR21)	32	RW	0000_0000h
58h	GPR22 General Purpose Register (GPR22)	32	RW	0000_0000h
5Ch	GPR23 General Purpose Register (GPR23)	32	RW	0000_0000h
60h	GPR24 General Purpose Register (GPR24)	32	RW	0000_0000h
64h	GPR25 General Purpose Register (GPR25)	32	RW	0000_0000h
68h	GPR26 General Purpose Register (GPR26)	32	RW	0000_4000h
84h	GPR33 General Purpose Register (GPR33)	32	RW	0000_0000h
88h	GPR34 General Purpose Register (GPR34)	32	RW	0000_0E00h
8Ch	GPR35 General Purpose Register (GPR35)	32	RW	0000_0000h
90h	GPR36 General Purpose Register (GPR36)	32	RW	0000_0000h
94h	GPR37 General Purpose Register (GPR37)	32	RW	0000_0000h
98h	GPR38 General Purpose Register (GPR38)	32	RW	0000_0000h
9Ch	GPR39 General Purpose Register (GPR39)	32	RW	0000_0000h
A0h	GPR40 General Purpose Register (GPR40)	32	R	0000_0000h
A4h	GPR41 General Purpose Register (GPR41)	32	R	0000_0000h

12.4.3.2 GPR0 General Purpose Register (GPR0)

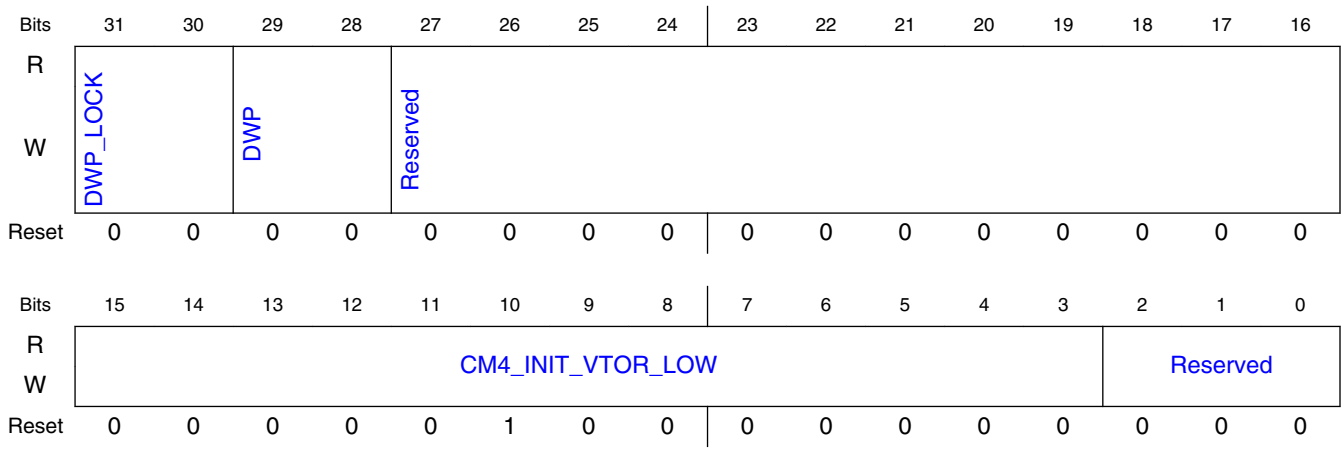
12.4.3.2.1 Offset

Register	Offset
GPR0	0h

12.4.3.2.2 Function

GPR Register

12.4.3.2.3 Diagram



12.4.3.2.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change CM4_INIT_VTOR_LOW field. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-3 CM4_INIT_VTOR_LOW	CM4 Vector table offset value lower bits out of reset The VTOR sets the offset of the vector table base address from memory address 0x00000000. This field contains bits 15:3 of CM4 VTOR. **NOTE:** When user uses HAB authenticated or encrypted boot function, NXP strongly suggest user to set the default value (ROM entry address) into INIT_VTOR register and lock the default value.
2-0 —	Reserved

12.4.3.3 GPR1 General Purpose Register (GPR1)

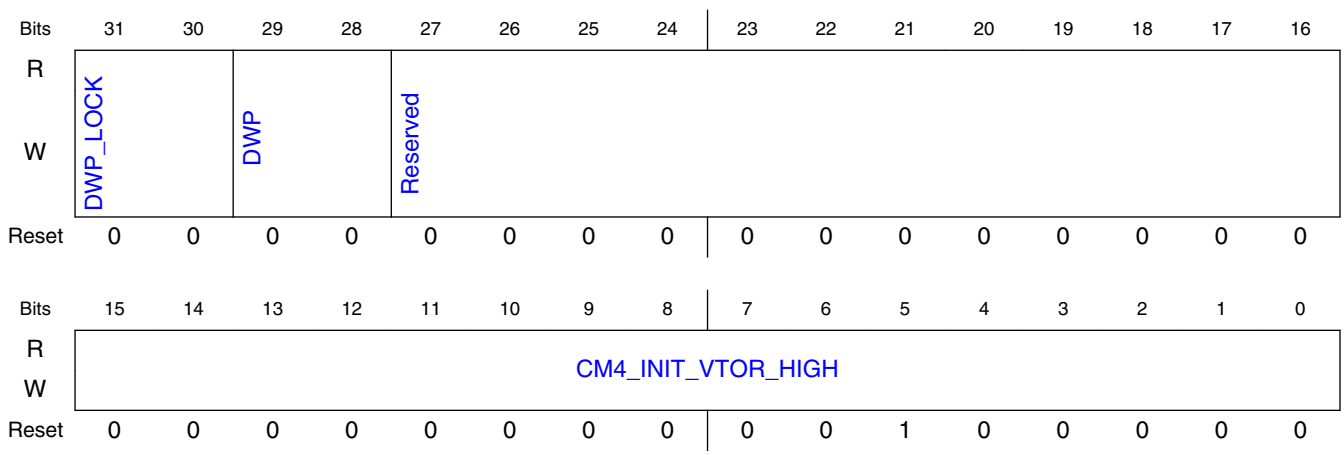
12.4.3.3.1 Offset

Register	Offset
GPR1	4h

12.4.3.3.2 Function

GPR Register

12.4.3.3.3 Diagram



12.4.3.3.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change CM4_INIT_VTOR_HIGH field. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 CM4_INIT_VTOR_HIGH	CM4 Vector table offset value higher bits out of reset The VTOR sets the offset of the vector table base address from memory address 0x00000000. This field contains bits 31:16 of CM4 VTOR. **NOTE:** When user uses HAB authenticated or encrypted boot function, NXP strongly suggest user to set the default value (ROM entry address) into INIT_VTOR register and lock the default value.

12.4.3.4 GPR2 General Purpose Register (GPR2)

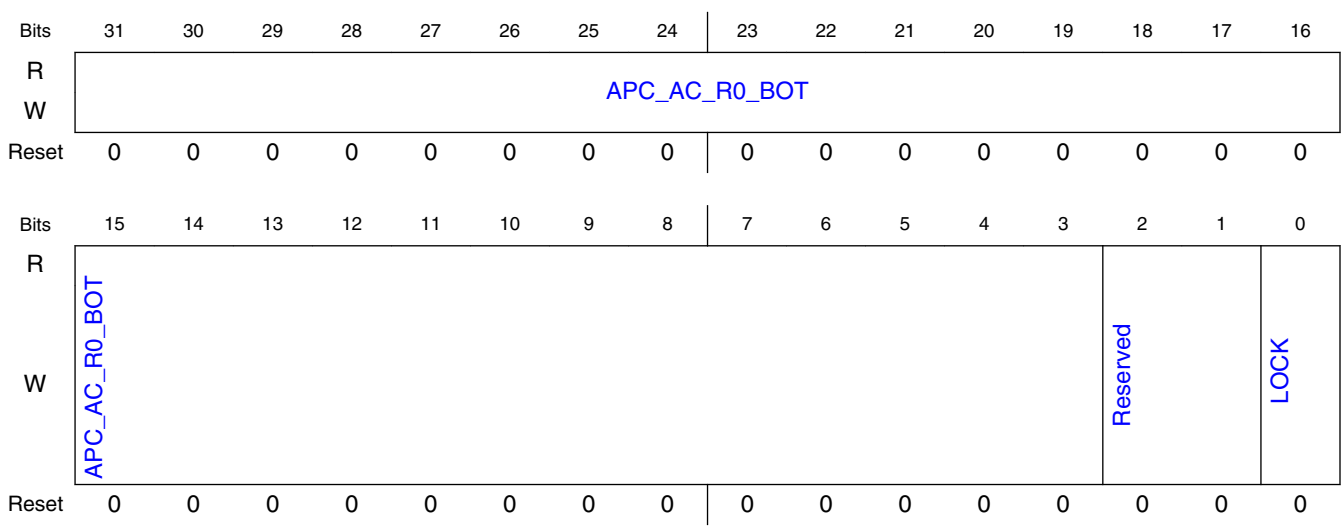
12.4.3.4.1 Offset

Register	Offset
GPR2	8h

12.4.3.4.2 Function

GPR Register

12.4.3.4.3 Diagram



12.4.3.4.4 Fields

Field	Function
31-3 APC_AC_R0_BOT	APC start address of memory region-0
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.5 GPR3 General Purpose Register (GPR3)

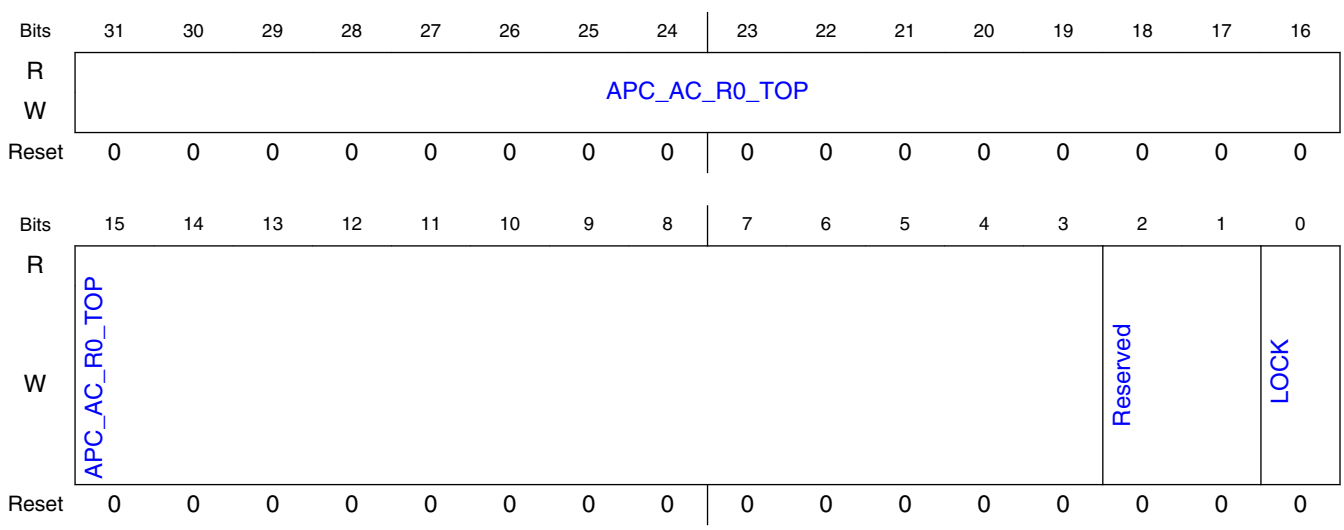
12.4.3.5.1 Offset

Register	Offset
GPR3	Ch

12.4.3.5.2 Function

GPR Register

12.4.3.5.3 Diagram



12.4.3.5.4 Fields

Field	Function
31-3 APC_AC_R0_TOP	APC end address of memory region-0
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.6 GPR4 General Purpose Register (GPR4)

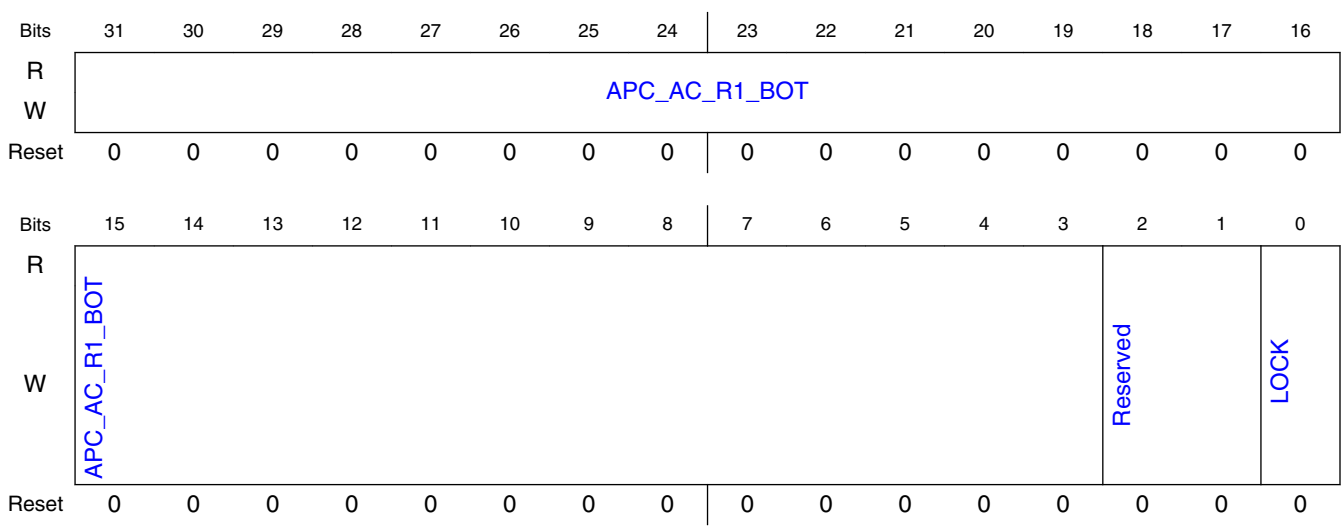
12.4.3.6.1 Offset

Register	Offset
GPR4	10h

12.4.3.6.2 Function

GPR Register

12.4.3.6.3 Diagram



12.4.3.6.4 Fields

Field	Function
31-3 APC_AC_R1_BOT	APC start address of memory region-1
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.7 GPR5 General Purpose Register (GPR5)

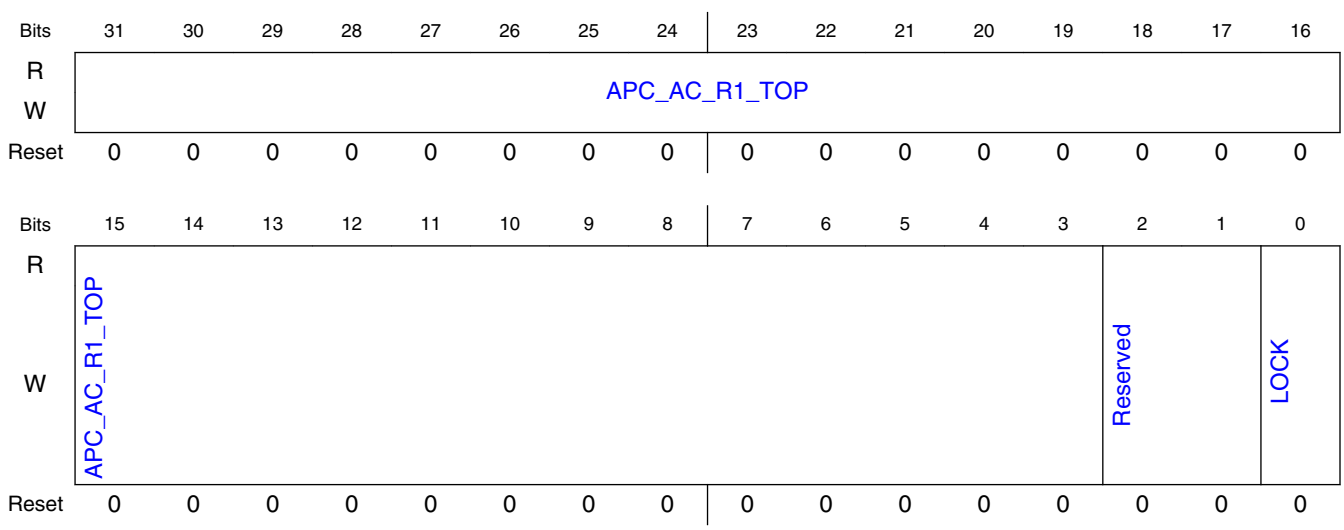
12.4.3.7.1 Offset

Register	Offset
GPR5	14h

12.4.3.7.2 Function

GPR Register

12.4.3.7.3 Diagram



12.4.3.7.4 Fields

Field	Function
31-3 APC_AC_R1_TOP	APC end address of memory region-1
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.8 GPR6 General Purpose Register (GPR6)

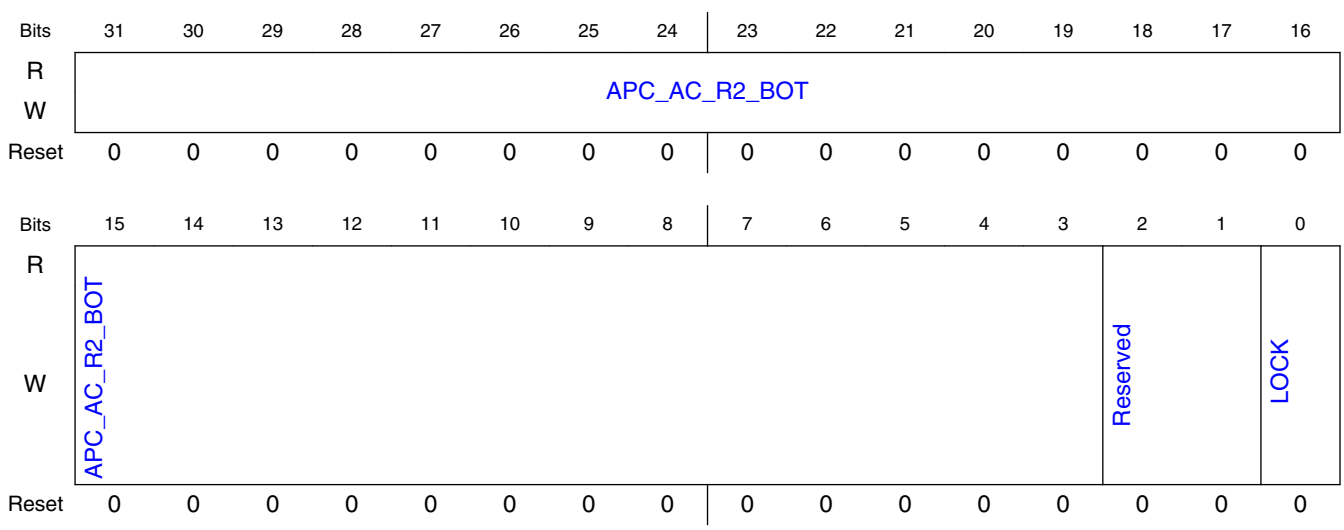
12.4.3.8.1 Offset

Register	Offset
GPR6	18h

12.4.3.8.2 Function

GPR Register

12.4.3.8.3 Diagram



12.4.3.8.4 Fields

Field	Function
31-3 APC_AC_R2_BOT	APC start address of memory region-2
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.9 GPR7 General Purpose Register (GPR7)

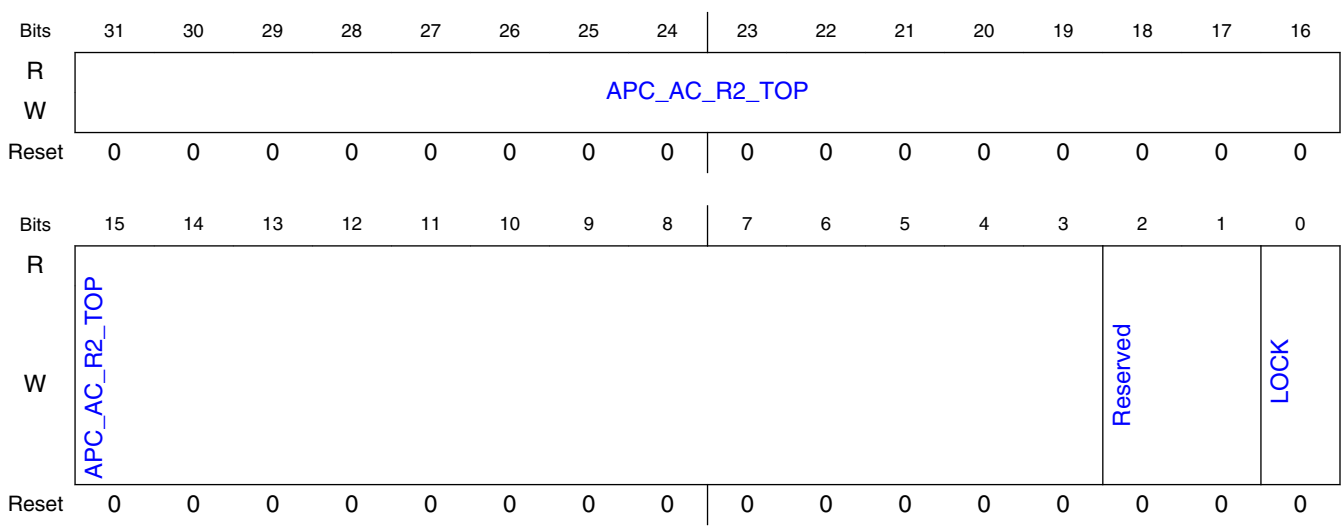
12.4.3.9.1 Offset

Register	Offset
GPR7	1Ch

12.4.3.9.2 Function

GPR Register

12.4.3.9.3 Diagram



12.4.3.9.4 Fields

Field	Function
31-3 APC_AC_R2_TOP	APC end address of memory region-2
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.10 GPR8 General Purpose Register (GPR8)

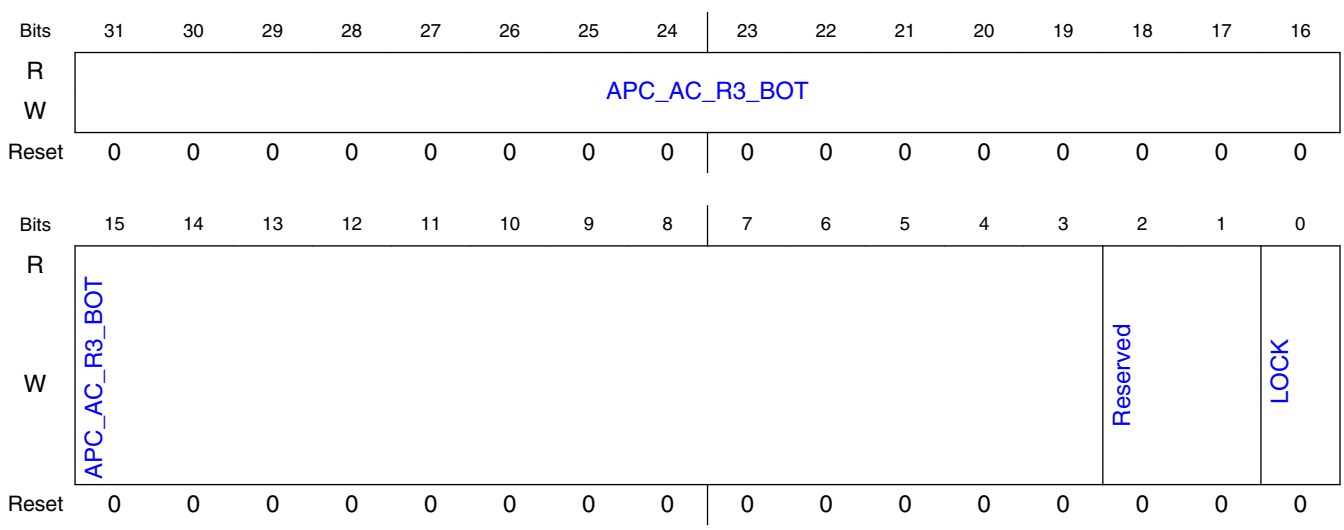
12.4.3.10.1 Offset

Register	Offset
GPR8	20h

12.4.3.10.2 Function

GPR Register

12.4.3.10.3 Diagram



12.4.3.10.4 Fields

Field	Function
31-3 APC_AC_R3_BOT	APC start address of memory region-3
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.11 GPR9 General Purpose Register (GPR9)

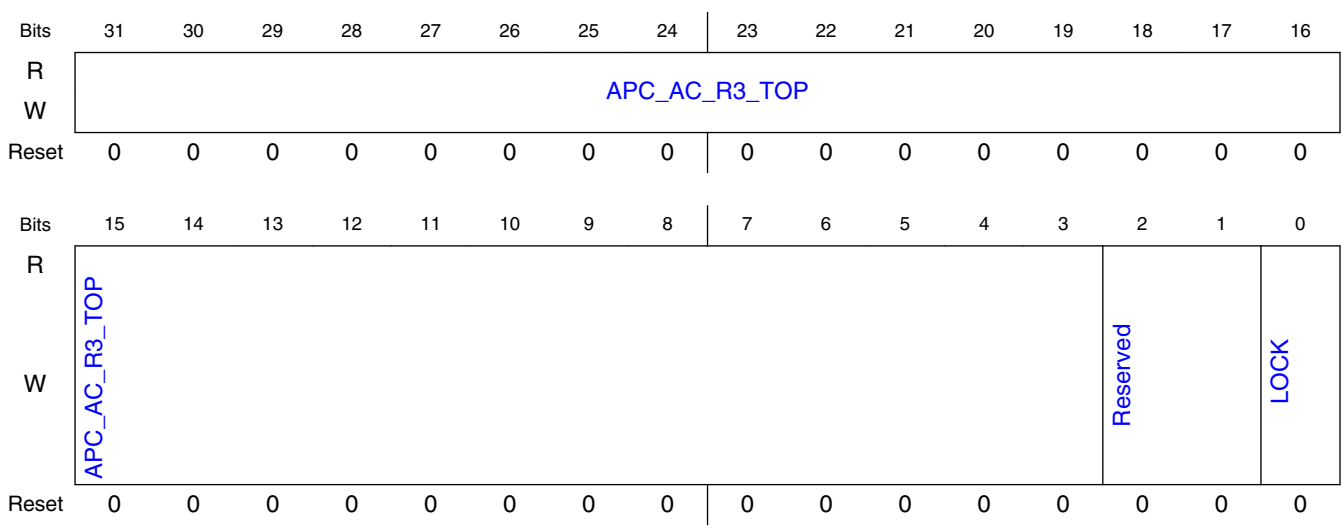
12.4.3.11.1 Offset

Register	Offset
GPR9	24h

12.4.3.11.2 Function

GPR Register

12.4.3.11.3 Diagram



12.4.3.11.4 Fields

Field	Function
31-3 APC_AC_R3_TOP	APC end address of memory region-3
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.12 GPR10 General Purpose Register (GPR10)

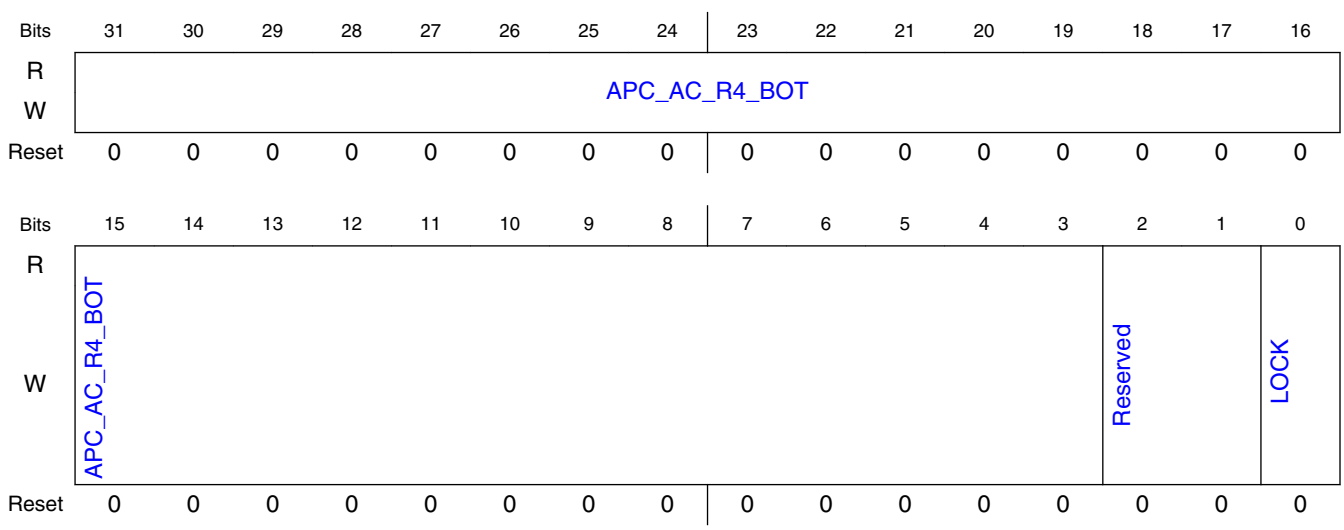
12.4.3.12.1 Offset

Register	Offset
GPR10	28h

12.4.3.12.2 Function

GPR Register

12.4.3.12.3 Diagram



12.4.3.12.4 Fields

Field	Function
31-3 APC_AC_R4_BOT	APC start address of memory region-4
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.13 GPR11 General Purpose Register (GPR11)

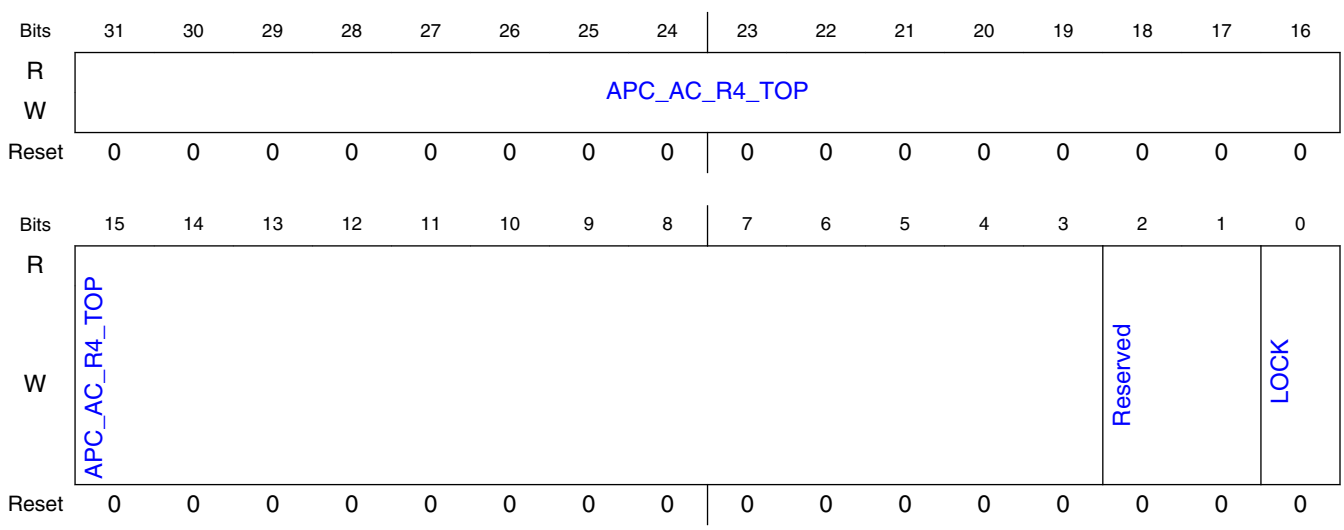
12.4.3.13.1 Offset

Register	Offset
GPR11	2Ch

12.4.3.13.2 Function

GPR Register

12.4.3.13.3 Diagram



12.4.3.13.4 Fields

Field	Function
31-3 APC_AC_R4_TOP	APC end address of memory region-4
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.14 GPR12 General Purpose Register (GPR12)

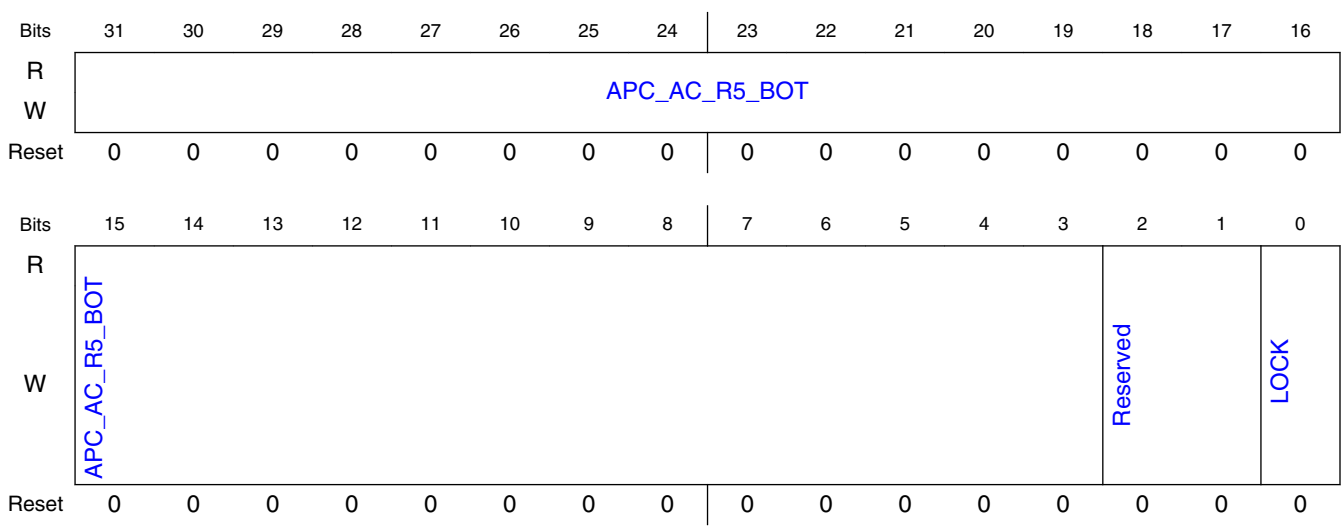
12.4.3.14.1 Offset

Register	Offset
GPR12	30h

12.4.3.14.2 Function

GPR Register

12.4.3.14.3 Diagram



12.4.3.14.4 Fields

Field	Function
31-3 APC_AC_R5_BOT	APC start address of memory region-5
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.15 GPR13 General Purpose Register (GPR13)

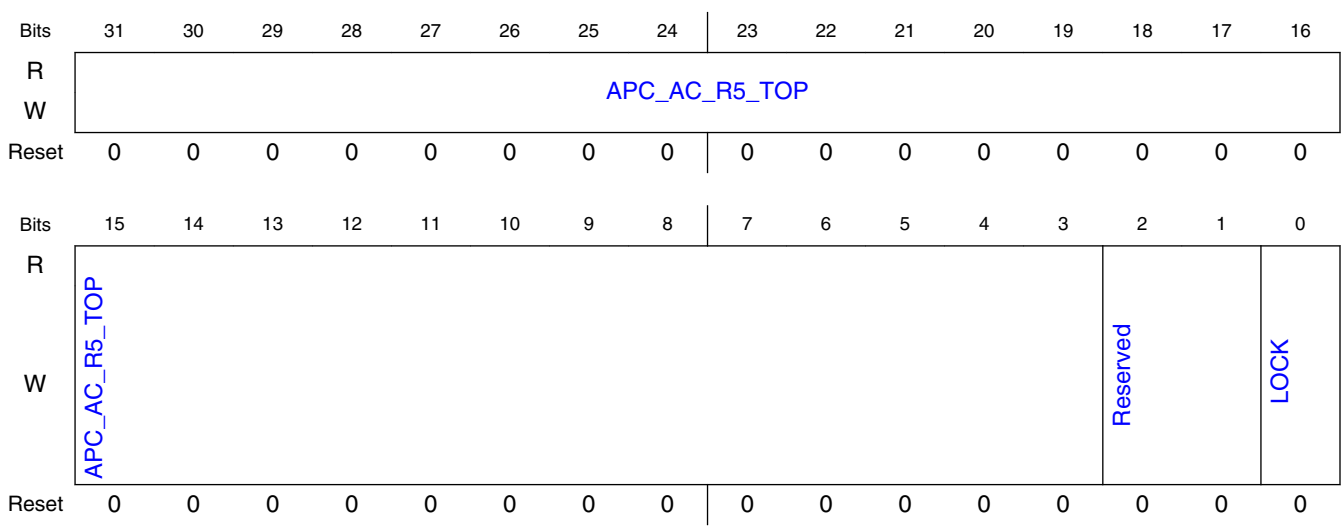
12.4.3.15.1 Offset

Register	Offset
GPR13	34h

12.4.3.15.2 Function

GPR Register

12.4.3.15.3 Diagram



12.4.3.15.4 Fields

Field	Function
31-3 APC_AC_R5_TOP	APC end address of memory region-5
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.16 GPR14 General Purpose Register (GPR14)

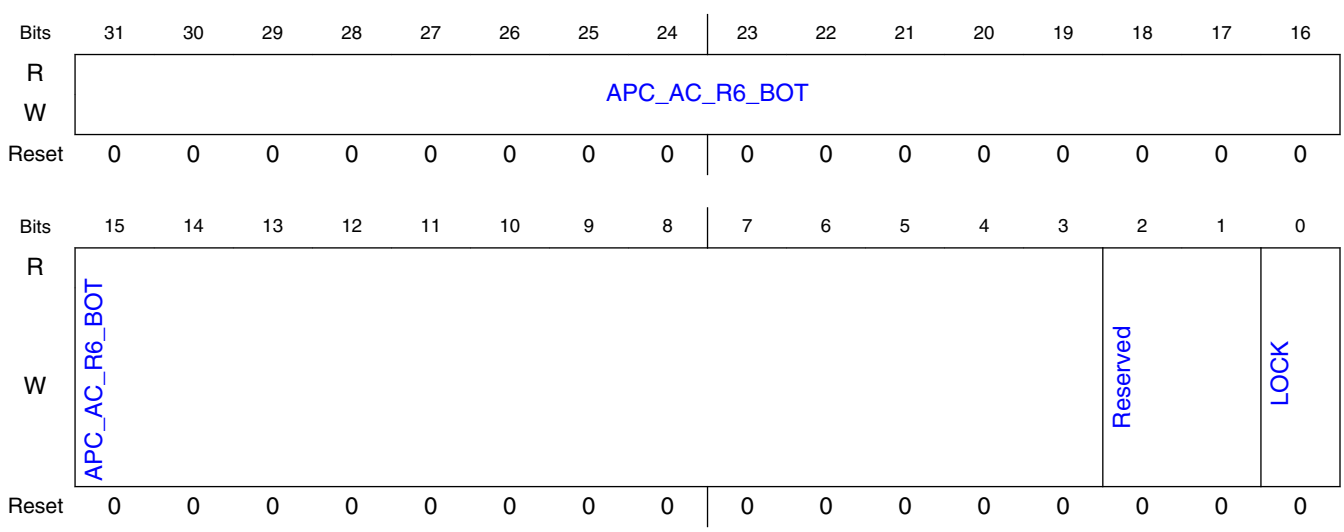
12.4.3.16.1 Offset

Register	Offset
GPR14	38h

12.4.3.16.2 Function

GPR Register

12.4.3.16.3 Diagram



12.4.3.16.4 Fields

Field	Function
31-3 APC_AC_R6_BOT	APC start address of memory region-6
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.17 GPR15 General Purpose Register (GPR15)

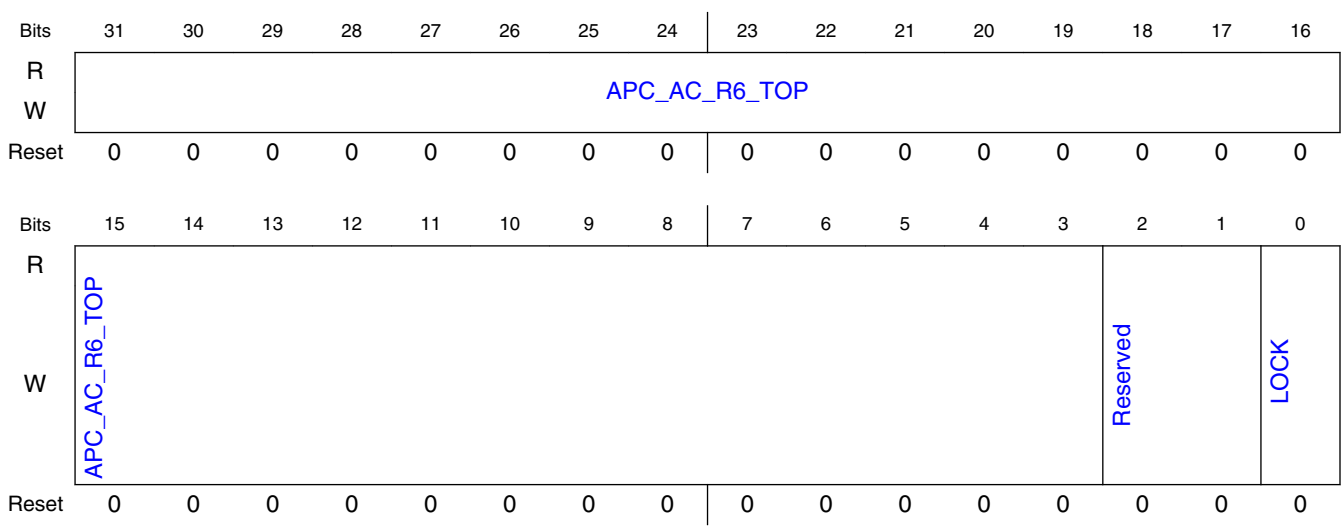
12.4.3.17.1 Offset

Register	Offset
GPR15	3Ch

12.4.3.17.2 Function

GPR Register

12.4.3.17.3 Diagram



12.4.3.17.4 Fields

Field	Function
31-3 APC_AC_R6_TOP	APC end address of memory region-6
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.18 GPR16 General Purpose Register (GPR16)

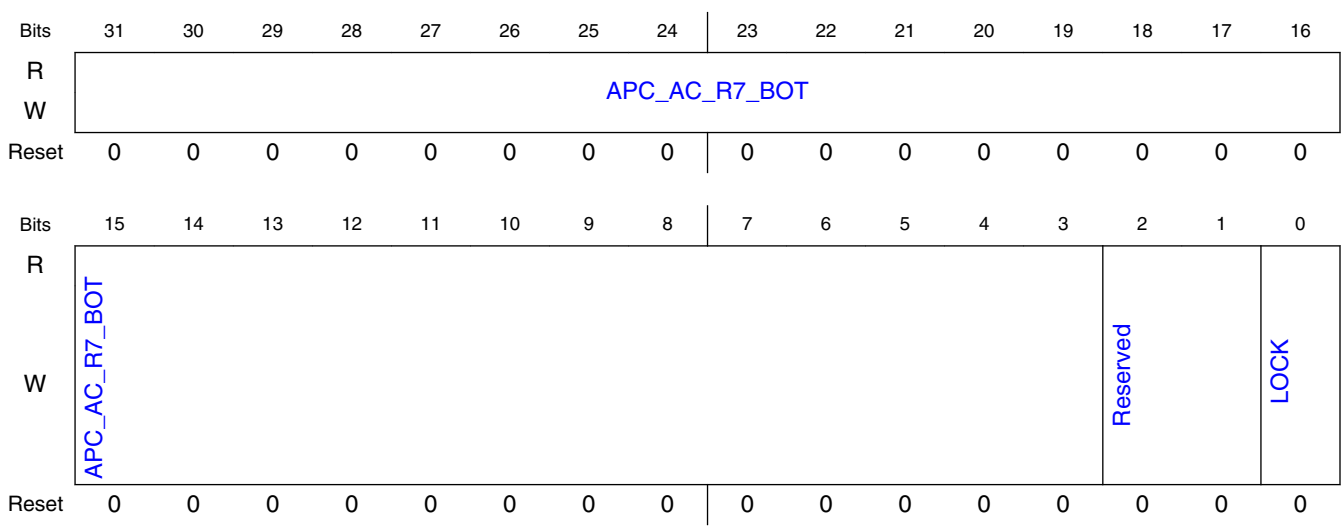
12.4.3.18.1 Offset

Register	Offset
GPR16	40h

12.4.3.18.2 Function

GPR Register

12.4.3.18.3 Diagram



12.4.3.18.4 Fields

Field	Function
31-3 APC_AC_R7_BOT	APC start address of memory region-7
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.19 GPR17 General Purpose Register (GPR17)

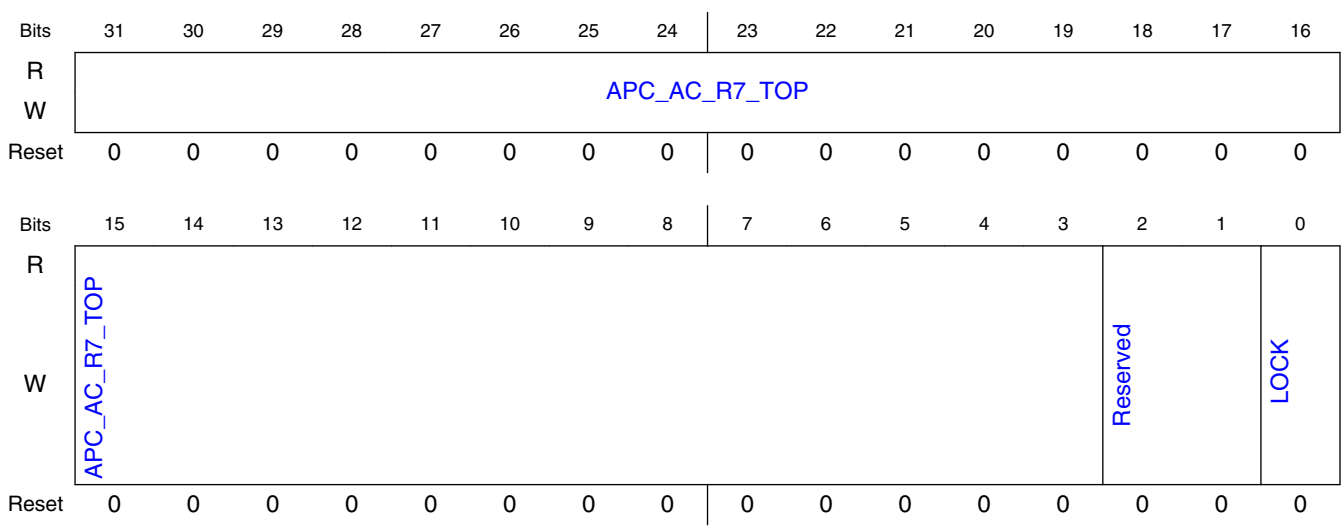
12.4.3.19.1 Offset

Register	Offset
GPR17	44h

12.4.3.19.2 Function

GPR Register

12.4.3.19.3 Diagram



12.4.3.19.4 Fields

Field	Function
31-3 APC_AC_R7_TOP	APC end address of memory region-7
2-1 —	Reserved
0 LOCK	Lock the write to bit 31:1 0b - Write access to bit 31:1 is not blocked 1b - Write access to bit 31:1 is blocked

12.4.3.20 GPR18 General Purpose Register (GPR18)

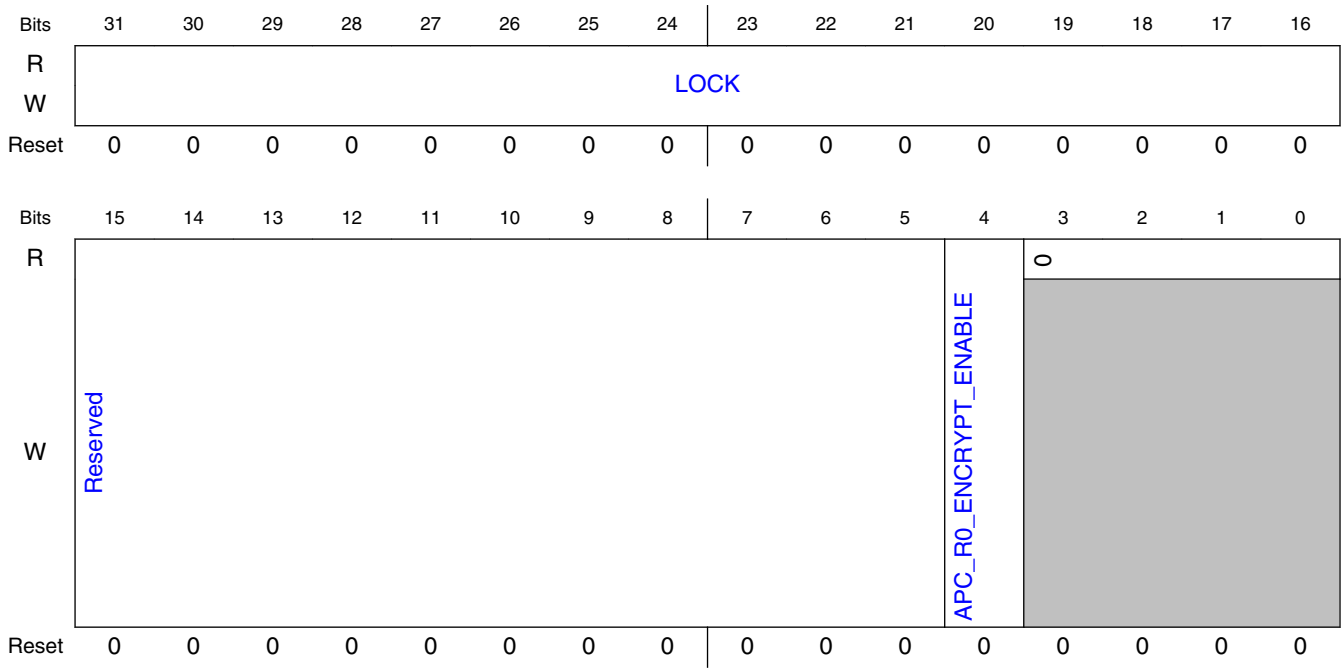
12.4.3.20.1 Offset

Register	Offset
GPR18	48h

12.4.3.20.2 Function

GPR Register

12.4.3.20.3 Diagram



12.4.3.20.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R0_ENCRYPT_ENABLE	APC memory region-0 encryption enable When high, access to memory region-0 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.21 GPR19 General Purpose Register (GPR19)

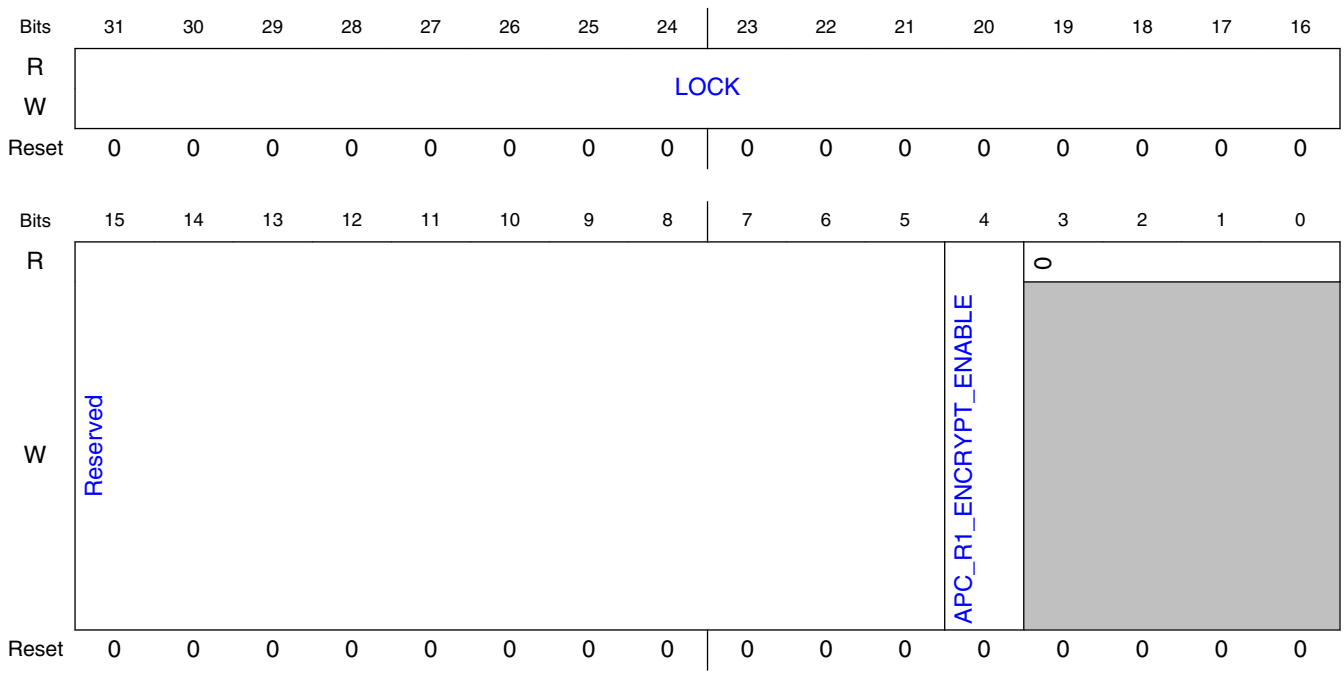
12.4.3.21.1 Offset

Register	Offset
GPR19	4Ch

12.4.3.21.2 Function

GPR Register

12.4.3.21.3 Diagram



12.4.3.21.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R1_ENCRYPT_ENABLE	APC memory region-1 encryption enable When high, access to memory region-1 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect

Table continues on the next page...

Field	Function
	1b - Encryption enabled
3-0 —	Reserved

12.4.3.22 GPR20 General Purpose Register (GPR20)

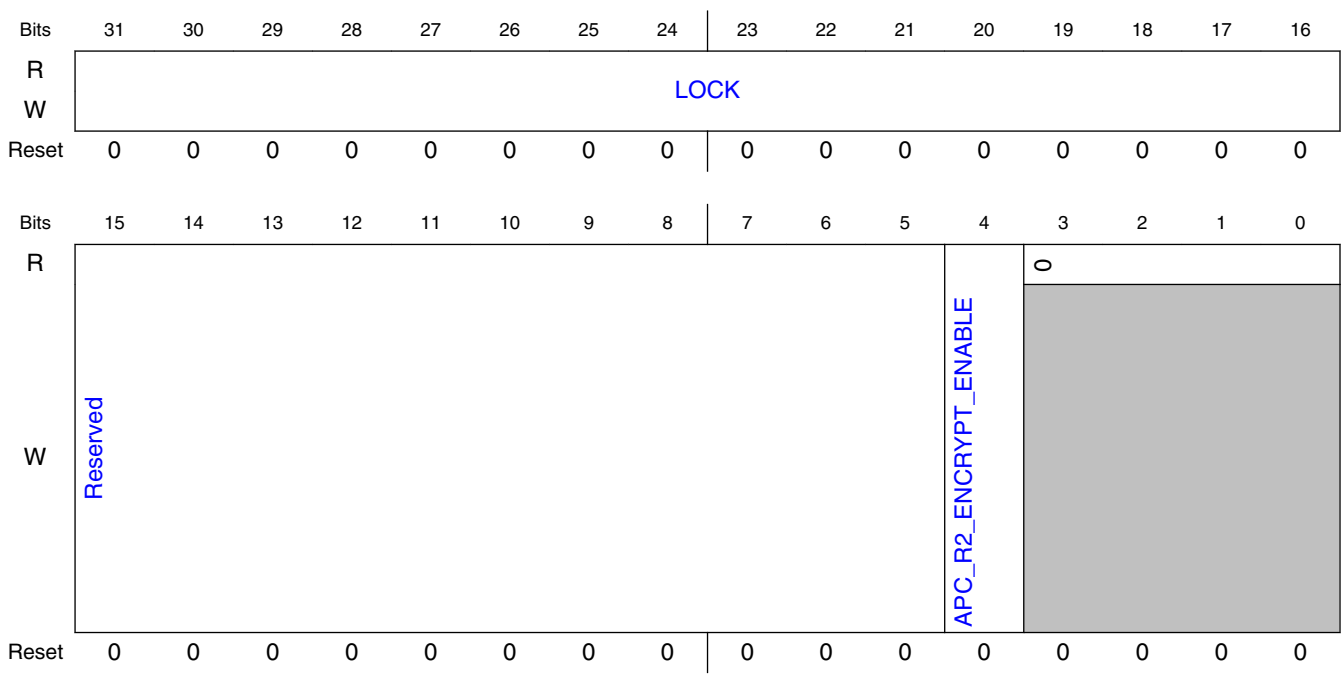
12.4.3.22.1 Offset

Register	Offset
GPR20	50h

12.4.3.22.2 Function

GPR Register

12.4.3.22.3 Diagram



12.4.3.22.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R2_ENCRYPT_ENABLE	APC memory region-2 encryption enable When high, access to memory region-2 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.23 GPR21 General Purpose Register (GPR21)

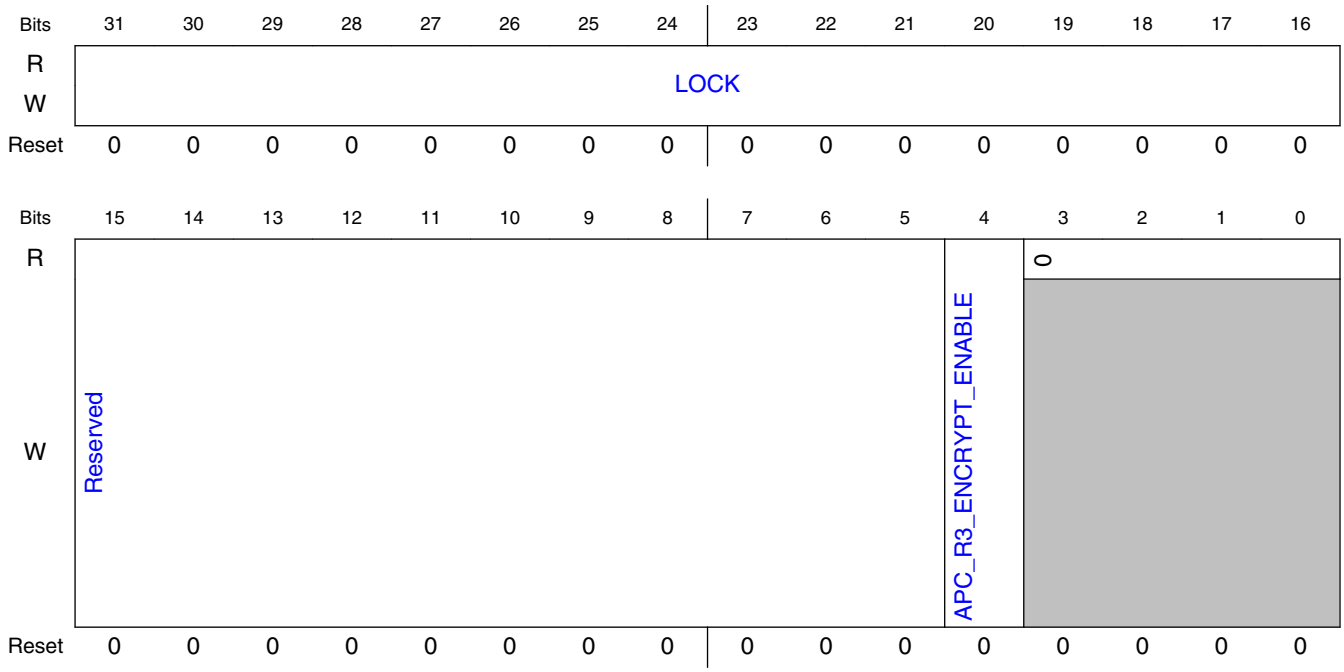
12.4.3.23.1 Offset

Register	Offset
GPR21	54h

12.4.3.23.2 Function

GPR Register

12.4.3.23.3 Diagram



12.4.3.23.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R3_ENCRYPT_ENABLE	APC memory region-3 encryption enable When high, access to memory region-3 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.24 GPR22 General Purpose Register (GPR22)

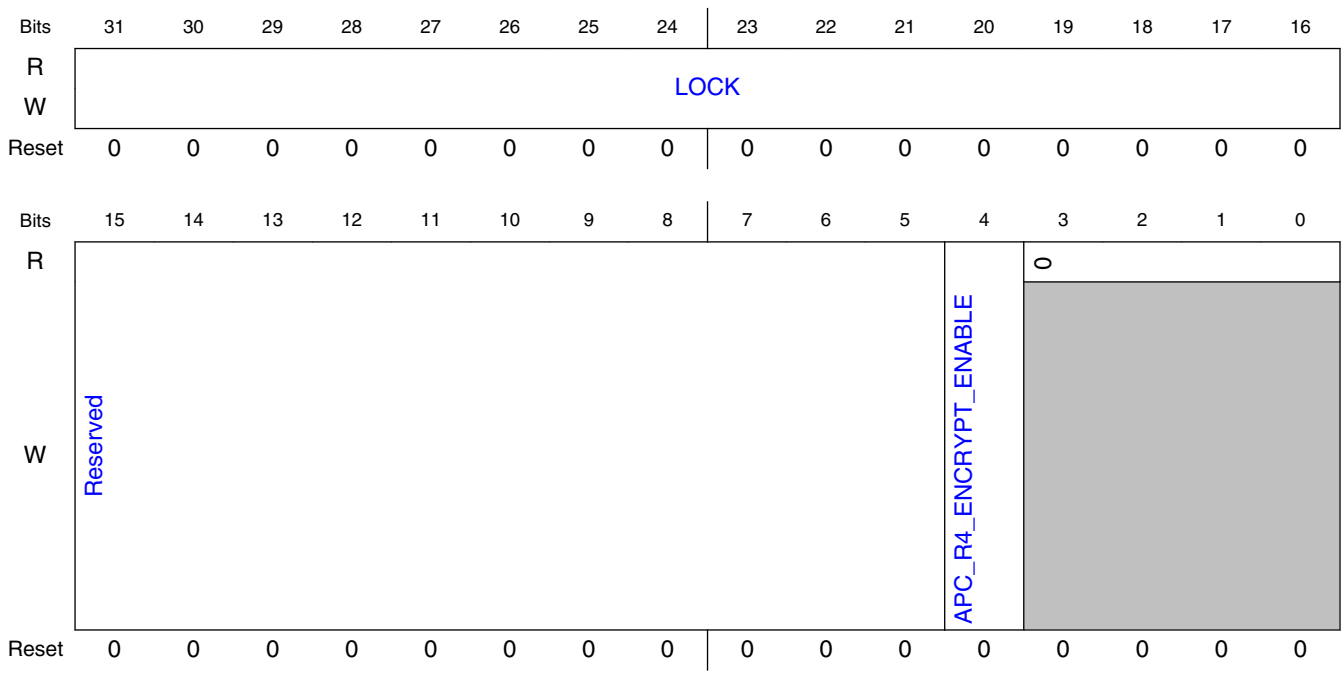
12.4.3.24.1 Offset

Register	Offset
GPR22	58h

12.4.3.24.2 Function

GPR Register

12.4.3.24.3 Diagram



12.4.3.24.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R4_ENCRYPT_ENABLE	APC memory region-4 encryption enable When high, access to memory region-4 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect

Table continues on the next page...

Field	Function
	1b - Encryption enabled
3-0 —	Reserved

12.4.3.25 GPR23 General Purpose Register (GPR23)

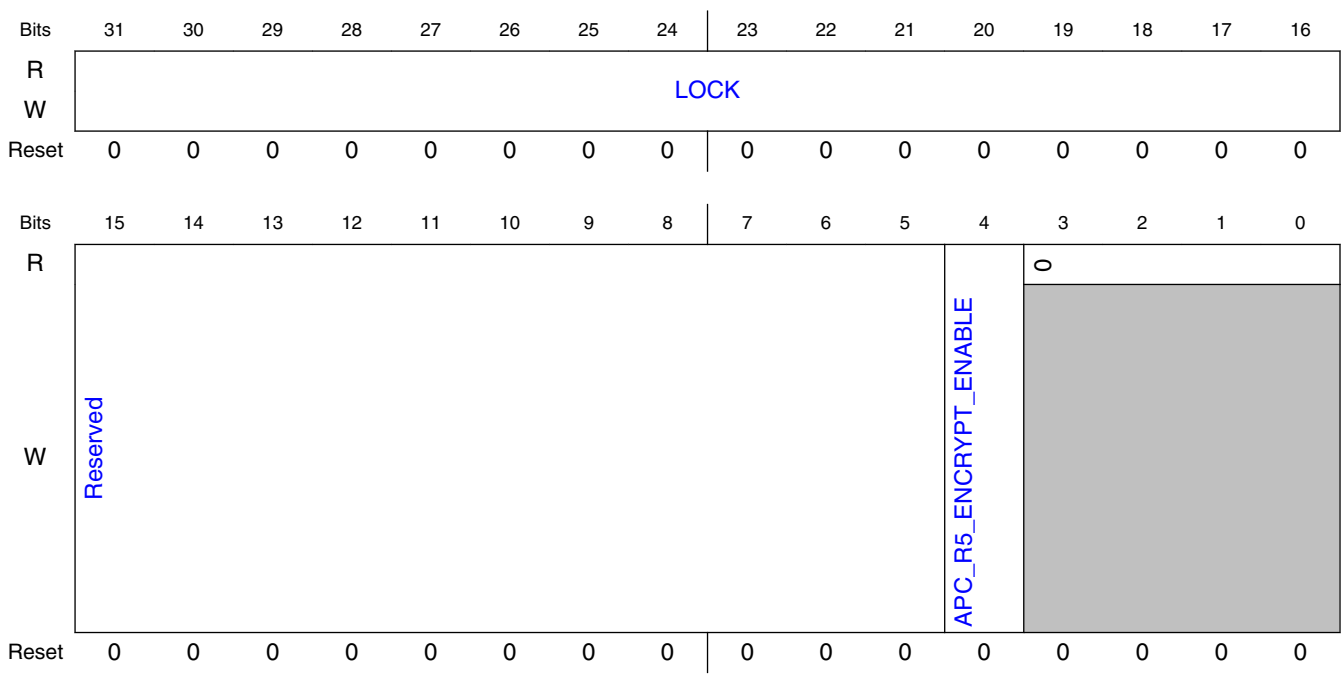
12.4.3.25.1 Offset

Register	Offset
GPR23	5Ch

12.4.3.25.2 Function

GPR Register

12.4.3.25.3 Diagram



12.4.3.25.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R5_ENCRYPT_ENABLE	APC memory region-5 encryption enable When high, access to memory region-5 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.26 GPR24 General Purpose Register (GPR24)

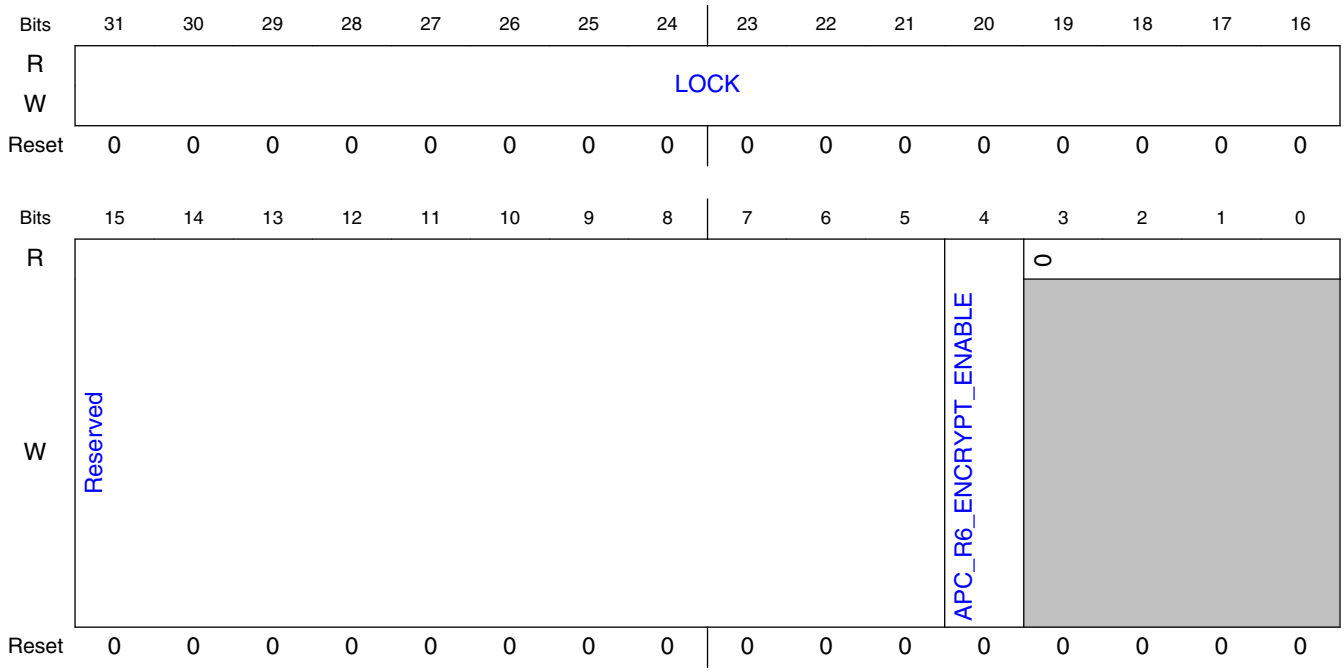
12.4.3.26.1 Offset

Register	Offset
GPR24	60h

12.4.3.26.2 Function

GPR Register

12.4.3.26.3 Diagram



12.4.3.26.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-5 —	Reserved
4 APC_R6_ENCRYPT_ENABLE	APC memory region-6 encryption enable When high, access to memory region-6 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.27 GPR25 General Purpose Register (GPR25)

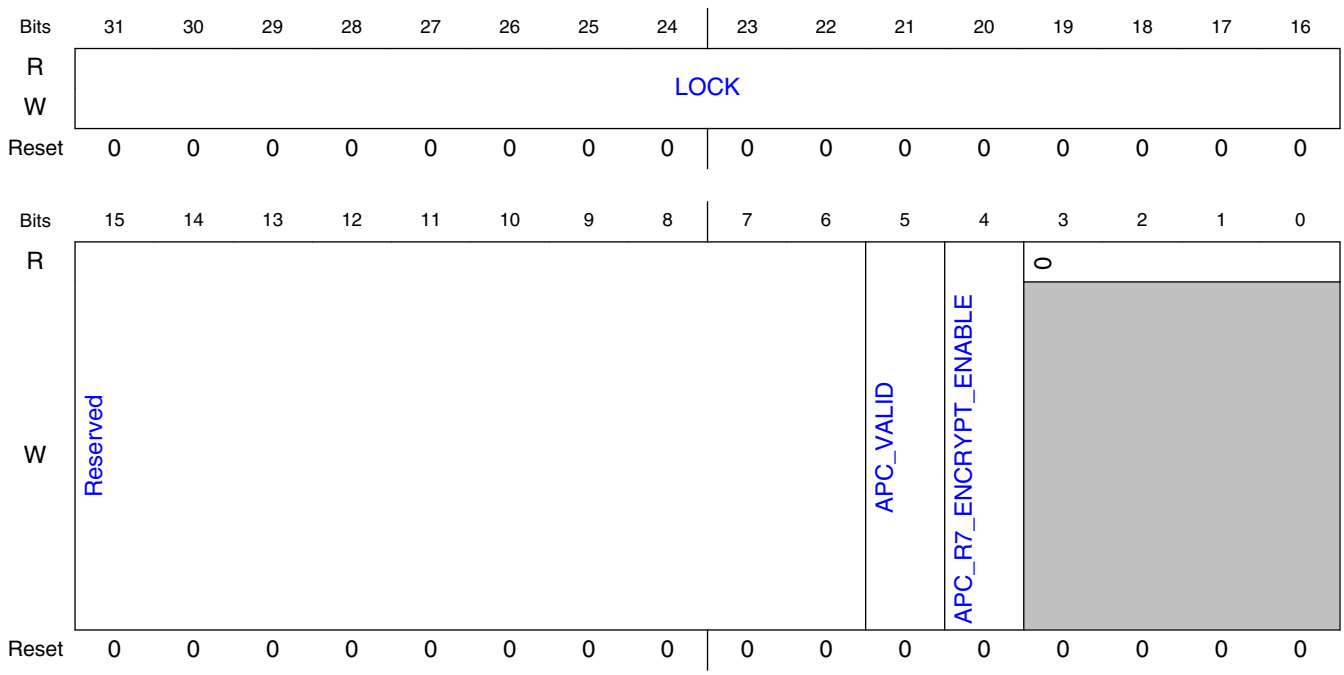
12.4.3.27.1 Offset

Register	Offset
GPR25	64h

12.4.3.27.2 Function

GPR Register

12.4.3.27.3 Diagram



12.4.3.27.4 Fields

Field	Function
31-16 LOCK	Lock the write to bit 15:0 Once LOCK bit i is set to '1', the write access to bit i will be blocked. If LOCK bit i is set to '0', there will be no change.
15-6 —	Reserved
5 APC_VALID	APC global enable bit 0b - No effect 1b - Enable encryption for GPRx[APC_x_ENCRYPT_ENABLE] (valid for GPR2-GPR25)
4	APC memory region-7 encryption enable

Table continues on the next page...

Field	Function
APC_R7_ENCRYPT_ENABLE	When high, access to memory region-7 will be routed to IEE. It impacts access from all the core and masters. 0b - No effect 1b - Encryption enabled
3-0 —	Reserved

12.4.3.28 GPR26 General Purpose Register (GPR26)

12.4.3.28.1 Offset

Register	Offset
GPR26	68h

12.4.3.28.2 Function

GPR Register

12.4.3.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DWP_LOCK		DWP		FIELD_0			CM7_INIT_VTOR								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CM7_INIT_VTOR															
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.3.28.4 Fields

Field	Function
31-30	Domain write protection lock

Table continues on the next page...

Field	Function
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change CM7_INIT_VTOR. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-25 FIELD_0	General purpose bits Reserved
24-0 CM7_INIT_VTOR	Vector table offset register out of reset. See the ARM v7-M Architecture Reference Manual for more information about the vector table offset register (VTOR). **NOTE:** When user uses HAB authenticated or encrypted boot function, NXP strongly suggest user to set the default value (ROM entry address) into INIT_VTOR register and lock the default value.

12.4.3.29 GPR33 General Purpose Register (GPR33)

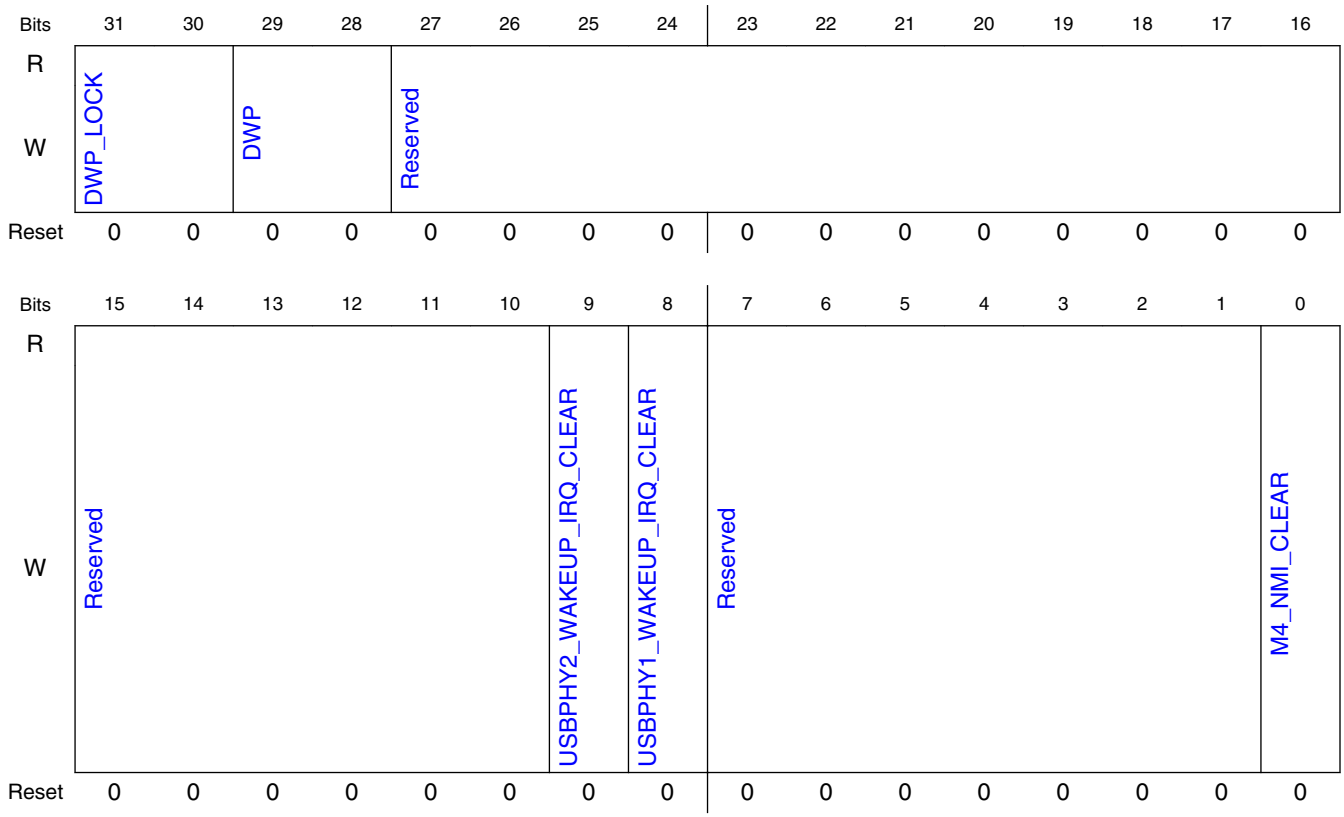
12.4.3.29.1 Offset

Register	Offset
GPR33	84h

12.4.3.29.2 Function

GPR Register

12.4.3.29.3 Diagram



12.4.3.29.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change FIELD_0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-10 —	Reserved
9	Clear USBPHY1 wakeup interrupt holding register

Table continues on the next page...

Field	Function
USBPHY2_WAKEUP_IRQ_CLEAR	The wakeup interrupt from USBPHY1 will be held internally until this bit is set to 1. Not that this bit need software to clear. Set it to 1 to clear the interrupt and following with writing it to 0 to complete the operation.
8 USBPHY1_WAKEUP_IRQ_CLEAR	Clear USBPHY1 wakeup interrupt holding register The wakeup interrupt from USBPHY1 will be held internally until this bit is set to 1. Not that this bit need software to clear. Set it to 1 to clear the interrupt and following with writing it to 0 to complete the operation.
7-1 —	Reserved
0 M4_NMI_CLEAR	Clear CM4 NMI holding register The NMI input from IO will be held internally until this bit is set to 1. Note that this bit need software to clear. Set it to 1 to clear the interrupt and following with writing it to 0 to complete the operation.

12.4.3.30 GPR34 General Purpose Register (GPR34)

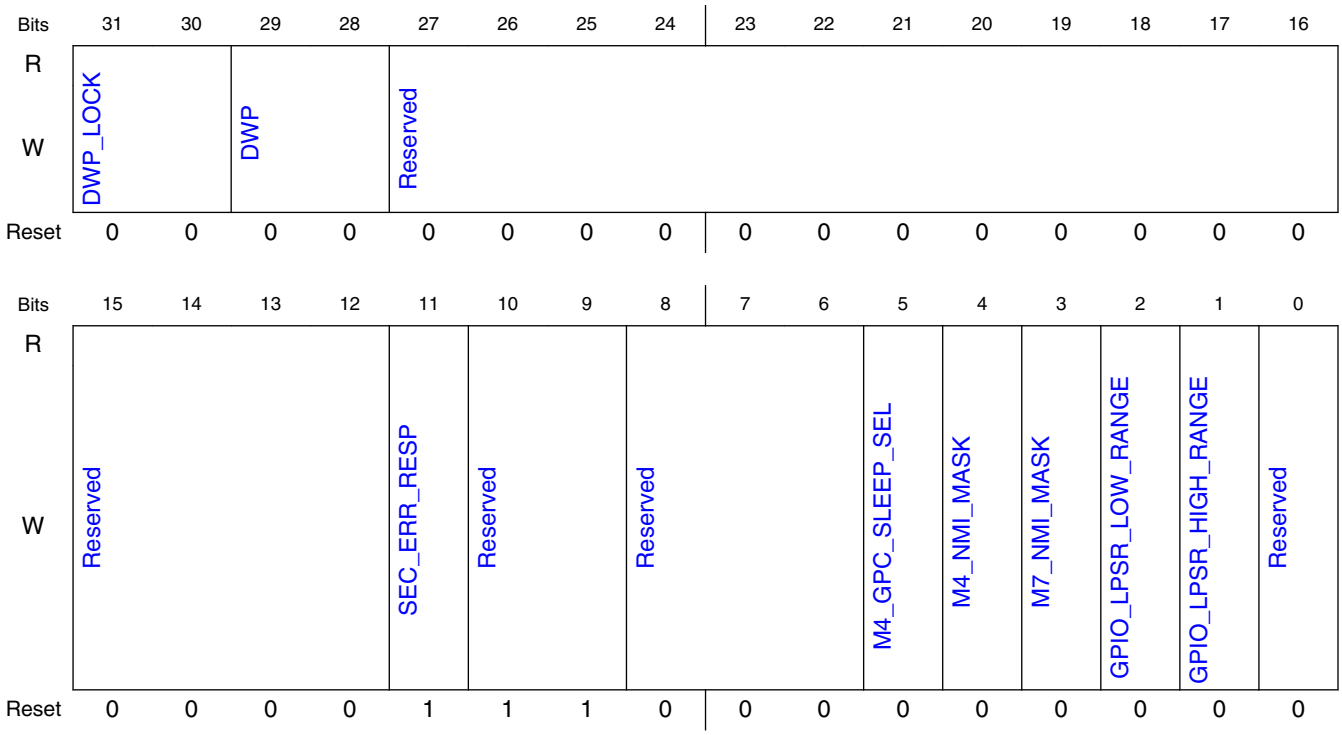
12.4.3.30.1 Offset

Register	Offset
GPR34	88h

12.4.3.30.2 Function

GPR Register

12.4.3.30.3 Diagram



12.4.3.30.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-12 —	Reserved
11 SEC_ERR_RESP	Security error response enable This bitfield is the security error response enable for all security gaskets on both AHB and AXI buses. The RDC defines access policies and the security gaskets on the buses are responsible to verify the access

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>against these policies. This bit defines what kind of bus response is returned when the access violates the policies.
0b - OKEY response
1b - SLVError (default)</td></tr>
<tr><td>10-9
—</td><td>Reserved</td></tr>
<tr><td>8-6
—</td><td>Reserved</td></tr>
<tr><td>5
M4_GPC_SLEEP_SEL</td><td>CM4 sleep request selection
This bit controls which kind of CM4 sleep request is sent to GPC to start sleep sequence.
0b - CM4 SLEEPDEEP is sent to GPC
1b - CM4 SLEEPING is sent to GPC</td></tr>
<tr><td>4
M4_NMI_MASK</td><td>Mask CM4 NMI pin input
0b - NMI input from IO to CM4 is not blocked
1b - NMI input from IO to CM4 is blocked</td></tr>
<tr><td>3
M7_NMI_MASK</td><td>Mask CM7 NMI pin input
0b - NMI input from IO to CM7 is not blocked
1b - NMI input from IO to CM7 is blocked</td></tr>
<tr><td>2
GPIO_LPSR_LOW_RANGE</td><td>GPIO_LPSR IO bank supply voltage range selection
<table>
<tr><th>GPIO_LPSR_HIGH_RANGE</th><th>GPIO_LPSR_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_LPSR_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_LPSR_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_LPSR_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
<tr><td>1
GPIO_LPSR_HIGH_RANGE</td><td>GPIO_LPSR IO bank supply voltage range selection
<table>
<tr><th>GPIO_LPSR_HIGH_RANGE</th><th>GPIO_LPSR_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_LPSR_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_LPSR_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_LPSR_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
<tr><td>0</td><td>Reserved</td></tr>
</table>

Field	Function
—	

12.4.3.31 GPR35 General Purpose Register (GPR35)

12.4.3.31.1 Offset

Register	Offset
GPR35	8Ch

12.4.3.31.2 Function

GPR Register

12.4.3.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWP_LOCK		DWP		Reserved	FLEXSPI2_STOP_REQ	FLEXSPI2_IPG_DOZE	FLEXSPI1_STOP_REQ	FLEXSPI1_IPG_DOZE	FLEXIO2_IPG_DOZE	FLEXIO1_IPG_DOZE	ENET1G_STOP_REQ	ENET1G_IPG_DOZE	ENET_STOP_REQ	ENET_IPG_DOZE	EDMA_LPSR_STOP_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDMA_STOP_REQ	0	CAN3_STOP_REQ	CAN3_IPG_DOZE	CAN2_STOP_REQ	CAN2_IPG_DOZE	CAN1_STOP_REQ	CAN1_IPG_DOZE	CAAM_STOP_REQ	CAAM_IPG_DOZE	ADC2_IPG_STOP_MODE	ADC2_STOP_REQ	ADC2_IPG_DOZE	ADC1_IPG_STOP_MODE	ADC1_STOP_REQ	ADC1_IPG_DOZE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.3.31.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved
26 FLEXSPI2_STOP_REQ	FLEXSPI2 stop request 0b - Stop request off 1b - Stop request on
25 FLEXSPI2_IPG_DOZE	FLEXSPI2 doze mode 0b - Not in doze mode 1b - In doze mode
24 FLEXSPI1_STOP_REQ	FLEXSPI1 stop request 0b - Stop request off 1b - Stop request on
23 FLEXSPI1_IPG_DOZE	FLEXSPI1 doze mode 0b - Not in doze mode 1b - In doze mode
22 FLEXIO2_IPG_DOZE	FLEXIO2 doze mode 0b - Not in doze mode 1b - In doze mode
21 FLEXIO1_IPG_DOZE	FLEXIO2 doze mode 0b - Not in doze mode 1b - In doze mode
20 ENET1G_STOP_REQ	ENET1G stop request 0b - Stop request off 1b - Stop request on
19 ENET1G_IPG_DOZE	ENET1G doze mode 0b - Not in doze mode 1b - In doze mode
18 ENET_STOP_REQ	ENET stop request 0b - Stop request off 1b - Stop request on

Table continues on the next page...

Field	Function
17 ENET_IPG_DOZE	ENET doze mode 0b - Not in doze mode 1b - In doze mode
16 EDMA_LPSR_STOP_REQ	EDMA_LPSR stop request 0b - Stop request off 1b - Stop request on
15 EDMA_STOP_REQ	EDMA stop request 0b - Stop request off 1b - Stop request on
14 —	Reserved
13 CAN3_STOP_REQ	CAN3 stop request 0b - Stop request off 1b - Stop request on
12 CAN3_IPG_DOZE	CAN3 doze mode 0b - Not in doze mode 1b - In doze mode
11 CAN2_STOP_REQ	CAN2 stop request 0b - Stop request off 1b - Stop request on
10 CAN2_IPG_DOZE	CAN2 doze mode 0b - Not in doze mode 1b - In doze mode
9 CAN1_STOP_REQ	CAN1 stop request 0b - Stop request off 1b - Stop request on
8 CAN1_IPG_DOZE	CAN1 doze mode 0b - Not in doze mode 1b - In doze mode
7 CAAM_STOP_REQ	CAAM stop request 0b - Stop request off 1b - Stop request on
6 CAAM_IPG_DOZE	CAN3 doze mode 0b - Not in doze mode 1b - In doze mode
5 ADC2_IPG_STOP_MODE	ADC2 stop mode selection. This bitfield cannot change when ADC2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 ADC2_STOP_REQ	ADC2 stop request 0b - Stop request off 1b - Stop request on
3	ADC2 doze mode 0b - Not in doze mode 1b - In doze mode

Table continues on the next page...

Field	Function
ADC2_IPG_DOZE	
2 ADC1_IPG_STOP_MODE	ADC1 stop mode selection. This bitfield cannot change when ADC1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1 ADC1_STOP_REQ	ADC1 stop request 0b - Stop request off 1b - Stop request on
0 ADC1_IPG_DOZE	ADC1 doze mode 0b - Not in doze mode 1b - In doze mode

12.4.3.32 GPR36 General Purpose Register (GPR36)

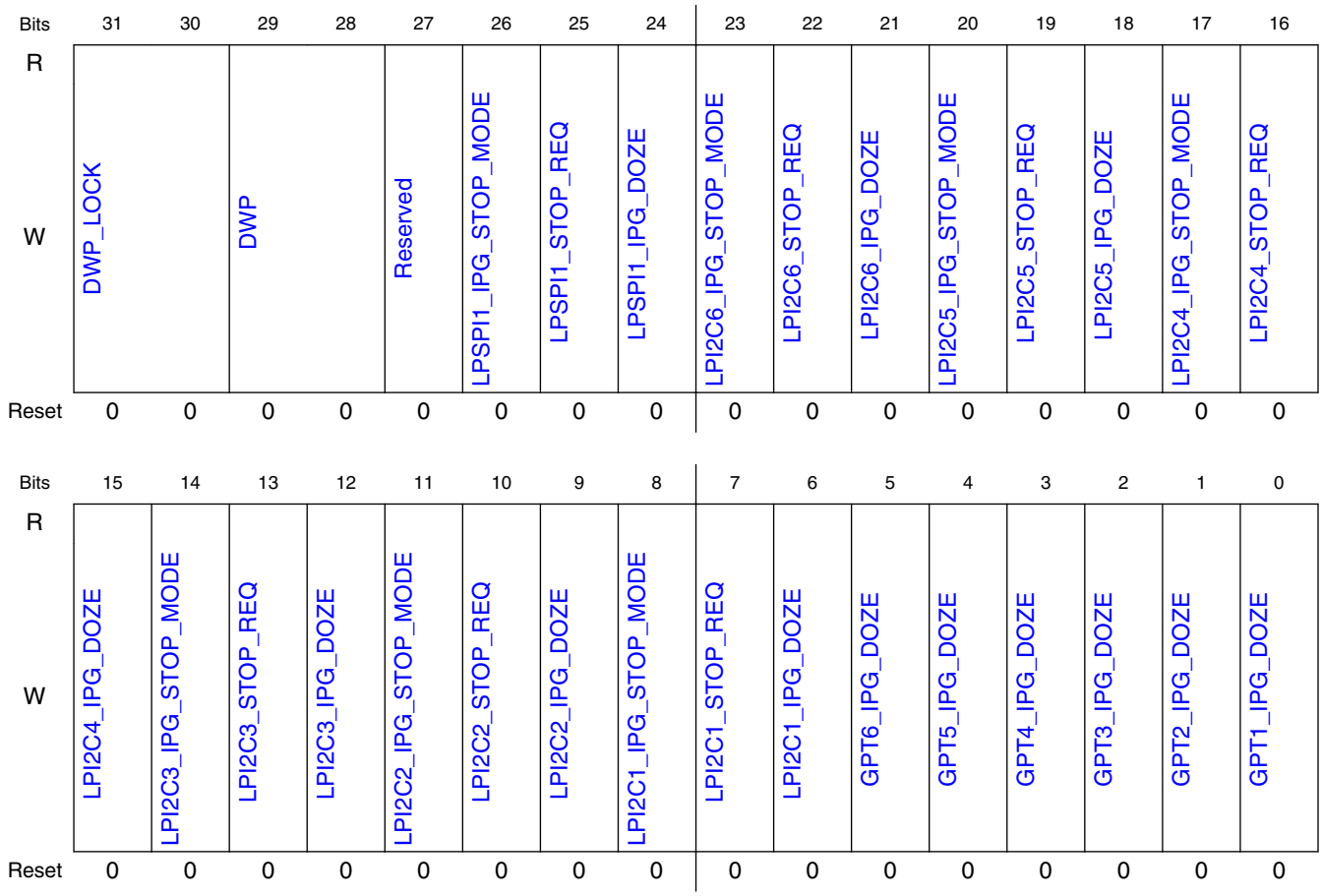
12.4.3.32.1 Offset

Register	Offset
GPR36	90h

12.4.3.32.2 Function

GPR Register

12.4.3.32.3 Diagram



12.4.3.32.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved

Table continues on the next page...

Field	Function
26 LPSPI1_IPG_STOP_MODE	LPSPI1 stop mode selection. This bitfield cannot change when LPSPI1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 LPSPI1_STOP_REQ	LPSPI1 stop request 0b - Stop request off 1b - Stop request on
24 LPSPI1_IPG_DOZE	LPSPI1 doze mode 0b - Not in doze mode 1b - In doze mode
23 LPI2C6_IPG_STOP_MODE	LPI2C6 stop mode selection. This bitfield cannot change when LPI2C6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPI2C6_STOP_REQ	LPI2C6 stop request 0b - Stop request off 1b - Stop request on
21 LPI2C6_IPG_DOZE	LPI2C6 doze mode 0b - Not in doze mode 1b - In doze mode
20 LPI2C5_IPG_STOP_MODE	LPI2C5 stop mode selection. This bitfield cannot change when LPI2C5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPI2C5_STOP_REQ	LPI2C5 stop request 0b - Stop request off 1b - Stop request on
18 LPI2C5_IPG_DOZE	LPI2C5 doze mode 0b - Not in doze mode 1b - In doze mode
17 LPI2C4_IPG_STOP_MODE	LPI2C4 stop mode selection. This bitfield cannot change when LPI2C4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPI2C4_STOP_REQ	LPI2C4 stop request 0b - Stop request off 1b - Stop request on
15 LPI2C4_IPG_DOZE	LPI2C4 doze mode 0b - Not in doze mode 1b - In doze mode
14 LPI2C3_IPG_STOP_MODE	LPI2C3 stop mode selection. This bitfield cannot change when LPI2C3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
13 LPI2C3_STOP_REQ	LPI2C3 stop request 0b - Stop request off 1b - Stop request on

Table continues on the next page...

Field	Function
12 LPI2C3_IPG_DOZE	LPI2C3 doze mode 0b - Not in doze mode 1b - In doze mode
11 LPI2C2_IPG_STOP_MODE	LPI2C2 stop mode selection. This bitfield cannot change when LPI2C2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPI2C2_STOP_REQ	LPI2C2 stop request 0b - Stop request off 1b - Stop request on
9 LPI2C2_IPG_DOZE	LPI2C2 doze mode 0b - Not in doze mode 1b - In doze mode
8 LPI2C1_IPG_STOP_MODE	LPI2C1 stop mode selection. This bitfield cannot change when LPI2C1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPI2C1_STOP_REQ	LPI2C1 stop request 0b - Stop request off 1b - Stop request on
6 LPI2C1_IPG_DOZE	LPI2C1 doze mode 0b - Not in doze mode 1b - In doze mode
5 GPT6_IPG_DOZE	GPT6 doze mode 0b - Not in doze mode 1b - In doze mode
4 GPT5_IPG_DOZE	GPT5 doze mode 0b - Not in doze mode 1b - In doze mode
3 GPT4_IPG_DOZE	GPT4 doze mode 0b - Not in doze mode 1b - In doze mode
2 GPT3_IPG_DOZE	GPT3 doze mode 0b - Not in doze mode 1b - In doze mode
1 GPT2_IPG_DOZE	GPT2 doze mode 0b - Not in doze mode 1b - In doze mode
0 GPT1_IPG_DOZE	GPT1 doze mode 0b - Not in doze mode 1b - In doze mode

12.4.3.33 GPR37 General Purpose Register (GPR37)

12.4.3.33.1 Offset

Register	Offset
GPR37	94h

12.4.3.33.2 Function

GPR Register

12.4.3.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DWP_LOCK		DWP		Reserved	LPUART4_IPG_STOP_MODE	LPUART4_STOP_REQ	LPUART4_IPG_DOZE	LPUART3_IPG_STOP_MODE	LPUART3_STOP_REQ	LPUART3_IPG_DOZE	LPUART2_IPG_STOP_MODE	LPUART2_STOP_REQ	LPUART2_IPG_DOZE	LPUART1_IPG_STOP_MODE	LPUART1_STOP_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	LPUART1_IPG_DOZE	LPSPI6_IPG_STOP_MODE	LPSPI6_STOP_REQ	LPSPI6_IPG_DOZE	LPSPI5_IPG_STOP_MODE	LPSPI5_STOP_REQ	LPSPI5_IPG_DOZE	LPSPI4_IPG_STOP_MODE	LPSPI4_STOP_REQ	LPSPI4_IPG_DOZE	LPSPI3_IPG_STOP_MODE	LPSPI3_STOP_REQ	LPSPI3_IPG_DOZE	LPSPI2_IPG_STOP_MODE	LPSPI2_STOP_REQ	LPSPI2_IPG_DOZE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.3.33.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved
26 LPUART4_IPG_STOP_MODE	LPUART4 stop mode selection. This bitfield cannot change when LPUART4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 LPUART4_STOP_REQ	LPUART4 stop request 0b - Stop request off 1b - Stop request on
24 LPUART4_IPG_DOZE	LPUART4 doze mode 0b - Not in doze mode 1b - In doze mode
23 LPUART3_IPG_STOP_MODE	LPUART3 stop mode selection. This bitfield cannot change when LPUART3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPUART3_STOP_REQ	LPUART3 stop request 0b - Stop request off 1b - Stop request on
21 LPUART3_IPG_DOZE	LPUART3 doze mode 0b - Not in doze mode 1b - In doze mode
20 LPUART2_IPG_STOP_MODE	LPUART2 stop mode selection. This bitfield cannot change when LPUART2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPUART2_STOP_REQ	LPUART2 stop request 0b - Stop request off 1b - Stop request on
18 LPUART2_IPG_DOZE	LPUART2 doze mode 0b - Not in doze mode 1b - In doze mode

Table continues on the next page...

Field	Function
17 LPUART1_IPG_STOP_MODE	LPUART1 stop mode selection. This bitfield cannot change when LPUART1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPUART1_STOP_REQ	LPUART1 stop request 0b - Stop request off 1b - Stop request on
15 LPUART1_IPG_DOZE	LPUART1 doze mode 0b - Not in doze mode 1b - In doze mode
14 LPSPI6_IPG_STOP_MODE	LPSPI6 stop mode selection. This bitfield cannot change when LPSPI6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
13 LPSPI6_STOP_REQ	LPSPI6 stop request 0b - Stop request off 1b - Stop request on
12 LPSPI6_IPG_DOZE	LPSPI6 doze mode 0b - Not in doze mode 1b - In doze mode
11 LPSPI5_IPG_STOP_MODE	LPSPI5 stop mode selection. This bitfield cannot change when LPSPI5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPSPI5_STOP_REQ	LPSPI5 stop request 0b - Stop request off 1b - Stop request on
9 LPSPI5_IPG_DOZE	LPSPI5 doze mode 0b - Not in doze mode 1b - In doze mode
8 LPSPI4_IPG_STOP_MODE	LPSPI4 stop mode selection. This bitfield cannot change when LPSPI4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPSPI4_STOP_REQ	LPSPI4 stop request 0b - Stop request off 1b - Stop request on
6 LPSPI4_IPG_DOZE	LPSPI4 doze mode 0b - Not in doze mode 1b - In doze mode
5 LPSPI3_IPG_STOP_MODE	LPSPI3 stop mode selection. This bitfield cannot change when LPSPI3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 LPSPI3_STOP_REQ	LPSPI3 stop request 0b - Stop request off 1b - Stop request on

Table continues on the next page...

Field	Function
3 LPSPI3_IPG_DOZE	LPSPI3 doze mode 0b - Not in doze mode 1b - In doze mode
2 LPSPI2_IPG_STOP_MODE	LPSPI2 stop mode selection. This bitfield cannot change when LPSPI2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1 LPSPI2_STOP_REQ	LPSPI2 stop request 0b - Stop request off 1b - Stop request on
0 LPSPI2_IPG_DOZE	LPSPI2 doze mode 0b - Not in doze mode 1b - In doze mode

12.4.3.34 GPR38 General Purpose Register (GPR38)

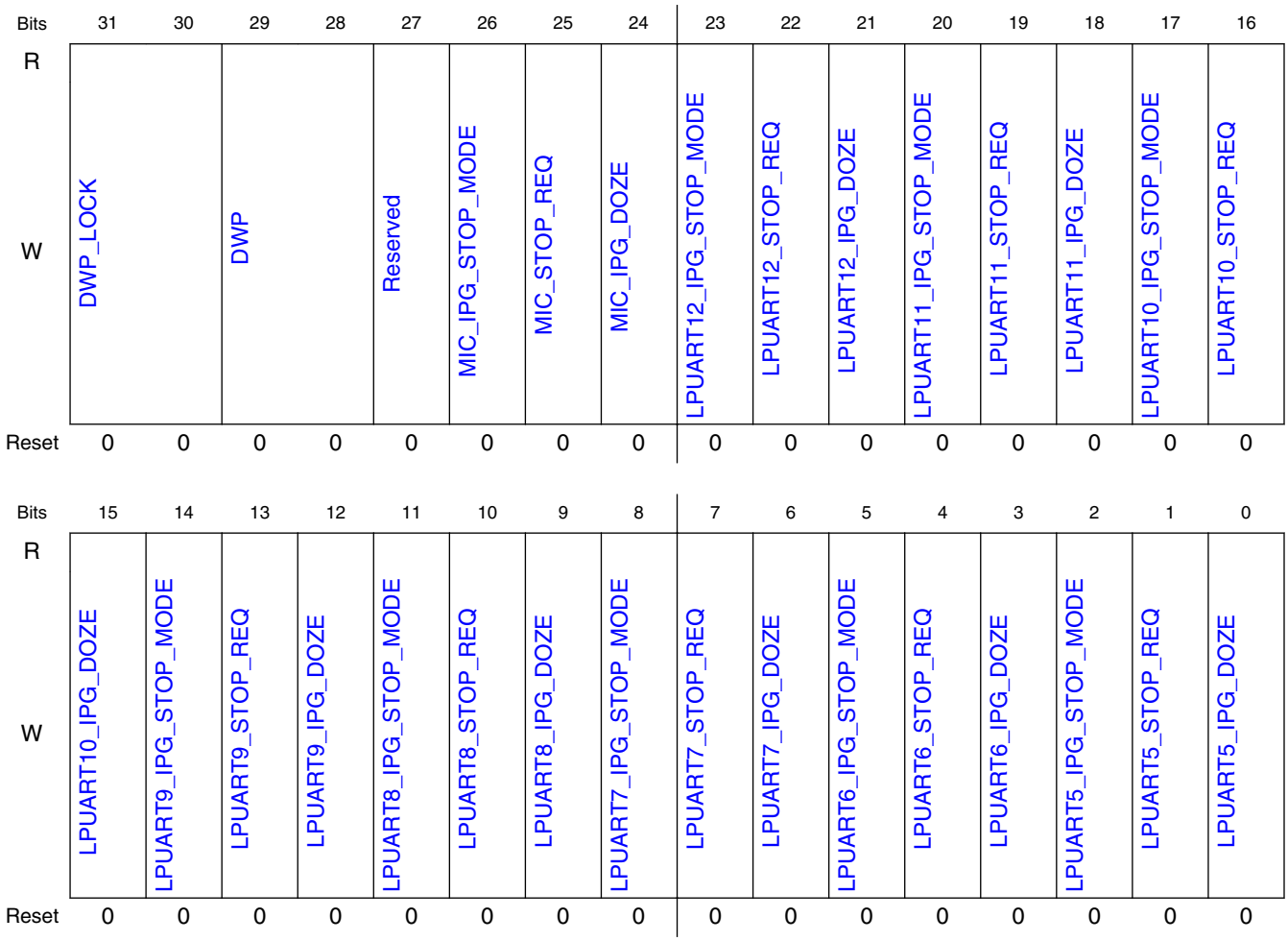
12.4.3.34.1 Offset

Register	Offset
GPR38	98h

12.4.3.34.2 Function

GPR Register

12.4.3.34.3 Diagram



12.4.3.34.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27	Reserved

Table continues on the next page...

Field	Function
—	
26 MIC_IPG_STOP_MODE	MIC stop mode selection. This bitfield cannot change when MIC_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 MIC_STOP_REQ	MIC stop request 0b - Stop request off 1b - Stop request on
24 MIC_IPG_DOZE	MIC doze mode 0b - Not in doze mode 1b - In doze mode
23 LPUART12_IPG_STOP_MODE	LPUART12 stop mode selection. This bitfield cannot change when LPUART12_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPUART12_STOP_REQ	LPUART12 stop request 0b - Stop request off 1b - Stop request on
21 LPUART12_IPG_DOZE	LPUART12 doze mode 0b - Not in doze mode 1b - In doze mode
20 LPUART11_IPG_STOP_MODE	LPUART11 stop mode selection. This bitfield cannot change when LPUART11_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPUART11_STOP_REQ	LPUART11 stop request 0b - Stop request off 1b - Stop request on
18 LPUART11_IPG_DOZE	LPUART11 doze mode 0b - Not in doze mode 1b - In doze mode
17 LPUART10_IPG_STOP_MODE	LPUART10 stop mode selection. This bitfield cannot change when LPUART10_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPUART10_STOP_REQ	LPUART10 stop request 0b - Stop request off 1b - Stop request on
15 LPUART10_IPG_DOZE	LPUART10 doze mode 0b - Not in doze mode 1b - In doze mode
14 LPUART9_IPG_STOP_MODE	LPUART9 stop mode selection. This bitfield cannot change when LPUART9_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
13 LPUART9_STOP_REQ	LPUART9 stop request 0b - Stop request off 1b - Stop request on

Table continues on the next page...

Field	Function
12 LPUART9_IPG_DOZE	LPUART9 doze mode 0b - Not in doze mode 1b - In doze mode
11 LPUART8_IPG_STOP_MODE	LPUART8 stop mode selection. This bitfield cannot change when LPUART8_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPUART8_STOP_REQ	LPUART8 stop request 0b - Stop request off 1b - Stop request on
9 LPUART8_IPG_DOZE	LPUART8 doze mode 0b - Not in doze mode 1b - In doze mode
8 LPUART7_IPG_STOP_MODE	LPUART7 stop mode selection. This bitfield cannot change when LPUART7_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPUART7_STOP_REQ	LPUART7 stop request 0b - Stop request off 1b - Stop request on
6 LPUART7_IPG_DOZE	LPUART7 doze mode 0b - Not in doze mode 1b - In doze mode
5 LPUART6_IPG_STOP_MODE	LPUART6 stop mode selection. This bitfield cannot change when LPUART6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 LPUART6_STOP_REQ	LPUART6 stop request 0b - Stop request off 1b - Stop request on
3 LPUART6_IPG_DOZE	LPUART6 doze mode 0b - Not in doze mode 1b - In doze mode
2 LPUART5_IPG_STOP_MODE	LPUART5 stop mode selection. This bitfield cannot change when LPUART5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1 LPUART5_STOP_REQ	LPUART5 stop request 0b - Stop request off 1b - Stop request on
0 LPUART5_IPG_DOZE	LPUART5 doze mode 0b - Not in doze mode 1b - In doze mode

12.4.3.35 GPR39 General Purpose Register (GPR39)

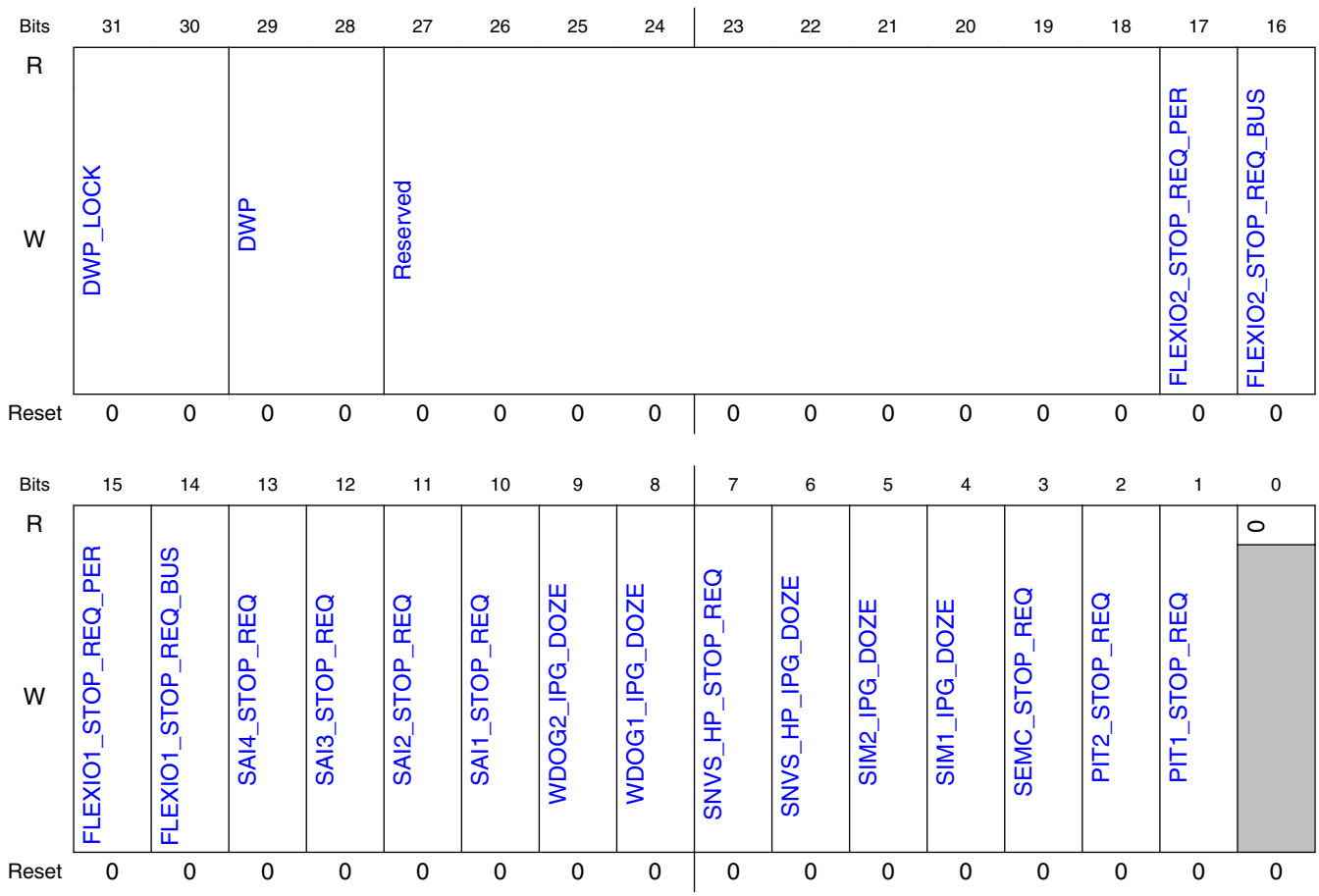
12.4.3.35.1 Offset

Register	Offset
GPR39	9Ch

12.4.3.35.2 Function

GPR Register

12.4.3.35.3 Diagram



12.4.3.35.4 Fields

Field	Function
31-30	Domain write protection lock

Table continues on the next page...

Field	Function
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-18 —	Reserved
17 FLEXIO2_STOP_REQ_PER	FLEXIO2 peripheral clock domain stop request 0b - Stop request off 1b - Stop request on
16 FLEXIO2_STOP_REQ_BUS	FLEXIO2 bus clock domain stop request 0b - Stop request off 1b - Stop request on
15 FLEXIO1_STOP_REQ_PER	FLEXIO1 peripheral clock domain stop request 0b - Stop request off 1b - Stop request on
14 FLEXIO1_STOP_REQ_BUS	FLEXIO1 bus clock domain stop request 0b - Stop request off 1b - Stop request on
13 SAI4_STOP_REQ	SAI4 stop request 0b - Stop request off 1b - Stop request on
12 SAI3_STOP_REQ	SAI3 stop request 0b - Stop request off 1b - Stop request on
11 SAI2_STOP_REQ	SAI2 stop request 0b - Stop request off 1b - Stop request on
10 SAI1_STOP_REQ	SAI1 stop request 0b - Stop request off 1b - Stop request on
9 WDOG2_IPG_DOZE	WDOG2 doze mode 0b - Not in doze mode 1b - In doze mode
8 WDOG1_IPG_DOZE	WDOG1 doze mode 0b - Not in doze mode 1b - In doze mode
7	SNVS_HP stop request 0b - Stop request off

Table continues on the next page...

Field	Function
SNVS_HP_STOP_REQ	1b - Stop request on
6 SNVS_HP_IPG_DOZE	SNVS_HP doze mode 0b - Not in doze mode 1b - In doze mode
5 SIM2_IPG_DOZE	SIM2 doze mode 0b - Not in doze mode 1b - In doze mode
4 SIM1_IPG_DOZE	SIM1 doze mode 0b - Not in doze mode 1b - In doze mode
3 SEMC_STOP_REQ	SEMC stop request 0b - Stop request off 1b - Stop request on
2 PIT2_STOP_REQ	PIT2 stop request 0b - Stop request off 1b - Stop request on
1 PIT1_STOP_REQ	PIT1 stop request 0b - Stop request off 1b - Stop request on
0 —	Reserved

12.4.3.36 GPR40 General Purpose Register (GPR40)

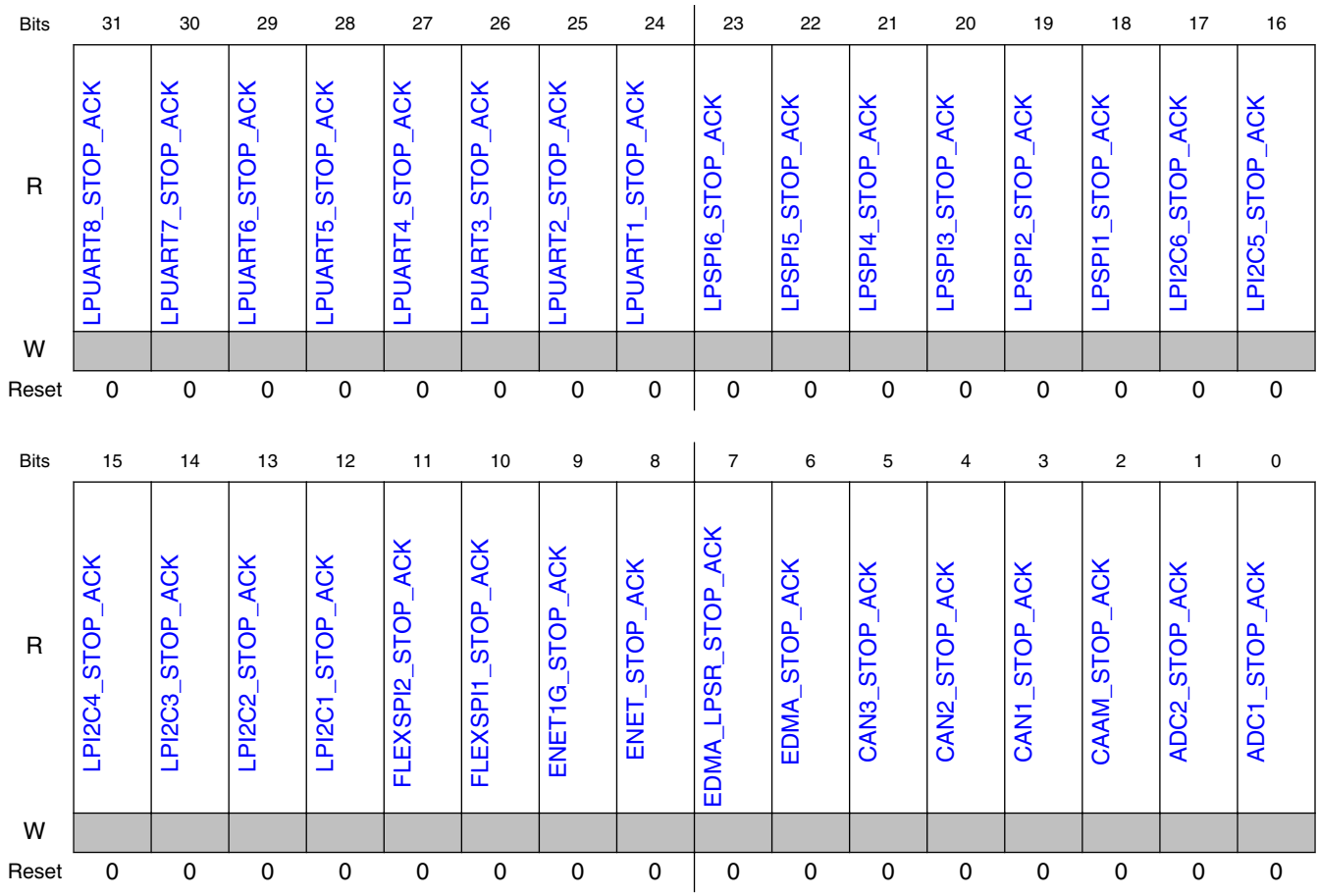
12.4.3.36.1 Offset

Register	Offset
GPR40	A0h

12.4.3.36.2 Function

GPR Register

12.4.3.36.3 Diagram



12.4.3.36.4 Fields

Field	Function
31 LPUART8_STOP_ACK	LPUART8 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
30 LPUART7_STOP_ACK	LPUART7 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
29 LPUART6_STOP_ACK	LPUART6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
28 LPUART5_STOP_ACK	LPUART5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
27	LPUART4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
LPUART4_STOP_ACK	
26 LPUART3_STOP_ACK	LPUART3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
25 LPUART2_STOP_ACK	LPUART2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
24 LPUART1_STOP_ACK	LPUART1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
23 LPSPI6_STOP_ACK	LPSPI6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
22 LPSPI5_STOP_ACK	LPSPI5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
21 LPSPI4_STOP_ACK	LPSPI4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
20 LPSPI3_STOP_ACK	LPSPI3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
19 LPSPI2_STOP_ACK	LPSPI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
18 LPSPI1_STOP_ACK	LPSPI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
17 LPI2C6_STOP_ACK	LPI2C6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
16 LPI2C5_STOP_ACK	LPI2C5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
15 LPI2C4_STOP_ACK	LPI2C4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
14 LPI2C3_STOP_ACK	LPI2C3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
13 LPI2C2_STOP_ACK	LPI2C2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
12 LPI2C1_STOP_ACK	LPI2C1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
11 FLEXSPI2_STOP_ACK	FLEXSPI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
10 FLEXSPI1_STOP_ACK	FLEXSPI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
9 ENET1G_STOP_ACK	ENET1G stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
8 ENET_STOP_ACK	ENET stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
7 EDMA_LPSR_STOP_ACK	EDMA_LPSR stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
6 EDMA_STOP_ACK	EDMA stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
5 CAN3_STOP_ACK	CAN3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
4 CAN2_STOP_ACK	CAN2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
3 CAN1_STOP_ACK	CAN1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
2 CAAM_STOP_ACK	CAAM stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
1 ADC2_STOP_ACK	ADC2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
0 ADC1_STOP_ACK	ADC1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

12.4.3.37 GPR41 General Purpose Register (GPR41)

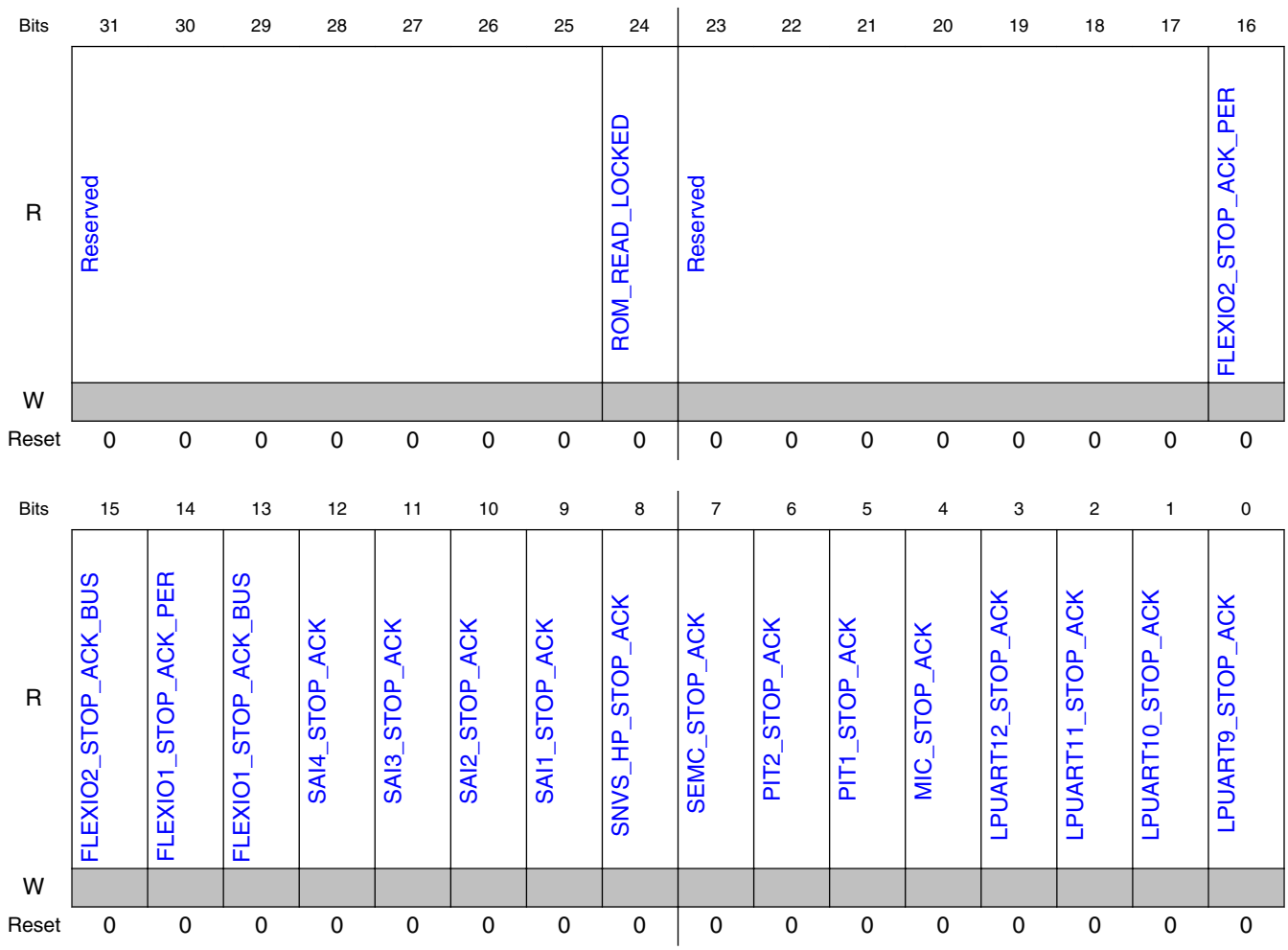
12.4.3.37.1 Offset

Register	Offset
GPR41	A4h

12.4.3.37.2 Function

GPR Register

12.4.3.37.3 Diagram



12.4.3.37.4 Fields

Field	Function
31-25	Reserved

Table continues on the next page...

Field	Function
—	
24 ROM_READ_LOCKED	ROM read lock status bit When is bit is 1, it indicates that the first 64KB of ROM code are locked and not readable anymore.
23-17 —	Reserved
16 FLEXIO2_STOP_ACK_PER	FLEXIO2 stop acknowledge of peripheral clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
15 FLEXIO2_STOP_ACK_BUS	FLEXIO2 stop acknowledge of bus clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
14 FLEXIO1_STOP_ACK_PER	FLEXIO1 stop acknowledge of peripheral clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
13 FLEXIO1_STOP_ACK_BUS	FLEXIO1 stop acknowledge of bus clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
12 SAI4_STOP_ACK	SAI4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
11 SAI3_STOP_ACK	SAI3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
10 SAI2_STOP_ACK	SAI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
9 SAI1_STOP_ACK	SAI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
8 SNVS_HP_STOP_ACK	SNVS_HP stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
7 SEMC_STOP_ACK	SEMC stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
6 PIT2_STOP_ACK	PIT2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
5 PIT1_STOP_ACK	PIT1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
4	MIC stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
MIC_STOP_ACK	
3 LPUART12_STOP_ACK	LPUART12 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
2 LPUART11_STOP_ACK	LPUART11 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
1 LPUART10_STOP_ACK	LPUART10 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
0 LPUART9_STOP_ACK	LPUART9 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

12.4.4 IOMUXC LPSR register descriptions

12.4.4.1 iomuxc_lpsr memory map

IOMUXC_LPSR base address: 40C0_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SW_MUX_CTL_PAD_GPIO_LPSR_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_00)	32	RW	0000_000Ah
4h	SW_MUX_CTL_PAD_GPIO_LPSR_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_01)	32	RW	0000_000Ah
8h	SW_MUX_CTL_PAD_GPIO_LPSR_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_02)	32	RW	0000_0000h
Ch	SW_MUX_CTL_PAD_GPIO_LPSR_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_03)	32	RW	0000_0000h
10h	SW_MUX_CTL_PAD_GPIO_LPSR_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_04)	32	RW	0000_000Ah
14h	SW_MUX_CTL_PAD_GPIO_LPSR_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_05)	32	RW	0000_000Ah
18h	SW_MUX_CTL_PAD_GPIO_LPSR_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_06)	32	RW	0000_000Ah
1Ch	SW_MUX_CTL_PAD_GPIO_LPSR_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_07)	32	RW	0000_000Ah

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
20h	SW_MUX_CTL_PAD_GPIO_LPSR_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_08)	32	RW	0000_000Ah
24h	SW_MUX_CTL_PAD_GPIO_LPSR_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_09)	32	RW	0000_000Ah
28h	SW_MUX_CTL_PAD_GPIO_LPSR_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_10)	32	RW	0000_0000h
2Ch	SW_MUX_CTL_PAD_GPIO_LPSR_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_11)	32	RW	0000_0000h
30h	SW_MUX_CTL_PAD_GPIO_LPSR_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_12)	32	RW	0000_0000h
34h	SW_MUX_CTL_PAD_GPIO_LPSR_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_13)	32	RW	0000_0000h
38h	SW_MUX_CTL_PAD_GPIO_LPSR_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_14)	32	RW	0000_0000h
3Ch	SW_MUX_CTL_PAD_GPIO_LPSR_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_15)	32	RW	0000_0000h
40h	SW_PAD_CTL_PAD_GPIO_LPSR_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_00)	32	RW	0000_0002h
44h	SW_PAD_CTL_PAD_GPIO_LPSR_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_01)	32	RW	0000_0002h
48h	SW_PAD_CTL_PAD_GPIO_LPSR_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_02)	32	RW	0000_0002h
4Ch	SW_PAD_CTL_PAD_GPIO_LPSR_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_03)	32	RW	0000_0002h
50h	SW_PAD_CTL_PAD_GPIO_LPSR_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_04)	32	RW	0000_0002h
54h	SW_PAD_CTL_PAD_GPIO_LPSR_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_05)	32	RW	0000_0002h
58h	SW_PAD_CTL_PAD_GPIO_LPSR_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_06)	32	RW	0000_0002h
5Ch	SW_PAD_CTL_PAD_GPIO_LPSR_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_07)	32	RW	0000_0002h
60h	SW_PAD_CTL_PAD_GPIO_LPSR_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_08)	32	RW	0000_0002h
64h	SW_PAD_CTL_PAD_GPIO_LPSR_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_09)	32	RW	0000_0002h
68h	SW_PAD_CTL_PAD_GPIO_LPSR_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_10)	32	RW	0000_000Eh
6Ch	SW_PAD_CTL_PAD_GPIO_LPSR_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_11)	32	RW	0000_0002h
70h	SW_PAD_CTL_PAD_GPIO_LPSR_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_12)	32	RW	0000_000Eh
74h	SW_PAD_CTL_PAD_GPIO_LPSR_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_13)	32	RW	0000_0002h
78h	SW_PAD_CTL_PAD_GPIO_LPSR_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_14)	32	RW	0000_0006h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7Ch	SW_PAD_CTL_PAD_GPIO_LPSR_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_15)	32	RW	0000_000Eh
80h	CAN3_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN3_IPP_IND_CANRX_SELECT_INPUT)	32	RW	0000_0000h
84h	LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
88h	LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
8Ch	LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
90h	LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
94h	LPSPI5_IPP_IND_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI5_IPP_IND_LPSPI_PCS_SELECT_INPUT_0)	32	RW	0000_0000h
98h	LPSPI5_IPP_IND_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SCK_SELECT_INPUT)	32	RW	0000_0000h
9Ch	LPSPI5_IPP_IND_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SDI_SELECT_INPUT)	32	RW	0000_0000h
A0h	LPSPI5_IPP_IND_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SDO_SELECT_INPUT)	32	RW	0000_0000h
A4h	LPUART11_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART11_IPP_IND_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
A8h	LPUART11_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART11_IPP_IND_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
ACh	LPUART12_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART12_IPP_IND_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
B0h	LPUART12_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART12_IPP_IND_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
B4h	MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0)	32	RW	0000_0000h
B8h	MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1)	32	RW	0000_0000h
BCh	MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2)	32	RW	0000_0000h
C0h	MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3)	32	RW	0000_0000h
C4h	NMI_GLUE_IPP_IND_NMI_SELECT_INPUT DAISY Register (NMI_GLUE_IPP_IND_NMI_SELECT_INPUT)	32	RW	0000_0000h
C8h	SAI4_IPG_CLK_SAI_MCLK_SELECT_INPUT DAISY Register (SAI4_IPG_CLK_SAI_MCLK_SELECT_INPUT)	32	RW	0000_0000h
CCh	SAI4_IPP_IND_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_RXBCLK_SELECT_INPUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
D0h	SAI4_IPP_IND_SAI_RXDATA_SELECT_INPUT_0 DAISY Register (SAI4_IPP_IND_SAI_RXDATA_SELECT_INPUT_0)	32	RW	0000_0000h
D4h	SAI4_IPP_IND_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_RXSYNC_SELECT_INPUT)	32	RW	0000_0000h
D8h	SAI4_IPP_IND_SAI_TXBCLK_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_TXBCLK_SELECT_INPUT)	32	RW	0000_0000h
DCh	SAI4_IPP_IND_SAI_TXSYNC_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_TXSYNC_SELECT_INPUT)	32	RW	0000_0000h

12.4.4.2 SW_MUX_CTL_PAD_GPIO_LPSR_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_00)

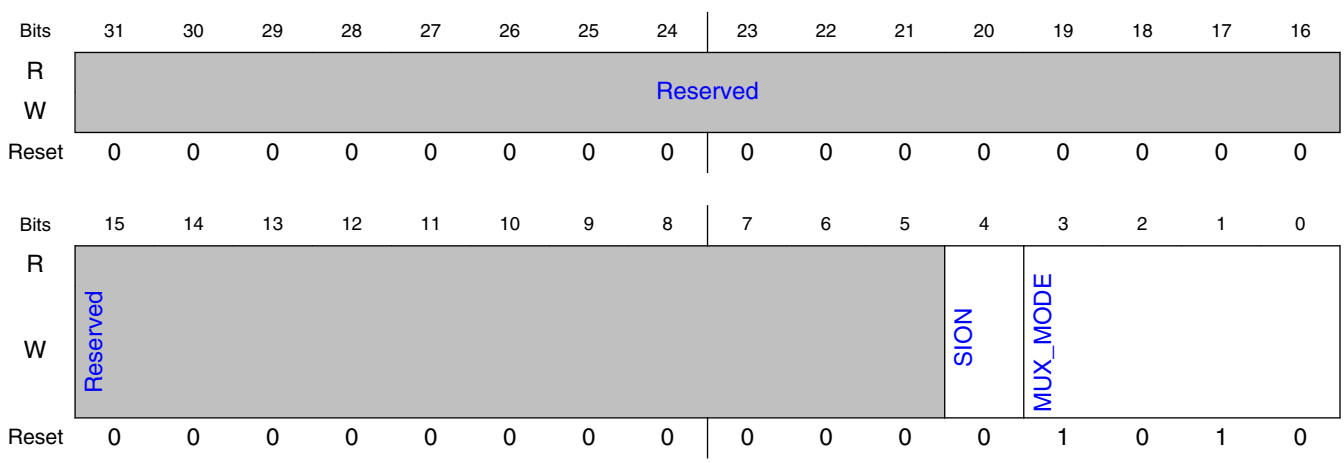
12.4.4.2.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_00	0h

12.4.4.2.2 Function

SW_MUX_CTL Register

12.4.4.2.3 Diagram



12.4.4.2.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_LPSR_00. 0000b - Select mux mode: ALT0 mux port: FLEXCAN3_TX of instance: FLEXCAN3 0001b - Select mux mode: ALT1 mux port: MIC_CLK of instance: MIC 0010b - Select mux mode: ALT2 mux port: MQS_RIGHT of instance: MQS 0011b - Select mux mode: ALT3 mux port: ARM_CM4_EVENTO of instance: CM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO00 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPUART12_TXD of instance: LPUART12 0111b - Select mux mode: ALT7 mux port: SAI4_MCLK of instance: SAI4 1010b - Select mux mode: ALT10 mux port: GPIO12_IO00 of instance: GPIO12

12.4.4.3 SW_MUX_CTL_PAD_GPIO_LPSR_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_01)

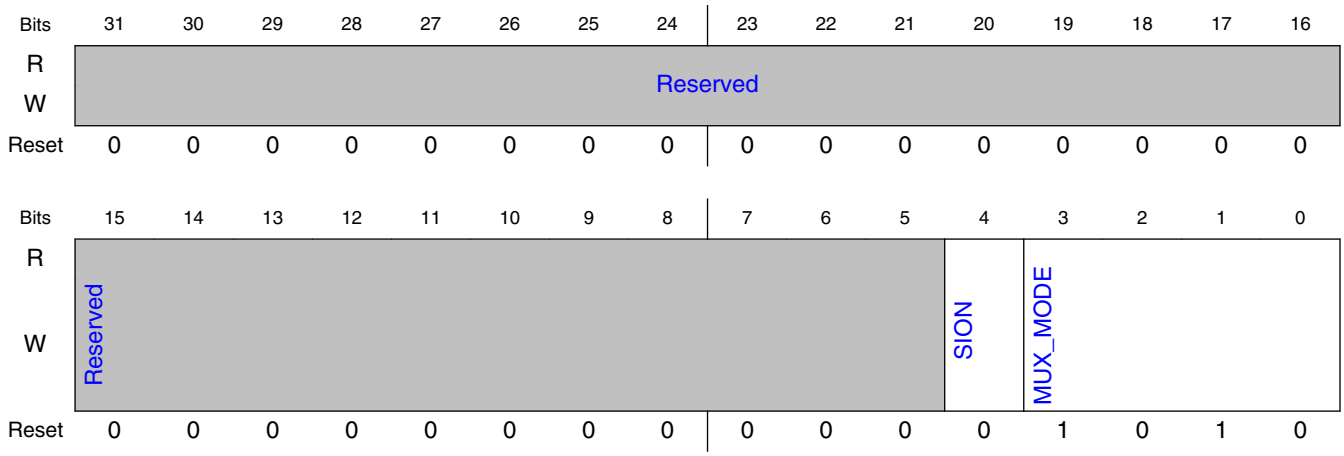
12.4.4.3.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_01	4h

12.4.4.3.2 Function

SW_MUX_CTL Register

12.4.4.3.3 Diagram



12.4.4.3.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_LPSR_01. 0000b - Select mux mode: ALT0 mux port: FLEXCAN3_RX of instance: FLEXCAN3 0001b - Select mux mode: ALT1 mux port: MIC_BITSTREAM0 of instance: MIC 0010b - Select mux mode: ALT2 mux port: MQS_LEFT of instance: MQS 0011b - Select mux mode: ALT3 mux port: ARM_CM4_EVENTI of instance: CM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO01 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPUART12_RXD of instance: LPUART12 1010b - Select mux mode: ALT10 mux port: GPIO12_IO01 of instance: GPIO12

12.4.4.4 SW_MUX_CTL_PAD_GPIO_LPSR_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_02)

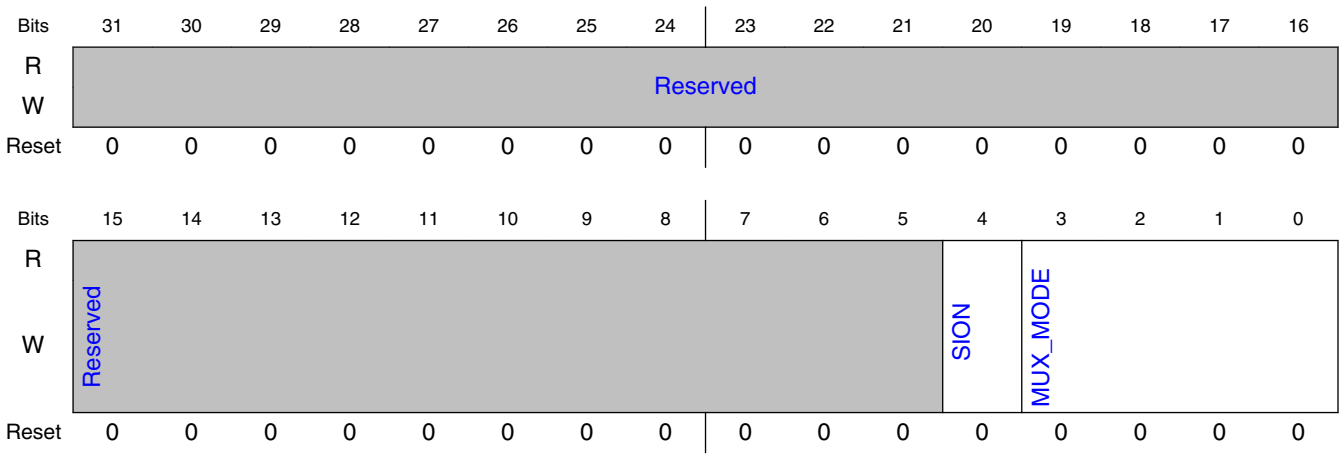
12.4.4.4.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_02	8h

12.4.4.4.2 Function

SW_MUX_CTL Register

12.4.4.4.3 Diagram



12.4.4.4.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_LPSR_02. 0000b - Select mux mode: ALT0 mux port: SRC_BOOT_MODE00 of instance: SRC 0001b - Select mux mode: ALT1 mux port: LPSPI5_SCK of instance: LPSPI5 0010b - Select mux mode: ALT2 mux port: SAI4_TX_DATA of instance: SAI4 0011b - Select mux mode: ALT3 mux port: MQS_RIGHT of instance: MQS 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO02 of instance: GPIO_MUX6 1010b - Select mux mode: ALT10 mux port: GPIO12_IO02 of instance: GPIO12

12.4.4.5 SW_MUX_CTL_PAD_GPIO_LPSR_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_03)

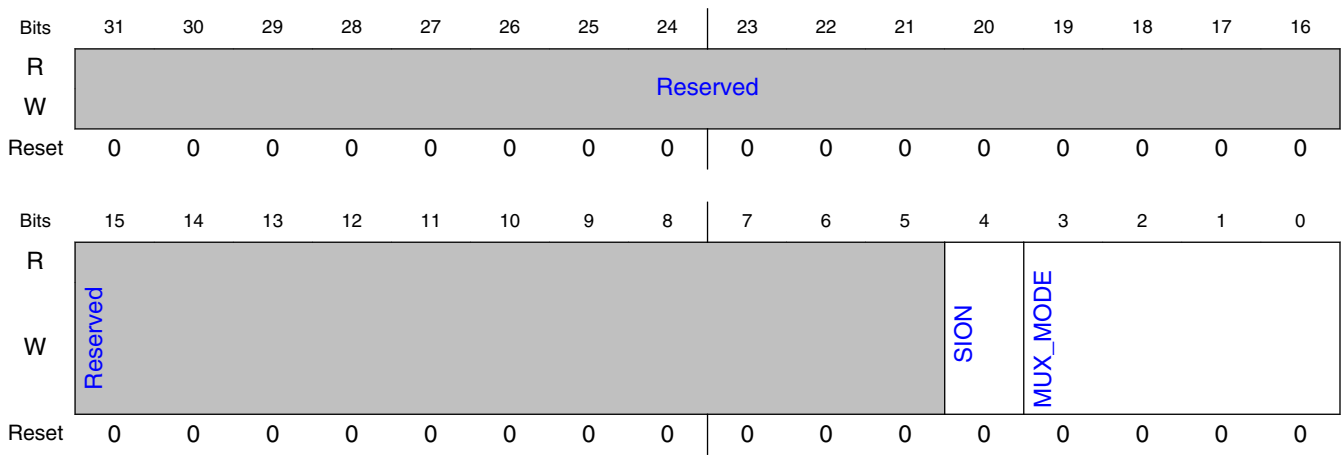
12.4.4.5.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_03	Ch

12.4.4.5.2 Function

SW_MUX_CTL Register

12.4.4.5.3 Diagram



12.4.4.5.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_LPSR_03. 0000b - Select mux mode: ALT0 mux port: SRC_BOOT_MODE01 of instance: SRC 0001b - Select mux mode: ALT1 mux port: LPSPI5_PCS0 of instance: LPSPI5 0010b - Select mux mode: ALT2 mux port: SAI4_TX_SYNC of instance: SAI4 0011b - Select mux mode: ALT3 mux port: MQS_LEFT of instance: MQS 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO03 of instance: GPIO_MUX6 1010b - Select mux mode: ALT10 mux port: GPIO12_IO03 of instance: GPIO12

12.4.4.6 SW_MUX_CTL_PAD_GPIO_LPSR_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_04)

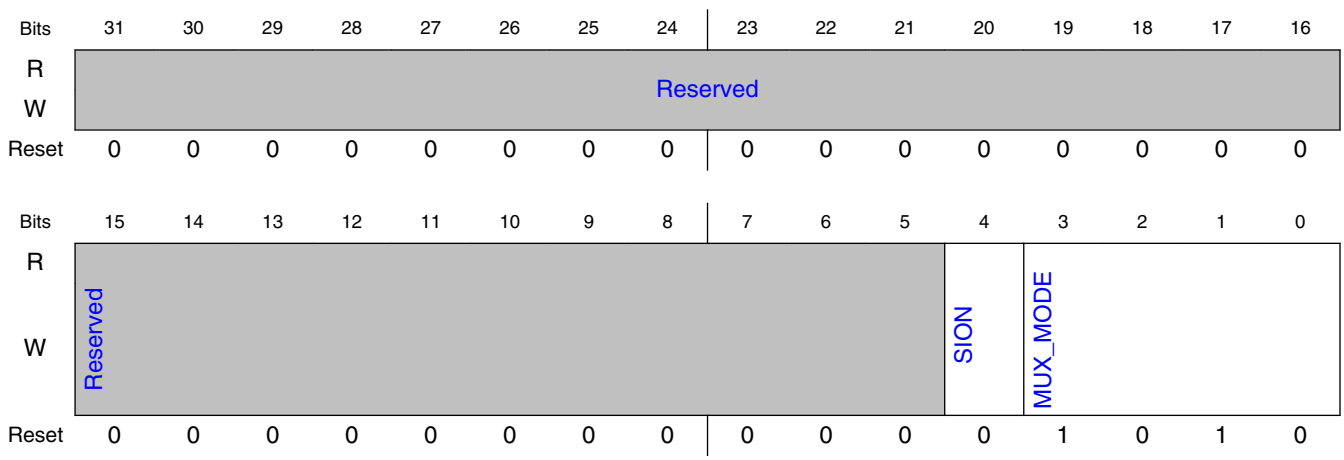
12.4.4.6.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_04	10h

12.4.4.6.2 Function

SW_MUX_CTL Register

12.4.4.6.3 Diagram



12.4.4.6.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_LPSR_04.

Field	Function
	0000b - Select mux mode: ALT0 mux port: LPI2C5_SDA of instance: LPI2C5 0001b - Select mux mode: ALT1 mux port: LPSPI5_SOUT of instance: LPSPI5 0010b - Select mux mode: ALT2 mux port: SAI4_TX_BCLK of instance: SAI4 0011b - Select mux mode: ALT3 mux port: LPUART12_RTS_B of instance: LPUART12 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO04 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPUART11_TXD of instance: LPUART11 1010b - Select mux mode: ALT10 mux port: GPIO12_IO04 of instance: GPIO12

12.4.4.7 SW_MUX_CTL_PAD_GPIO_LPSR_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_05)

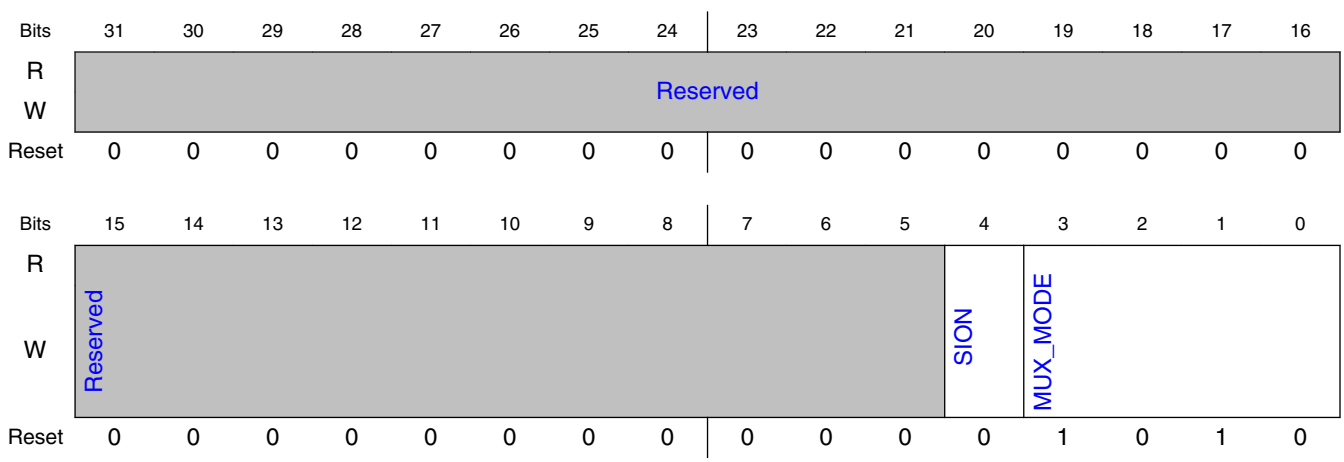
12.4.4.7.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_05	14h

12.4.4.7.2 Function

SW_MUX_CTL Register

12.4.4.7.3 Diagram



12.4.4.7.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_LPSR_05. 0000b - Select mux mode: ALT0 mux port: LPI2C5_SCL of instance: LPI2C5 0001b - Select mux mode: ALT1 mux port: LPSPI5_SIN of instance: LPSPI5 0010b - Select mux mode: ALT2 mux port: SAI4_MCLK of instance: SAI4 0011b - Select mux mode: ALT3 mux port: LPUART12_CTS_B of instance: LPUART12 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO05 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPUART11_RXD of instance: LPUART11 0111b - Select mux mode: NMI_GLUE_NMI of instance: nmi_glue 1010b - Select mux mode: ALT10 mux port: GPIO12_IO05 of instance: GPIO12

12.4.4.8 SW_MUX_CTL_PAD_GPIO_LPSR_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_06)

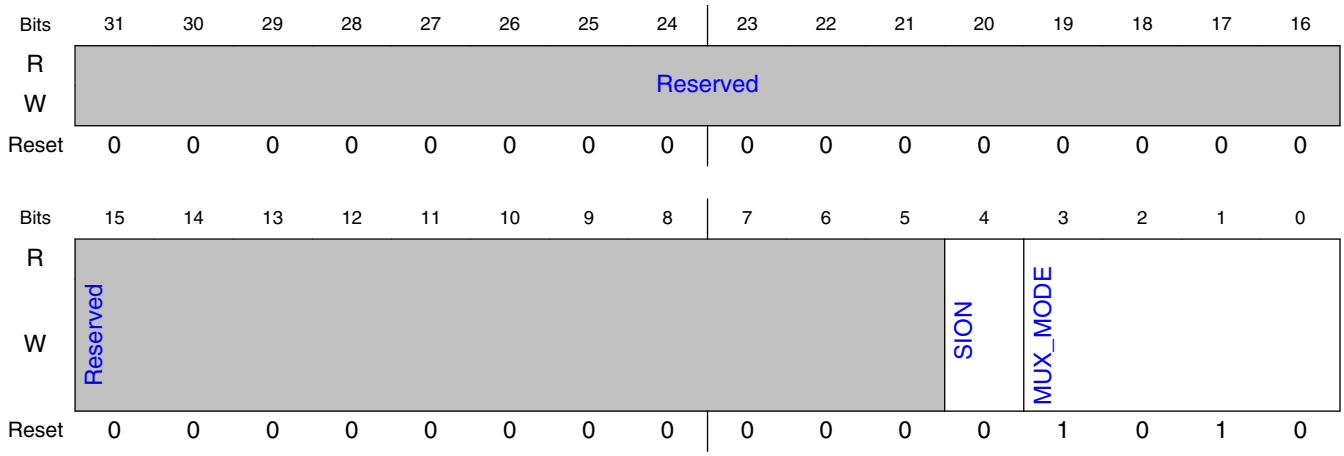
12.4.4.8.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_06	18h

12.4.4.8.2 Function

SW_MUX_CTL Register

12.4.4.8.3 Diagram



12.4.4.8.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_LPSR_06. 0000b - Select mux mode: ALT0 mux port: LPI2C6_SDA of instance: LPI2C6 0010b - Select mux mode: ALT2 mux port: SAI4_RX_DATA of instance: SAI4 0011b - Select mux mode: ALT3 mux port: LPUART12_TXD of instance: LPUART12 0100b - Select mux mode: ALT4 mux port: LPSPI6_PCS3 of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO06 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: FLEXCAN3_TX of instance: FLEXCAN3 0111b - Select mux mode: ALT7 mux port: PIT2_TRIGGER3 of instance: PIT2 1000b - Select mux mode: ALT8 mux port: LPSPI5_PCS1 of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO06 of instance: GPIO12

12.4.4.9 SW_MUX_CTL_PAD_GPIO_LPSR_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_07)

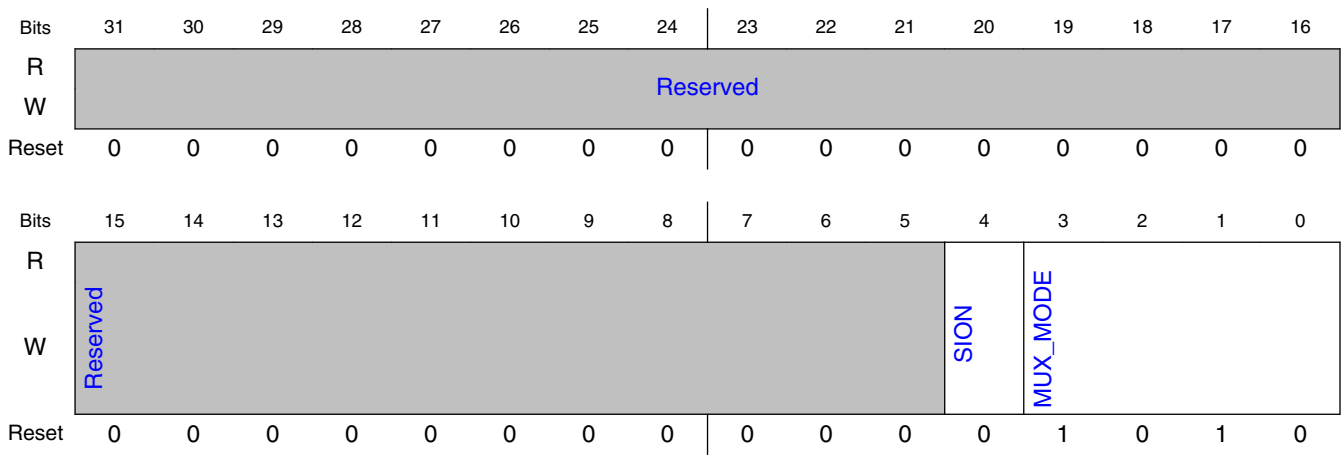
12.4.4.9.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_07	1Ch

12.4.4.9.2 Function

SW_MUX_CTL Register

12.4.4.9.3 Diagram



12.4.4.9.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_LPSR_07. 0000b - Select mux mode: ALT0 mux port: LPI2C6_SCL of instance: LPI2C6 0010b - Select mux mode: ALT2 mux port: SAI4_RX_BCLK of instance: SAI4 0011b - Select mux mode: ALT3 mux port: LPUART12_RXD of instance: LPUART12 0100b - Select mux mode: ALT4 mux port: LPSPI6_PCS2 of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO07 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: FLEXCAN3_RX of instance: FLEXCAN3 0111b - Select mux mode: ALT7 mux port: PIT2_TRIGGER2 of instance: PIT2 1000b - Select mux mode: ALT8 mux port: LPSPI5_PCS2 of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO07 of instance: GPIO12

12.4.4.10 SW_MUX_CTL_PAD_GPIO_LPSR_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_08)

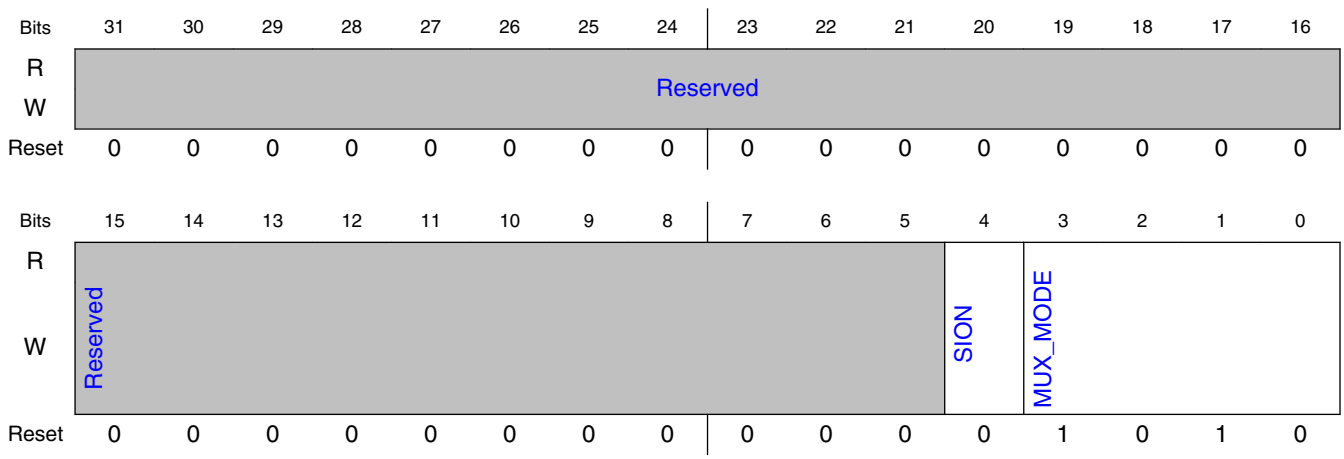
12.4.4.10.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_08	20h

12.4.4.10.2 Function

SW_MUX_CTL Register

12.4.4.10.3 Diagram



12.4.4.10.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_08
3-0	MUX Mode Select Field.

Field	Function
MUX_MODE	Select 1 of 10 iomux modes to be used for pad: GPIO_LPSR_08. 0000b - Select mux mode: ALT0 mux port: LPUART11_TXD of instance: LPUART11 0001b - Select mux mode: ALT1 mux port: FLEXCAN3_TX of instance: FLEXCAN3 0010b - Select mux mode: ALT2 mux port: SAI4_RX_SYNC of instance: SAI4 0011b - Select mux mode: ALT3 mux port: MIC_CLK of instance: MIC 0100b - Select mux mode: ALT4 mux port: LPSPI6_PCS1 of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO08 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPI2C5_SDA of instance: LPI2C5 0111b - Select mux mode: ALT7 mux port: PIT2_TRIGGER1 of instance: PIT2 1000b - Select mux mode: ALT8 mux port: LPSPI5_PCS3 of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO08 of instance: GPIO12

12.4.4.11 SW_MUX_CTL_PAD_GPIO_LPSR_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_09)

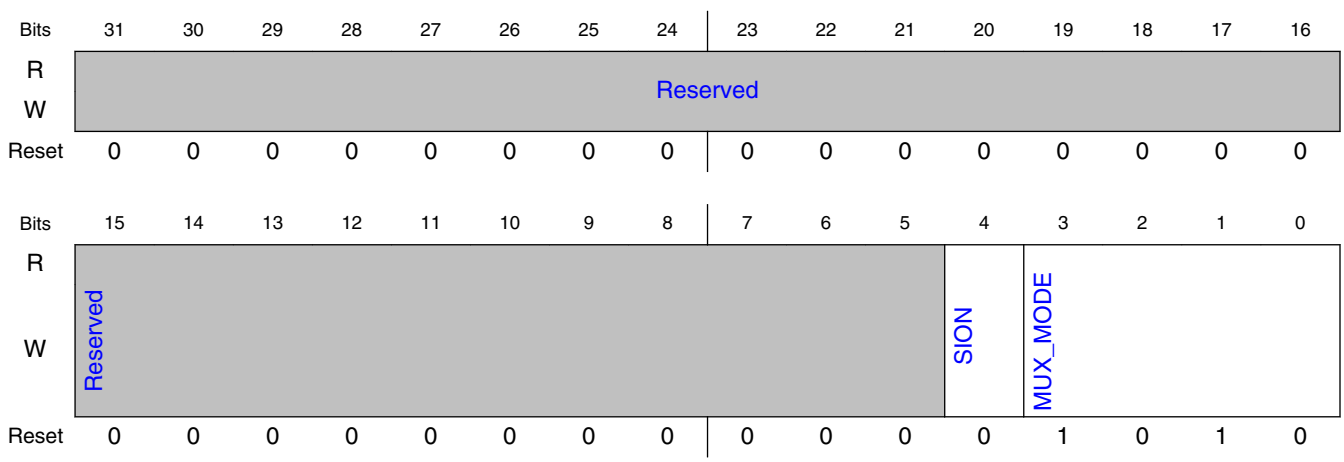
12.4.4.11.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_09	24h

12.4.4.11.2 Function

SW_MUX_CTL Register

12.4.4.11.3 Diagram



12.4.4.11.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_LPSR_09. 0000b - Select mux mode: ALT0 mux port: LPUART11_RXD of instance: LPUART11 0001b - Select mux mode: ALT1 mux port: FLEXCAN3_RX of instance: FLEXCAN3 0010b - Select mux mode: ALT2 mux port: PIT2_TRIGGER0 of instance: PIT2 0011b - Select mux mode: ALT3 mux port: MIC_BITSTREAM0 of instance: MIC 0100b - Select mux mode: ALT4 mux port: LPSPI6_PCS0 of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO09 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPI2C5_SCL of instance: LPI2C5 0111b - Select mux mode: ALT7 mux port: SAI4_TX_DATA of instance: SAI4 1010b - Select mux mode: ALT10 mux port: GPIO12_IO09 of instance: GPIO12

12.4.4.12 SW_MUX_CTL_PAD_GPIO_LPSR_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_10)

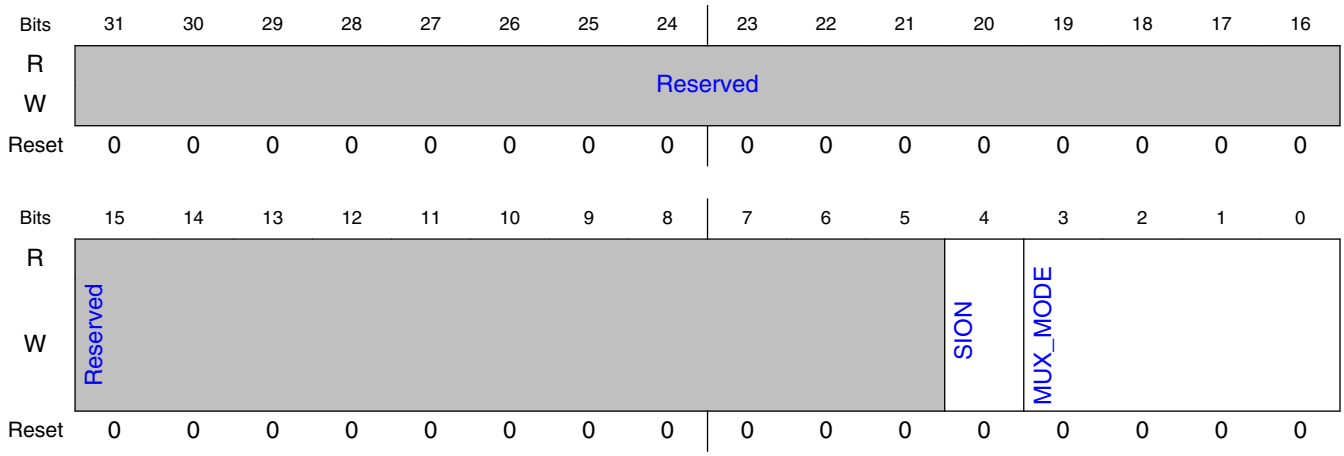
12.4.4.12.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_10	28h

12.4.4.12.2 Function

SW_MUX_CTL Register

12.4.4.12.3 Diagram



12.4.4.12.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_LPSR_10. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_TRSTB of instance: JTAG_MUX 0001b - Select mux mode: ALT1 mux port: LPUART11_CTS_B of instance: LPUART11 0010b - Select mux mode: ALT2 mux port: LPI2C6_SDA of instance: LPI2C6 0011b - Select mux mode: ALT3 mux port: MIC_BITSTREAM1 of instance: MIC 0100b - Select mux mode: ALT4 mux port: LPSPI6_SCK of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO10 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPI2C5_SCLS of instance: LPI2C5 0111b - Select mux mode: ALT7 mux port: SAI4_TX_SYNC of instance: SAI4 1000b - Select mux mode: ALT8 mux port: LPUART12_TXD of instance: LPUART12 1010b - Select mux mode: ALT10 mux port: GPIO12_IO10 of instance: GPIO12

12.4.4.13 SW_MUX_CTL_PAD_GPIO_LPSR_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_11)

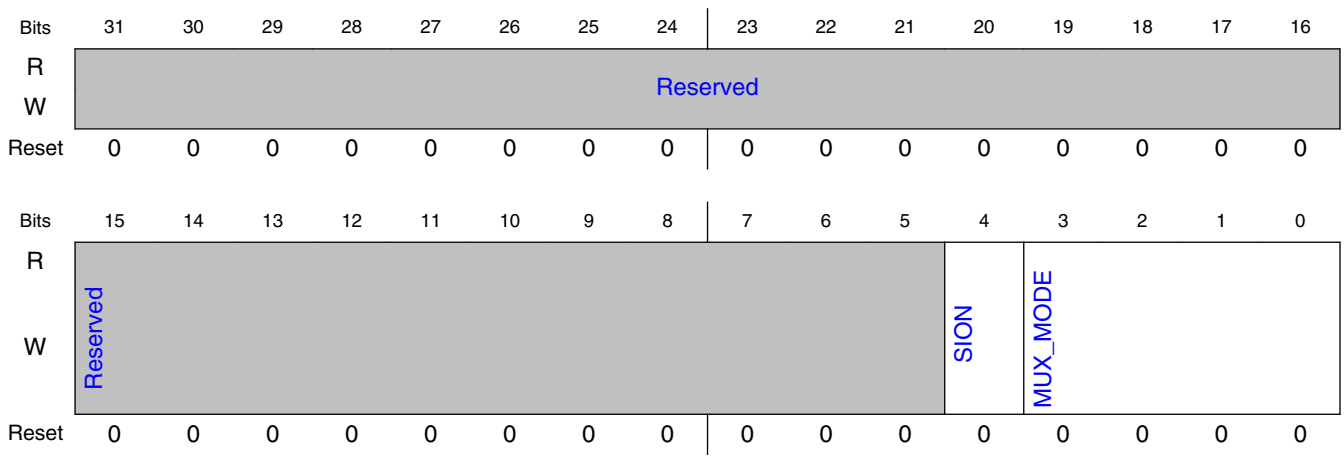
12.4.4.13.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_11	2Ch

12.4.4.13.2 Function

SW_MUX_CTL Register

12.4.4.13.3 Diagram



12.4.4.13.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_LPSR_11. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_TDO of instance: JTAG_MUX 0001b - Select mux mode: ALT1 mux port: LPUART11_RTS_B of instance: LPUART11 0010b - Select mux mode: ALT2 mux port: LPI2C6_SCL of instance: LPI2C6 0011b - Select mux mode: ALT3 mux port: MIC_BITSTREAM2 of instance: MIC 0100b - Select mux mode: ALT4 mux port: LPSPI6_SOUT of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO11 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPI2C5_SDAS of instance: LPI2C5 0111b - Select mux mode: ALT7 mux port: ARM_TRACE_SWO of instance: ARM

Field	Function
	1000b - Select mux mode: ALT8 mux port: LPUART12_RXD of instance: LPUART12 1010b - Select mux mode: ALT10 mux port: GPIO12_IO11 of instance: GPIO12

12.4.4.14 SW_MUX_CTL_PAD_GPIO_LPSR_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_12)

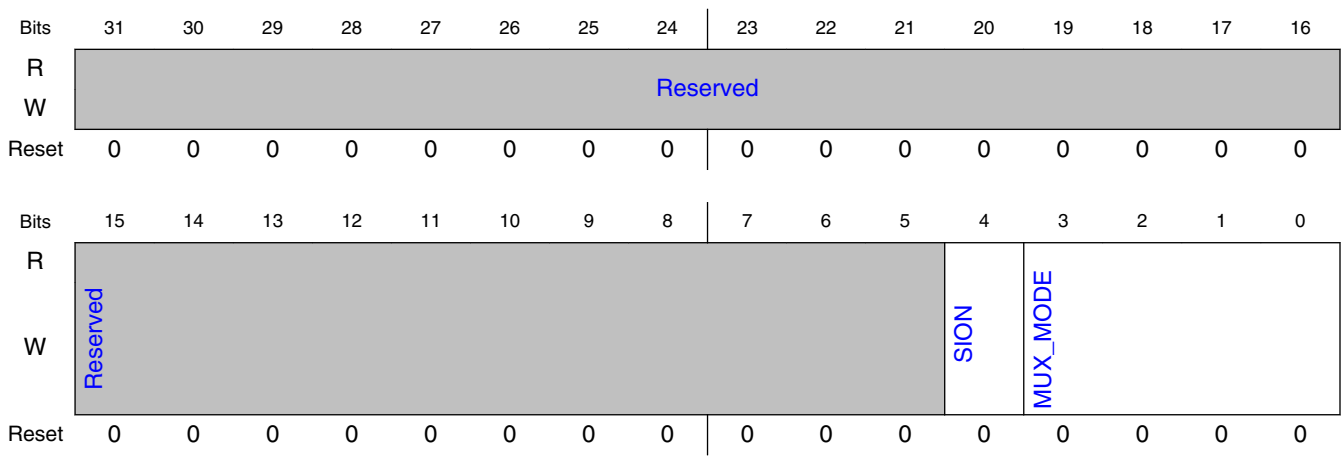
12.4.4.14.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_12	30h

12.4.4.14.2 Function

SW_MUX_CTL Register

12.4.4.14.3 Diagram



12.4.4.14.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_12
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_LPSR_12. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_TDI of instance: JTAG_MUX 0001b - Select mux mode: ALT1 mux port: PIT2_TRIGGER0 of instance: PIT2 0011b - Select mux mode: ALT3 mux port: MIC_BITSTREAM3 of instance: MIC 0100b - Select mux mode: ALT4 mux port: LPSPI6_SIN of instance: LPSPI6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO12 of instance: GPIO_MUX6 0110b - Select mux mode: ALT6 mux port: LPI2C5_HREQ of instance: LPI2C5 0111b - Select mux mode: ALT7 mux port: SAI4_TX_BCLK of instance: SAI4 1000b - Select mux mode: ALT8 mux port: LPSPI5_SCK of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO12 of instance: GPIO12

12.4.4.15 SW_MUX_CTL_PAD_GPIO_LPSR_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_13)

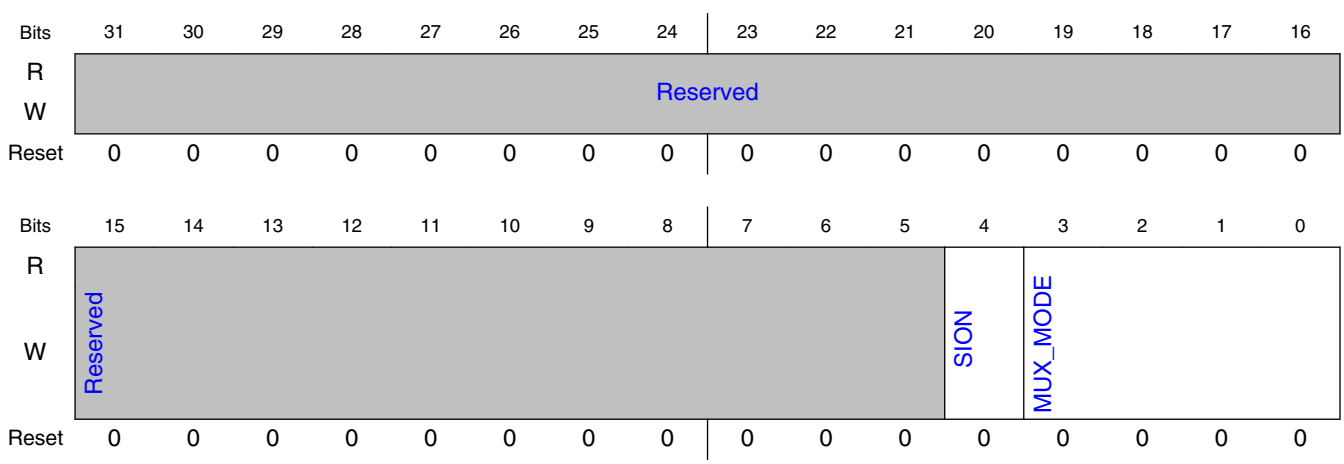
12.4.4.15.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_13	34h

12.4.4.15.2 Function

SW_MUX_CTL Register

12.4.4.15.3 Diagram



12.4.4.15.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_13
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_LPSR_13. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_MOD of instance: JTAG_MUX 0001b - Select mux mode: ALT1 mux port: MIC_BITSTREAM1 of instance: MIC 0010b - Select mux mode: ALT2 mux port: PIT2_TRIGGER1 of instance: PIT2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO13 of instance: GPIO_MUX6 0111b - Select mux mode: ALT7 mux port: SAI4_RX_DATA of instance: SAI4 1000b - Select mux mode: ALT8 mux port: LPSPI5_PCS0 of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO13 of instance: GPIO12

12.4.4.16 SW_MUX_CTL_PAD_GPIO_LPSR_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_14)

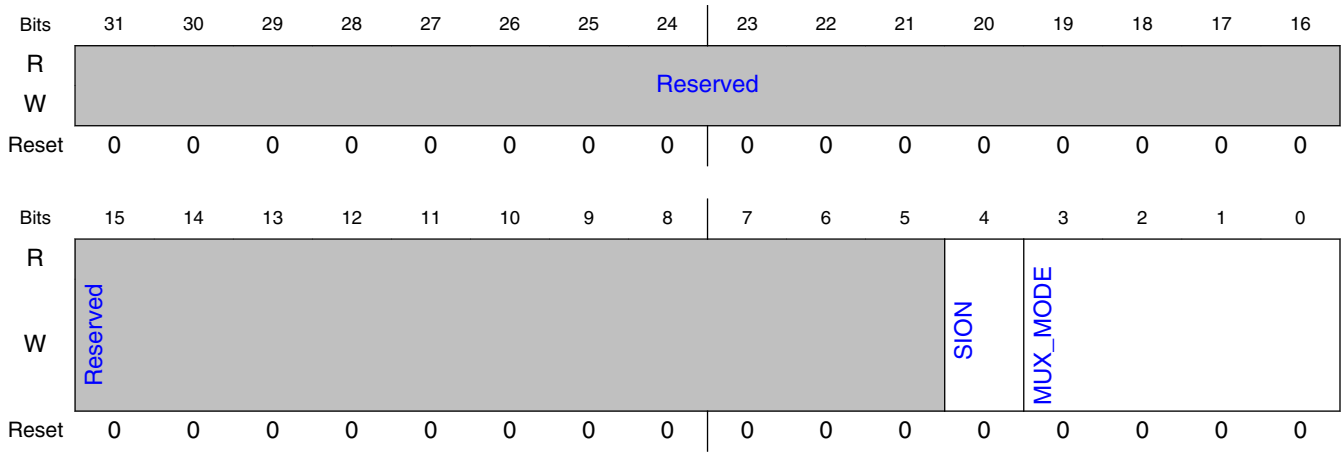
12.4.4.16.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_14	38h

12.4.4.16.2 Function

SW_MUX_CTL Register

12.4.4.16.3 Diagram



12.4.4.16.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_14
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_LPSR_14. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_TCK of instance: JTAG_MUX/SWD_CLK 0001b - Select mux mode: ALT1 mux port: MIC_BITSTREAM2 of instance: MIC 0010b - Select mux mode: ALT2 mux port: PIT2_TRIGGER2 of instance: PIT2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO14 of instance: GPIO_MUX6 0111b - Select mux mode: ALT7 mux port: SAI4_RX_BCLK of instance: SAI4 1000b - Select mux mode: ALT8 mux port: LPSPI5_SOUT of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO14 of instance: GPIO12

12.4.4.17 SW_MUX_CTL_PAD_GPIO_LPSR_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_LPSR_15)

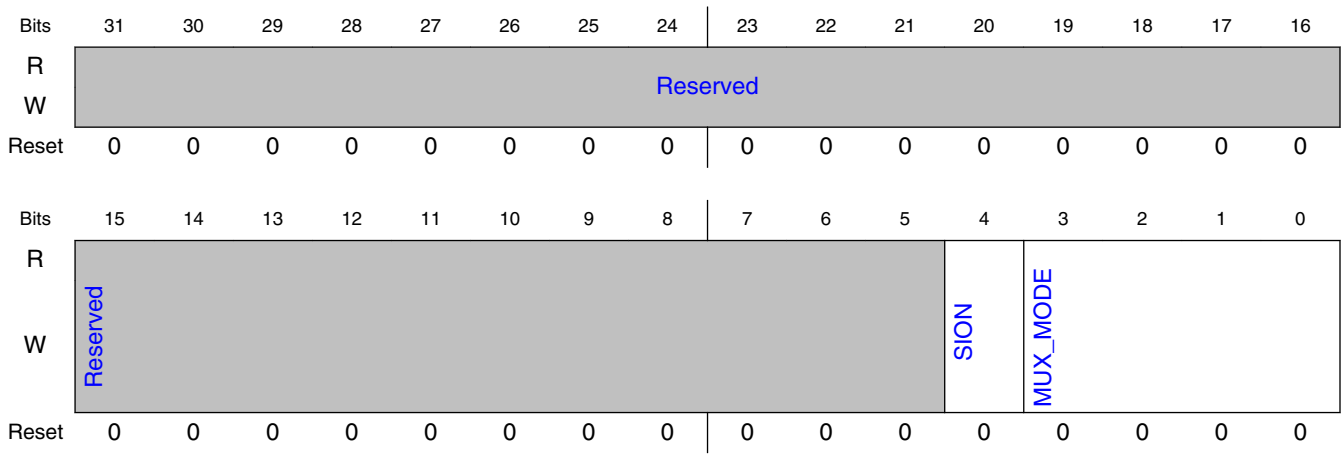
12.4.4.17.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_LPSR_15	3Ch

12.4.4.17.2 Function

SW_MUX_CTL Register

12.4.4.17.3 Diagram



12.4.4.17.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_LPSR_15
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_LPSR_15. 0000b - Select mux mode: ALT0 mux port: JTAG_MUX_TMS of instance: JTAG_MUX/SWD_DIO 0001b - Select mux mode: ALT1 mux port: MIC_BITSTREAM3 of instance: MIC 0010b - Select mux mode: ALT2 mux port: PIT2_TRIGGER3 of instance: PIT2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX6_IO15 of instance: GPIO_MUX6 0111b - Select mux mode: ALT7 mux port: SAI4_RX_SYNC of instance: SAI4 1000b - Select mux mode: ALT8 mux port: LPSPI5_SIN of instance: LPSPI5 1010b - Select mux mode: ALT10 mux port: GPIO12_IO15 of instance: GPIO12

12.4.4.18 SW_PAD_CTL_PAD_GPIO_LPSR_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_00)

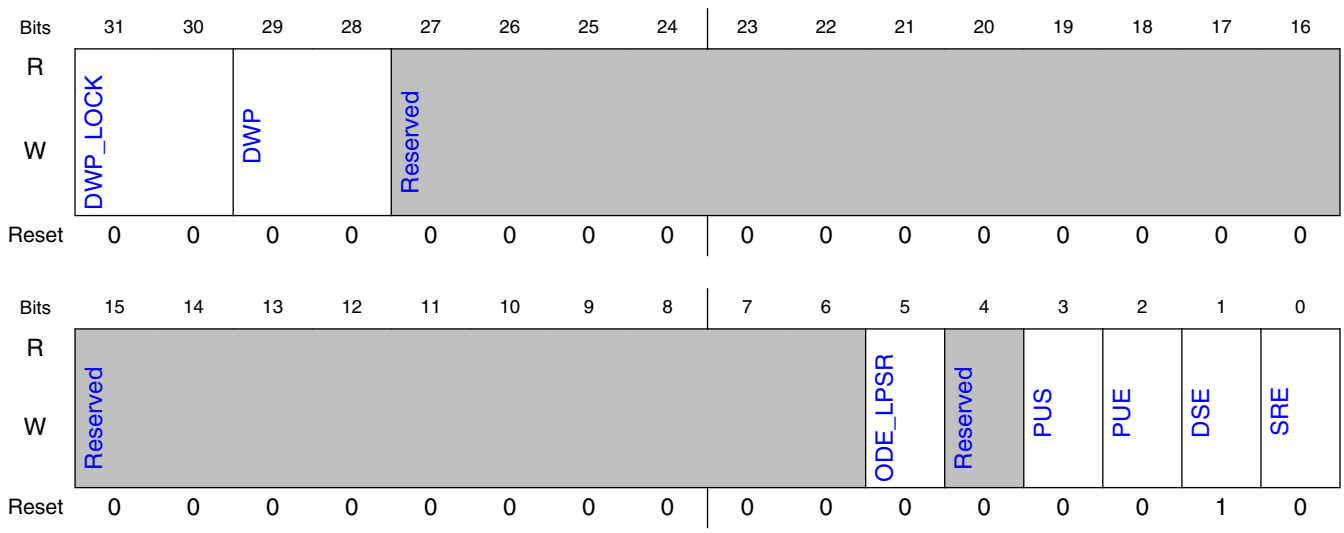
12.4.4.18.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_00	40h

12.4.4.18.2 Function

SW_PAD_CTL Register

12.4.4.18.3 Diagram



12.4.4.18.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6	-

Table continues on the next page...

Field	Function
—	Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_00 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_00 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_00 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_00 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_00 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.19 SW_PAD_CTL_PAD_GPIO_LPSR_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_01)

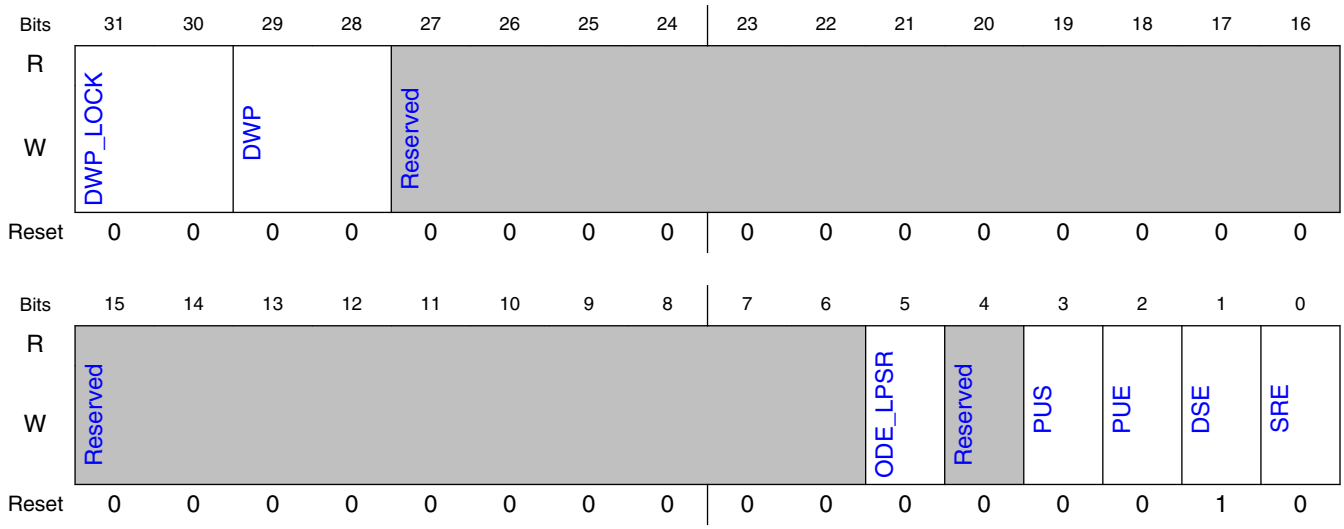
12.4.4.19.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_01	44h

12.4.4.19.2 Function

SW_PAD_CTL Register

12.4.4.19.3 Diagram



12.4.4.19.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_01 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_01 0b - Weak pull down 1b - Weak pull up
2	Pull Select Field

Table continues on the next page...

Field	Function
PUE	Select one out of next values for pad: GPIO_LPSR_01 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_01 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_01 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.20 SW_PAD_CTL_PAD_GPIO_LPSR_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_02)

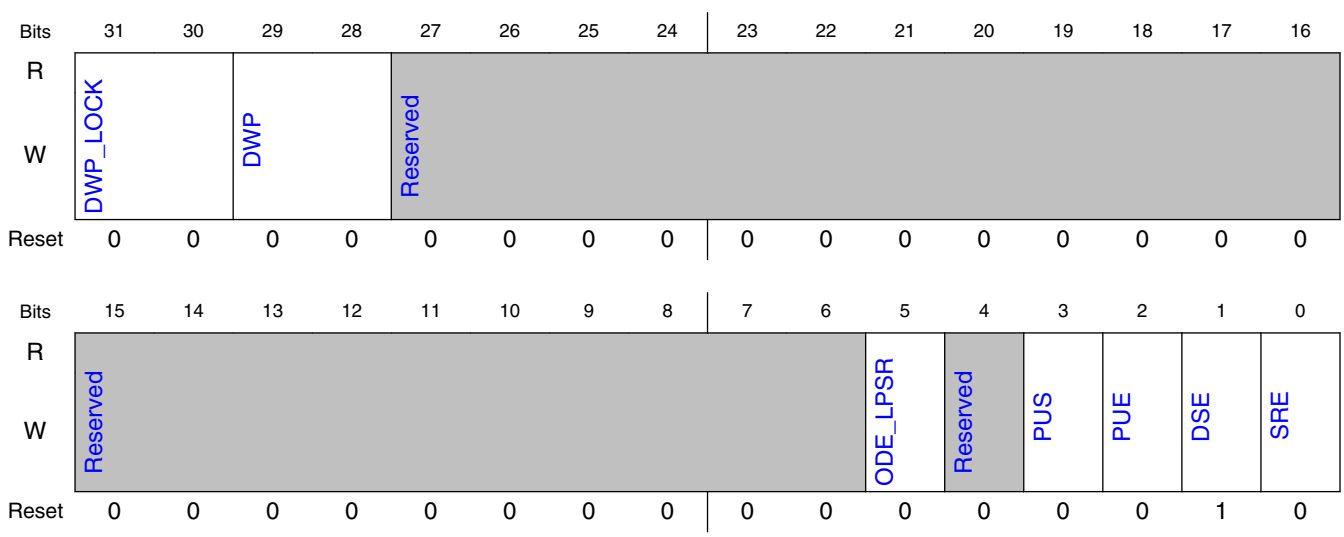
12.4.4.20.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_02	48h

12.4.4.20.2 Function

SW_PAD_CTL Register

12.4.4.20.3 Diagram



12.4.4.20.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_02 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_02 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_02 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_02 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_02 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.21 SW_PAD_CTL_PAD_GPIO_LPSR_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_03)

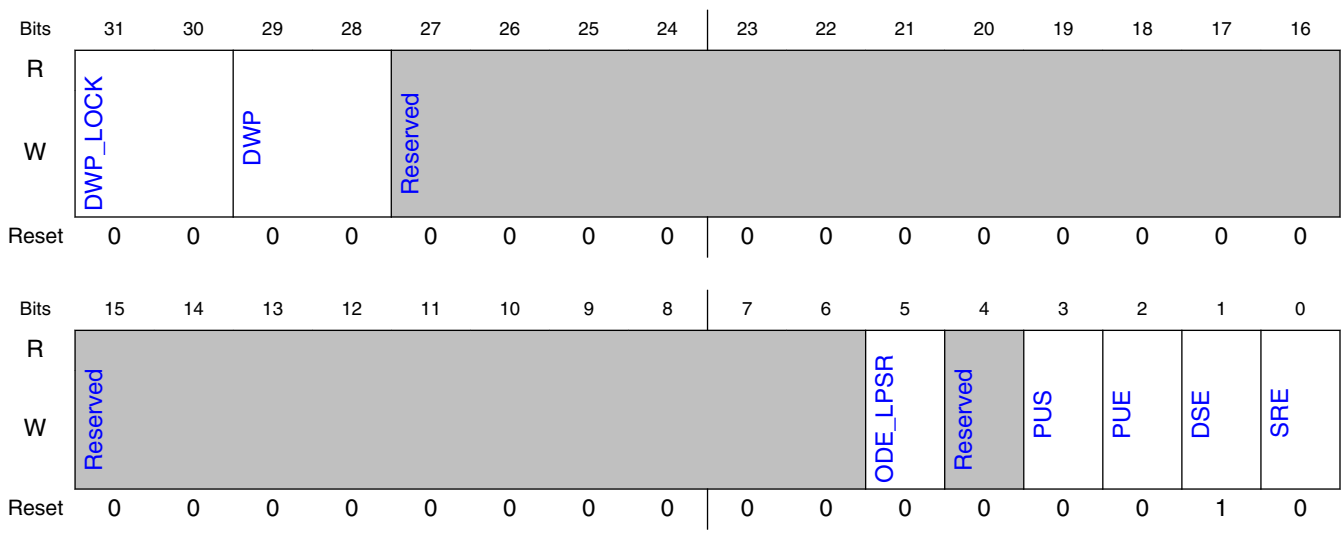
12.4.4.21.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_03	4Ch

12.4.4.21.2 Function

SW_PAD_CTL Register

12.4.4.21.3 Diagram



12.4.4.21.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden.

Table continues on the next page...

Field	Function
	00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_03 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_03 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_03 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_03 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_03 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.22 SW_PAD_CTL_PAD_GPIO_LPSR_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_04)

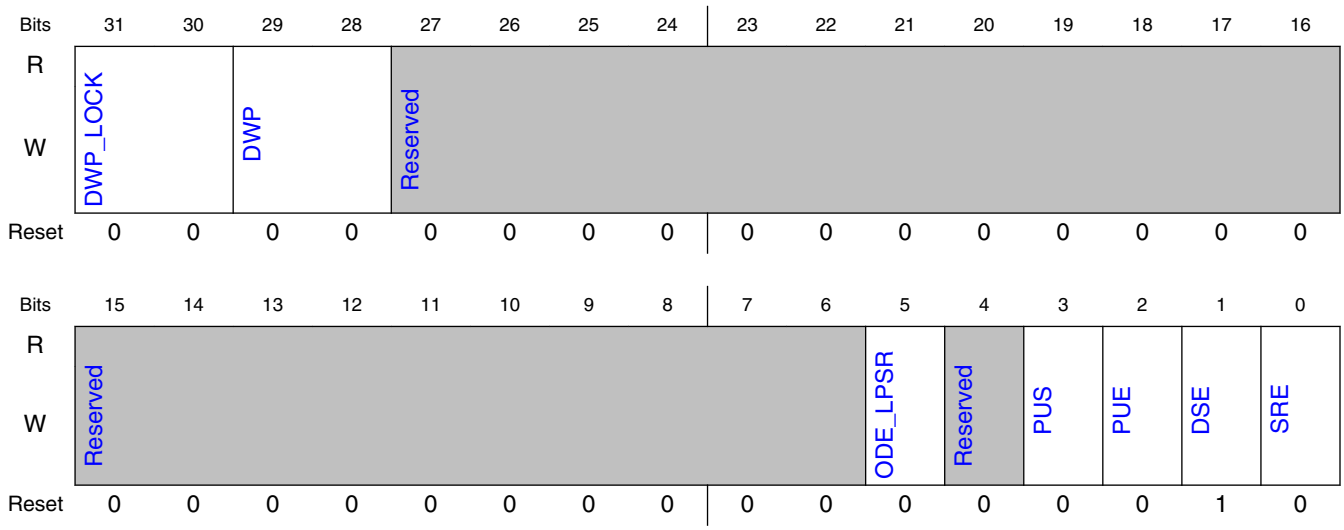
12.4.4.22.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_04	50h

12.4.4.22.2 Function

SW_PAD_CTL Register

12.4.4.22.3 Diagram



12.4.4.22.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_04 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_04 0b - Weak pull down

Table continues on the next page...

Field	Function
	1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_04 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_04 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_04 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.23 SW_PAD_CTL_PAD_GPIO_LPSR_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_05)

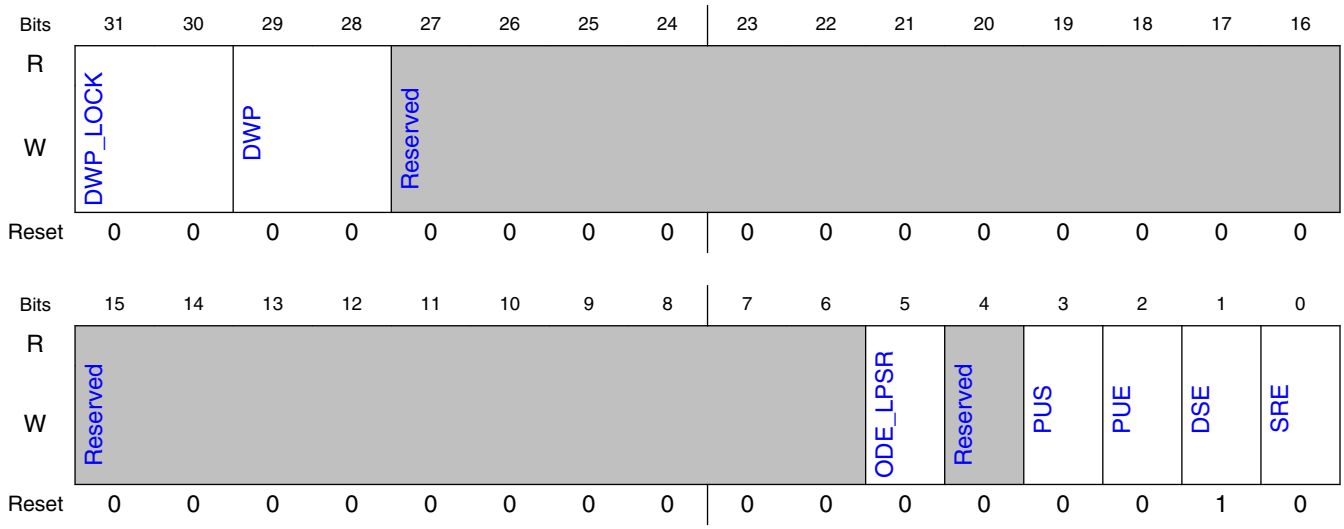
12.4.4.23.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_05	54h

12.4.4.23.2 Function

SW_PAD_CTL Register

12.4.4.23.3 Diagram



12.4.4.23.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_05 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_05 0b - Weak pull down 1b - Weak pull up
2	Pull Select Field

Table continues on the next page...

Field	Function
PUE	Select one out of next values for pad: GPIO_LPSR_05 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_05 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_05 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.24 SW_PAD_CTL_PAD_GPIO_LPSR_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_06)

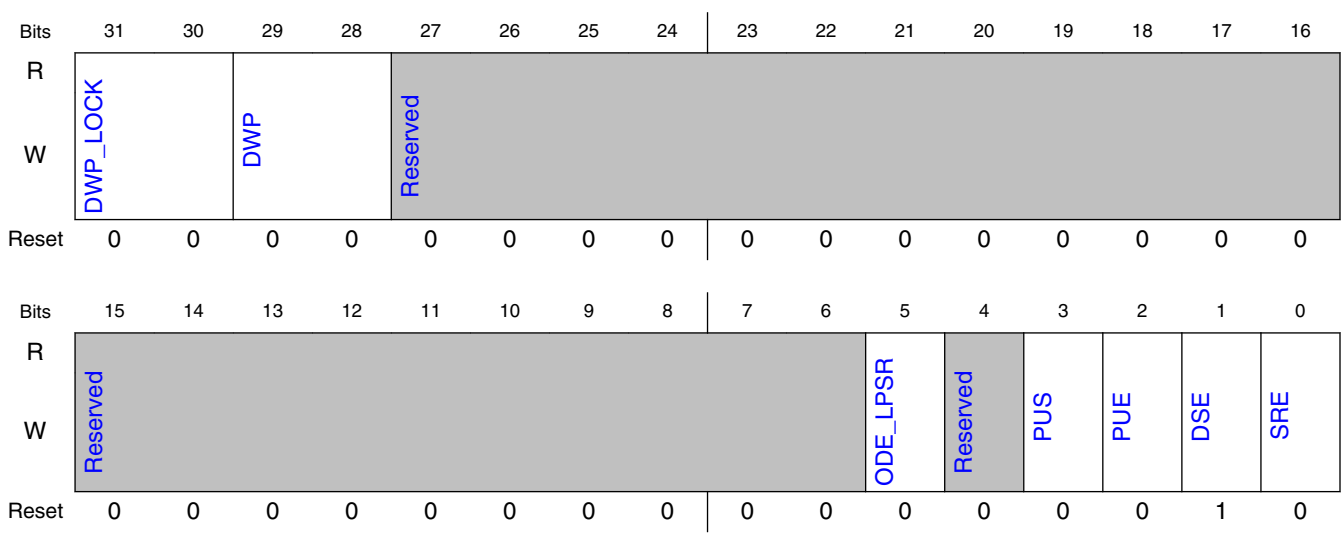
12.4.4.24.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_06	58h

12.4.4.24.2 Function

SW_PAD_CTL Register

12.4.4.24.3 Diagram



12.4.4.24.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_06 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_06 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_06 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_06 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_06 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.25 SW_PAD_CTL_PAD_GPIO_LPSR_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_07)

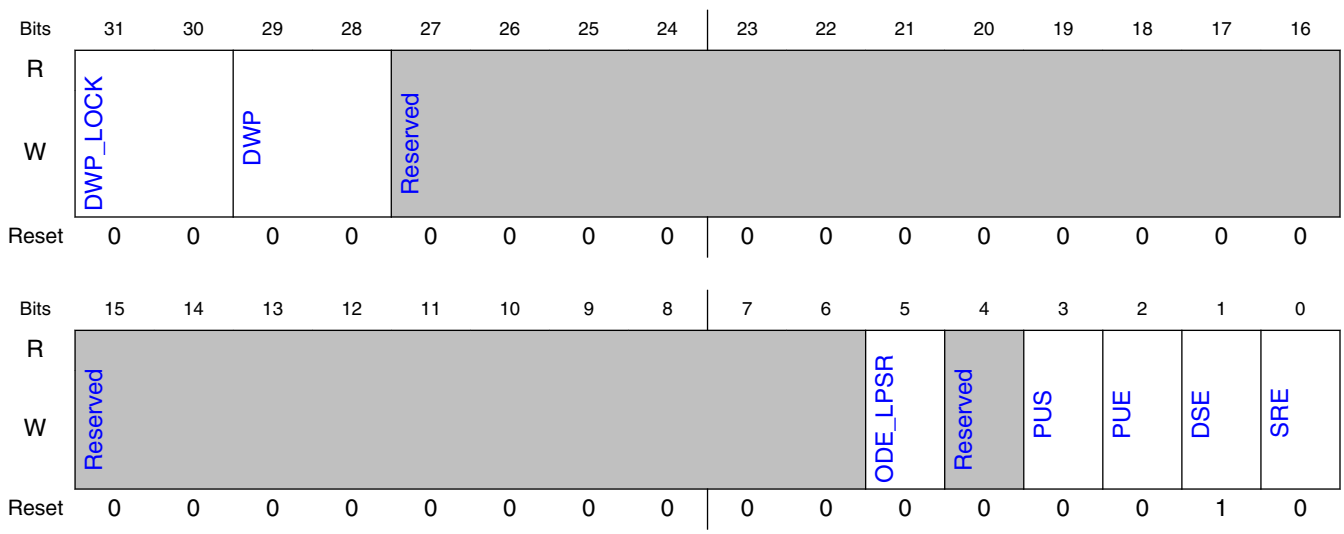
12.4.4.25.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_07	5Ch

12.4.4.25.2 Function

SW_PAD_CTL Register

12.4.4.25.3 Diagram



12.4.4.25.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden.

Table continues on the next page...

Field	Function
	00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_07 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_07 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_07 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_07 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_07 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.26 SW_PAD_CTL_PAD_GPIO_LPSR_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_08)

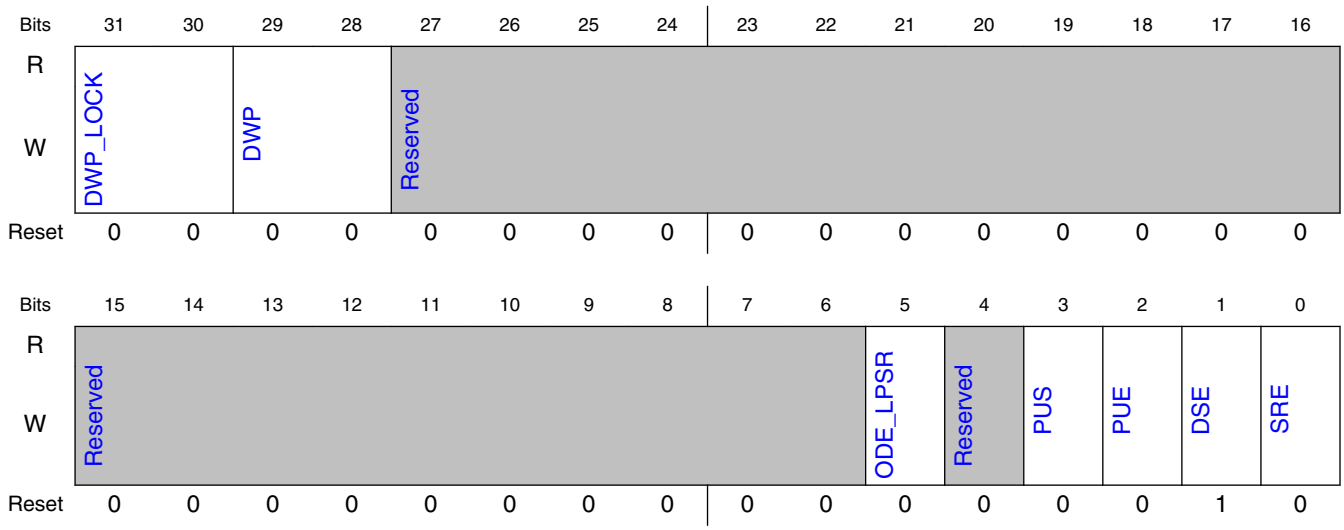
12.4.4.26.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_08	60h

12.4.4.26.2 Function

SW_PAD_CTL Register

12.4.4.26.3 Diagram



12.4.4.26.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_08 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_08 0b - Weak pull down

Table continues on the next page...

Field	Function
	1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_08 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_08 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_08 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.27 SW_PAD_CTL_PAD_GPIO_LPSR_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_09)

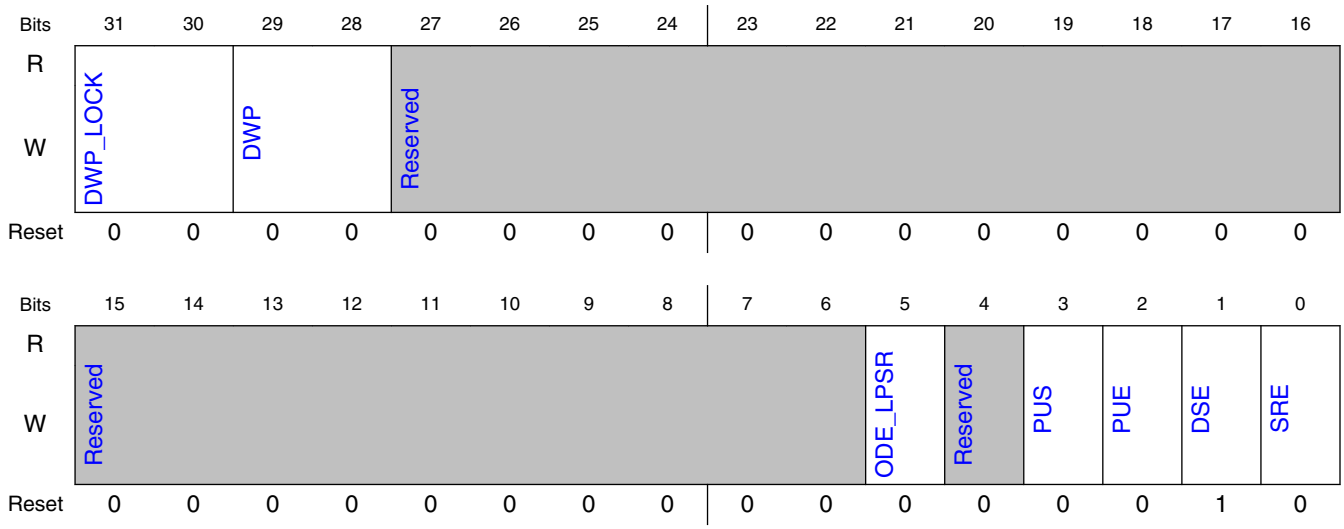
12.4.4.27.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_09	64h

12.4.4.27.2 Function

SW_PAD_CTL Register

12.4.4.27.3 Diagram



12.4.4.27.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_09 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_09 0b - Weak pull down 1b - Weak pull up
2	Pull Select Field

Table continues on the next page...

Field	Function
PUE	Select one out of next values for pad: GPIO_LPSR_09 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_09 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_09 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.28 SW_PAD_CTL_PAD_GPIO_LPSR_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_10)

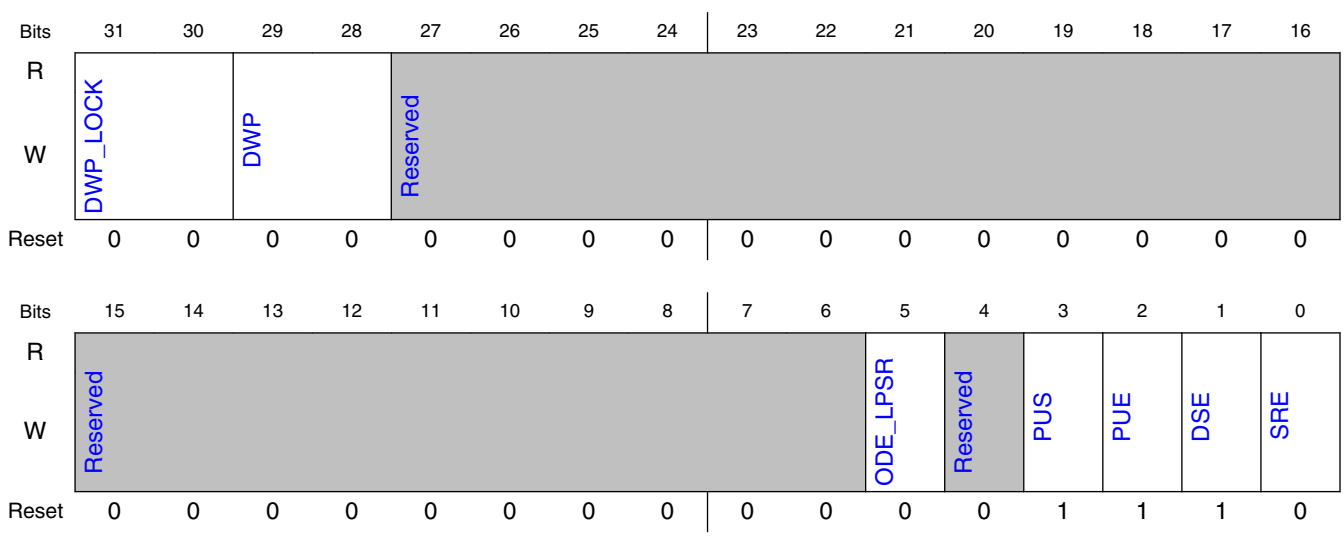
12.4.4.28.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_10	68h

12.4.4.28.2 Function

SW_PAD_CTL Register

12.4.4.28.3 Diagram



12.4.4.28.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_10 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_10 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_10 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_10 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_10 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.29 SW_PAD_CTL_PAD_GPIO_LPSR_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_11)

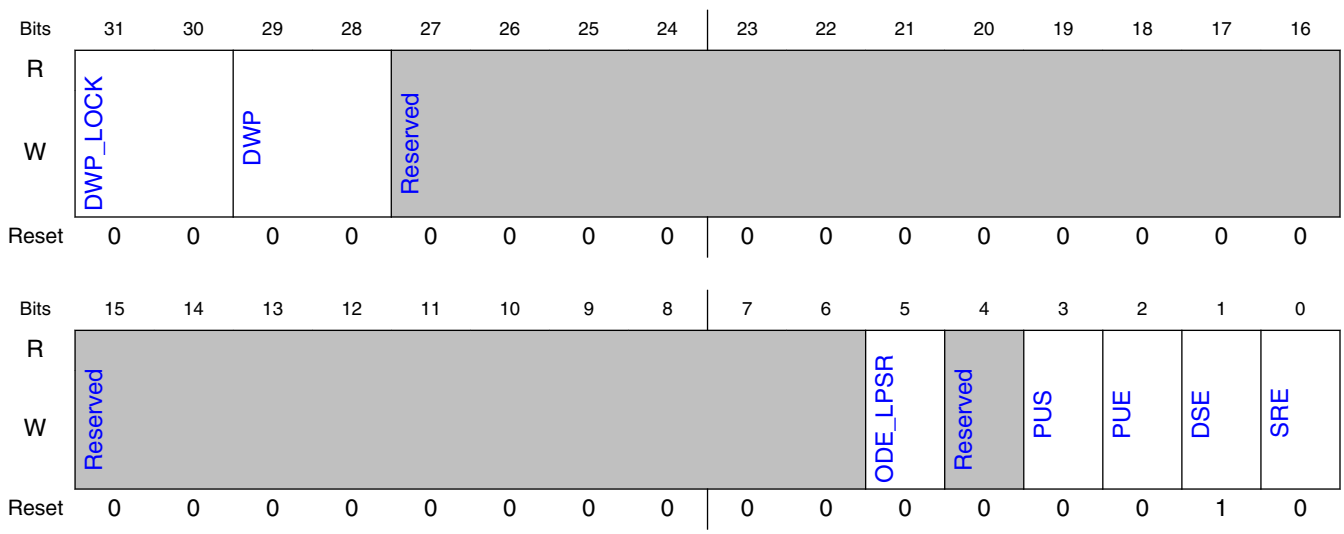
12.4.4.29.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_11	6Ch

12.4.4.29.2 Function

SW_PAD_CTL Register

12.4.4.29.3 Diagram



12.4.4.29.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden.

Table continues on the next page...

Field	Function
	00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_11 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_11 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_11 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_11 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_11 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.30 SW_PAD_CTL_PAD_GPIO_LPSR_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_12)

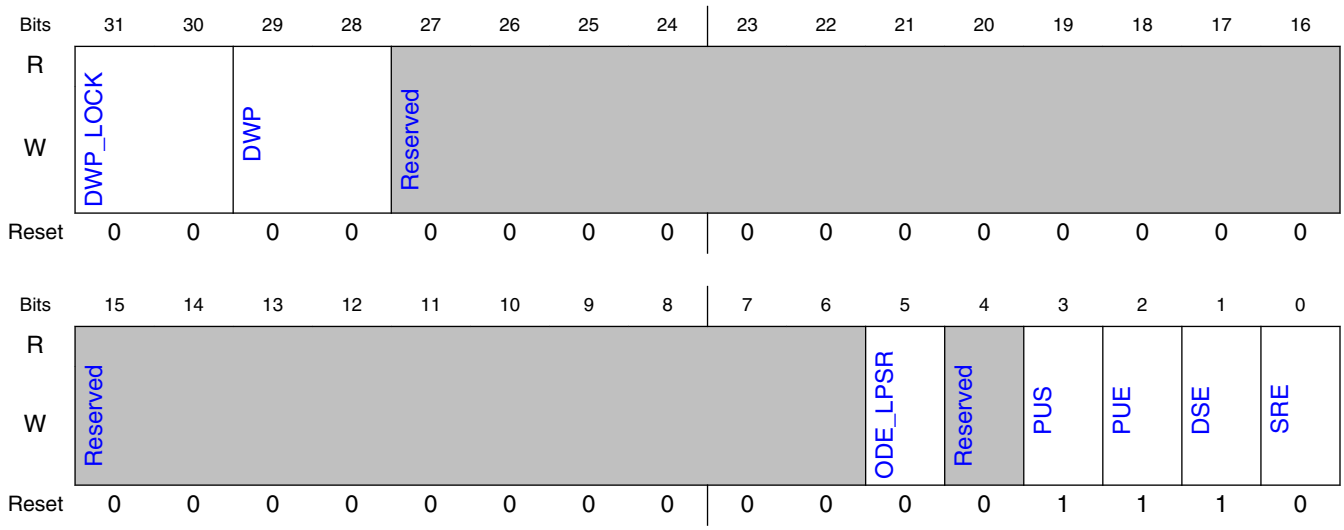
12.4.4.30.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_12	70h

12.4.4.30.2 Function

SW_PAD_CTL Register

12.4.4.30.3 Diagram



12.4.4.30.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_12 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_12 0b - Weak pull down

Table continues on the next page...

Field	Function
	1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_12 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_12 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_12 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.31 SW_PAD_CTL_PAD_GPIO_LPSR_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_13)

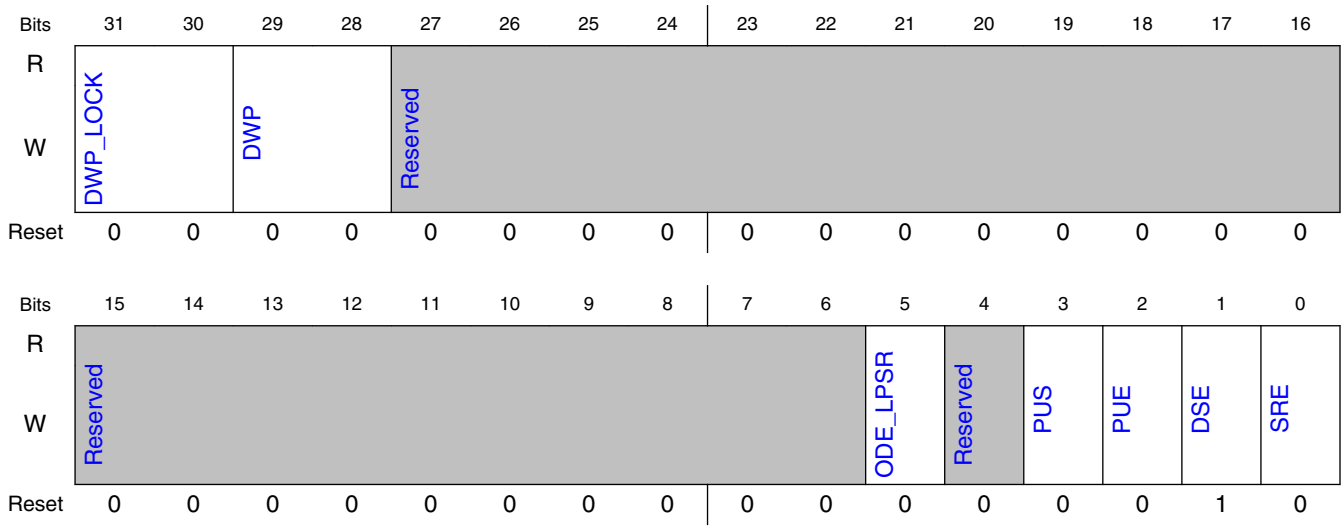
12.4.4.31.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_13	74h

12.4.4.31.2 Function

SW_PAD_CTL Register

12.4.4.31.3 Diagram



12.4.4.31.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_13 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_13 0b - Weak pull down 1b - Weak pull up
2	Pull Select Field

Table continues on the next page...

Field	Function
PUE	Select one out of next values for pad: GPIO_LPSR_13 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_13 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_13 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.32 SW_PAD_CTL_PAD_GPIO_LPSR_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_14)

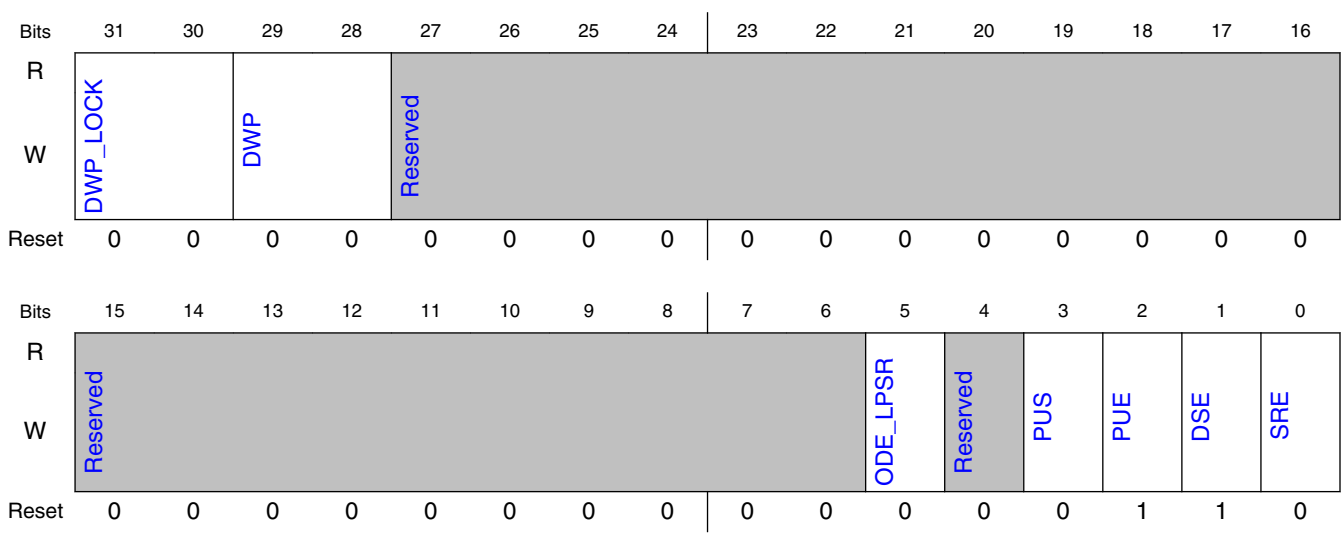
12.4.4.32.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_14	78h

12.4.4.32.2 Function

SW_PAD_CTL Register

12.4.4.32.3 Diagram



12.4.4.32.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_14 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_14 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_14 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_14 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_14 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.33 SW_PAD_CTL_PAD_GPIO_LPSR_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_LPSR_15)

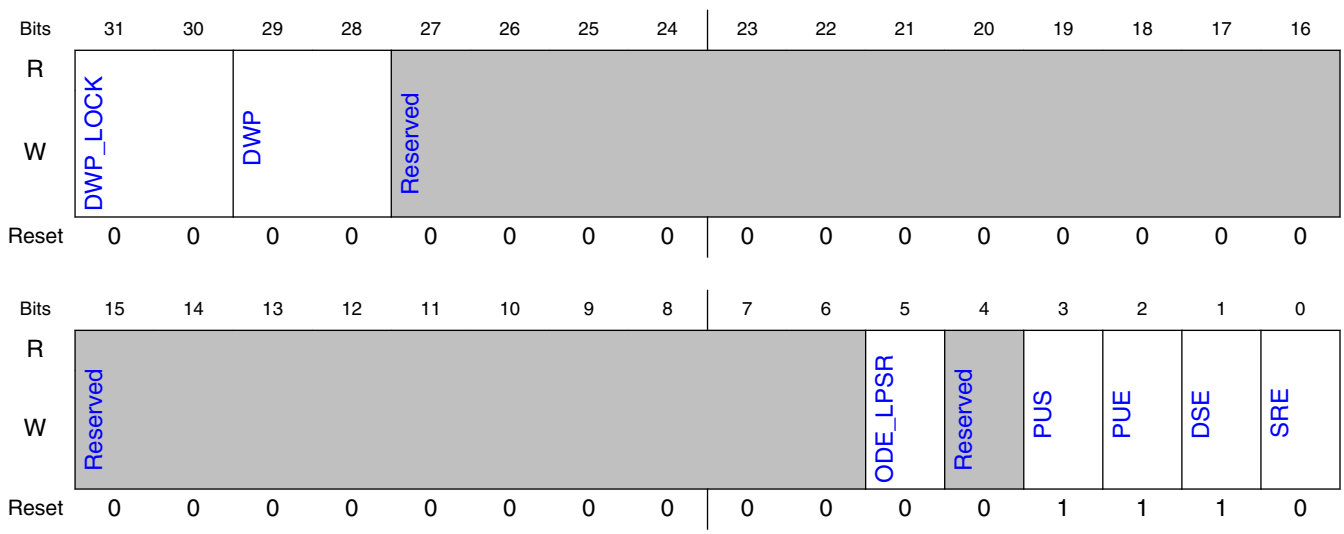
12.4.4.33.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_LPSR_15	7Ch

12.4.4.33.2 Function

SW_PAD_CTL Register

12.4.4.33.3 Diagram



12.4.4.33.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden.

Table continues on the next page...

Field	Function
	00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-6 —	- Reserved
5 ODE_LPSR	Open Drain LPSR Field Select one out of next values for pad: GPIO_LPSR_15 0b - Disabled 1b - Enabled
4 —	Reserved
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_LPSR_15 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull Select Field Select one out of next values for pad: GPIO_LPSR_15 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_LPSR_15 0b - normal driver 1b - high driver
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_LPSR_15 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.4.34 CAN3_IPP_IND_CANRX_SELECT_INPUT DAISY Register (CAN3_IPP_IND_CANRX_SELECT_INPUT)

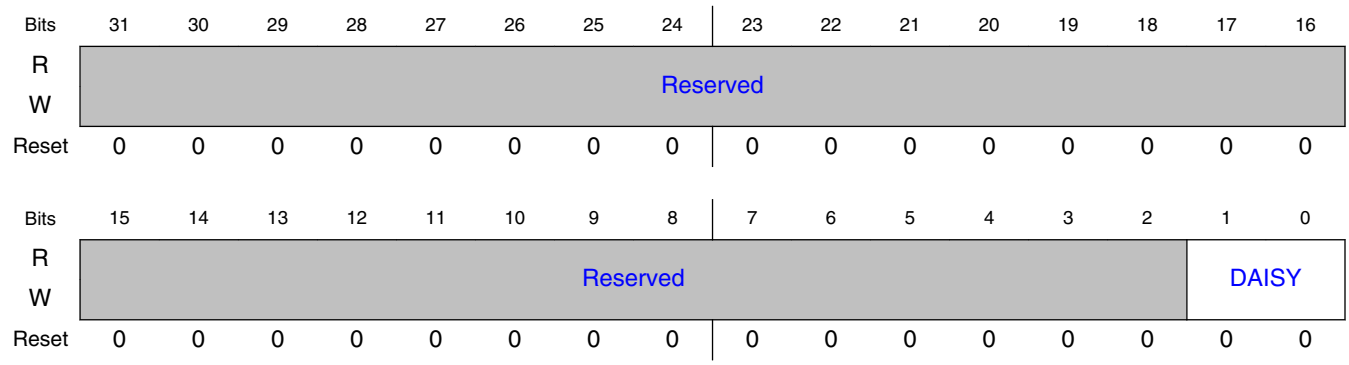
12.4.4.34.1 Offset

Register	Offset
CAN3_IPP_IND_CANRX_SELECT_INPUT	80h

12.4.4.34.2 Function

DAISY Register

12.4.4.34.3 Diagram



12.4.4.34.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: can3, In Pin: ipp_ind_canrx 00b - Selecting Pad: GPIO_LPSR_01 for Mode: ALT0 01b - Selecting Pad: GPIO_LPSR_07 for Mode: ALT6 10b - Selecting Pad: GPIO_LPSR_09 for Mode: ALT1

12.4.4.35 LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT)

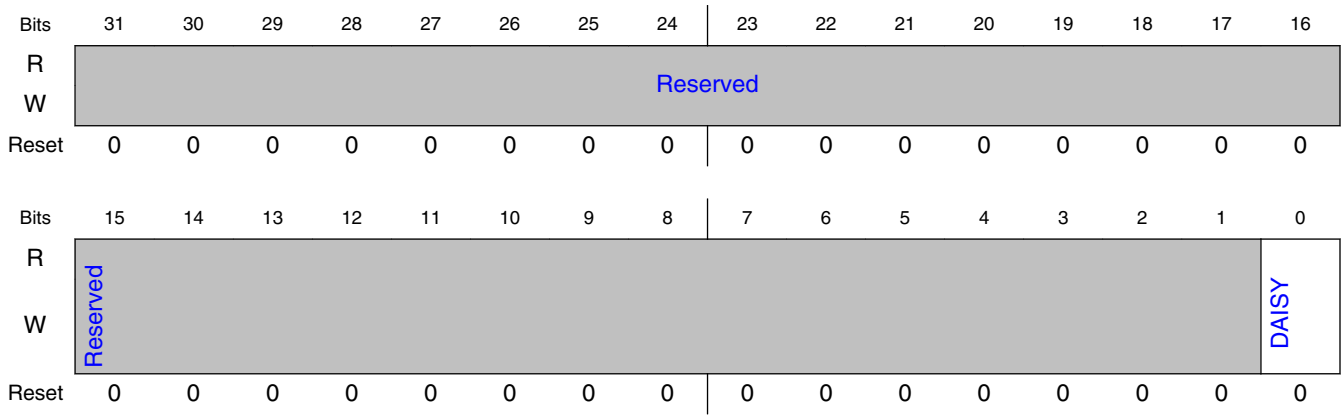
12.4.4.35.1 Offset

Register	Offset
LPI2C5_IPP_IND_LPI2C_SCL_SELECT_INPUT	84h

12.4.4.35.2 Function

DAISY Register

12.4.4.35.3 Diagram



12.4.4.35.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c5, In Pin: ipp_ind_lpi2c_scl 0b - Selecting Pad: GPIO_LPSR_05 for Mode: ALT0 1b - Selecting Pad: GPIO_LPSR_09 for Mode: ALT6

12.4.4.36 LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT)

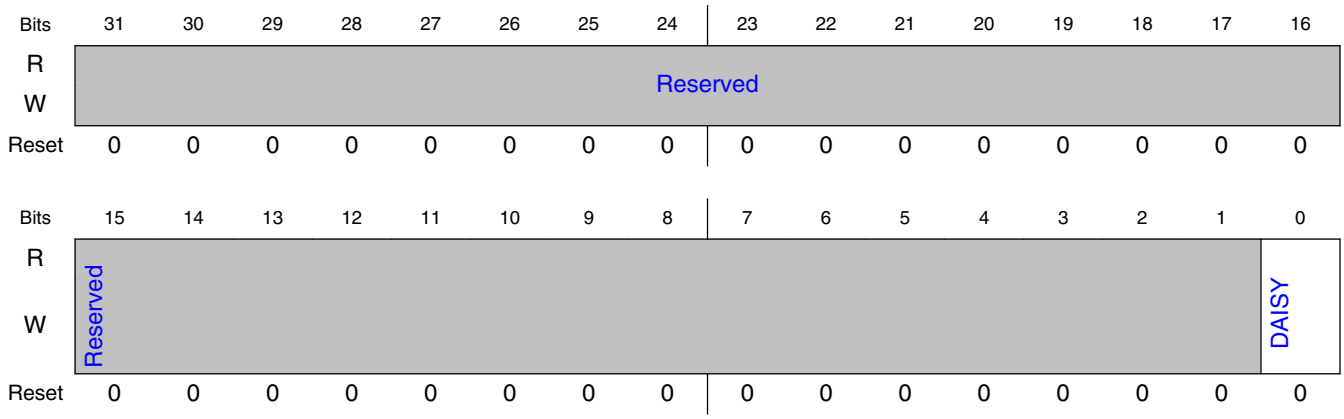
12.4.4.36.1 Offset

Register	Offset
LPI2C5_IPP_IND_LPI2C_SDA_SELECT_INPUT	88h

12.4.4.36.2 Function

DAISY Register

12.4.4.36.3 Diagram



12.4.4.36.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c5, In Pin: ipp_ind_lpi2c_sda 0b - Selecting Pad: GPIO_LPSR_04 for Mode: ALT0 1b - Selecting Pad: GPIO_LPSR_08 for Mode: ALT6

12.4.4.37 LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT)

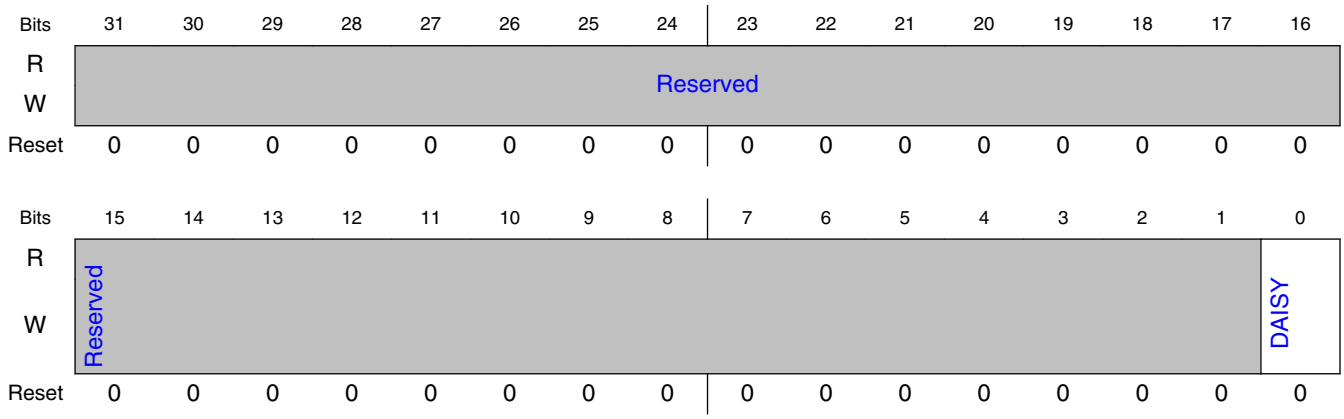
12.4.4.37.1 Offset

Register	Offset
LPI2C6_IPP_IND_LPI2C_SCL_SELECT_INPUT	8Ch

12.4.4.37.2 Function

DAISY Register

12.4.4.37.3 Diagram



12.4.4.37.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c6, In Pin: ipp_ind_lpi2c_scl 0b - Selecting Pad: GPIO_LPSR_07 for Mode: ALT0 1b - Selecting Pad: GPIO_LPSR_11 for Mode: ALT2

12.4.4.38 LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT)

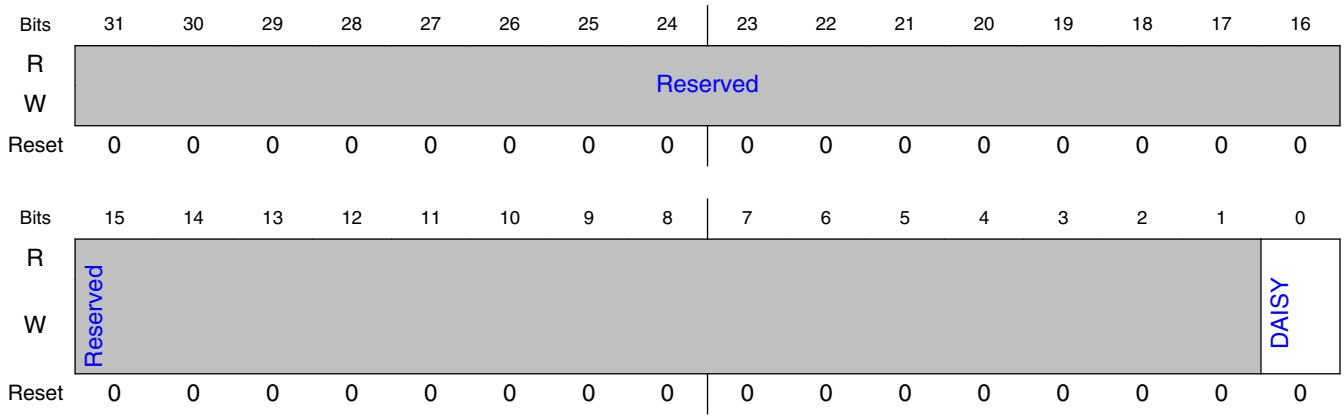
12.4.4.38.1 Offset

Register	Offset
LPI2C6_IPP_IND_LPI2C_SDA_SELECT_INPUT	90h

12.4.4.38.2 Function

DAISY Register

12.4.4.38.3 Diagram



12.4.4.38.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c6, In Pin: ipp_ind_lpi2c_sda 0b - Selecting Pad: GPIO_LPSR_06 for Mode: ALT0 1b - Selecting Pad: GPIO_LPSR_10 for Mode: ALT2

12.4.4.39 LPSPI5_IPP_IND_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI5_IPP_IND_LPSPI_PCS_SELECT_INPUT_0)

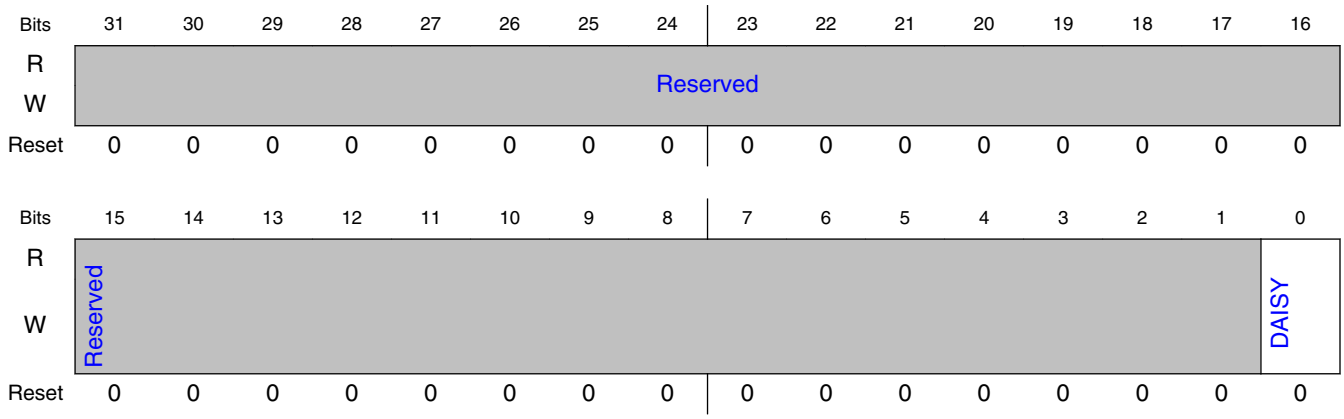
12.4.4.39.1 Offset

Register	Offset
LPSPI5_IPP_IND_LPSPI_PCS_SELECT_INPUT_0	94h

12.4.4.39.2 Function

DAISY Register

12.4.4.39.3 Diagram



12.4.4.39.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi5, In Pin: ipp_ind_lpspi_pcs[0] 0b - Selecting Pad: GPIO_LPSR_03 for Mode: ALT1 1b - Selecting Pad: GPIO_LPSR_13 for Mode: ALT8

12.4.4.40 LPSPI5_IPP_IND_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SCK_SELECT_INPUT)

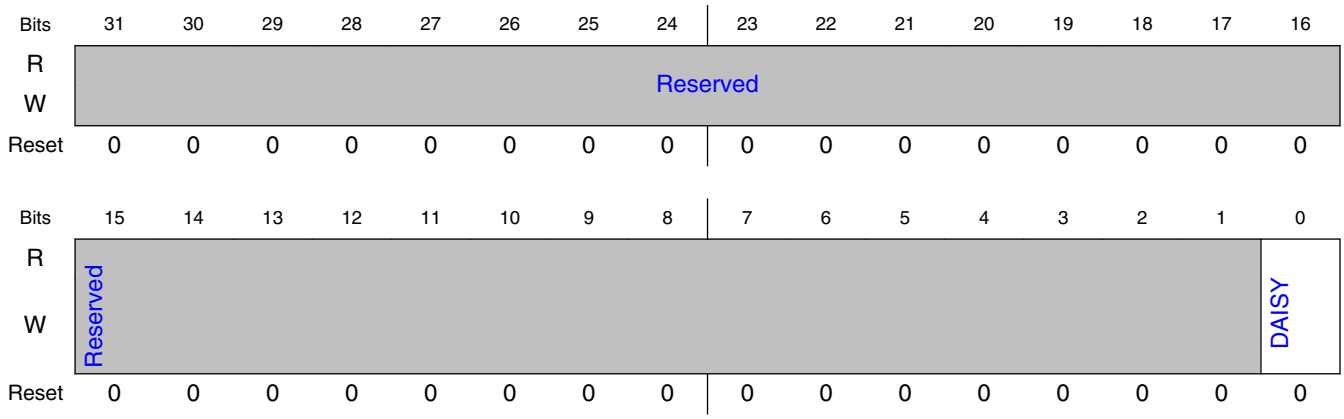
12.4.4.40.1 Offset

Register	Offset
LPSPI5_IPP_IND_LPSPI_SCK_SELECT_INPUT	98h

12.4.4.40.2 Function

DAISY Register

12.4.4.40.3 Diagram



12.4.4.40.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi5, In Pin: ipp_ind_lpspi_sck 0b - Selecting Pad: GPIO_LPSR_02 for Mode: ALT1 1b - Selecting Pad: GPIO_LPSR_12 for Mode: ALT8

12.4.4.41 LPSPI5_IPP_IND_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SDI_SELECT_INPUT)

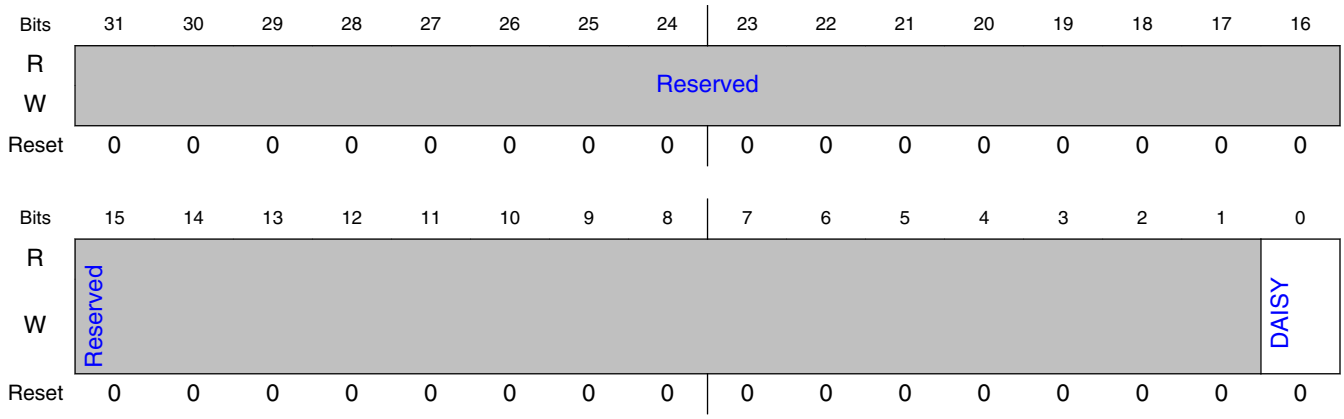
12.4.4.41.1 Offset

Register	Offset
LPSPI5_IPP_IND_LPSPI_SDI_SELECT_INPUT	9Ch

12.4.4.41.2 Function

DAISY Register

12.4.4.41.3 Diagram



12.4.4.41.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi5, In Pin: ipp_ind_lpspi_sdi 0b - Selecting Pad: GPIO_LPSR_05 for Mode: ALT1 1b - Selecting Pad: GPIO_LPSR_15 for Mode: ALT8

12.4.4.42 LPSPI5_IPP_IND_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI5_IPP_IND_LPSPI_SDO_SELECT_INPUT)

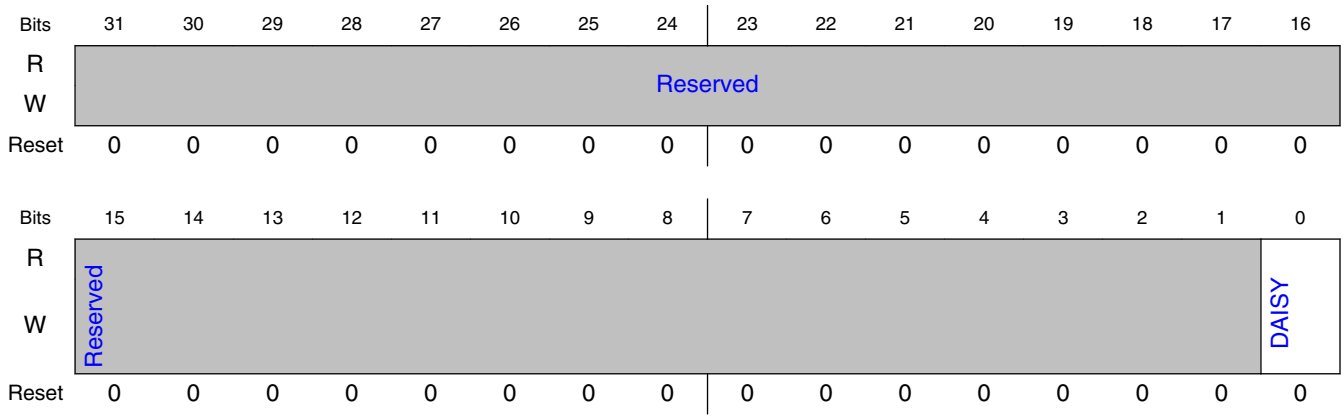
12.4.4.42.1 Offset

Register	Offset
LPSPI5_IPP_IND_LPSPI_SDO_SELECT_INPUT	A0h

12.4.4.42.2 Function

DAISY Register

12.4.4.42.3 Diagram



12.4.4.42.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi5, In Pin: ipp_ind_lpspi_sdo 0b - Selecting Pad: GPIO_LPSR_04 for Mode: ALT1 1b - Selecting Pad: GPIO_LPSR_14 for Mode: ALT8

12.4.4.43 LPUART11_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART11_IPP_IND_LPUART_RXD_SELECT_INPUT)

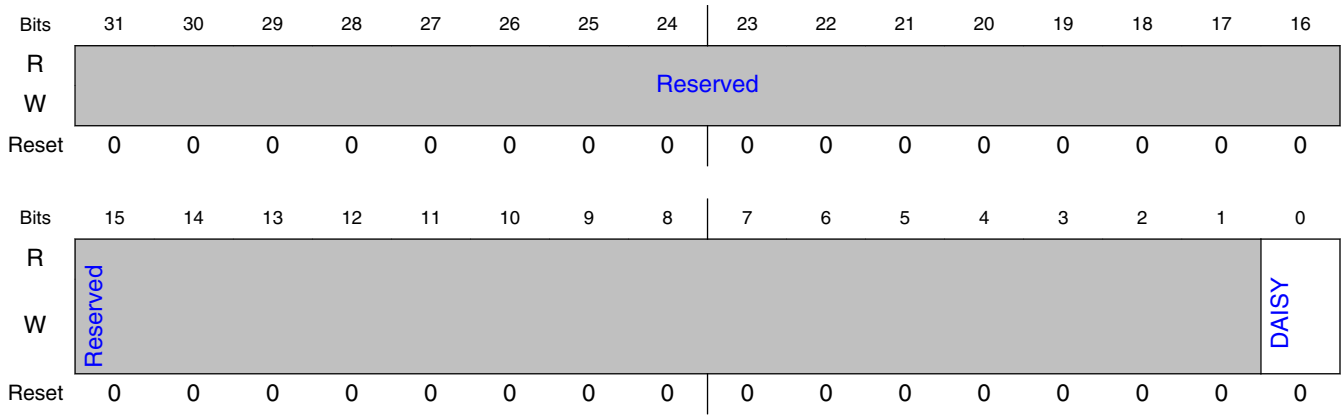
12.4.4.43.1 Offset

Register	Offset
LPUART11_IPP_IND_LPUART_RXD_SELECT_INPUT	A4h

12.4.4.43.2 Function

DAISY Register

12.4.4.43.3 Diagram



12.4.4.43.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart11, In Pin: ipp_ind_lpuart_rxd 0b - Selecting Pad: GPIO_LPSR_05 for Mode: ALT6 1b - Selecting Pad: GPIO_LPSR_09 for Mode: ALT0

12.4.4.44 LPUART11_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART11_IPP_IND_LPUART_TXD_SELECT_INPUT)

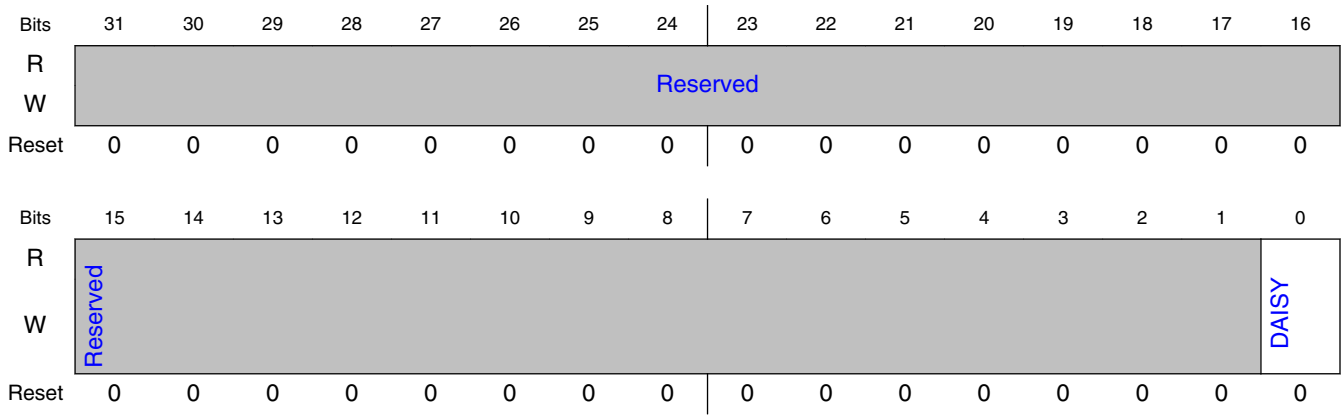
12.4.4.44.1 Offset

Register	Offset
LPUART11_IPP_IND_LPUART_TXD_SELECT_INPUT	A8h

12.4.4.44.2 Function

DAISY Register

12.4.4.44.3 Diagram



12.4.4.44.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart11, In Pin: ipp_ind_lpuart_txd 0b - Selecting Pad: GPIO_LPSR_04 for Mode: ALT6 1b - Selecting Pad: GPIO_LPSR_08 for Mode: ALT0

12.4.4.45 LPUART12_IPP_IND_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART12_IPP_IND_LPUART_RXD_SELECT_INPUT)

12.4.4.45.1 Offset

Register	Offset
LPUART12_IPP_IND_LPUART_RXD_SELECT_INPUT	ACh

12.4.4.45.2 Function

DAISY Register

12.4.4.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.4.45.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart12, In Pin: ipp_ind_lpuart_rxd 00b - Selecting Pad: GPIO_LPSR_01 for Mode: ALT6 01b - Selecting Pad: GPIO_LPSR_07 for Mode: ALT3 10b - Selecting Pad: GPIO_LPSR_11 for Mode: ALT8

12.4.4.46 LPUART12_IPP_IND_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART12_IPP_IND_LPUART_TXD_SELECT_INPUT)

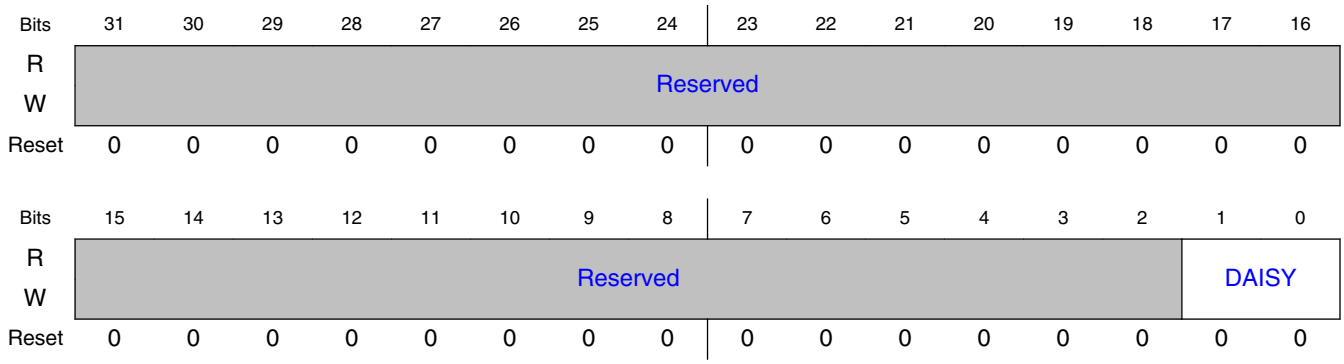
12.4.4.46.1 Offset

Register	Offset
LPUART12_IPP_IND_LPUART_TXD_SELECT_INPUT	B0h

12.4.4.46.2 Function

DAISY Register

12.4.4.46.3 Diagram



12.4.4.46.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart12, In Pin: ipp_ind_lpuart_txd 00b - Selecting Pad: GPIO_LPSR_00 for Mode: ALT6 01b - Selecting Pad: GPIO_LPSR_06 for Mode: ALT3 10b - Selecting Pad: GPIO_LPSR_10 for Mode: ALT8

12.4.4.47 MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0)

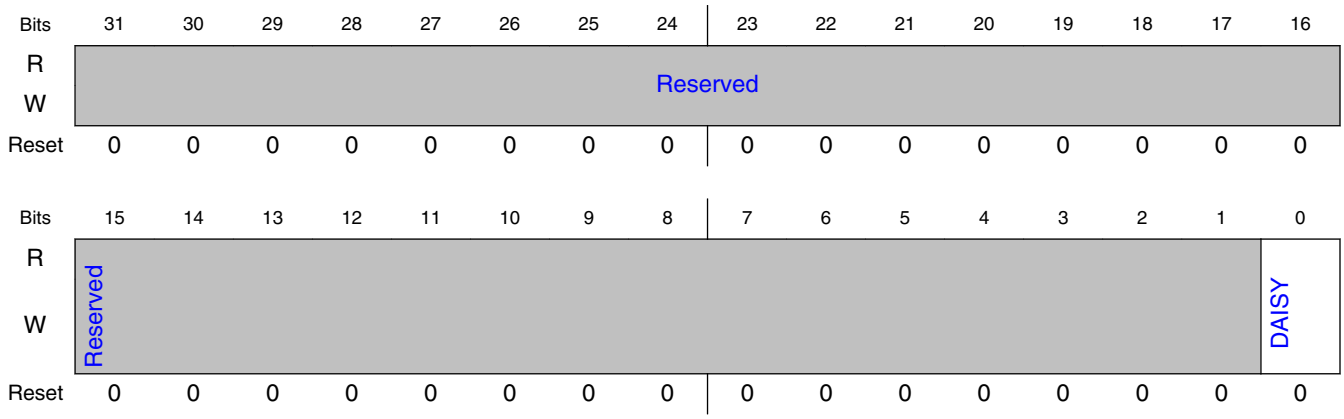
12.4.4.47.1 Offset

Register	Offset
MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_0	B4h

12.4.4.47.2 Function

DAISY Register

12.4.4.47.3 Diagram



12.4.4.47.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mic, In Pin: ipp_ind_mic_pdm_bitstream[0] 0b - Selecting Pad: GPIO_LPSR_01 for Mode: ALT1 1b - Selecting Pad: GPIO_LPSR_09 for Mode: ALT3

12.4.4.48 MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1)

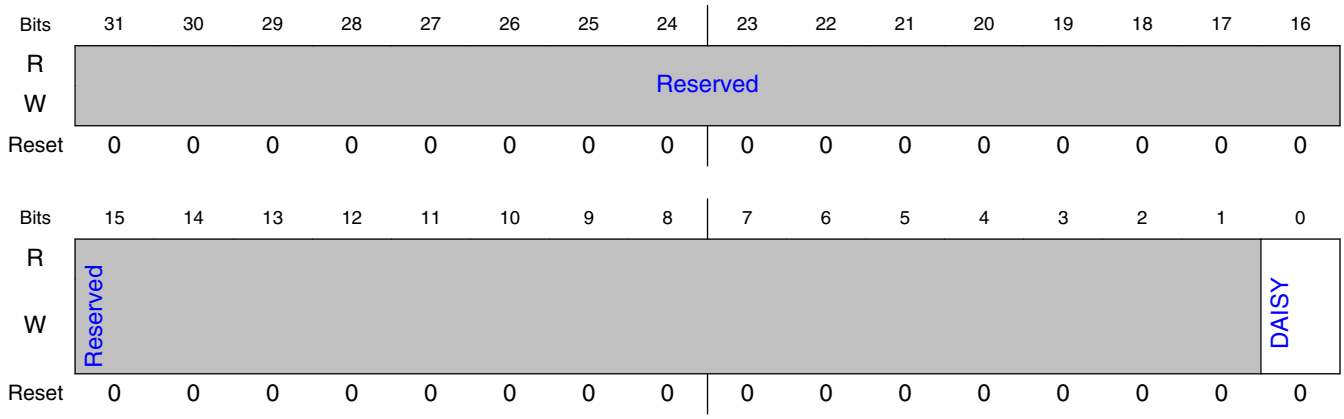
12.4.4.48.1 Offset

Register	Offset
MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_1	B8h

12.4.4.48.2 Function

DAISY Register

12.4.4.48.3 Diagram



12.4.4.48.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mic, In Pin: ipp_ind_mic_pdm_bitstream[1] 0b - Selecting Pad: GPIO_LPSR_10 for Mode: ALT3 1b - Selecting Pad: GPIO_LPSR_13 for Mode: ALT1

12.4.4.49 MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2)

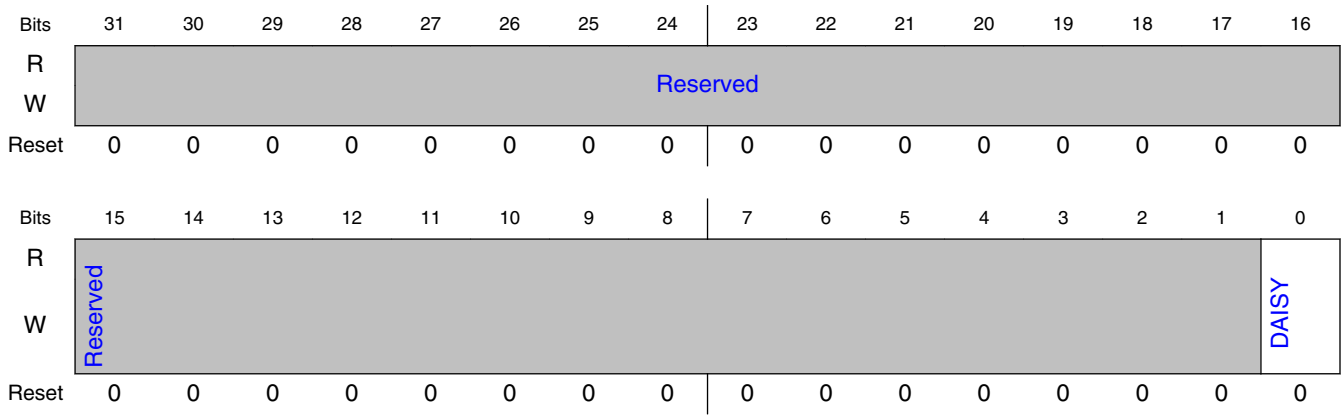
12.4.4.49.1 Offset

Register	Offset
MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_2	BCh

12.4.4.49.2 Function

DAISY Register

12.4.4.49.3 Diagram



12.4.4.49.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mic, In Pin: ipp_ind_mic_pdm_bitstream[2] 0b - Selecting Pad: GPIO_LPSR_11 for Mode: ALT3 1b - Selecting Pad: GPIO_LPSR_14 for Mode: ALT1

12.4.4.50 MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3 DAISY Register (MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3)

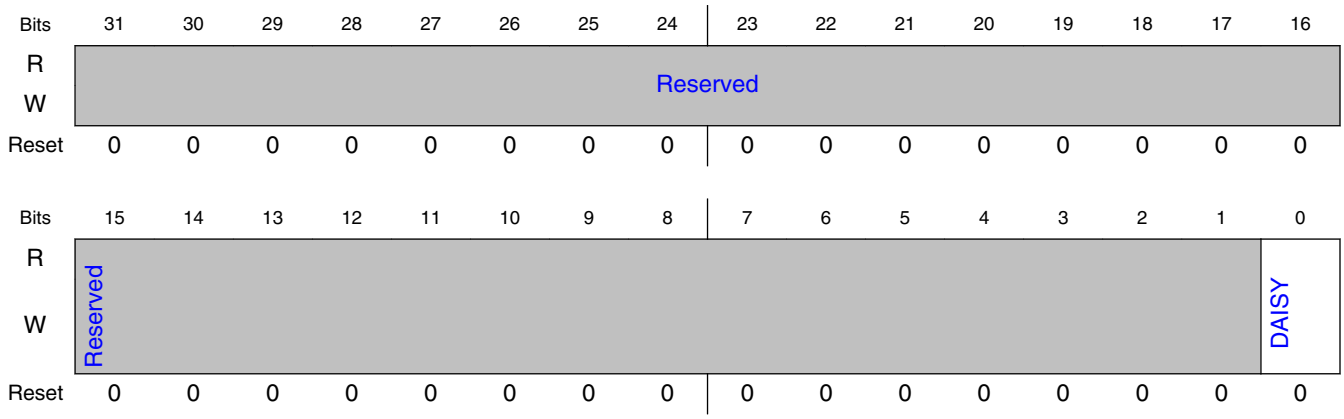
12.4.4.50.1 Offset

Register	Offset
MIC_IPP_IND_MIC_PDM_BITSTREAM_SELECT_INPUT_3	C0h

12.4.4.50.2 Function

DAISY Register

12.4.4.50.3 Diagram



12.4.4.50.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mic, In Pin: ipp_ind_mic_pdm_bitstream[3] 0b - Selecting Pad: GPIO_LPSR_12 for Mode: ALT3 1b - Selecting Pad: GPIO_LPSR_15 for Mode: ALT1

12.4.4.51 NMI_GLUE_IPP_IND_NMI_SELECT_INPUT DAISY Register (NMI_GLUE_IPP_IND_NMI_SELECT_INPUT)

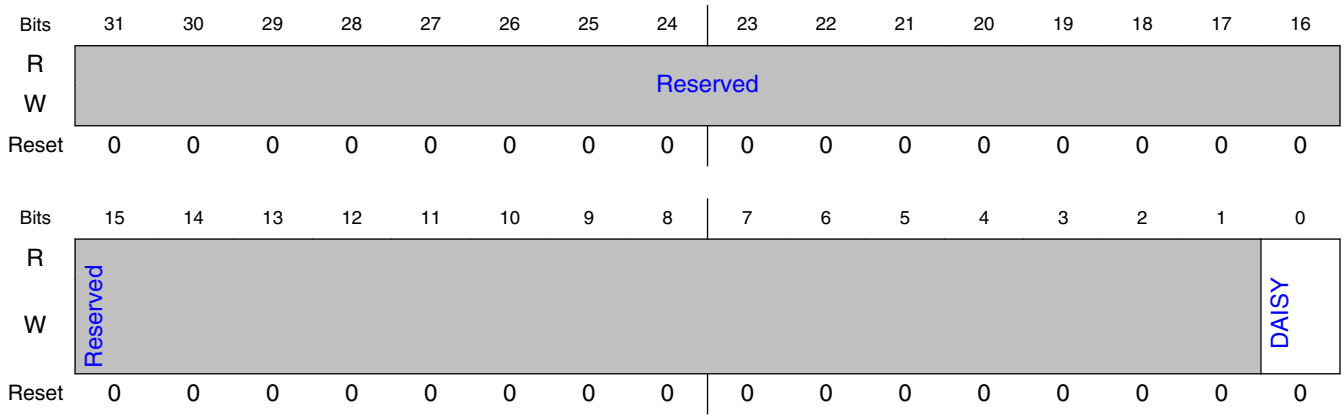
12.4.4.51.1 Offset

Register	Offset
NMI_GLUE_IPP_IND_NMI_SELECT_INPUT	C4h

12.4.4.51.2 Function

DAISY Register

12.4.4.51.3 Diagram



12.4.4.51.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: nmi_glue, In Pin: ipp_ind_nmi 0b - Selecting Pad: GPIO_LPSR_05 for Mode: ALT7 1b - Selecting Pad: WAKEUP_DIG for Mode: ALT7

12.4.4.52 SAI4_IPG_CLK_SAI_MCLK_SELECT_INPUT DAISY Register (SAI4_IPG_CLK_SAI_MCLK_SELECT_INPUT)

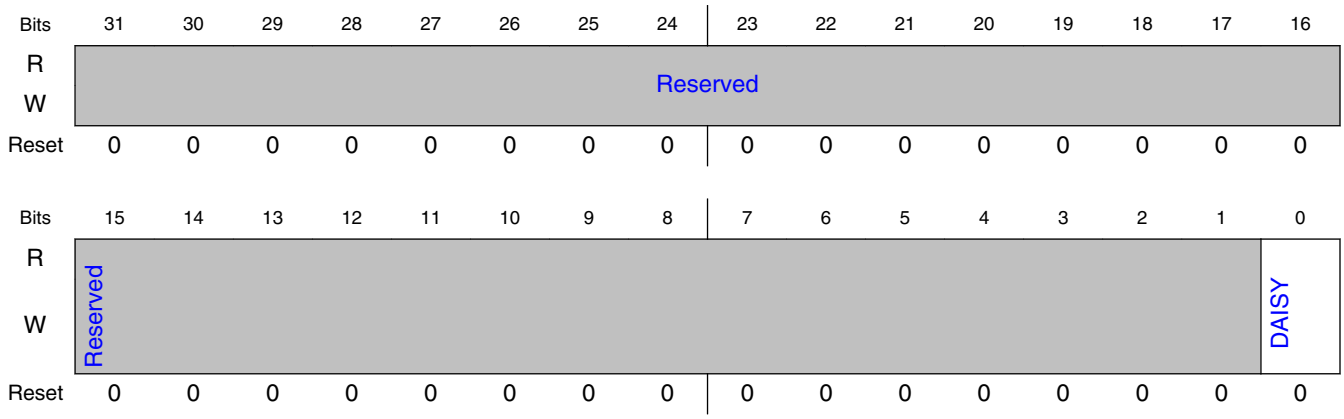
12.4.4.52.1 Offset

Register	Offset
SAI4_IPG_CLK_SAI_MCLK_SELECT_INPUT	C8h

12.4.4.52.2 Function

DAISY Register

12.4.4.52.3 Diagram



12.4.4.52.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipg_clk_sai_mclk 0b - Selecting Pad: GPIO_LPSR_00 for Mode: ALT7 1b - Selecting Pad: GPIO_LPSR_05 for Mode: ALT2

12.4.4.53 SAI4_IPP_IND_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_RXBCLK_SELECT_INPUT)

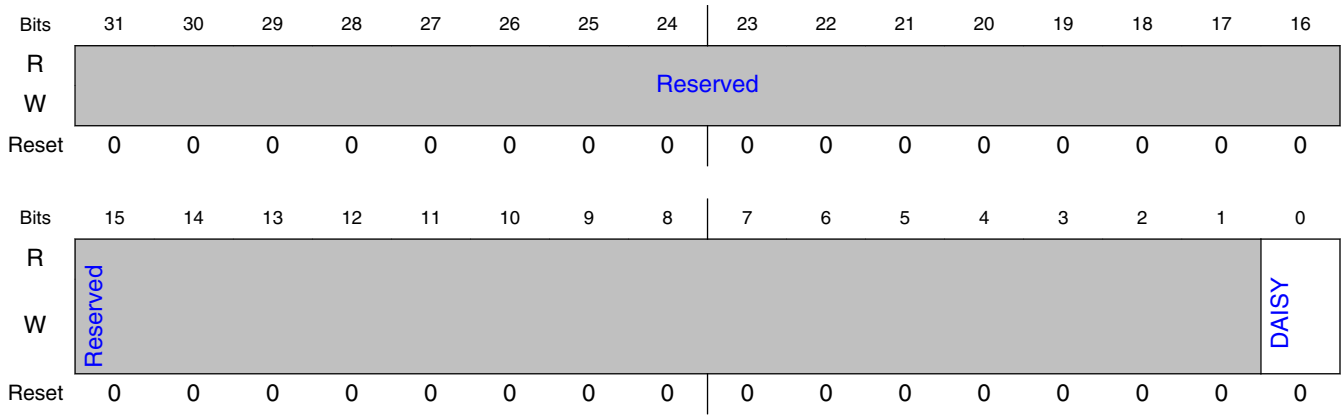
12.4.4.53.1 Offset

Register	Offset
SAI4_IPP_IND_SAI_RXBCLK_SELECT_INPUT	CCh

12.4.4.53.2 Function

DAISY Register

12.4.4.53.3 Diagram



12.4.4.53.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipp_ind_sai_rxbclk 0b - Selecting Pad: GPIO_LPSR_07 for Mode: ALT2 1b - Selecting Pad: GPIO_LPSR_14 for Mode: ALT7

12.4.4.54 SAI4_IPP_IND_SAI_RXDATA_SELECT_INPUT_0 DAISY Register (SAI4_IPP_IND_SAI_RXDATA_SELECT_INPUT_0)

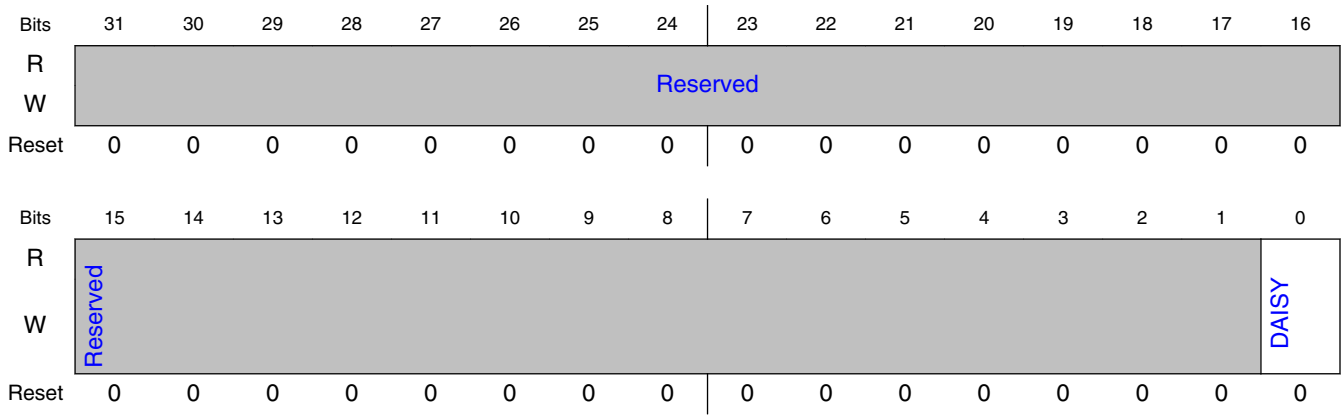
12.4.4.54.1 Offset

Register	Offset
SAI4_IPP_IND_SAI_RXDATA_SELECT_INPUT_0	D0h

12.4.4.54.2 Function

DAISY Register

12.4.4.54.3 Diagram



12.4.4.54.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipp_ind_sai_rxdata[0] 0b - Selecting Pad: GPIO_LPSR_06 for Mode: ALT2 1b - Selecting Pad: GPIO_LPSR_13 for Mode: ALT7

12.4.4.55 SAI4_IPP_IND_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_RXSYNC_SELECT_INPUT)

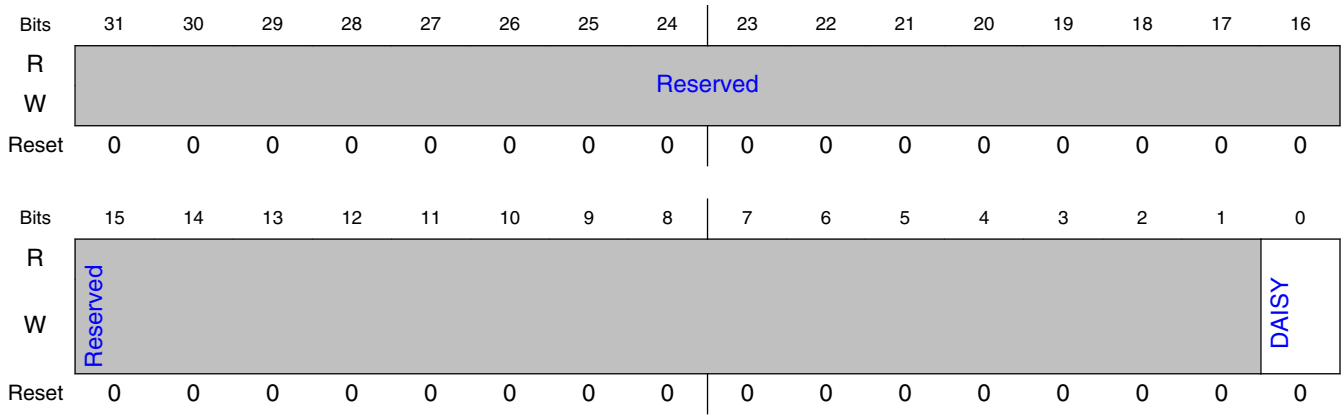
12.4.4.55.1 Offset

Register	Offset
SAI4_IPP_IND_SAI_RXSYNC_SELECT_INPUT	D4h

12.4.4.55.2 Function

DAISY Register

12.4.4.55.3 Diagram



12.4.4.55.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipp_ind_sai_rxsync 0b - Selecting Pad: GPIO_LPSR_08 for Mode: ALT2 1b - Selecting Pad: GPIO_LPSR_15 for Mode: ALT7

12.4.4.56 SAI4_IPP_IND_SAI_TXBCLK_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_TXBCLK_SELECT_INPUT)

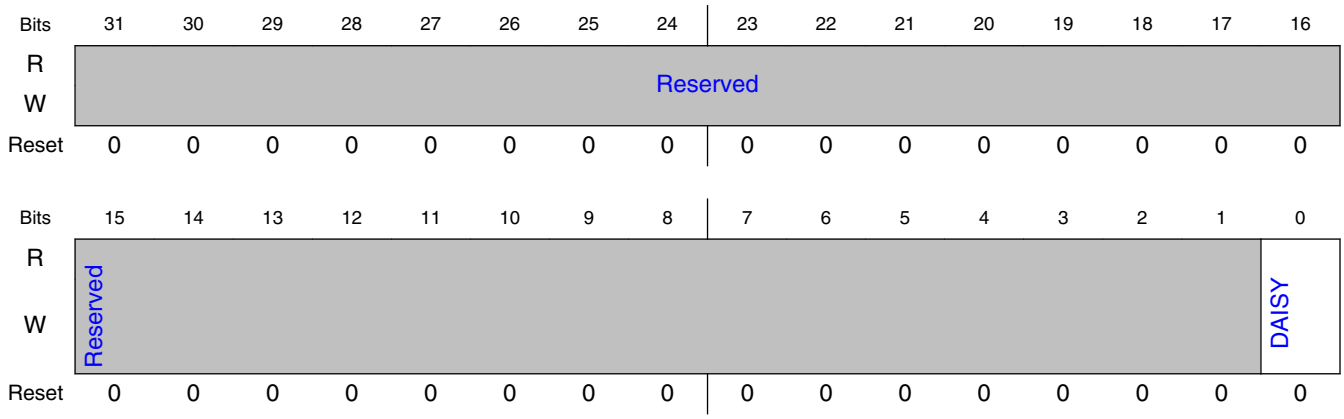
12.4.4.56.1 Offset

Register	Offset
SAI4_IPP_IND_SAI_TXBCLK_SELECT_INPUT	D8h

12.4.4.56.2 Function

DAISY Register

12.4.4.56.3 Diagram



12.4.4.56.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipp_ind_sai_txbclk 0b - Selecting Pad: GPIO_LPSR_04 for Mode: ALT2 1b - Selecting Pad: GPIO_LPSR_12 for Mode: ALT7

12.4.4.57 SAI4_IPP_IND_SAI_TXSYNC_SELECT_INPUT DAISY Register (SAI4_IPP_IND_SAI_TXSYNC_SELECT_INPUT)

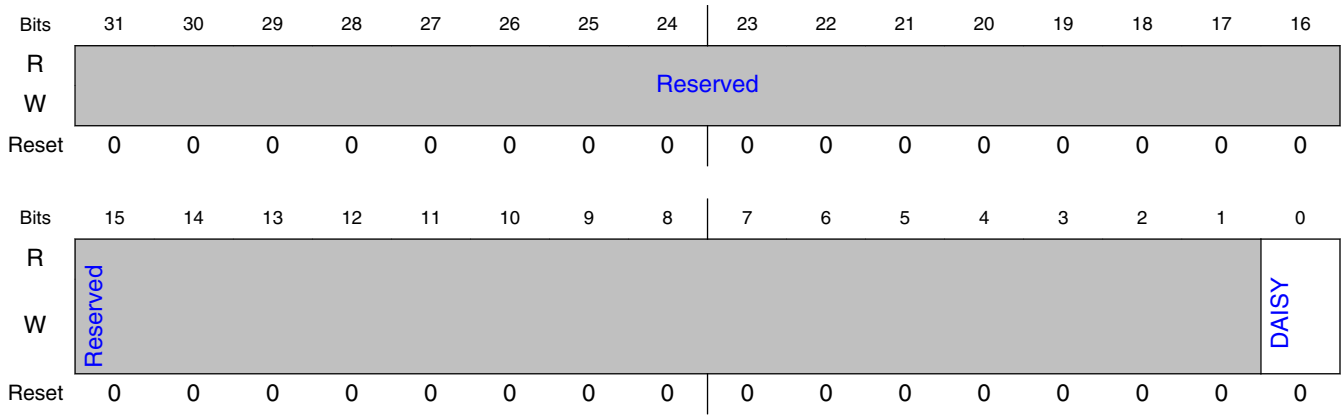
12.4.4.57.1 Offset

Register	Offset
SAI4_IPP_IND_SAI_TXSYNC_SELECT_INPUT	DCh

12.4.4.57.2 Function

DAISY Register

12.4.4.57.3 Diagram



12.4.4.57.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai4, In Pin: ipp_ind_sai_txsync 0b - Selecting Pad: GPIO_LPSR_03 for Mode: ALT2 1b - Selecting Pad: GPIO_LPSR_10 for Mode: ALT7

12.4.5 IOMUXC GPR register descriptions

12.4.5.1 iomuxc_gpr memory map

IOMUXC_GPR base address: 400E_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPR0 General Purpose Register (GPR0)	32	RW	0000_0018h
4h	GPR1 General Purpose Register (GPR1)	32	RW	0000_0000h
8h	GPR2 General Purpose Register (GPR2)	32	RW	0000_0000h
Ch	GPR3 General Purpose Register (GPR3)	32	RW	0000_0000h
10h	GPR4 General Purpose Register (GPR4)	32	RW	0000_0000h
14h	GPR5 General Purpose Register (GPR5)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
18h	GPR6 General Purpose Register (GPR6)	32	RW	0000_0000h
1Ch	GPR7 General Purpose Register (GPR7)	32	RW	0000_0000h
20h	GPR8 General Purpose Register (GPR8)	32	RW	0000_0000h
24h	GPR9 General Purpose Register (GPR9)	32	RW	0000_0000h
28h	GPR10 General Purpose Register (GPR10)	32	RW	0000_0000h
2Ch	GPR11 General Purpose Register (GPR11)	32	RW	0000_0000h
30h	GPR12 General Purpose Register (GPR12)	32	RW	0000_0000h
34h	GPR13 General Purpose Register (GPR13)	32	RW	0000_0000h
38h	GPR14 General Purpose Register (GPR14)	32	RW	0000_0000h
3Ch	GPR15 General Purpose Register (GPR15)	32	RW	0000_0000h
40h	GPR16 General Purpose Register (GPR16)	32	RW	0000_AA03h
44h	GPR17 General Purpose Register (GPR17)	32	RW	0000_0000h
48h	GPR18 General Purpose Register (GPR18)	32	RW	0000_0000h
50h	GPR20 General Purpose Register (GPR20)	32	RW	0000_0000h
54h	GPR21 General Purpose Register (GPR21)	32	RW	0000_0000h
58h	GPR22 General Purpose Register (GPR22)	32	RW	0000_0000h
5Ch	GPR23 General Purpose Register (GPR23)	32	RW	0000_0000h
60h	GPR24 General Purpose Register (GPR24)	32	RW	0000_0000h
64h	GPR25 General Purpose Register (GPR25)	32	RW	0000_0000h
68h	GPR26 General Purpose Register (GPR26)	32	RW	0000_0000h
6Ch	GPR27 General Purpose Register (GPR27)	32	RW	0000_0000h
70h	GPR28 General Purpose Register (GPR28)	32	RW	0000_0000h
74h	GPR29 General Purpose Register (GPR29)	32	RW	0000_0001h
78h	GPR30 General Purpose Register (GPR30)	32	RW	0000_0001h
7Ch	GPR31 General Purpose Register (GPR31)	32	RW	0000_0012h
80h	GPR32 General Purpose Register (GPR32)	32	RW	0000_0000h
84h	GPR33 General Purpose Register (GPR33)	32	RW	0000_0000h
88h	GPR34 General Purpose Register (GPR34)	32	RW	0000_0000h
8Ch	GPR35 General Purpose Register (GPR35)	32	RW	0000_0000h
90h	GPR36 General Purpose Register (GPR36)	32	RW	0000_0000h
94h	GPR37 General Purpose Register (GPR37)	32	RW	0000_0017h
98h	GPR38 General Purpose Register (GPR38)	32	RW	0000_0000h
9Ch	GPR39 General Purpose Register (GPR39)	32	RW	0000_0000h
A0h	GPR40 General Purpose Register (GPR40)	32	RW	0000_0000h
A4h	GPR41 General Purpose Register (GPR41)	32	RW	0000_0000h
A8h	GPR42 General Purpose Register (GPR42)	32	RW	0000_0000h
ACh	GPR43 General Purpose Register (GPR43)	32	RW	0000_0000h
B0h	GPR44 General Purpose Register (GPR44)	32	RW	0000_0000h
B4h	GPR45 General Purpose Register (GPR45)	32	RW	0000_0000h
B8h	GPR46 General Purpose Register (GPR46)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
BCh	GPR47 General Purpose Register (GPR47)	32	RW	0000_0000h
C0h	GPR48 General Purpose Register (GPR48)	32	RW	0000_0000h
C4h	GPR49 General Purpose Register (GPR49)	32	RW	0000_0000h
C8h	GPR50 General Purpose Register (GPR50)	32	RW	0000_0000h
CCh	GPR51 General Purpose Register (GPR51)	32	RW	0000_0000h
D0h	GPR52 General Purpose Register (GPR52)	32	RW	0000_0000h
D4h	GPR53 General Purpose Register (GPR53)	32	RW	0000_0000h
D8h	GPR54 General Purpose Register (GPR54)	32	RW	0000_0000h
DCh	GPR55 General Purpose Register (GPR55)	32	RW	0000_0000h
ECh	GPR59 General Purpose Register (GPR59)	32	RW	0000_0550h
F8h	GPR62 General Purpose Register (GPR62)	32	RW	0000_02DBh
FCh	GPR63 General Purpose Register (GPR63)	32	R	0000_0000h
100h	GPR64 General Purpose Register (GPR64)	32	RW	0000_4000h
104h	GPR65 General Purpose Register (GPR65)	32	RW	0000_4000h
108h	GPR66 General Purpose Register (GPR66)	32	RW	0000_4000h
10Ch	GPR67 General Purpose Register (GPR67)	32	RW	0000_4000h
110h	GPR68 General Purpose Register (GPR68)	32	RW	0000_4000h
114h	GPR69 General Purpose Register (GPR69)	32	RW	0000_0000h
118h	GPR70 General Purpose Register (GPR70)	32	RW	0000_0000h
11Ch	GPR71 General Purpose Register (GPR71)	32	RW	0000_0000h
120h	GPR72 General Purpose Register (GPR72)	32	RW	0000_0000h
124h	GPR73 General Purpose Register (GPR73)	32	RW	0000_0000h
128h	GPR74 General Purpose Register (GPR74)	32	RW	0000_0000h
12Ch	GPR75 General Purpose Register (GPR75)	32	R	0000_0000h
130h	GPR76 General Purpose Register (GPR76)	32	R	0000_0000h

12.4.5.2 GPR0 General Purpose Register (GPR0)

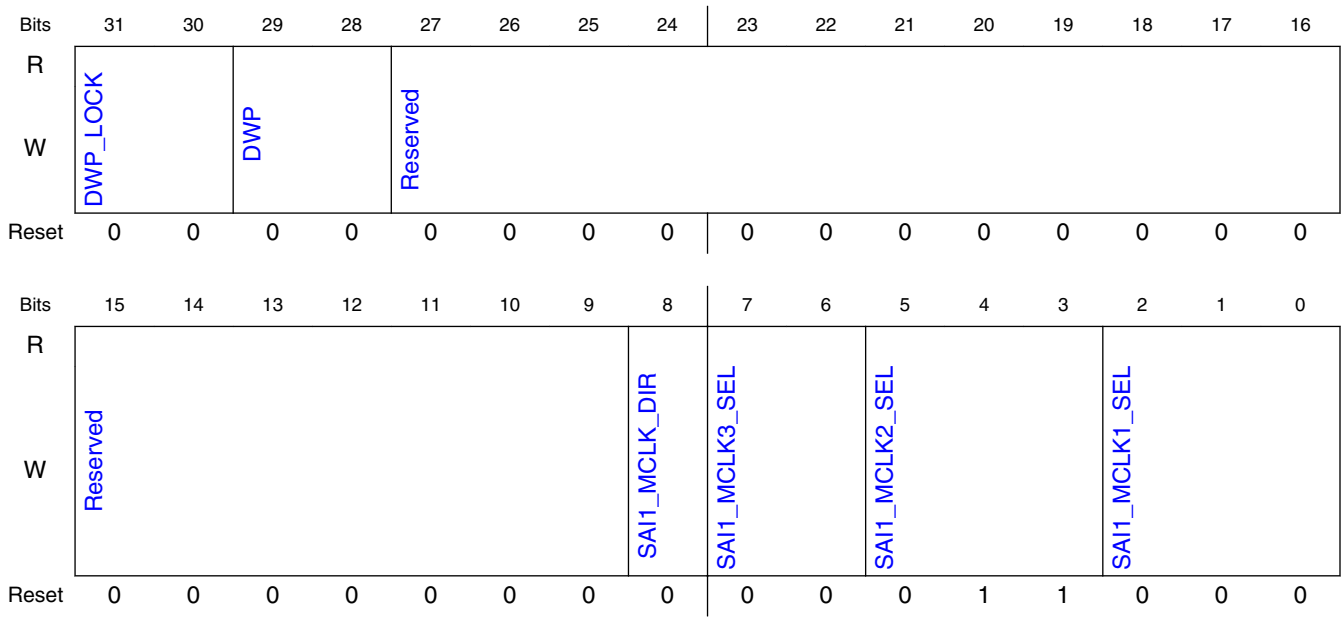
12.4.5.2.1 Offset

Register	Offset
GPR0	0h

12.4.5.2.2 Function

GPR Register

12.4.5.2.3 Diagram



12.4.5.2.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-9 —	Reserved
8 SAI1_MCLK_DIR	SAI1_MCLK signal direction control 0: SAI1_MCLK is input signal 1: SAI1_MCLK is output signal
7-6	SAI1 MCLK3 source select

Table continues on the next page...

Field	Function
SAI1_MCLK3_SEL	See the Audio subsystem clocking diagram in the Audio Overview Chapter for more information. 00: SPDIF_CLK_ROOT 01: spdif_tx_clk2 10: spdif_srclk 11: spdif_outclock
5-3 SAI1_MCLK2_SEL	SAI1 MCLK2 source select See the Audio subsystem clocking diagram in the Audio Overview Chapter for more information. 000 SAI1_CLK_ROOT 001: SAI2_CLK_ROOT 010: SAI3_CLK_ROOT 011: sai1_ipg_clk_sai_mclk 100: sai2_ipg_clk_sai_mclk 101: sai3_ipg_clk_sai_mclk 110: Reserved 111: Reserved
2-0 SAI1_MCLK1_SEL	SAI1 MCLK1 source select 000 SAI1_CLK_ROOT See the Audio subsystem clocking diagram in the Audio Overview Chapter for more information. 001: SAI2_CLK_ROOT 010: SAI3_CLK_ROOT 011: sai1_ipg_clk_sai_mclk 100: sai2_ipg_clk_sai_mclk 101: sai3_ipg_clk_sai_mclk 110: Reserved 111: Reserved

12.4.5.3 GPR1 General Purpose Register (GPR1)

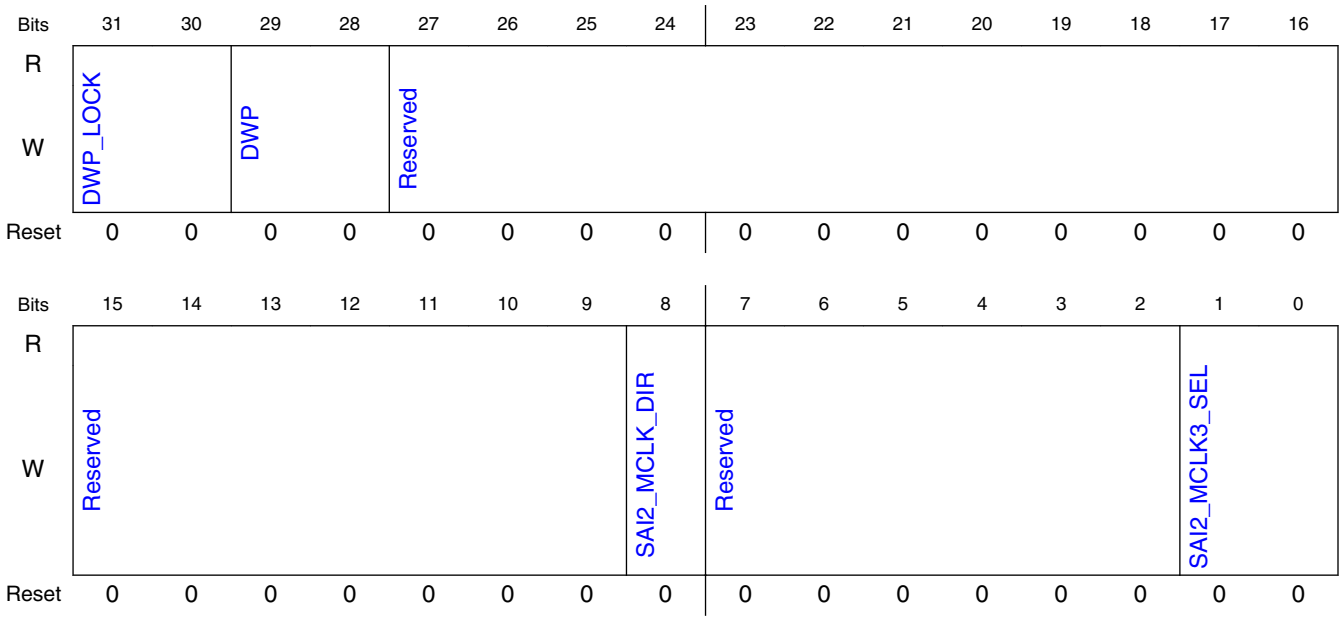
12.4.5.3.1 Offset

Register	Offset
GPR1	4h

12.4.5.3.2 Function

GPR Register

12.4.5.3.3 Diagram



12.4.5.3.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-9 —	Reserved
8 SAI2_MCLK_DIR	SAI2_MCLK signal direction control 0: SAI2_MCLK is input signal 1: SAI2_MCLK is output signal
7-2 —	Reserved

Table continues on the next page...

Field	Function
1-0 SAI2_MCLK3_SEL	SAI2 MCLK3 source select See the Audio subsystem clocking diagram in the Audio Overview Chapter for more information. 00: SPDIF_CLK_ROOT 01: spdif_tx_clk2 10: spdif_srclk 11: spdif_outclock

12.4.5.4 GPR2 General Purpose Register (GPR2)

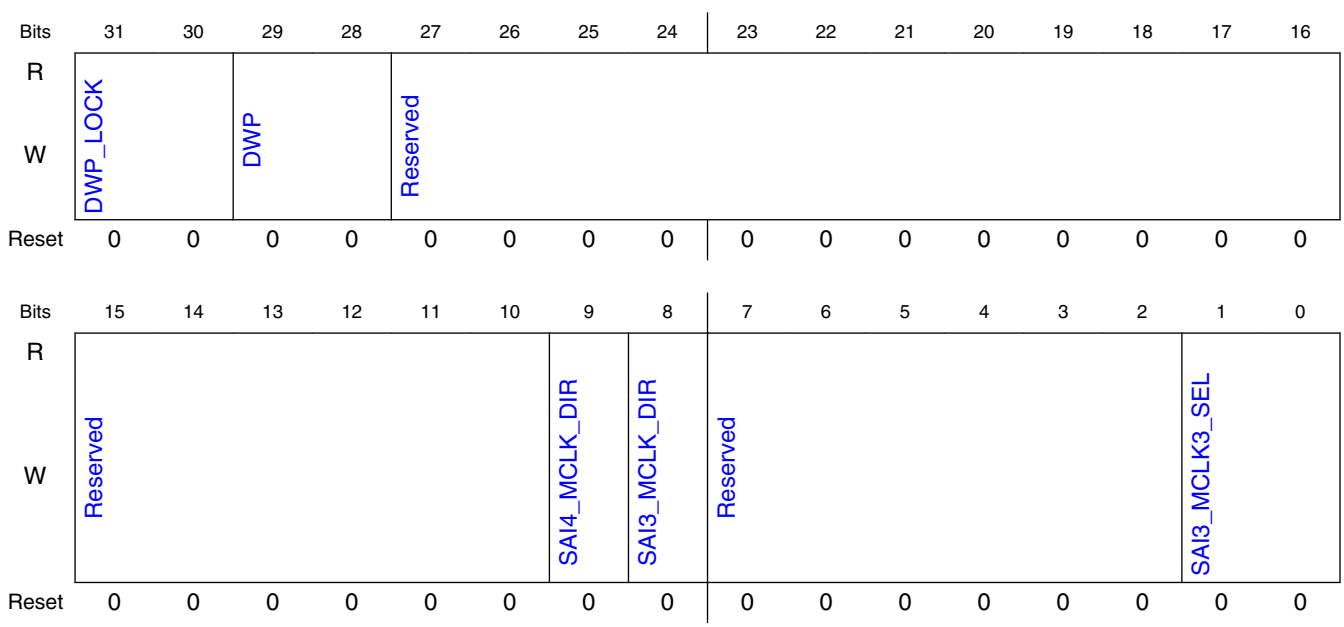
12.4.5.4.1 Offset

Register	Offset
GPR2	8h

12.4.5.4.2 Function

GPR Register

12.4.5.4.3 Diagram



12.4.5.4.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-10 —	Reserved
9 SAI4_MCLK_DIR	SAI4_MCLK signal direction control 0: SAI4_MCLK is input signal 1: SAI4_MCLK is output signal
8 SAI3_MCLK_DIR	SAI3_MCLK signal direction control 0: SAI3_MCLK is input signal 1: SAI3_MCLK is output signal
7-2 —	Reserved
1-0 SAI3_MCLK3_SEL	SAI3 MCLK3 source select See the Audio subsystem clocking diagram in the Audio Overview Chapter for more information. 00: SPDIF_CLK_ROOT 01: spdif_tx_clk2 10: spdif_srclk 11: spdif_outclock

12.4.5.5 GPR3 General Purpose Register (GPR3)

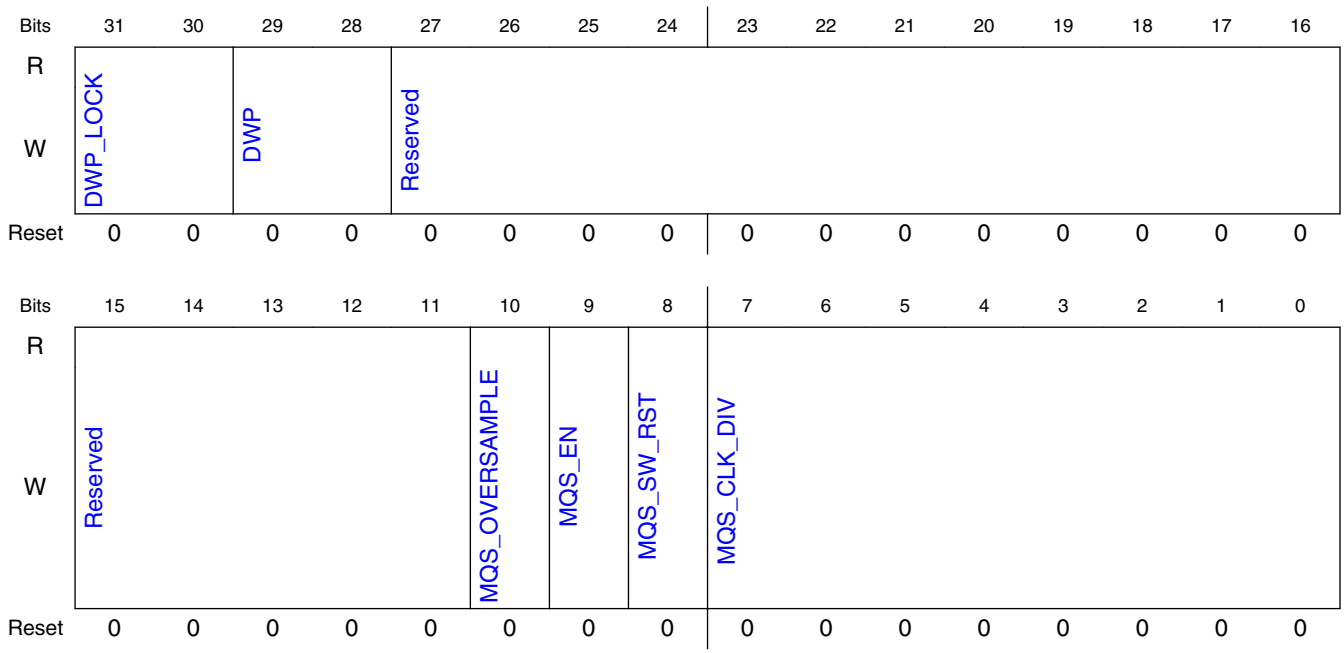
12.4.5.5.1 Offset

Register	Offset
GPR3	Ch

12.4.5.5.2 Function

GPR Register

12.4.5.5.3 Diagram



12.4.5.5.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-11 —	Reserved

Table continues on the next page...

Field	Function
10 MQS_OVERSAMPLE	Medium Quality Sound (MQS) Oversample Used to control the PWM oversampling rate compared with mclk. 0: 32 1: 64
9 MQS_EN	MQS enable 0: Disable MQS 1: Enable MQS
8 MQS_SW_RST	MQS software reset 0: Exit software reset for MQS 1: Enable software reset for MQS
7-0 MQS_CLK_DIV	Divider ratio control for mclk from hmclk. mclk frequency = 1/(n+1) * hmclk frequency 00000000: mclk frequency = hmclk frequency 00000001: mclk frequency = 1/2 * hmclk frequency 00000010: mclk frequency = 1/3 * hmclk frequency 11111111: mclk frequency = 1/256 * hmclk frequency

12.4.5.6 GPR4 General Purpose Register (GPR4)

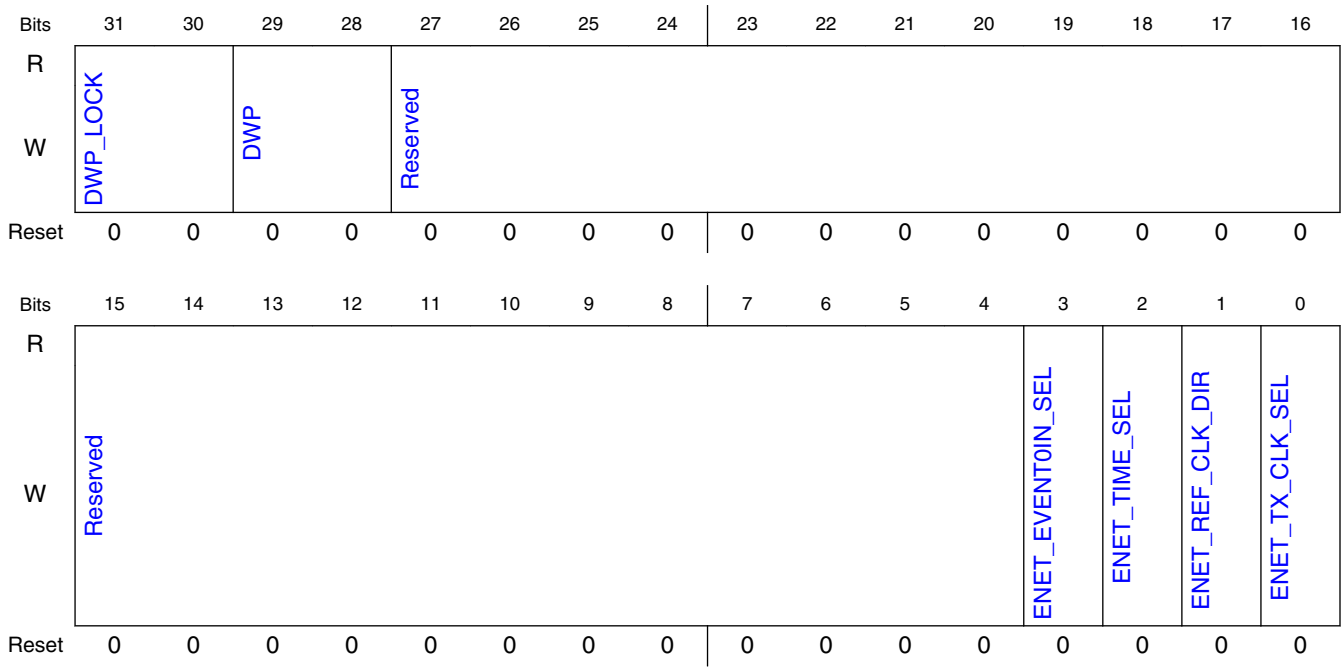
12.4.5.6.1 Offset

Register	Offset
GPR4	10h

12.4.5.6.2 Function

GPR Register

12.4.5.6.3 Diagram



12.4.5.6.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-4 —	Reserved
3 ENET_EVENT0IN_SEL	ENET ENET_1588_EVENT0_IN source select 0: ENET_1588_EVENT0_IN input is from pad 1: ENET_1588_EVENT0_IN input is from GPT2 COMPARE1 output
2	ENET master timer source select

Table continues on the next page...

Field	Function
ENET_TIME_SEL	IEEE 1588 master counter value is used when IEEE 1588 circuitry for ENET is in slave mode. 0: master counter value is from ENET1G module 1: master counter value is from ENET_QOS module
1 ENET_REF_CLK_DIR	ENET_REF_CLK direction control This bitfield controls the direction of ENET_REF_CLK. ENET_REF_CLK is the 50MHz RMII clock. It should be set together with the corresponding IOMUXC SW_MUX_CTL_PAD_xx SION bit so that ENET1_CLK_ROOT can provide 50MHz RMII clock to both ENET and board. 0: ENET_REF_CLK is input 1: ENET_REF_CLK is output driven by ENET1_CLK_ROOT
0 ENET_TX_CLK_SEL	ENET TX_CLK select This bit should always be set to 1 to use the 25MHz MII clock 0: Not supported 1: ENET TX_CLK is from pad

12.4.5.7 GPR5 General Purpose Register (GPR5)

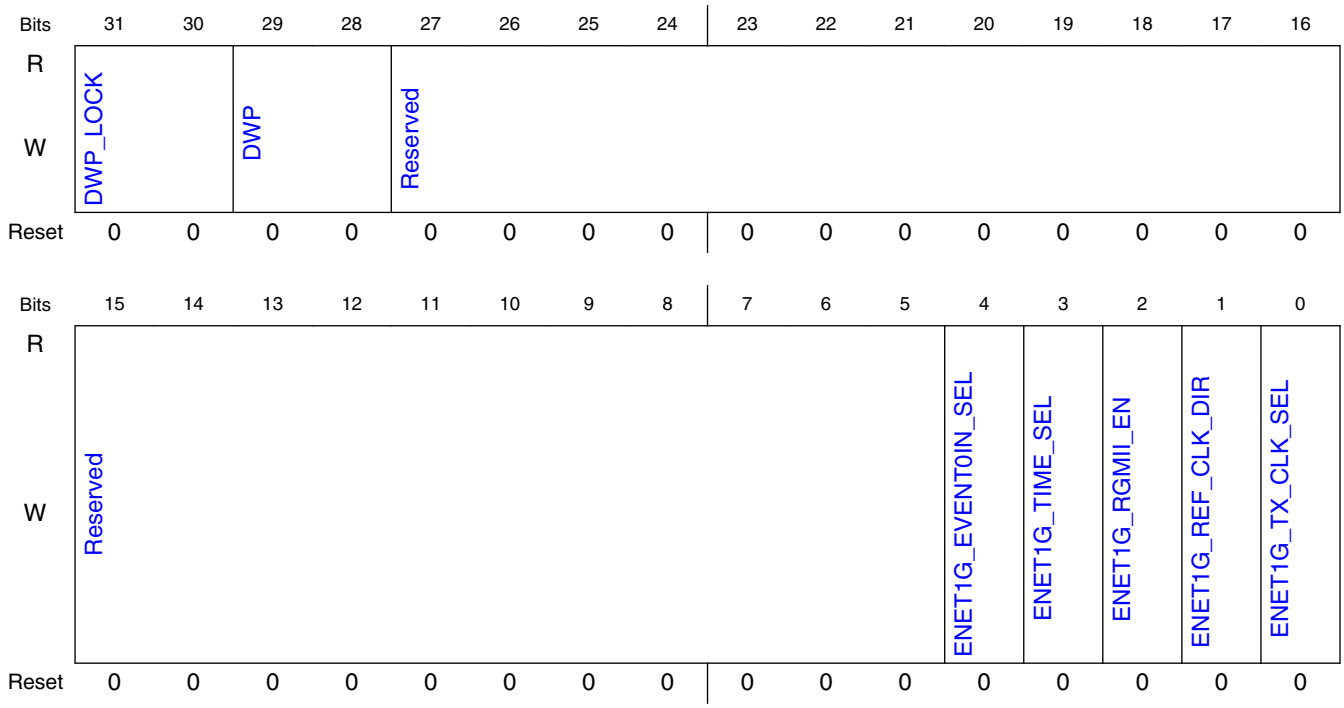
12.4.5.7.1 Offset

Register	Offset
GPR5	14h

12.4.5.7.2 Function

GPR Register

12.4.5.7.3 Diagram



12.4.5.7.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-5 —	Reserved
4 ENET1G_EVENT0IN_SEL	ENET1G ENET_1588_EVENT0_IN source select 0: ENET_1588_EVENT0_IN input is from pad 1 ENET_1588_EVENT0_IN input is from GPT2_COMPARE2 output

Table continues on the next page...

Field	Function
3 ENET1G_TIME_SEL	ENET1G master timer source select IEEE 1588 master counter value is used when IEEE 1588 circuitry for ENET1G is in slave mode. 0: master counter value is from ENET module 1: master counter value is from ENET_QOS module
2 ENET1G_RGMII_EN	ENET1G RGMII TX clock output enable Set this bit to enable ENET1G RGMII TX clock output on TX_CLK_IO pad. It should be set in RGMII mode, and be cleared in MII mode as in MII mode TX_CLK_IO is input.
1 ENET1G_REF_CLK_DIR	ENET1G_REF_CLK direction control This bitfield controls the direction of ENET1G_REF_CLK. ENET1G_REF_CLK is the 50MHz RMII clock. It should be set together with the corresponding IOMUXC SW_MUX_CTL_PAD_xx SION bit so that ENET1_CLK_ROOT can provide 50MHz RMII clock to both ENET and board. 0: ENET1G_REF_CLK is input 1: ENET1G_REF_CLK is output driven by ENET2_CLK_ROOT
0 ENET1G_TX_CLK_SEL	ENET1G TX_CLK select This bit should be set to 1 in MII mode to use the 25MHz TX clock. This bit should be set to 0 in RGMII mode to use ENET2_CLK_ROOT as the 125MHZ TX clock. 0 ENET1G TX_CLK is driven by ENET2_CLK_ROOT 1: ENET1G TX_CLK is from pad

12.4.5.8 GPR6 General Purpose Register (GPR6)

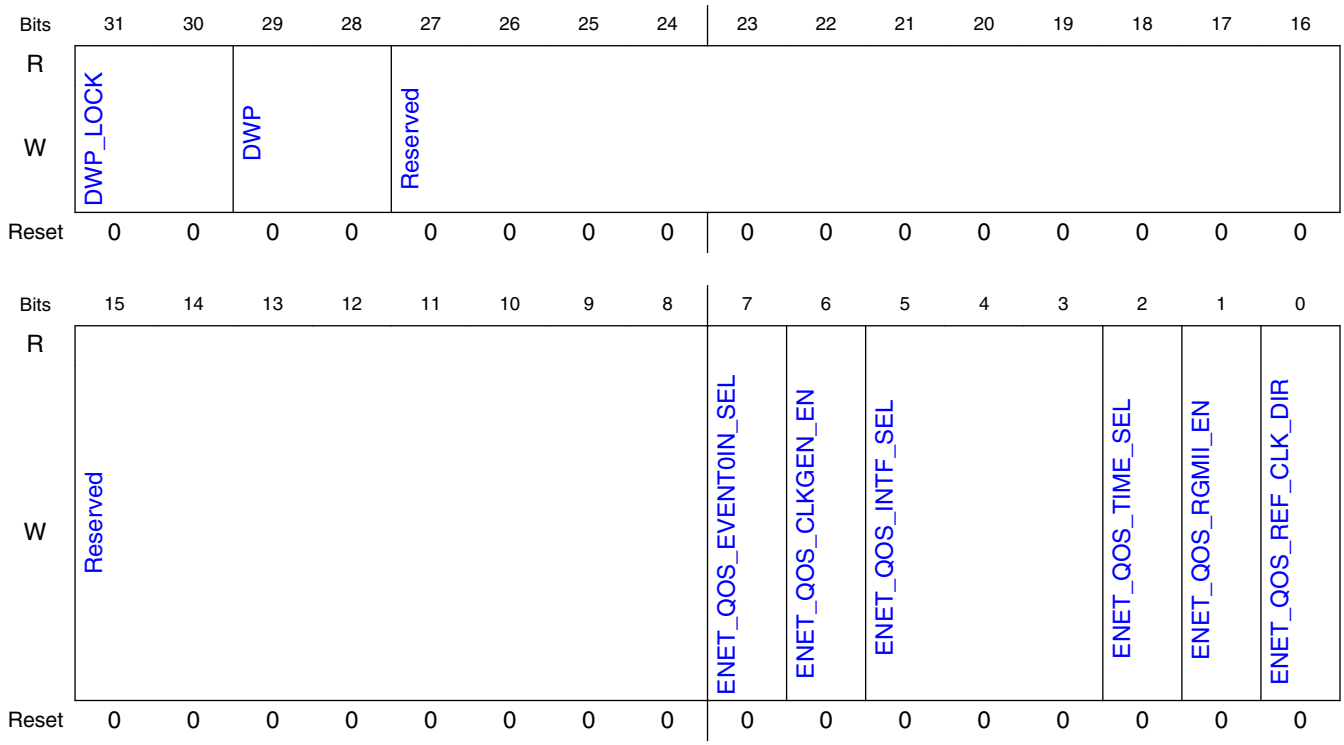
12.4.5.8.1 Offset

Register	Offset
GPR6	18h

12.4.5.8.2 Function

GPR Register

12.4.5.8.3 Diagram



12.4.5.8.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-8 —	Reserved
7	ENET_QOS ENET_1588_EVENT0_IN source select 0: ENET_1588_EVENT0_IN input is from pad

Table continues on the next page...

Field	Function
ENET_QOS_EVENT0IN_SEL	1: ENET_1588_EVENT0_IN input is from GPT3_COMPARE1 output
6 ENET_QOS_CLKGEN_EN	ENET_QOS clock generator enable The ENET_QOS clock generator provides ENET_QOS TX/RX clocks according to the PHY interface type and speed. Set this bit to enable the clock generator. It's required by all the three modes. In RGMII mode, ENET_QOS_CLK_ROOT is the source of TX clock and will drive ENET_QOS TX_CLK pad given ENET_QOS_RGMII_EN is set. In MII mode, the TX clock is input from TX_CLK. In RMII mode, the TX clock is input from REF_CLK.
5-3 ENET_QOS_INTF_SEL	ENET_QOS PHY Interface Select Selects the PHY interface of the MAC. This config is sampled only during ENET_QOS reset assertion, and it is ignored after that. User can assert ENET_QOS reset via ENET_QOS software reset bit. 000: MII 001: RGMII 100: RMII
2 ENET_QOS_TIME_SEL	ENET_QOS master timer source select IEEE 1588 master counter value is used when IEEE 1588 circuitry for ENET_QOS is in slave mode. 0: master counter value is from ENET module 1 master counter value is from ENET_1G module
1 ENET_QOS_RGMII_EN	ENET_QOS RGMII TX clock output enable Set this bit to enable ENET_QOS RGMII TX clock output on TX_CLK pad. It should be set in RGMII mode, and be cleared in MII mode as in MII mode TX_CLK is input.
0 ENET_QOS_REF_CLK_DIR	ENET_QOS_REF_CLK direction control This bitfield controls the direction of ENET_QOS_REF_CLK. ENET_QOS_REF_CLK is the 50MHz RMII clock. It should be set together with the corresponding IOMUXC SW_MUX_CTL_PAD_xx SION bit so that ENET1_CLK_ROOT can provide 50MHz RMII clock to both ENET and board. 0: ENET_QOS_REF_CLK is input 1 ENET_QOS_REF_CLK is output driven by ENET_QOS_CLK_ROOT

12.4.5.9 GPR7 General Purpose Register (GPR7)

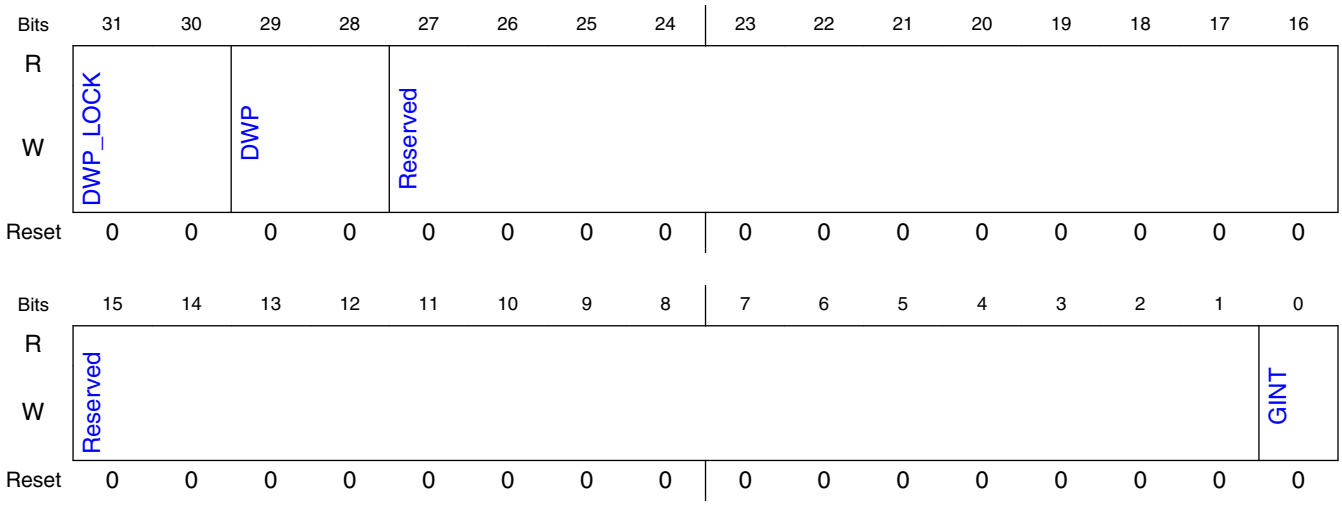
12.4.5.9.1 Offset

Register	Offset
GPR7	1Ch

12.4.5.9.2 Function

GPR Register

12.4.5.9.3 Diagram



12.4.5.9.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 GINT	Global interrupt This is the global interrupt bit. It is connected to CM7/CM4 IRQ#53 and GPC. 0 Global interrupt request is not asserted 1: Global interrupt request is asserted. Interrupt is issued to CM7/CM4 IRQ#53 and GPC

12.4.5.10 GPR8 General Purpose Register (GPR8)

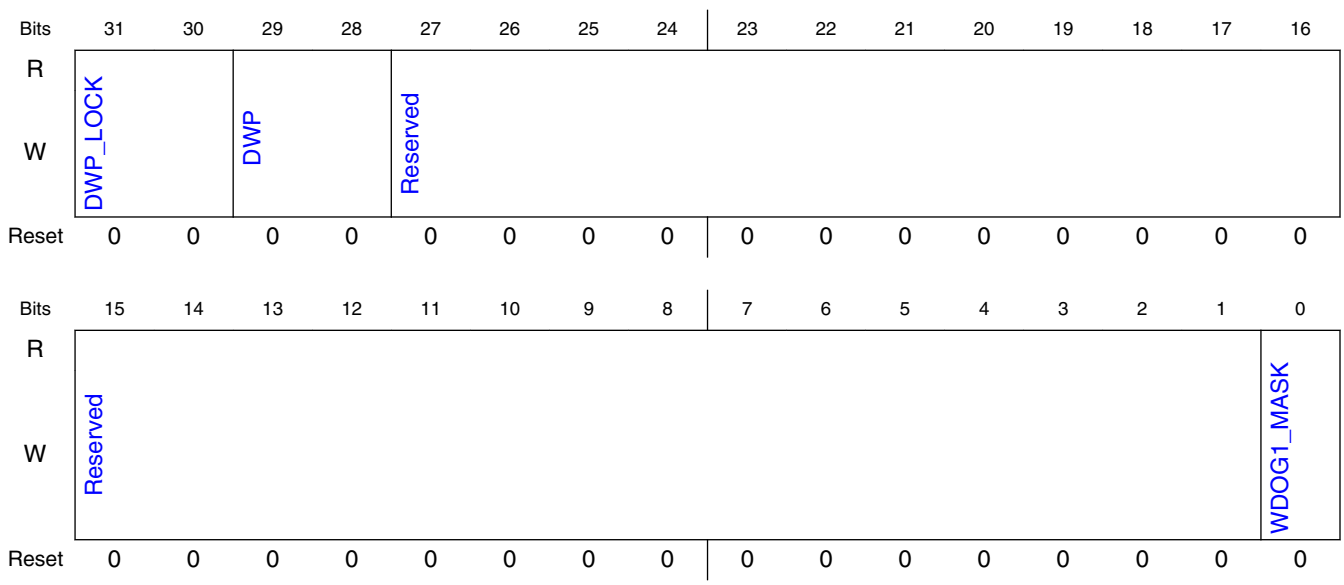
12.4.5.10.1 Offset

Register	Offset
GPR8	20h

12.4.5.10.2 Function

GPR Register

12.4.5.10.3 Diagram



12.4.5.10.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed

Table continues on the next page...

Field	Function
	01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 WDOG1_MASK	WDOG1 timeout mask for WDOG_ANY WDOG_ANY pad is driven by WDOG1 timeout OR WDOG2 timeout. Set this bit will mask WDOG1 timeout from WDOG_ANY.

12.4.5.11 GPR9 General Purpose Register (GPR9)

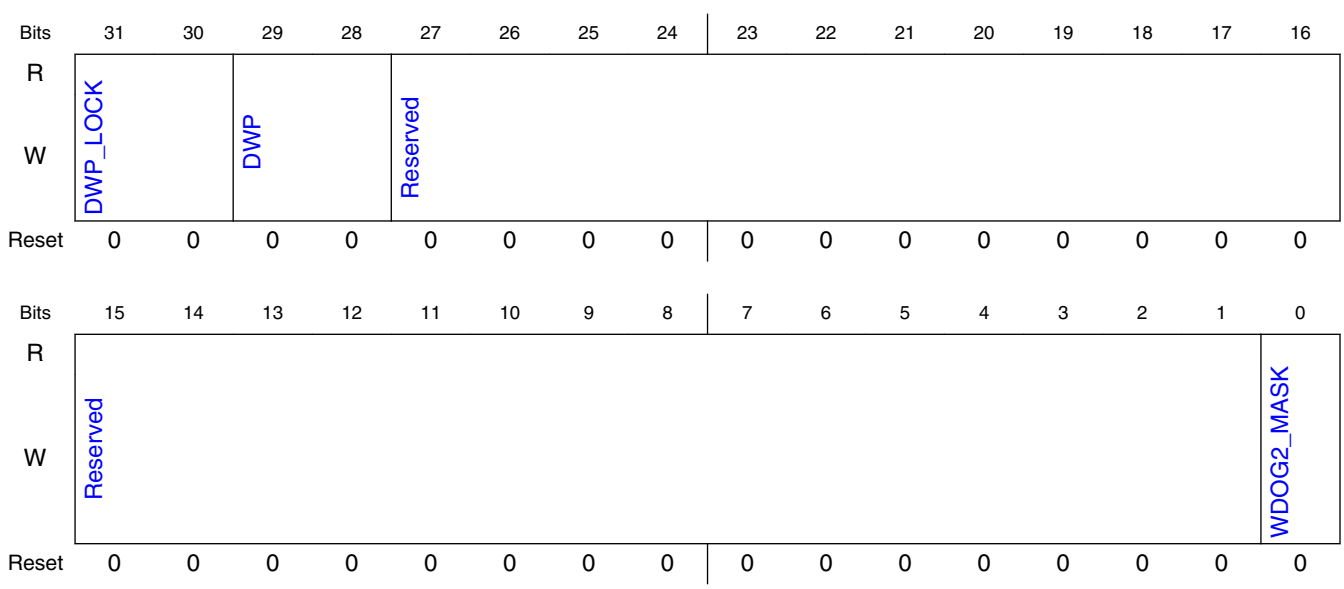
12.4.5.11.1 Offset

Register	Offset
GPR9	24h

12.4.5.11.2 Function

GPR Register

12.4.5.11.3 Diagram



12.4.5.11.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 WDOG2_MASK	WDOG2 timeout mask for WDOG_ANY WDOG_ANY pad is driven by WDOG1 timeout OR WDOG2 timeout. Set this bit will mask WDOG2 timeout from WDOG_ANY.

12.4.5.12 GPR10 General Purpose Register (GPR10)

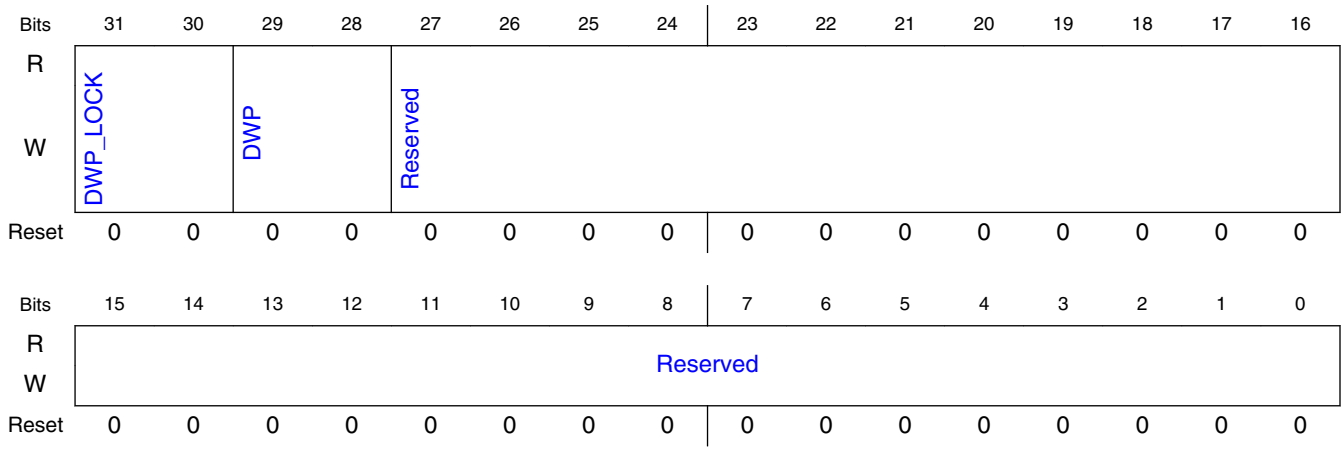
12.4.5.12.1 Offset

Register	Offset
GPR10	28h

12.4.5.12.2 Function

GPR Register

12.4.5.12.3 Diagram



12.4.5.12.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.13 GPR11 General Purpose Register (GPR11)

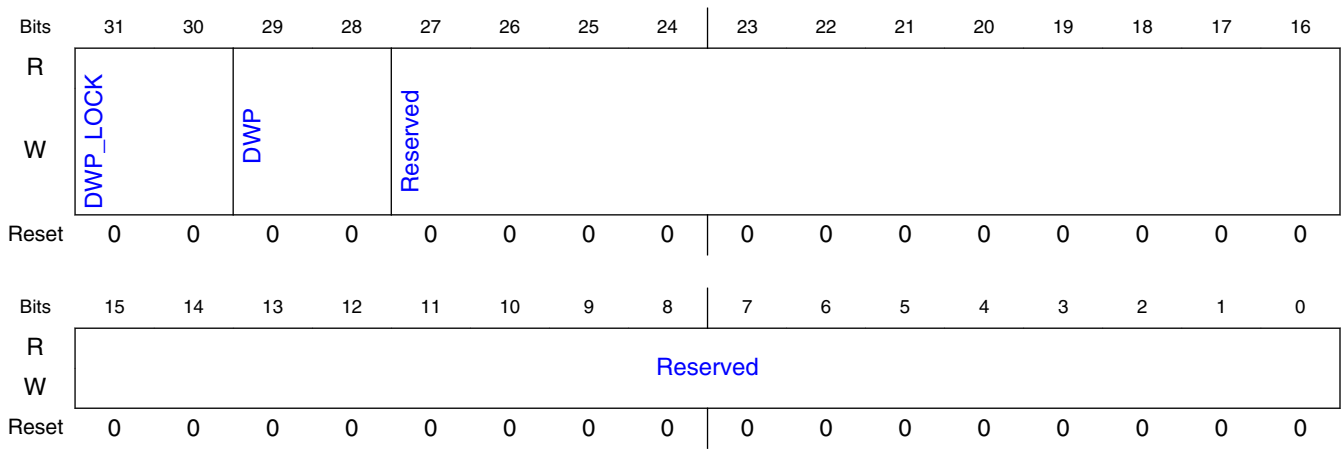
12.4.5.13.1 Offset

Register	Offset
GPR11	2Ch

12.4.5.13.2 Function

GPR Register

12.4.5.13.3 Diagram



12.4.5.13.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.14 GPR12 General Purpose Register (GPR12)

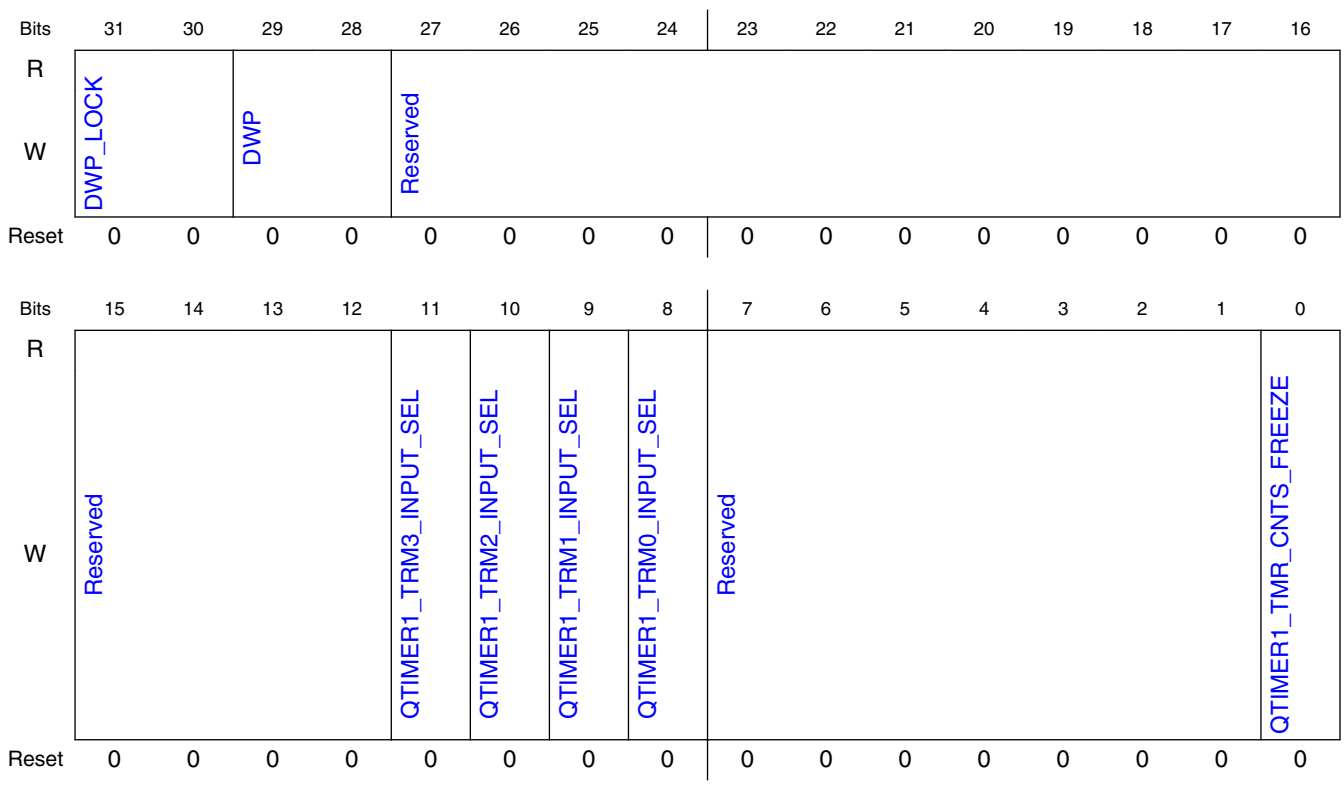
12.4.5.14.1 Offset

Register	Offset
GPR12	30h

12.4.5.14.2 Function

GPR Register

12.4.5.14.3 Diagram



12.4.5.14.4 Fields

Field	Function
31-30	Domain write protection lock
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked

Table continues on the next page...

Field	Function
	11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-12 —	Reserved
11 QTIMER1_TRM3_INPUT_SEL	QTIMER1 TMR3 input select 0: input from IOMUX; 1: input from XBAR
10 QTIMER1_TRM2_INPUT_SEL	QTIMER1 TMR2 input select 0: input from IOMUX 1: input from XBAR
9 QTIMER1_TRM1_INPUT_SEL	QTIMER1 TMR1 input select 0: input from IOMUX 1: input from XBAR
8 QTIMER1_TRM0_INPUT_SEL	QTIMER1 TMR0 input select 0: input from IOMUX 1: input from XBAR
7-1 —	Reserved
0 QTIMER1_TMR_CNTS_FREEZE	QTIMER1 timer counter freeze Setting this bit resets counters and output pins of QTIMER1. It is not self-clearing. 0: Timer counter works normally 1 Reset counter and ouput flags

12.4.5.15 GPR13 General Purpose Register (GPR13)

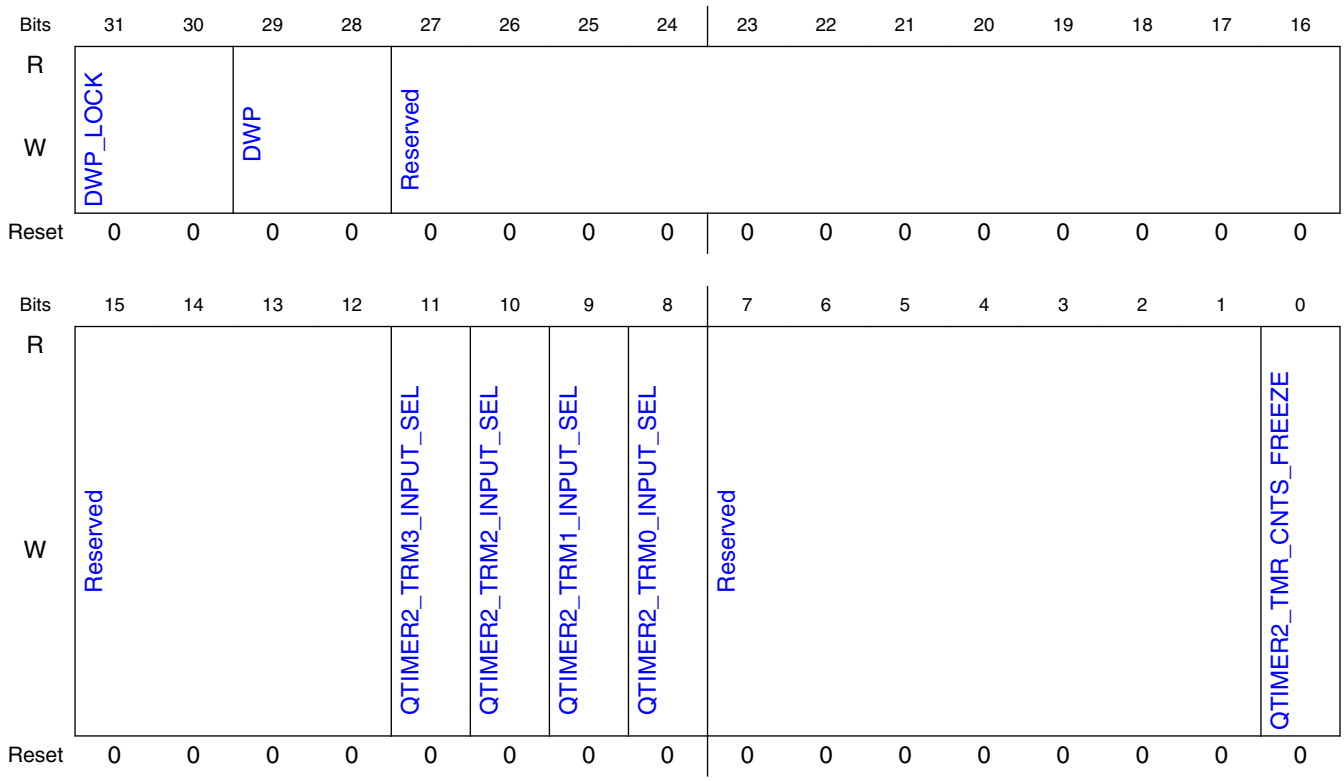
12.4.5.15.1 Offset

Register	Offset
GPR13	34h

12.4.5.15.2 Function

GPR Register

12.4.5.15.3 Diagram



12.4.5.15.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved

Table continues on the next page...

Field	Function
15-12 —	Reserved
11 QTIMER2_TRM3_INPUT_SEL	QTIMER2 TMR3 input select 0: input from IOMUX 1: input from XBAR
10 QTIMER2_TRM2_INPUT_SEL	QTIMER2 TMR2 input select 0: input from IOMUX 1: input from XBAR
9 QTIMER2_TRM1_INPUT_SEL	QTIMER2 TMR1 input select 0: input from IOMUX 1: input from XBAR
8 QTIMER2_TRM0_INPUT_SEL	QTIMER2 TMR0 input select 0: input from IOMUX 1: input from XBAR
7-1 —	Reserved
0 QTIMER2_TMR_CNTS_FREEZE	QTIMER2 timer counter freeze Setting this bit resets counters and output pins of QTIMER2. It is not self-clearing. 0: Timer counter works normally 1 Reset counter and ouput flags

12.4.5.16 GPR14 General Purpose Register (GPR14)

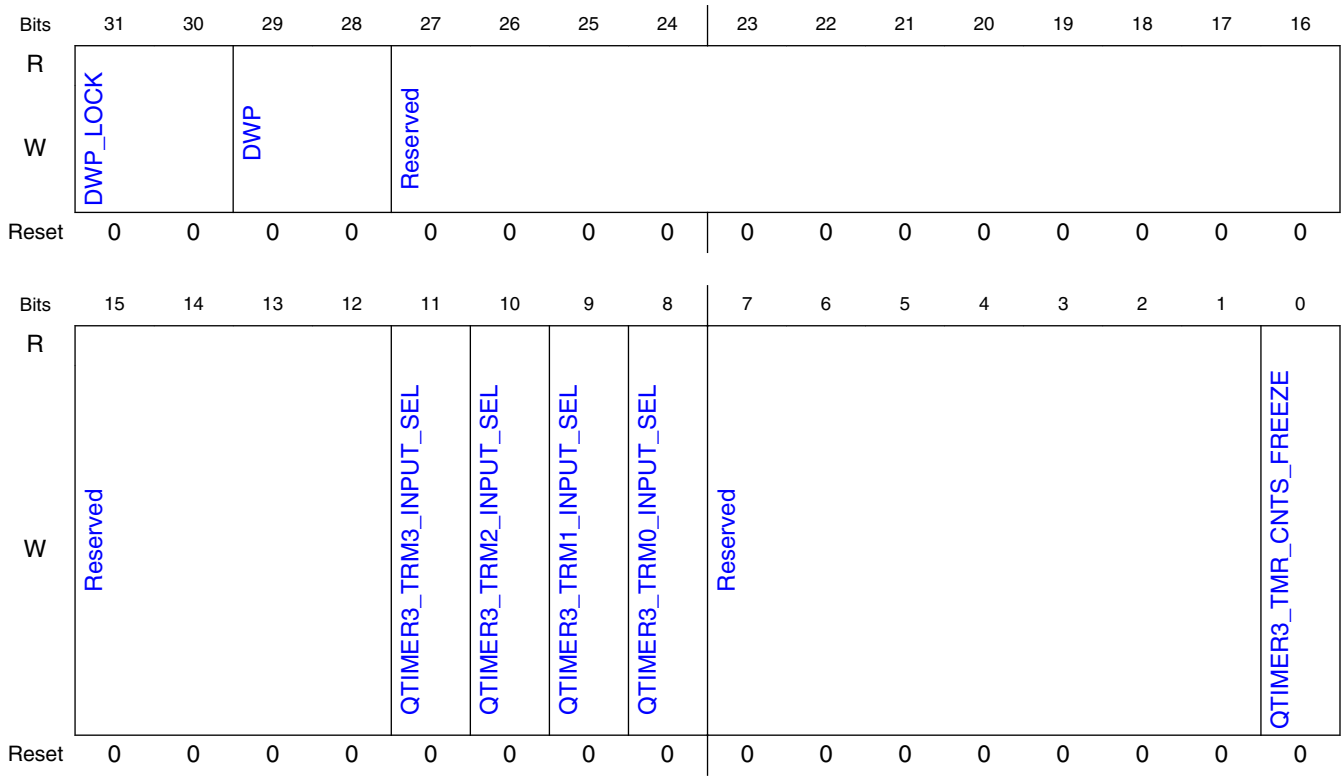
12.4.5.16.1 Offset

Register	Offset
GPR14	38h

12.4.5.16.2 Function

GPR Register

12.4.5.16.3 Diagram



12.4.5.16.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-12 —	Reserved
11	QTIMER3 TMR3 input select

Table continues on the next page...

Field	Function
QTIMER3_TRM3_INPUT_SEL	0: input from IOMUX 1: input from XBAR
10 QTIMER3_TRM2_INPUT_SEL	QTIMER3 TMR2 input select 0: input from IOMUX 1: input from XBAR
9 QTIMER3_TRM1_INPUT_SEL	QTIMER3 TMR1 input select 0: input from IOMUX 1: input from XBAR
8 QTIMER3_TRM0_INPUT_SEL	QTIMER3 TMR0 input select 0: input from IOMUX 1: input from XBAR
7-1 —	Reserved
0 QTIMER3_TMR_CNTS_FREEZE	QTIMER3 timer counter freeze Setting this bit resets counters and output pins of QTIMER3. It is not self-clearing. 0: Timer counter works normally 1 Reset counter and ouput flags

12.4.5.17 GPR15 General Purpose Register (GPR15)

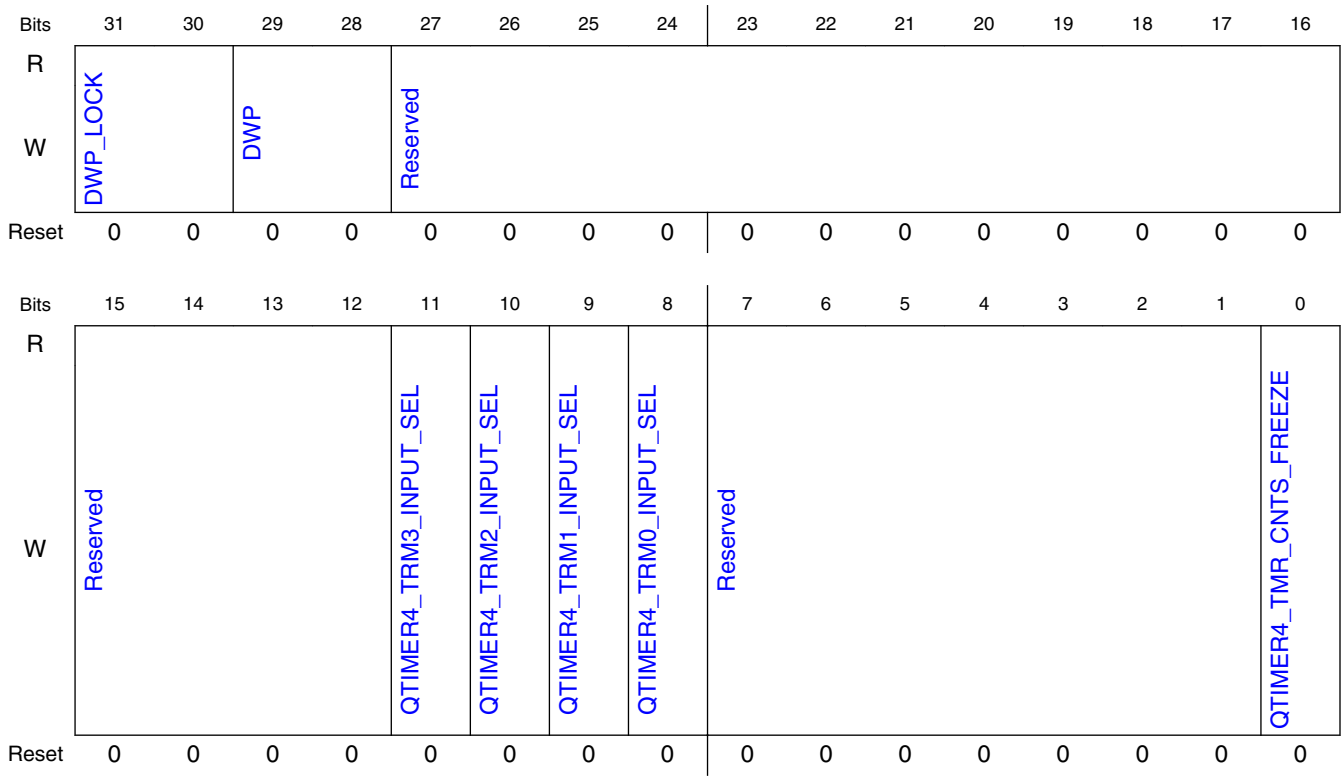
12.4.5.17.1 Offset

Register	Offset
GPR15	3Ch

12.4.5.17.2 Function

GPR Register

12.4.5.17.3 Diagram



12.4.5.17.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-12 —	Reserved
11	QTIMER4 TMR3 input select

Table continues on the next page...

Field	Function
QTIMER4_TRM3_INPUT_SEL	0: input from IOMUX 1: input from XBAR
10 QTIMER4_TRM2_INPUT_SEL	QTIMER4 TMR2 input select 0: input from IOMUX 1: input from XBAR
9 QTIMER4_TRM1_INPUT_SEL	QTIMER4 TMR1 input select 0: input from IOMUX 1: input from XBAR
8 QTIMER4_TRM0_INPUT_SEL	QTIMER4 TMR0 input select 0: input from IOMUX 1: input from XBAR
7-1 —	Reserved
0 QTIMER4_TMR_CNTS_FREEZE	QTIMER4 timer counter freeze Setting this bit resets counters and output pins of QTIMER1. It is not self-clearing. 0: Timer counter works normally 1 Reset counter and ouput flags

12.4.5.18 GPR16 General Purpose Register (GPR16)

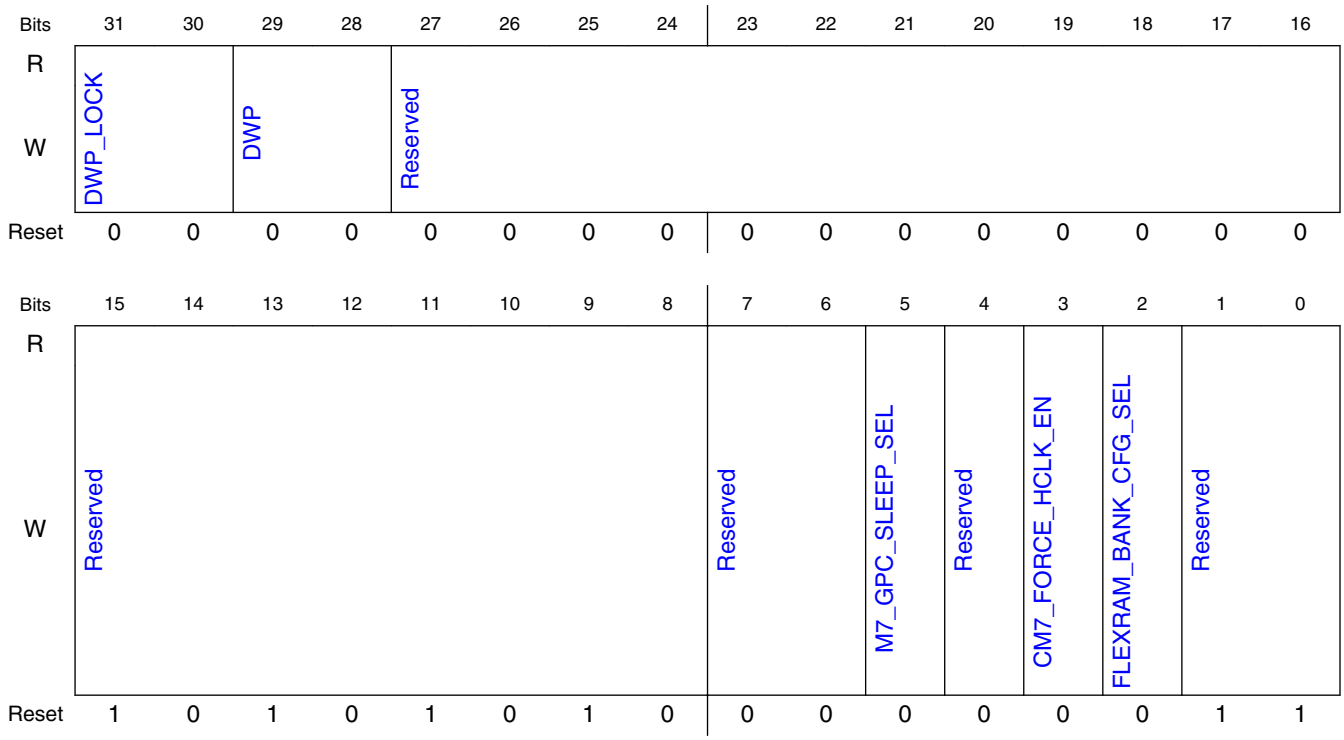
12.4.5.18.1 Offset

Register	Offset
GPR16	40h

12.4.5.18.2 Function

GPR Register

12.4.5.18.3 Diagram



12.4.5.18.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-8 —	Reserved
7-6 —	Reserved

Table continues on the next page...

Field	Function
5 M7_GPC_SLEEP_SEL	CM7 sleep request selection This bit controls which kind of CM7 sleep request is sent to GPC to start sleep sequence. 0: CM7 SLEEPDEEP is sent to GPC 1: CM7 SLEEPING is sent to GPC
4 —	Reserved
3 CM7_FORCE_HCLK_EN	CM7 platform AHB clock enable This bitfield determines whether or not the AHB clock is running when CM7 is sleeping. If the AHB clock is not enabled with this bit, the TCM is not accessible. 0: AHB clock is not running (gated) when CM7 is sleeping and TCM is not accessible 1 AHB clock is running (enabled) when CM7 is sleeping and TCM is accessible
2 FLEXRAM_BANK_CFG_SEL	FlexRAM bank config source select 0: use fuse value to configure 1: use FLEXRAM_BANK_CFG to configure
1-0 —	Reserved

12.4.5.19 GPR17 General Purpose Register (GPR17)

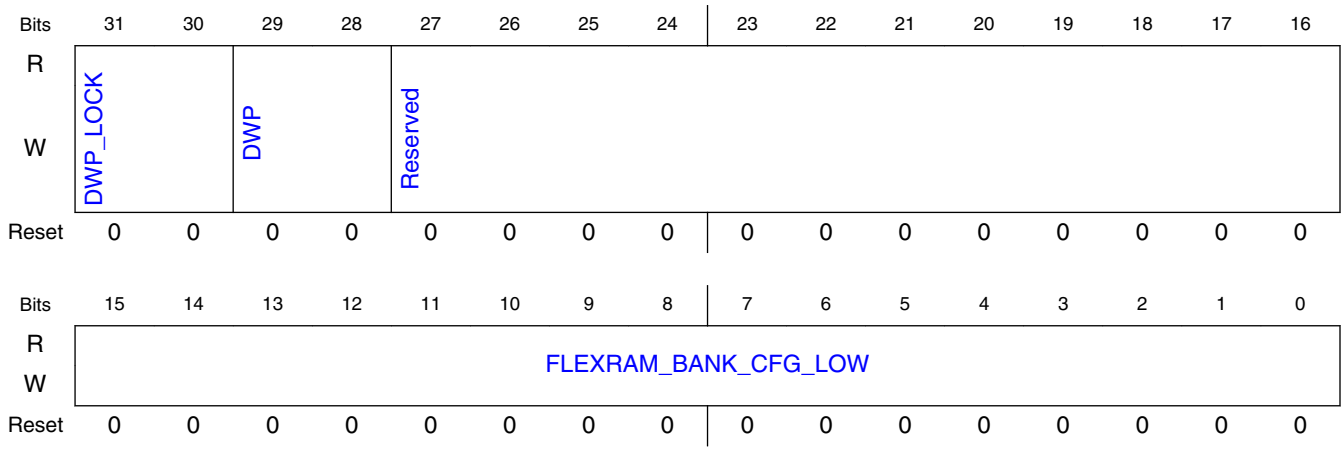
12.4.5.19.1 Offset

Register	Offset
GPR17	44h

12.4.5.19.2 Function

GPR Register

12.4.5.19.3 Diagram



12.4.5.19.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 FLEXRAM_BANK_CFG_LOW	FlexRAM bank config value GPR_FLEXRAM_BANK_CFG[2n+1 : 2n], where n = 0, 1, ..., 7 00: RAM bank n is not used 01: RAM bank n is OCRAM 10: RAM bank n is DTCM 11: RAM bank n is ITCM

12.4.5.20 GPR18 General Purpose Register (GPR18)

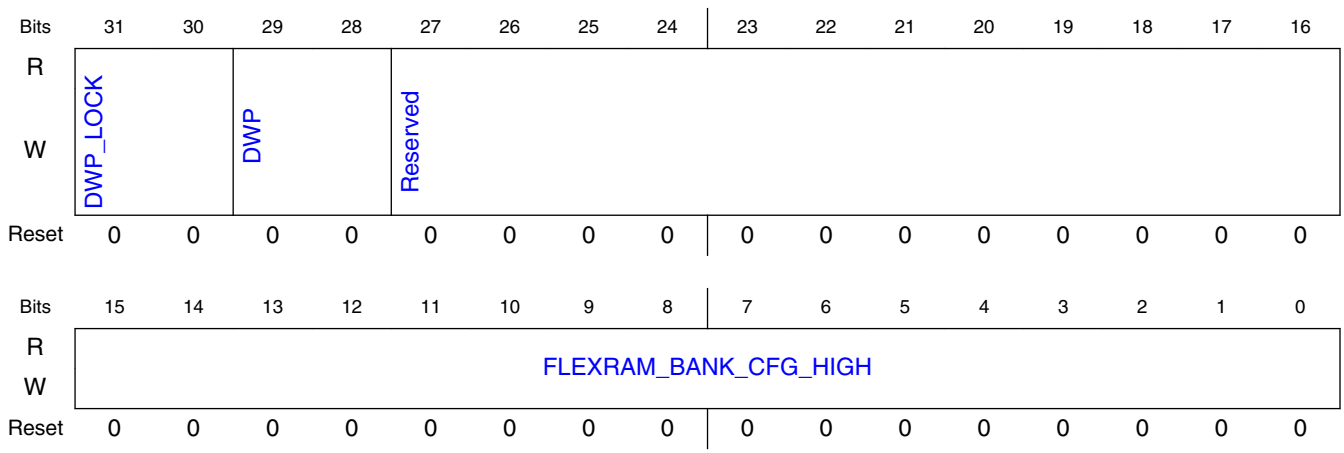
12.4.5.20.1 Offset

Register	Offset
GPR18	48h

12.4.5.20.2 Function

GPR Register

12.4.5.20.3 Diagram



12.4.5.20.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0	FlexRAM bank config value

Field	Function
FLEXRAM_BANK_CFG_HIGH	GPR_FLEXRAM_BANK_CFG[2n+1 : 2n], where n = 8, 9, ..., 15 00: RAM bank n is not used 01: RAM bank n is OCRAM 10: RAM bank n is DTCM 11: RAM bank n is ITCM

12.4.5.21 GPR20 General Purpose Register (GPR20)

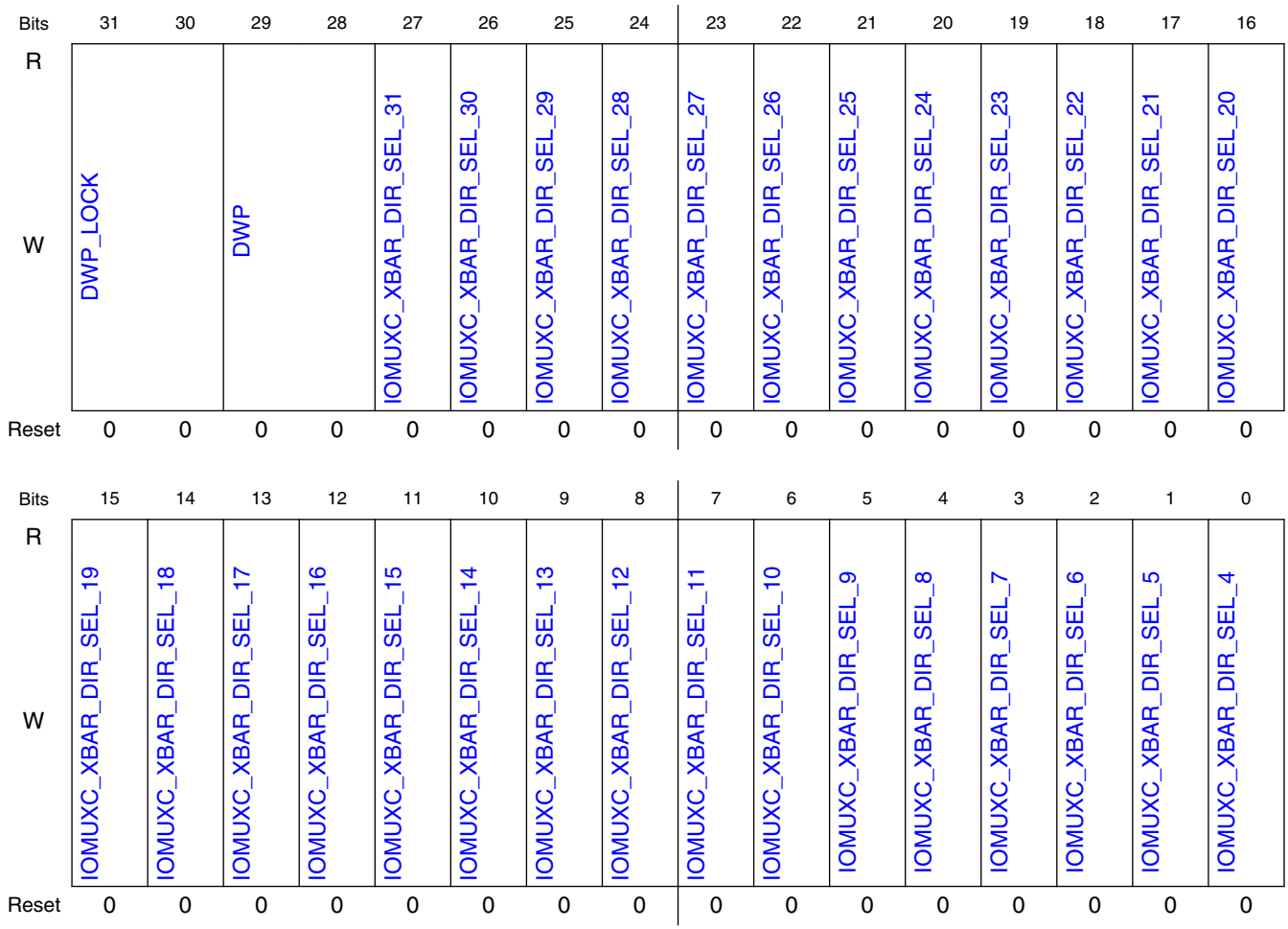
12.4.5.21.1 Offset

Register	Offset
GPR20	50h

12.4.5.21.2 Function

GPR Register

12.4.5.21.3 Diagram



12.4.5.21.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27	IOMUXC XBAR_INOUT31 function direction select

Table continues on the next page...

Field	Function
IOMUXC_XBAR_DIR_SEL_31	0: XBAR_INOUT as input 1: XBAR_INOUT as output
26 IOMUXC_XBAR_DIR_SEL_30	IOMUXC XBAR_INOUT30 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
25 IOMUXC_XBAR_DIR_SEL_29	IOMUXC XBAR_INOUT29 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
24 IOMUXC_XBAR_DIR_SEL_28	IOMUXC XBAR_INOUT28 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
23 IOMUXC_XBAR_DIR_SEL_27	IOMUXC XBAR_INOUT27 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
22 IOMUXC_XBAR_DIR_SEL_26	IOMUXC XBAR_INOUT26 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
21 IOMUXC_XBAR_DIR_SEL_25	IOMUXC XBAR_INOUT25 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
20 IOMUXC_XBAR_DIR_SEL_24	IOMUXC XBAR_INOUT24 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
19 IOMUXC_XBAR_DIR_SEL_23	IOMUXC XBAR_INOUT23 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
18 IOMUXC_XBAR_DIR_SEL_22	IOMUXC XBAR_INOUT22 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
17 IOMUXC_XBAR_DIR_SEL_21	IOMUXC XBAR_INOUT21 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
16 IOMUXC_XBAR_DIR_SEL_20	IOMUXC XBAR_INOUT20 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
15 IOMUXC_XBAR_DIR_SEL_19	IOMUXC XBAR_INOUT19 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
14	IOMUXC XBAR_INOUT18 function direction select

Table continues on the next page...

Field	Function
IOMUXC_XBAR_DIR_SEL_18	0: XBAR_INOUT as input 1: XBAR_INOUT as output
13 IOMUXC_XBAR_DIR_SEL_17	IOMUXC XBAR_INOUT17 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
12 IOMUXC_XBAR_DIR_SEL_16	IOMUXC XBAR_INOUT16 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
11 IOMUXC_XBAR_DIR_SEL_15	IOMUXC XBAR_INOUT15 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
10 IOMUXC_XBAR_DIR_SEL_14	IOMUXC XBAR_INOUT14 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
9 IOMUXC_XBAR_DIR_SEL_13	IOMUXC XBAR_INOUT13 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
8 IOMUXC_XBAR_DIR_SEL_12	IOMUXC XBAR_INOUT12 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
7 IOMUXC_XBAR_DIR_SEL_11	IOMUXC XBAR_INOUT11 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
6 IOMUXC_XBAR_DIR_SEL_10	IOMUXC XBAR_INOUT10 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
5 IOMUXC_XBAR_DIR_SEL_9	IOMUXC XBAR_INOUT9 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
4 IOMUXC_XBAR_DIR_SEL_8	IOMUXC XBAR_INOUT8 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
3 IOMUXC_XBAR_DIR_SEL_7	IOMUXC XBAR_INOUT7 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
2 IOMUXC_XBAR_DIR_SEL_6	IOMUXC XBAR_INOUT6 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
1	IOMUXC XBAR_INOUT5 function direction select

Table continues on the next page...

Field	Function
IOMUXC_XBAR_DIR_SEL_5	0: XBAR_INOUT as input 1: XBAR_INOUT as output
0 IOMUXC_XBAR_DIR_SEL_4	IOMUXC XBAR_INOUT4 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output

12.4.5.22 GPR21 General Purpose Register (GPR21)

12.4.5.22.1 Offset

Register	Offset
GPR21	54h

12.4.5.22.2 Function

GPR Register

12.4.5.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWP_LOCK		DWP		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					IOMUXC_XBAR_DIR_SEL_42	IOMUXC_XBAR_DIR_SEL_41	IOMUXC_XBAR_DIR_SEL_40	IOMUXC_XBAR_DIR_SEL_39	IOMUXC_XBAR_DIR_SEL_38	IOMUXC_XBAR_DIR_SEL_37	IOMUXC_XBAR_DIR_SEL_36	IOMUXC_XBAR_DIR_SEL_35	IOMUXC_XBAR_DIR_SEL_34	IOMUXC_XBAR_DIR_SEL_33	IOMUXC_XBAR_DIR_SEL_32
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.5.22.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-11 —	Reserved
10 IOMUXC_XBAR_DIR_SEL_42	IOMUXC XBAR_INOUT42 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
9 IOMUXC_XBAR_DIR_SEL_41	IOMUXC XBAR_INOUT41 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
8 IOMUXC_XBAR_DIR_SEL_40	IOMUXC XBAR_INOUT40 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
7 IOMUXC_XBAR_DIR_SEL_39	IOMUXC XBAR_INOUT39 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
6 IOMUXC_XBAR_DIR_SEL_38	IOMUXC XBAR_INOUT38 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
5 IOMUXC_XBAR_DIR_SEL_37	IOMUXC XBAR_INOUT37 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
4 IOMUXC_XBAR_DIR_SEL_36	IOMUXC XBAR_INOUT36 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
3	IOMUXC XBAR_INOUT35 function direction select

Table continues on the next page...

Field	Function
IOMUXC_XBAR_DIR_SEL_35	0: XBAR_INOUT as input 1: XBAR_INOUT as output
2 IOMUXC_XBAR_DIR_SEL_34	IOMUXC XBAR_INOUT34 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
1 IOMUXC_XBAR_DIR_SEL_33	IOMUXC XBAR_INOUT33 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output
0 IOMUXC_XBAR_DIR_SEL_32	IOMUXC XBAR_INOUT32 function direction select 0: XBAR_INOUT as input 1: XBAR_INOUT as output

12.4.5.23 GPR22 General Purpose Register (GPR22)

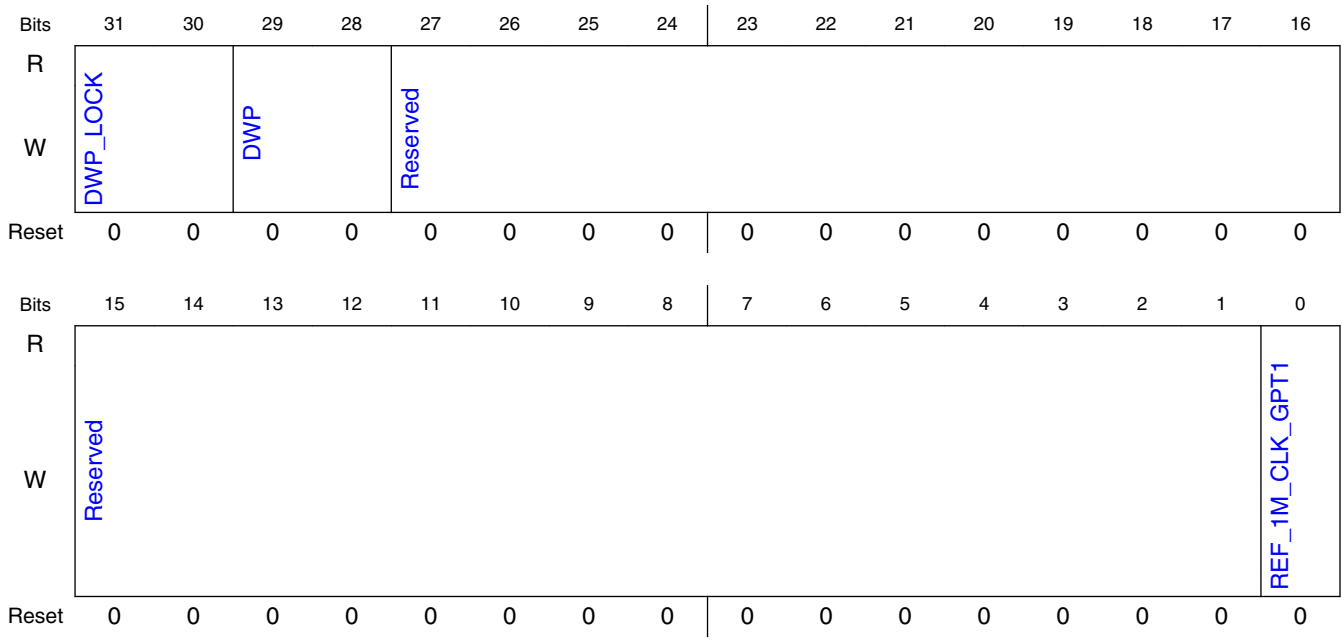
12.4.5.23.1 Offset

Register	Offset
GPR22	58h

12.4.5.23.2 Function

GPR Register

12.4.5.23.3 Diagram



12.4.5.23.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 REF_1M_CLK_GPT1	GPT1 1 MHz clock source select 0: GPT1 ipg_clk_highfreq driven by GPT1_CLK_ROOT 1: GPT1 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.24 GPR23 General Purpose Register (GPR23)

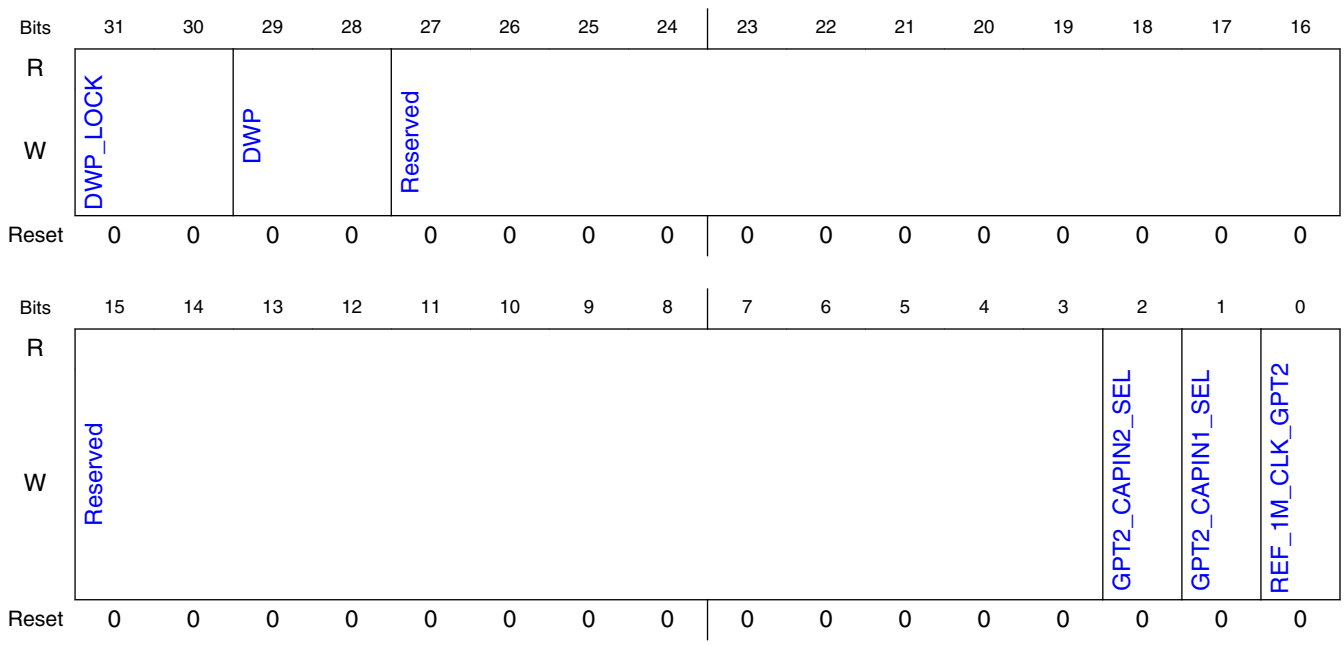
12.4.5.24.1 Offset

Register	Offset
GPR23	5Ch

12.4.5.24.2 Function

GPR Register

12.4.5.24.3 Diagram



12.4.5.24.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked

Table continues on the next page...

Field	Function
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-3 —	Reserved
2 GPT2_CAPIN2_SEL	GPT2 input capture channel 2 source select 0: source from pad 1: source from ENET1G_1588_EVENT1_OUT
1 GPT2_CAPIN1_SEL	GPT2 input capture channel 1 source select 0 source from pad 1 source from ENET_1588_EVENT1_OUT
0 REF_1M_CLK_GPT2	GPT2 1 MHz clock source select 0: GPT2 ipg_clk_highfreq driven by GPT2_CLK_ROOT 1: GPT2 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.25 GPR24 General Purpose Register (GPR24)

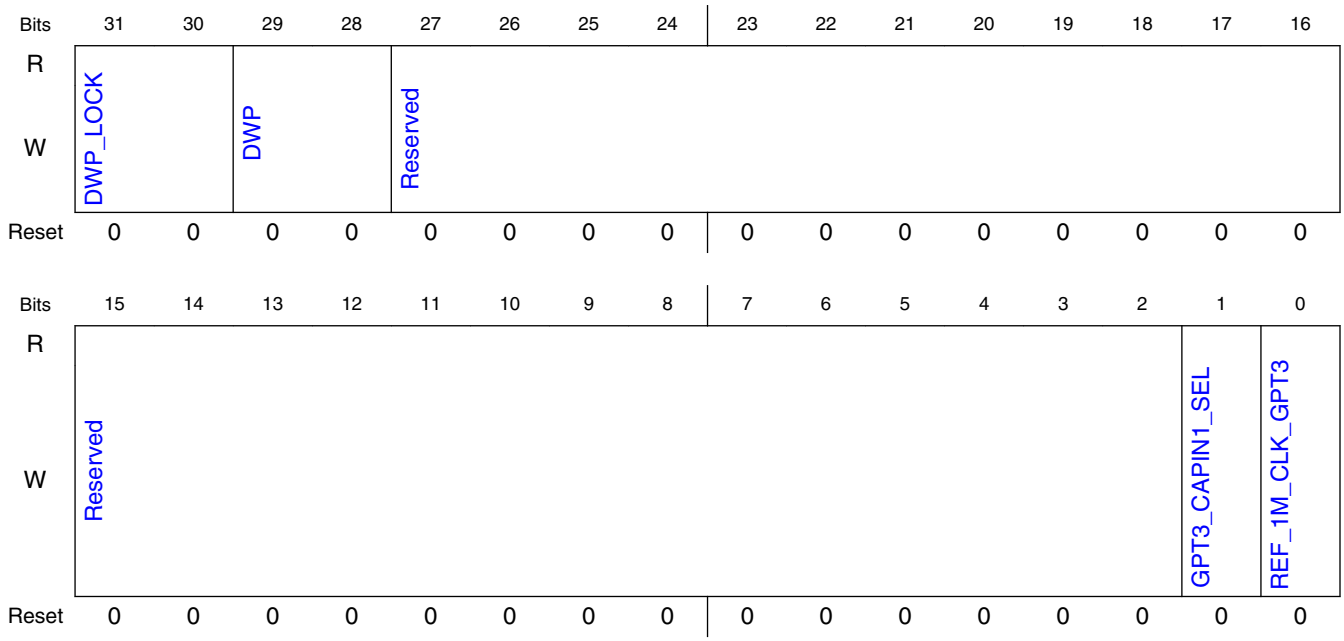
12.4.5.25.1 Offset

Register	Offset
GPR24	60h

12.4.5.25.2 Function

GPR Register

12.4.5.25.3 Diagram



12.4.5.25.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-2 —	Reserved
1 GPT3_CAPIN1_SEL	GPT3 input capture channel 1 source select 0: source from pad 1: source from ENET_QOS ptp_pps_o[1]
0	GPT3 1 MHz clock source select 0: GPT3 ipg_clk_highfreq driven by GPT3_CLK_ROOT

Field	Function
REF_1M_CLK_GPT3	1: GPT3 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.26 GPR25 General Purpose Register (GPR25)

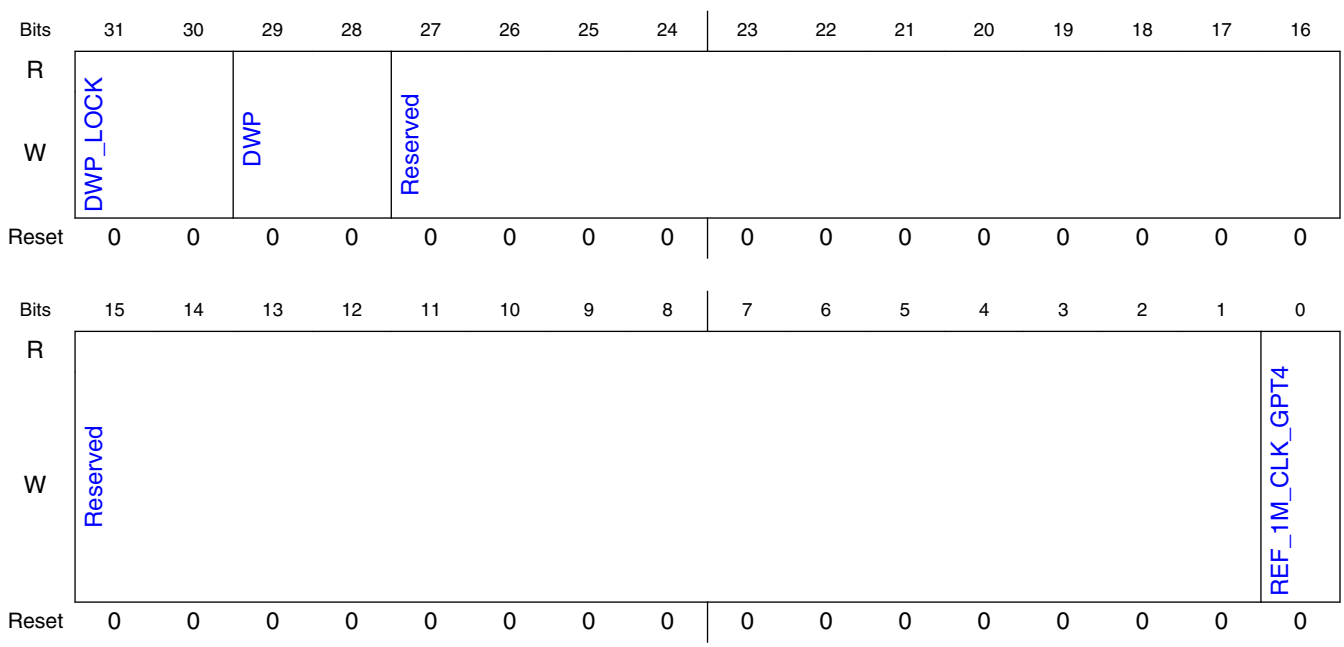
12.4.5.26.1 Offset

Register	Offset
GPR25	64h

12.4.5.26.2 Function

GPR Register

12.4.5.26.3 Diagram



12.4.5.26.4 Fields

Field	Function
31-30	Domain write protection lock

Table continues on the next page...

Field	Function
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 REF_1M_CLK_GPT4	GPT4 1 MHz clock source select 0: GPT4 ipg_clk_highfreq driven by GPT4_CLK_ROOT 1: GPT4 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.27 GPR26 General Purpose Register (GPR26)

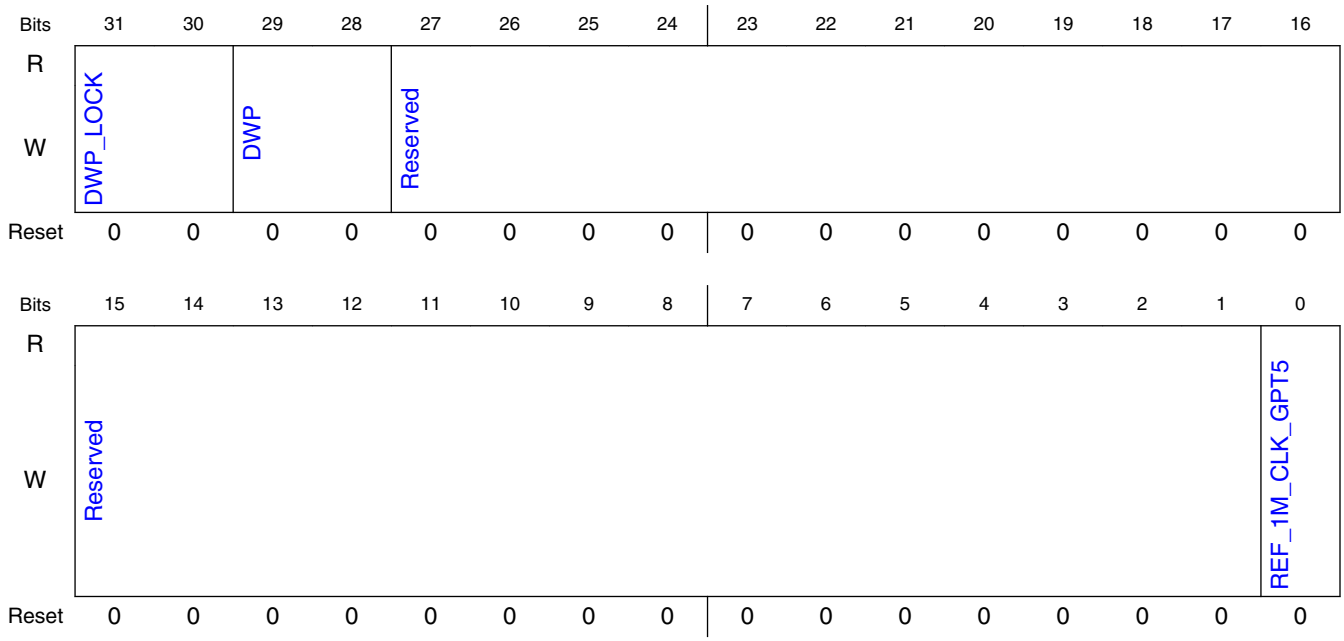
12.4.5.27.1 Offset

Register	Offset
GPR26	68h

12.4.5.27.2 Function

GPR Register

12.4.5.27.3 Diagram



12.4.5.27.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 REF_1M_CLK_GPT5	GPT5 1 MHz clock source select 0: GPT5 ipg_clk_highfreq driven by GPT5_CLK_ROOT 1: GPT5 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.28 GPR27 General Purpose Register (GPR27)

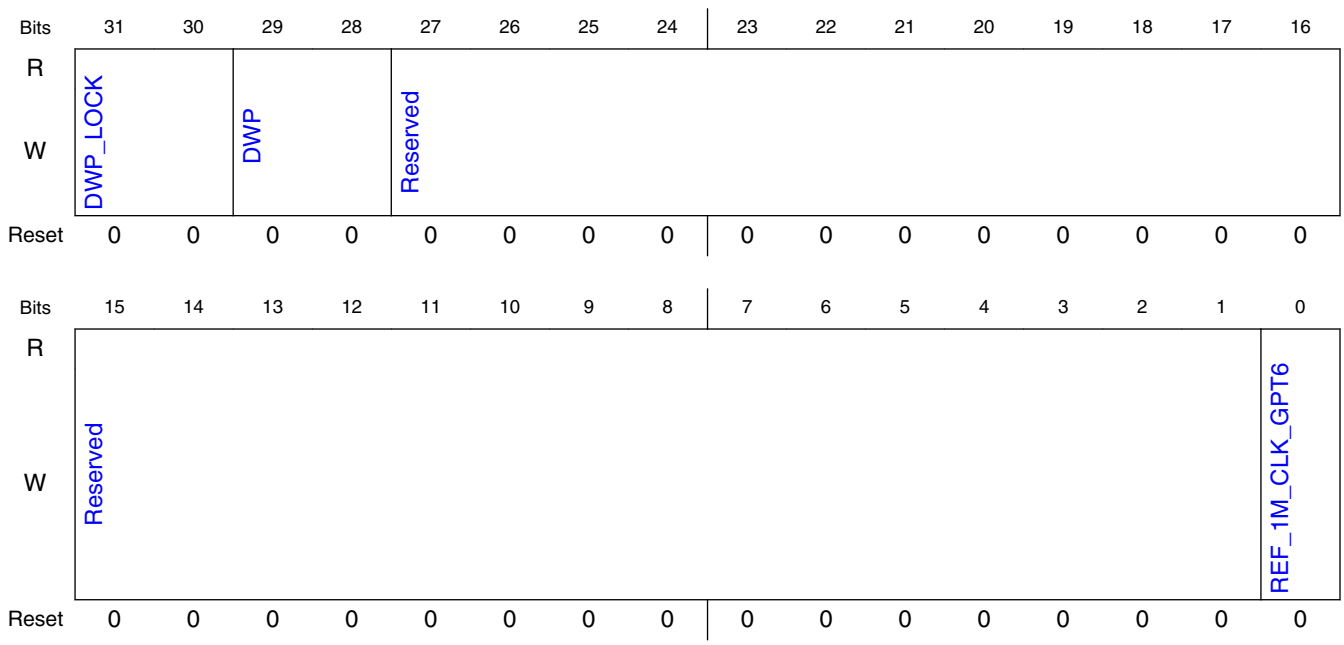
12.4.5.28.1 Offset

Register	Offset
GPR27	6Ch

12.4.5.28.2 Function

GPR Register

12.4.5.28.3 Diagram



12.4.5.28.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked

Table continues on the next page...

Field	Function
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 REF_1M_CLK_GPT6	GPT6 1 MHz clock source select 0: GPT6 ipg_clk_highfreq driven by GPT6_CLK_ROOT 1: GPT6 ipg_clk_highfreq driven by 1 MHz clock derived from RCOSC_400M

12.4.5.29 GPR28 General Purpose Register (GPR28)

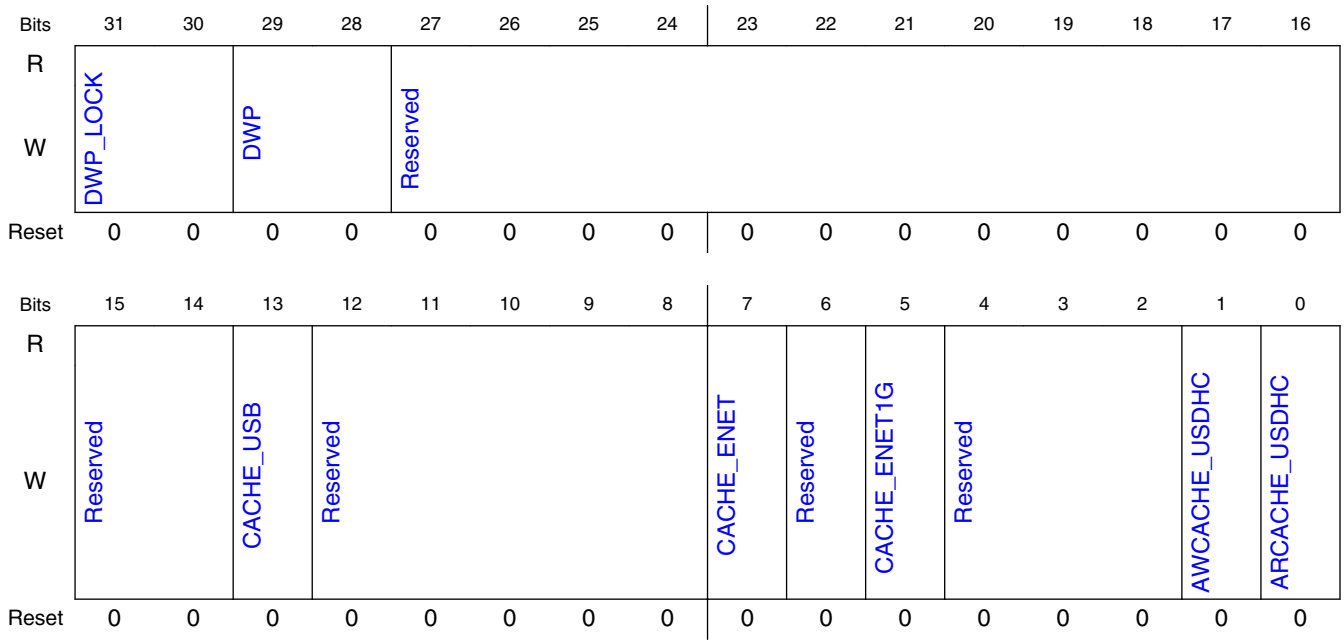
12.4.5.29.1 Offset

Register	Offset
GPR28	70h

12.4.5.29.2 Function

GPR Register

12.4.5.29.3 Diagram



12.4.5.29.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-14 —	Reserved
13 CACHE_USB	USB block cacheable attribute value of AXI transactions If the current transaction is cacheable, the bus infrastructure (NIC) will reorganize the transaction to optimize bus performance. The NIC will combine two 32-bit USB transactions into one 64-bit transaction. 0: Cacheable attribute is off for read/write transactions 1: Cacheable attribute is on for read/write transactions

Table continues on the next page...

Field	Function
12-8 —	Reserved
7 CACHE_ENET	ENET block cacheable attribute value of AXI transactions If the transaction is cacheable, the bus infrastructure (NIC) will reorganize the transaction to optimize bus performance. The NIC will combine two 32-bit ENET transactions into one 64-bit transaction. 0: Cacheable attribute is off for read/write transactions 1: Cacheable attribute is on for read/write transactions
6 —	Reserved
5 CACHE_ENET1G	
4-2 —	Reserved
1 AWCACHE_USDHC	uSDHC block cacheable attribute value of AXI write transactions If the current transaction is cacheable, the bus infrastructure (NIC) will reorganize the transaction to optimize bus performance. The NIC will combine two 32-bit uSDHC transactions into one 64-bit transaction. 0: Cacheable attribute is off for write transactions 1: Cacheable attribute is on for write transactions
0 ARCACHE_USDHC	uSDHC block cacheable attribute value of AXI read transactions If the current transaction is cacheable, the bus infrastructure (NIC) will reorganize the transaction to optimize bus performance. The NIC will combine two 32-bit uSDHC transactions into one 64-bit transaction. 0: Cacheable attribute is off for read transactions 1: Cacheable attribute is on for read transactions

12.4.5.30 GPR29 General Purpose Register (GPR29)

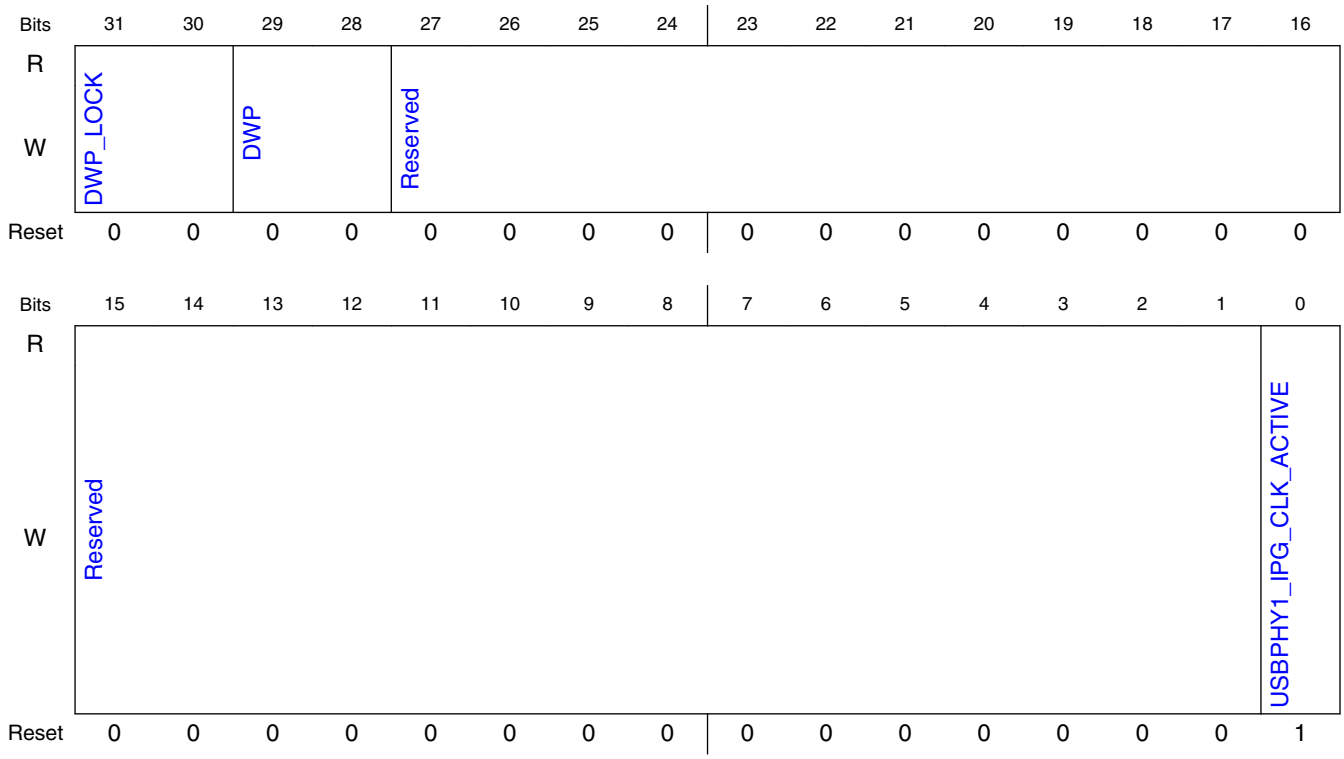
12.4.5.30.1 Offset

Register	Offset
GPR29	74h

12.4.5.30.2 Function

GPR Register

12.4.5.30.3 Diagram



12.4.5.30.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 USBPHY1_IPG_CLK_ACTIVE	USBPHY1 register access clock enable Clearing this bit to 0 will stop USBPHY1 register access clock to save power.

12.4.5.31 GPR30 General Purpose Register (GPR30)

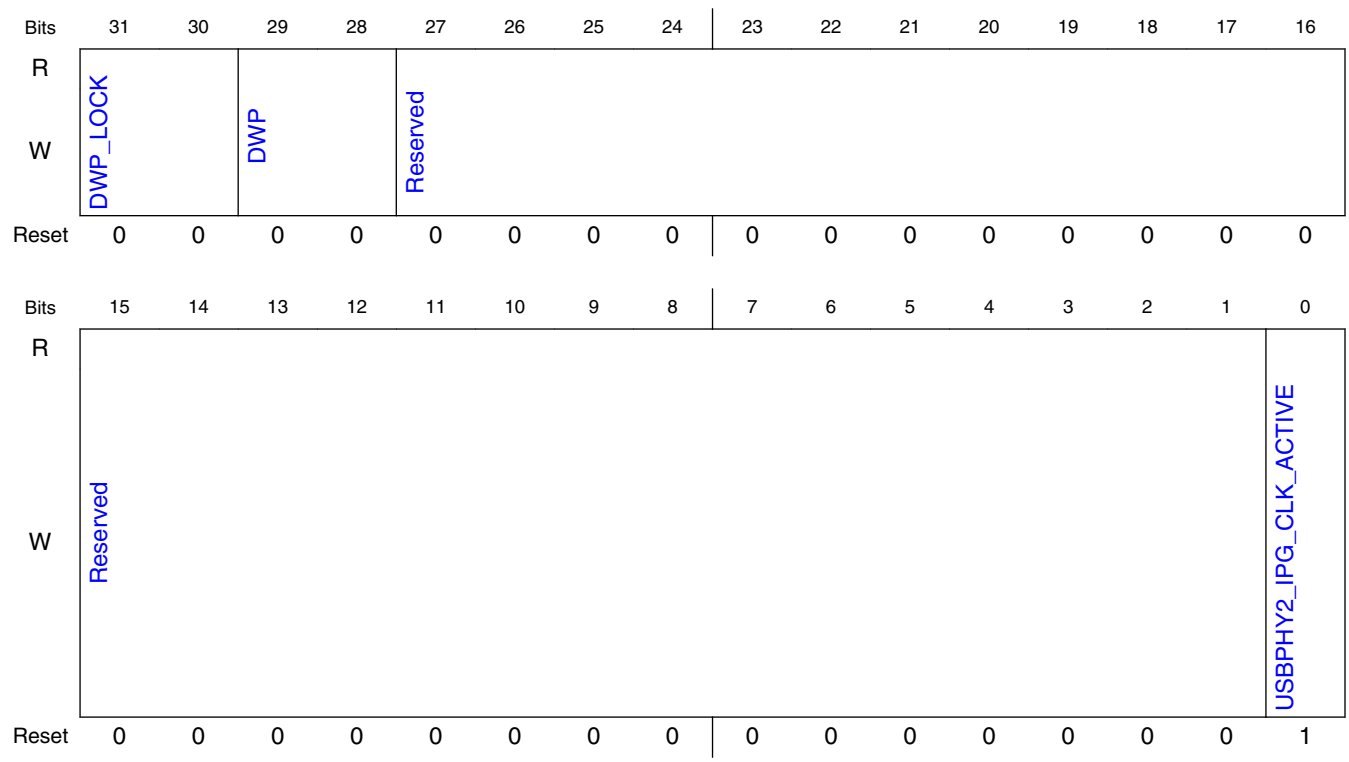
12.4.5.31.1 Offset

Register	Offset
GPR30	78h

12.4.5.31.2 Function

GPR Register

12.4.5.31.3 Diagram



12.4.5.31.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked.

Table continues on the next page...

Field	Function
	00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 USBPHY2_IPG_CLK_ACTIVE	USBPHY2 register access clock enable Clearing this bit to 0 will stop USBPHY2 register access clock to save power.

12.4.5.32 GPR31 General Purpose Register (GPR31)

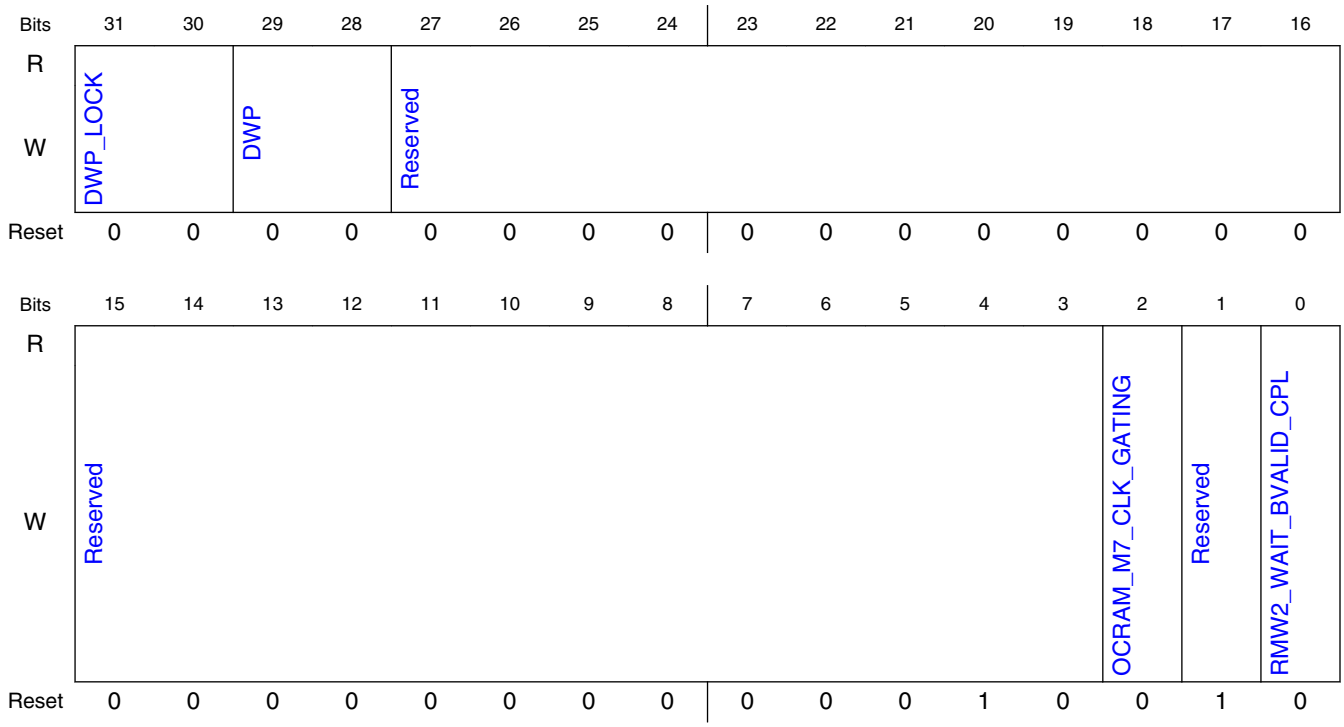
12.4.5.32.1 Offset

Register	Offset
GPR31	7Ch

12.4.5.32.2 Function

GPR Register.

12.4.5.32.3 Diagram



12.4.5.32.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-3 —	Reserved
2 OCRAM_M7_CLK_GATING	OCRAM M7 clock gating enable Set this bit to 1 to enable auto clock gating for OCRAM M7, which is the OCRAM of FlexRAM. When this bit is high, and no access on OCRAM of FlexRAM, the OCRAM clock is gated off.

Table continues on the next page...

Field	Function
1 —	Reserved
0 RMW2_WAIT_BVALID_CPL	OCRAM M7 RMW wait enable If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.33 GPR32 General Purpose Register (GPR32)

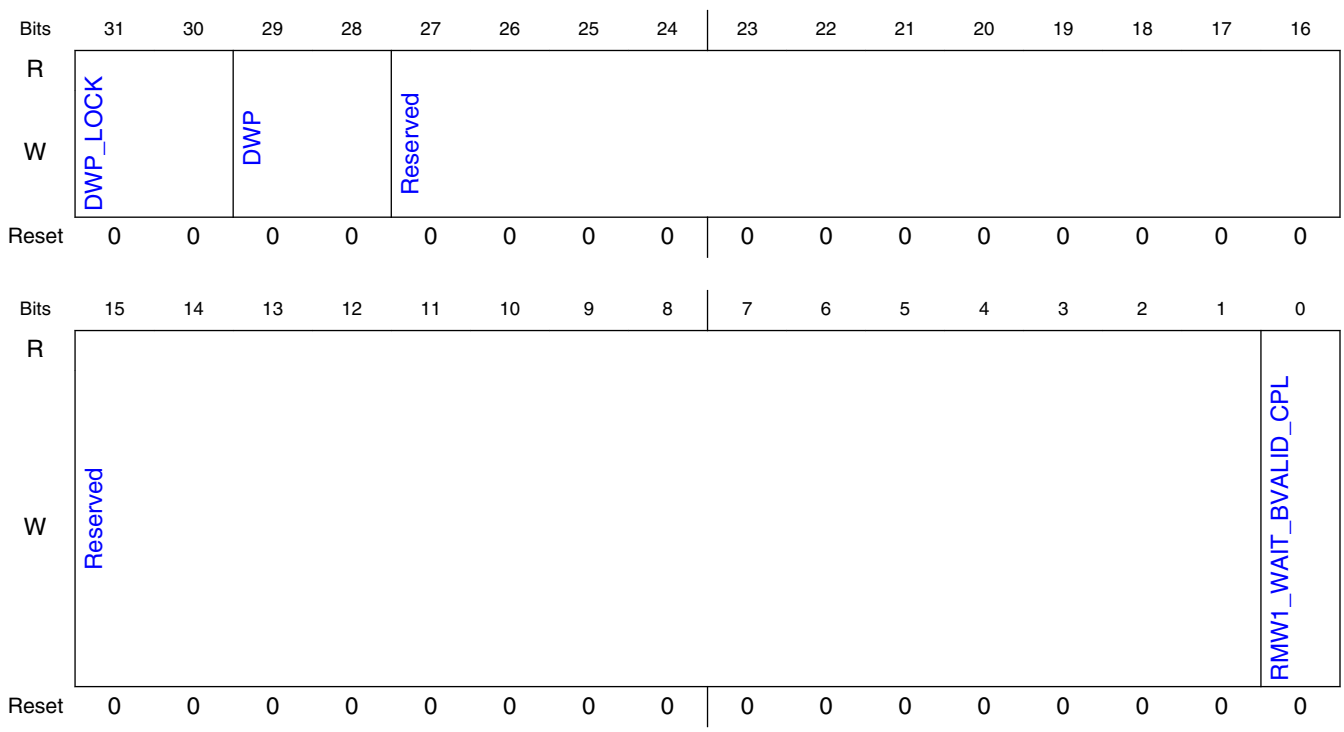
12.4.5.33.1 Offset

Register	Offset
GPR32	80h

12.4.5.33.2 Function

GPR Register

12.4.5.33.3 Diagram



12.4.5.33.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 RMW1_WAIT_BVALID_CPL	OCRAM1 RMW wait enable If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.34 GPR33 General Purpose Register (GPR33)

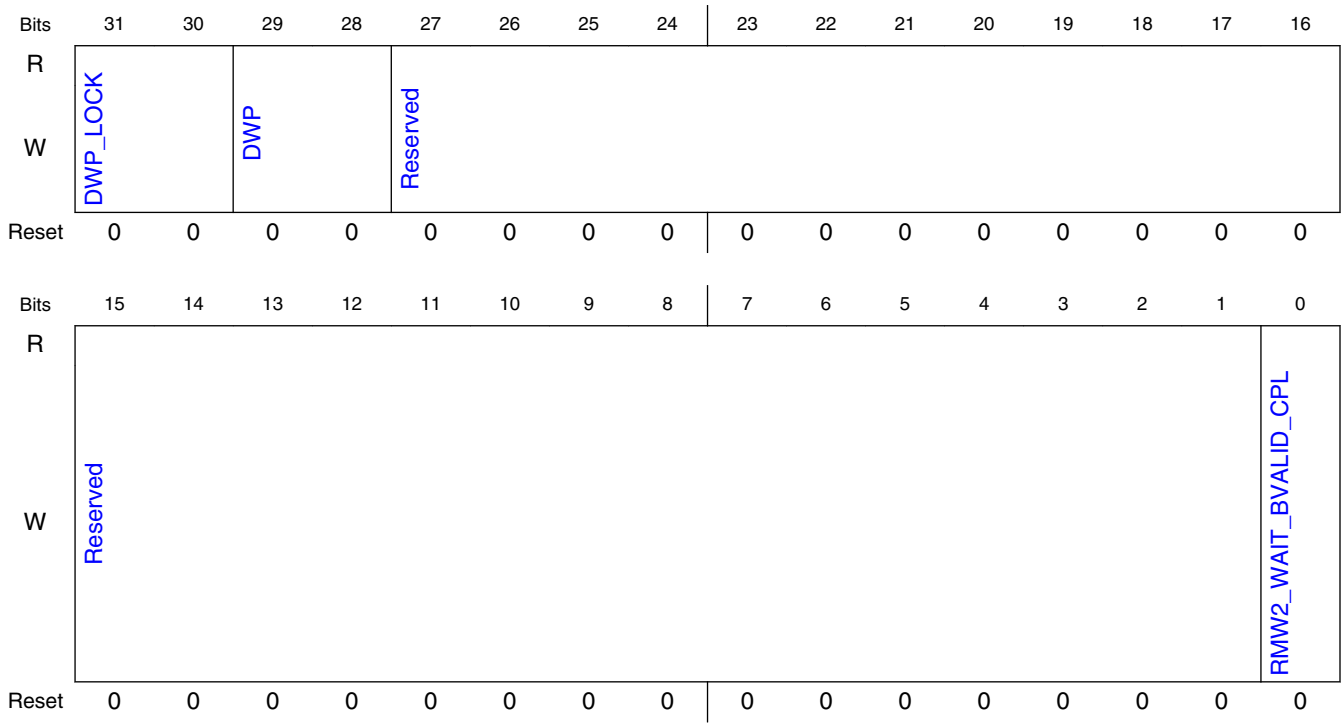
12.4.5.34.1 Offset

Register	Offset
GPR33	84h

12.4.5.34.2 Function

GPR Register

12.4.5.34.3 Diagram



12.4.5.34.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0 RMW2_WAIT_BVALID_CPL	OCRAM2 RMW wait enable If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.35 GPR34 General Purpose Register (GPR34)

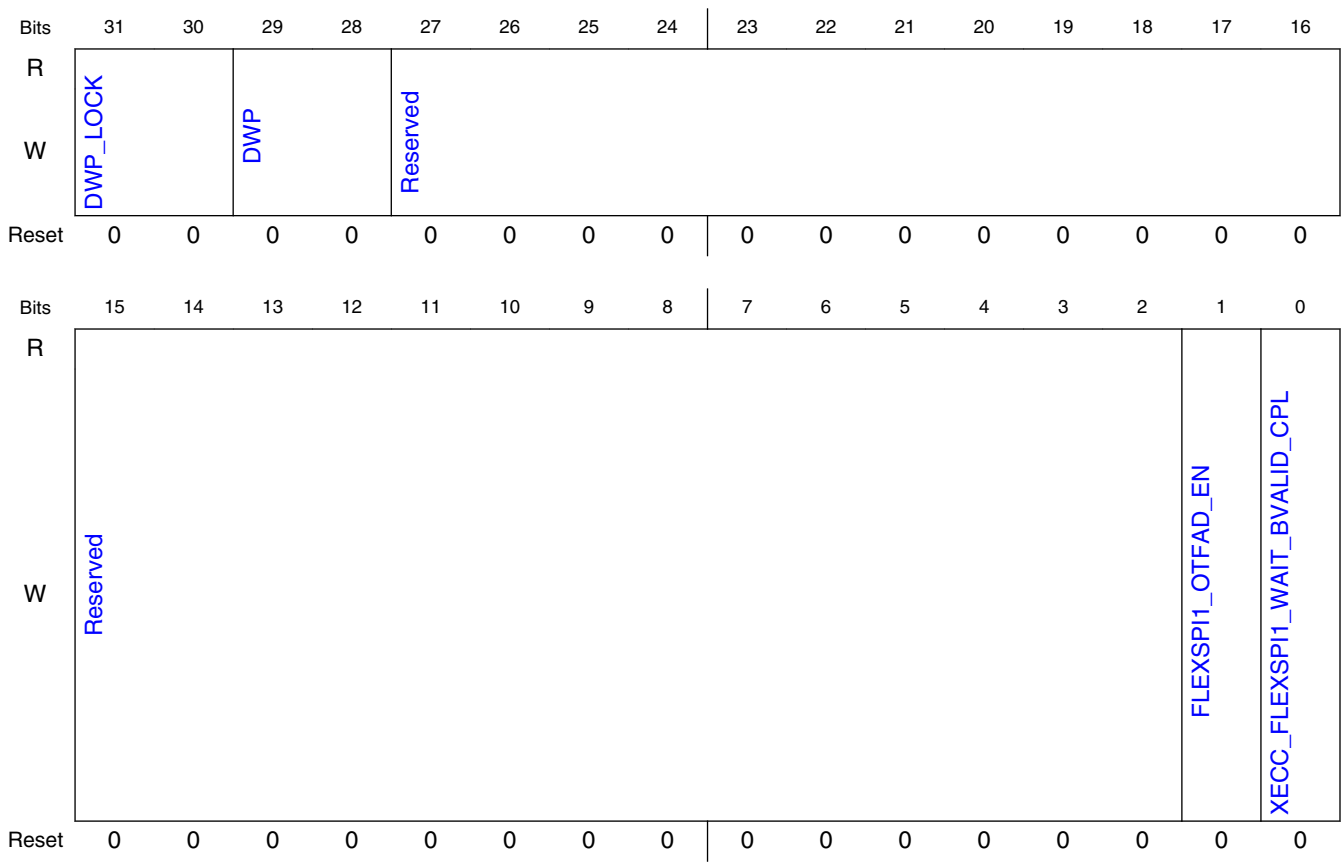
12.4.5.35.1 Offset

Register	Offset
GPR34	88h

12.4.5.35.2 Function

GPR Register

12.4.5.35.3 Diagram



12.4.5.35.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-2 —	Reserved
1 FLEXSPI1_OTFAD_EN	FlexSPI1 OTFAD enable This bit enables the OTFAD operation when it's set to 1. It has the same effect as OTFAD_CR[GE] bit.
0 XECC_FLEXSPI1_WAIT_BVALID_CPL	XECC_FLEXSPI1 RMW wait enable If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.36 GPR35 General Purpose Register (GPR35)

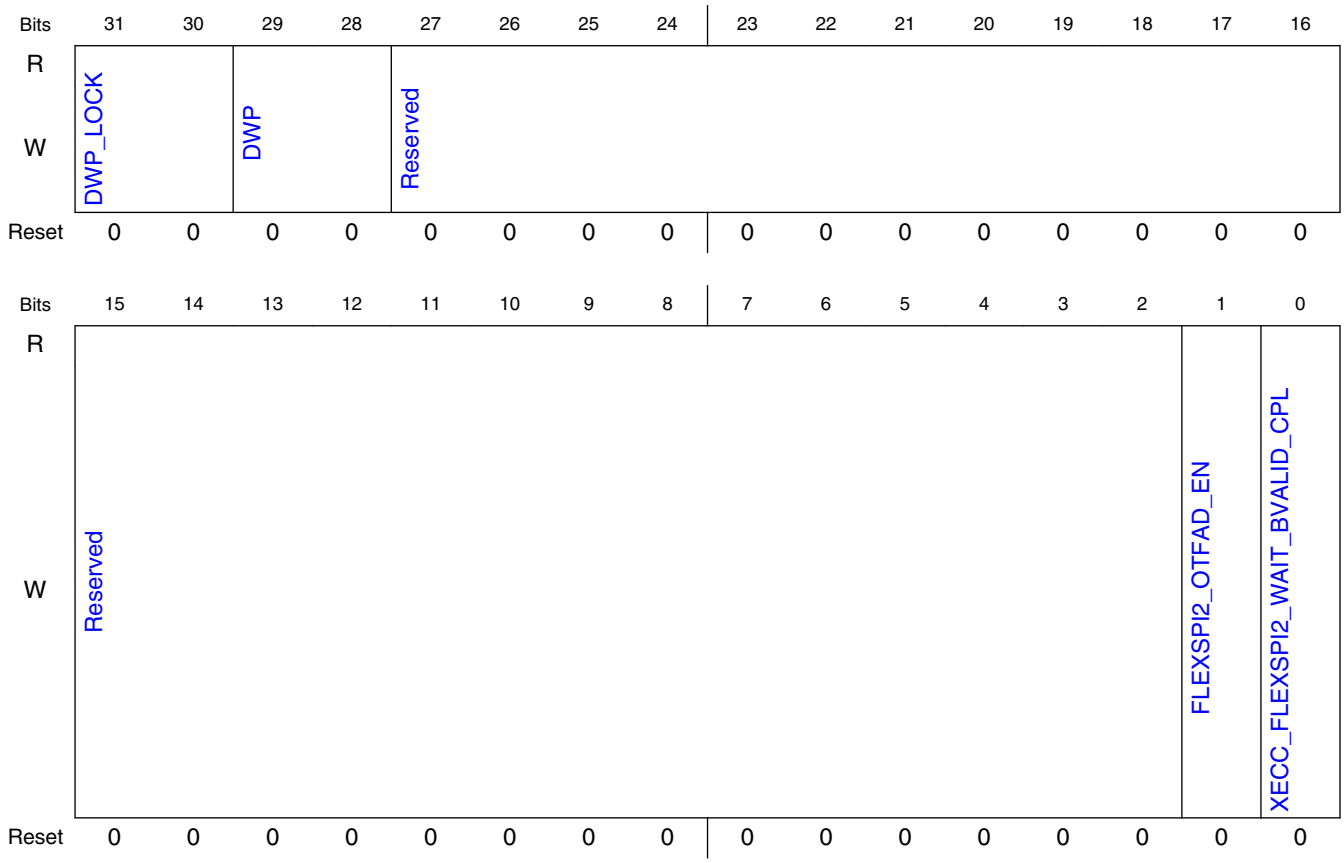
12.4.5.36.1 Offset

Register	Offset
GPR35	8Ch

12.4.5.36.2 Function

GPR Register

12.4.5.36.3 Diagram



12.4.5.36.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-2	Reserved

Table continues on the next page...

Field	Function
—	
1 FLEXSPI2_OTFAD_EN	FlexSPI2 OTFAD enable This bit enables the OTFAD operation when it's set to 1. It has the same effect as OTFAD_CR[GE] bit.
0 XECC_FLEXSPI2_WAIT_BVALID_CPL	XECC_FLEXSPI2 RMW wait enable If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.37 GPR36 General Purpose Register (GPR36)

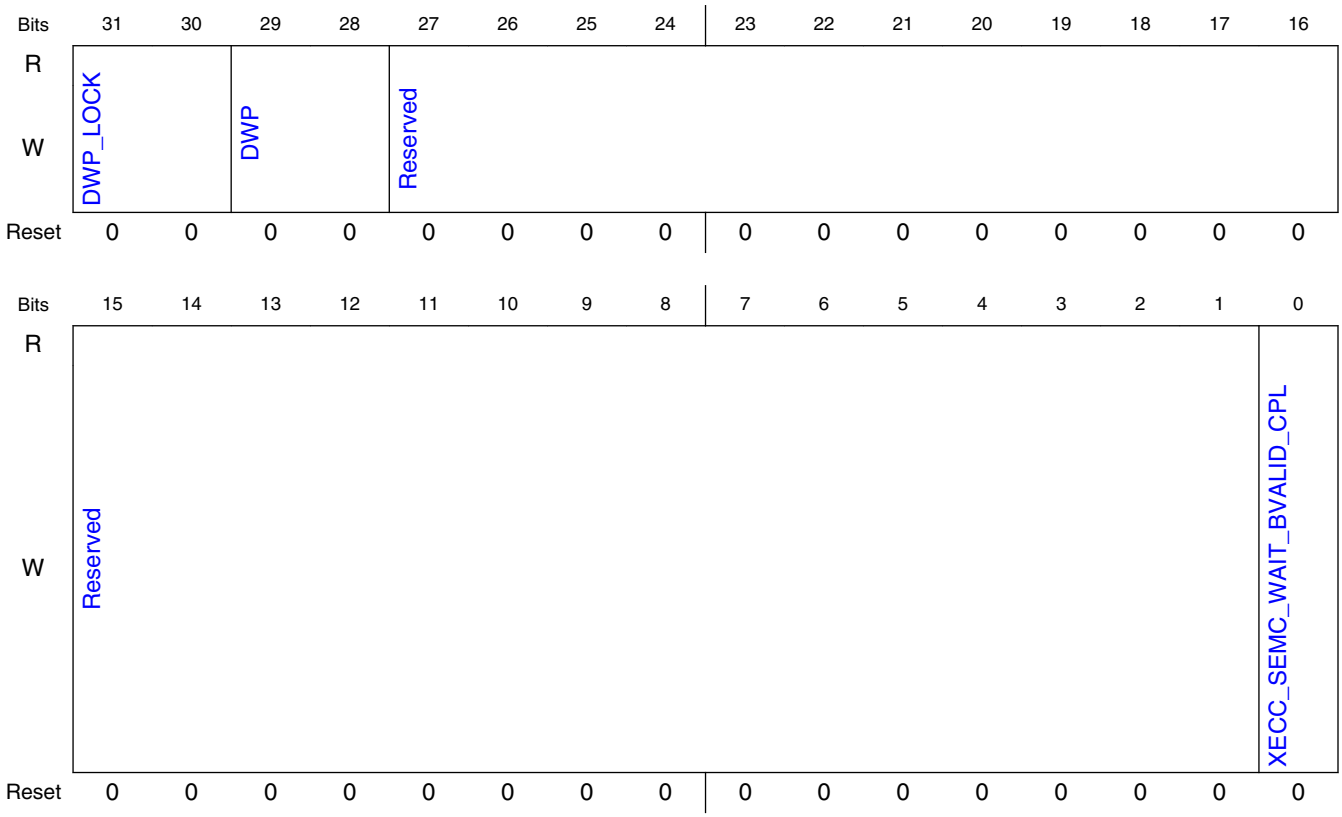
12.4.5.37.1 Offset

Register	Offset
GPR36	90h

12.4.5.37.2 Function

GPR Register

12.4.5.37.3 Diagram



12.4.5.37.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-1 —	Reserved
0	XECC_SEMC RMW wait enable

Field	Function
XECC_SEMC_WAIT_BVALID_CPL	If this bit is set to 1, RMW will write back next data only after current write response is received. It only affects the speed of data transfer.

12.4.5.38 GPR37 General Purpose Register (GPR37)

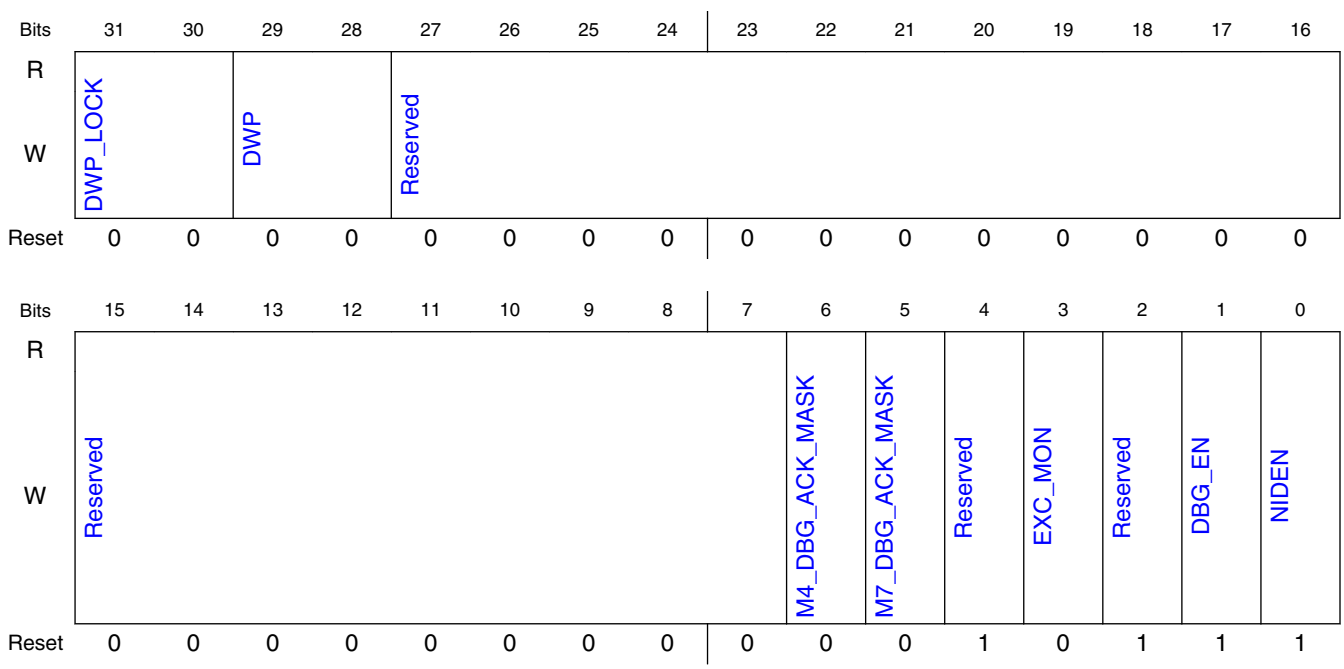
12.4.5.38.1 Offset

Register	Offset
GPR37	94h

12.4.5.38.2 Function

GPR Register

12.4.5.38.3 Diagram



12.4.5.38.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-7 —	Reserved
6 M4_DBG_ACK_MASK	CM4 debug halt mask Once set to 1, peripheral won't be halted when CM4 is halted by debugger.
5 M7_DBG_ACK_MASK	CM7 debug halt mask Once set to 1, peripheral won't be halted when CM7 is halted by debugger.
4 —	Reserved
3 EXC_MON	Exclusive monitor response select of illegal command This bit sets the bus response value when CM7 or CM4 exclusive access fails. 0: OKAY response 1: SLVError response
2 —	Reserved
1 DBG_EN	ARM invasive debug enable Invasive debug is Arm concept. See the Arm v7-M Architecture Reference Manual for more information. This bit applies to both CM7 and CM4. 0: Debug turned off 1: Debug enabled (default)
0 NIDEN	ARM non-secure (non-invasive) debug enable Non-invasive debug is Arm concept. See the Arm v7-M Architecture Reference Manual for more information. This bit applies to both CM7 and CM4. 0: Debug turned off 1: Debug enabled (default)

12.4.5.39 GPR38 General Purpose Register (GPR38)

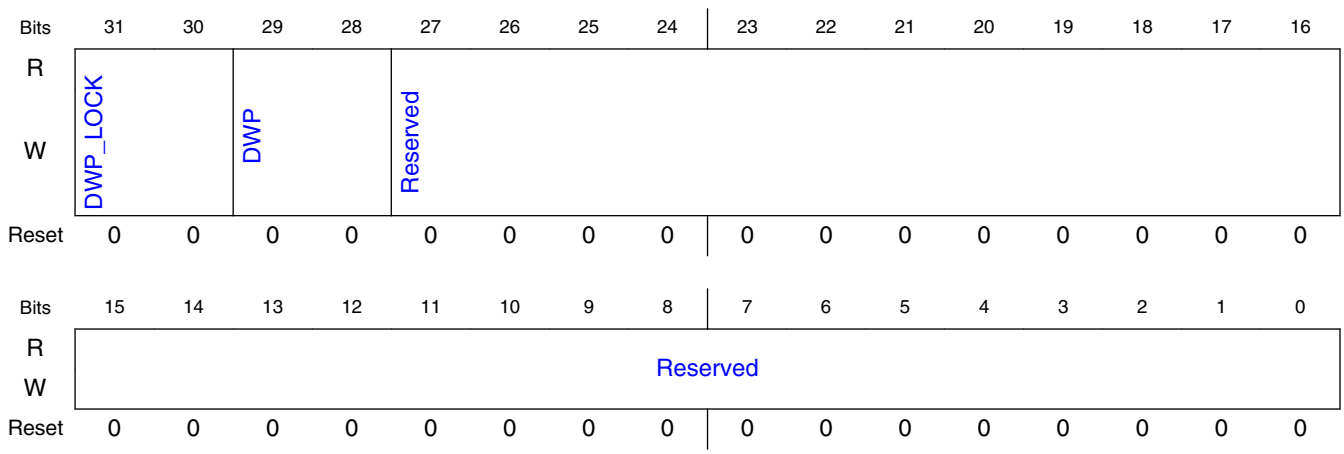
12.4.5.39.1 Offset

Register	Offset
GPR38	98h

12.4.5.39.2 Function

GPR Register

12.4.5.39.3 Diagram



12.4.5.39.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden

Table continues on the next page...

Field	Function
	11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.40 GPR39 General Purpose Register (GPR39)

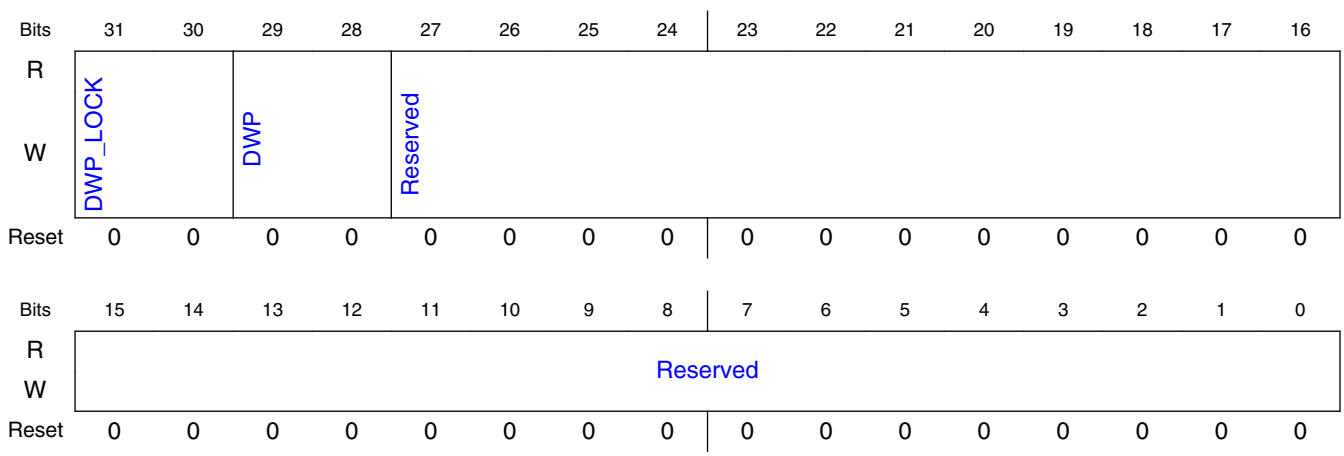
12.4.5.40.1 Offset

Register	Offset
GPR39	9Ch

12.4.5.40.2 Function

GPR Register

12.4.5.40.3 Diagram



12.4.5.40.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked

Table continues on the next page...

Field	Function
	01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.41 GPR40 General Purpose Register (GPR40)

12.4.5.41.1 Offset

Register	Offset
GPR40	A0h

12.4.5.41.2 Function

GPR Register

12.4.5.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWP_LOCK		DWP		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO_MUX2_GPIO_SEL_LOW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.5.41.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 GPIO_MUX2_GPIO_SEL_LOW	GPIO2 and CM7_GPIO2 share same IO MUX function, GPIO_MUX2 selects one GPIO function. This register controls GPIO_MUX2 to select GPIO2 or CM7_GPIO2. For bit n, 1b0 - GPIO2[n] is selected; 1b1 - CM7_GPIO2[n] is selected.

12.4.5.42 GPR41 General Purpose Register (GPR41)

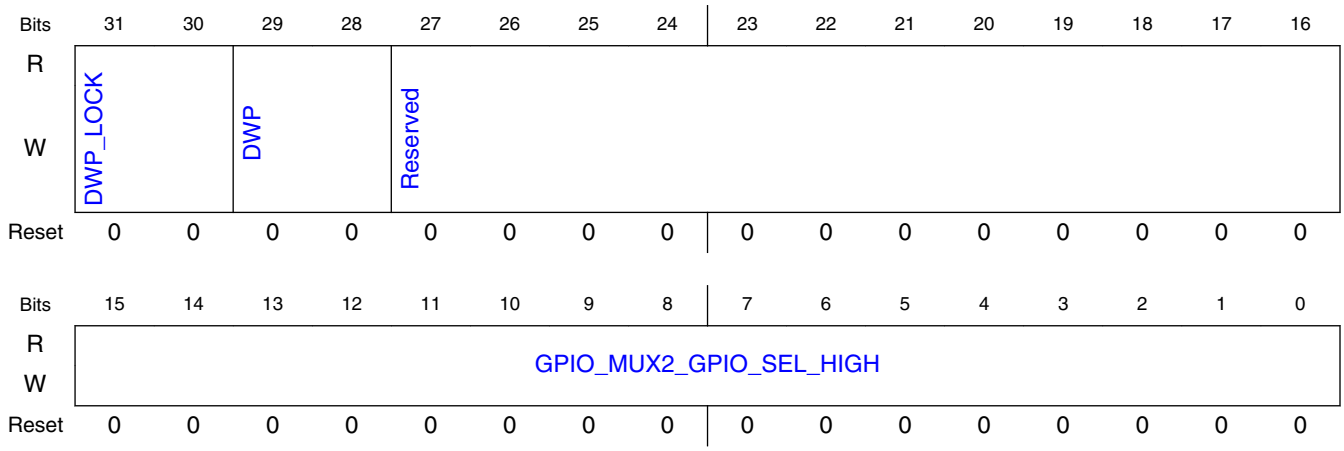
12.4.5.42.1 Offset

Register	Offset
GPR41	A4h

12.4.5.42.2 Function

GPR Register

12.4.5.42.3 Diagram



12.4.5.42.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 GPIO_MUX2_GPIO_SEL_HIGH	GPIO2 and CM7_GPIO2 share same IO MUX function, GPIO_MUX2 selects one GPIO function. This register controls GPIO_MUX2 to select GPIO2 or fast CM7_GPIO2. For bit n, 1b0 - GPIO2[16+n] is selected; 1b1 - CM7_GPIO2[16+n] is selected.

12.4.5.43 GPR42 General Purpose Register (GPR42)

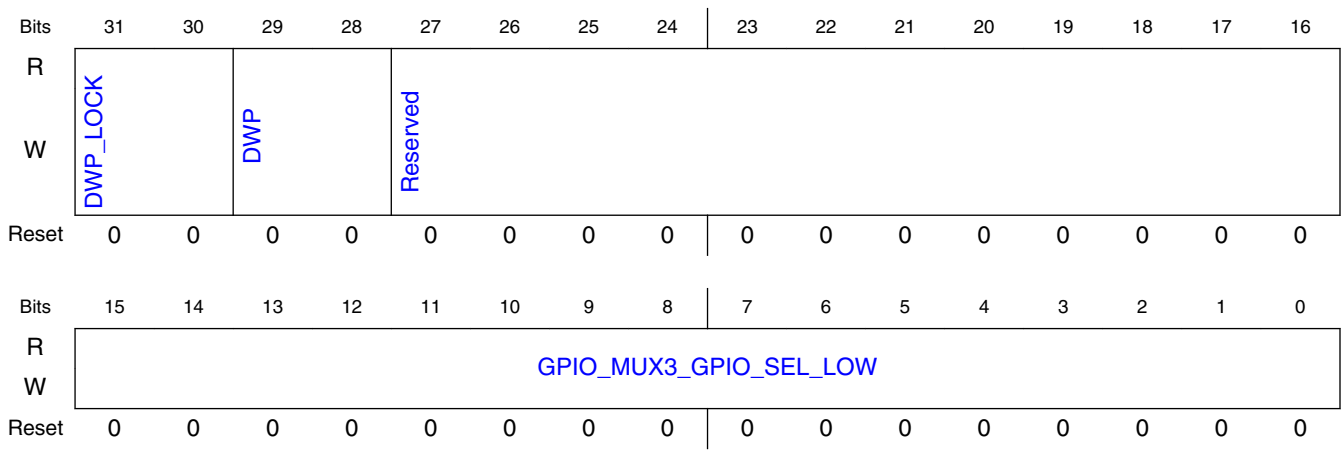
12.4.5.43.1 Offset

Register	Offset
GPR42	A8h

12.4.5.43.2 Function

GPR Register

12.4.5.43.3 Diagram



12.4.5.43.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0	GPIO3 and CM7_GPIO3 share same IO MUX function, GPIO_MUX3 selects one GPIO function.

Field	Function
GPIO_MUX3_GPIO_SEL_LOW	This register controls GPIO_MUX3 to select GPIO3 or fast CM7_GPIO3. For bit n, 1b0 - GPIO3[n] is selected; 1b1 - CM7_GPIO3[n] is selected.

12.4.5.44 GPR43 General Purpose Register (GPR43)

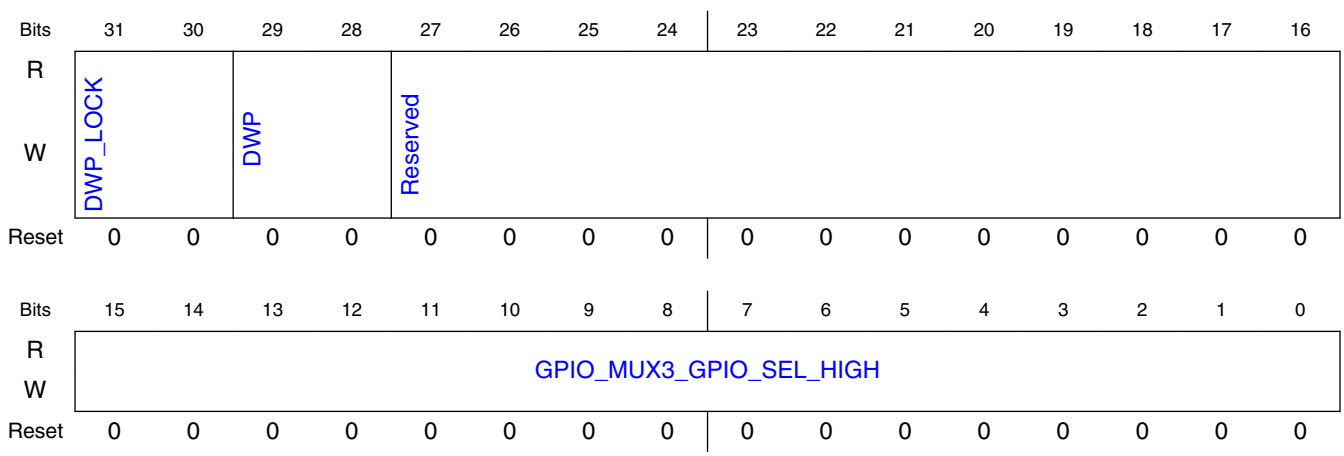
12.4.5.44.1 Offset

Register	Offset
GPR43	ACh

12.4.5.44.2 Function

GPR Register

12.4.5.44.3 Diagram



12.4.5.44.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked

Table continues on the next page...

Field	Function
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 GPIO_MUX3_GPIO_SEL_HIGH	GPIO3 and CM7_GPIO3 share same IO MUX function, GPIO_MUX3 selects one GPIO function. This register controls GPIO_MUX3 to select GPIO3 or CM7_GPIO3. For bit n, 1b0 - GPIO3[16+n] is selected; 1b1 - CM7_GPIO3[16+n] is selected.

12.4.5.45 GPR44 General Purpose Register (GPR44)

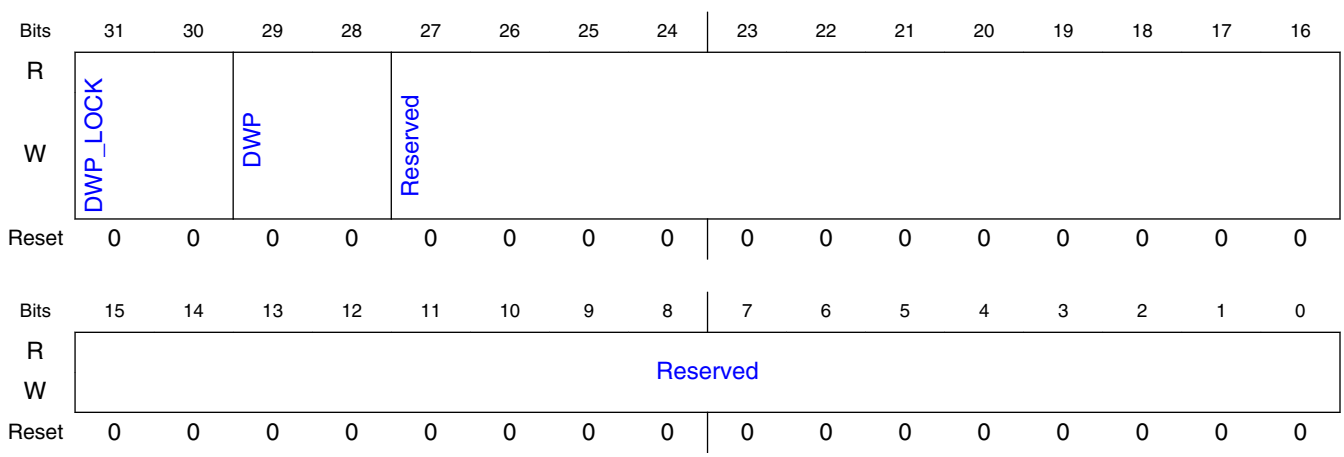
12.4.5.45.1 Offset

Register	Offset
GPR44	B0h

12.4.5.45.2 Function

GPR Register

12.4.5.45.3 Diagram



12.4.5.45.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.46 GPR45 General Purpose Register (GPR45)

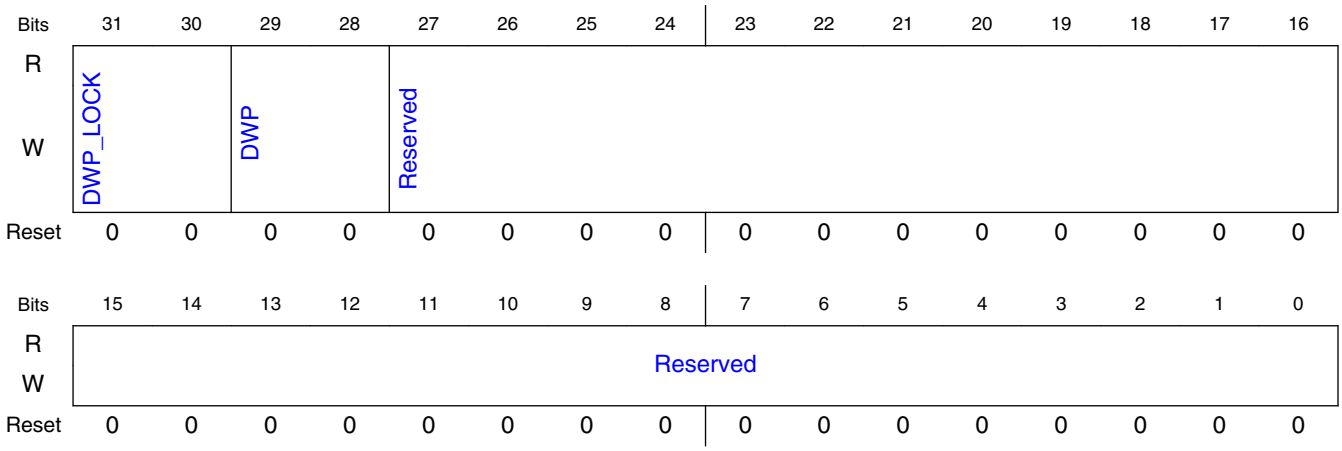
12.4.5.46.1 Offset

Register	Offset
GPR45	B4h

12.4.5.46.2 Function

GPR Register

12.4.5.46.3 Diagram



12.4.5.46.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.47 GPR46 General Purpose Register (GPR46)

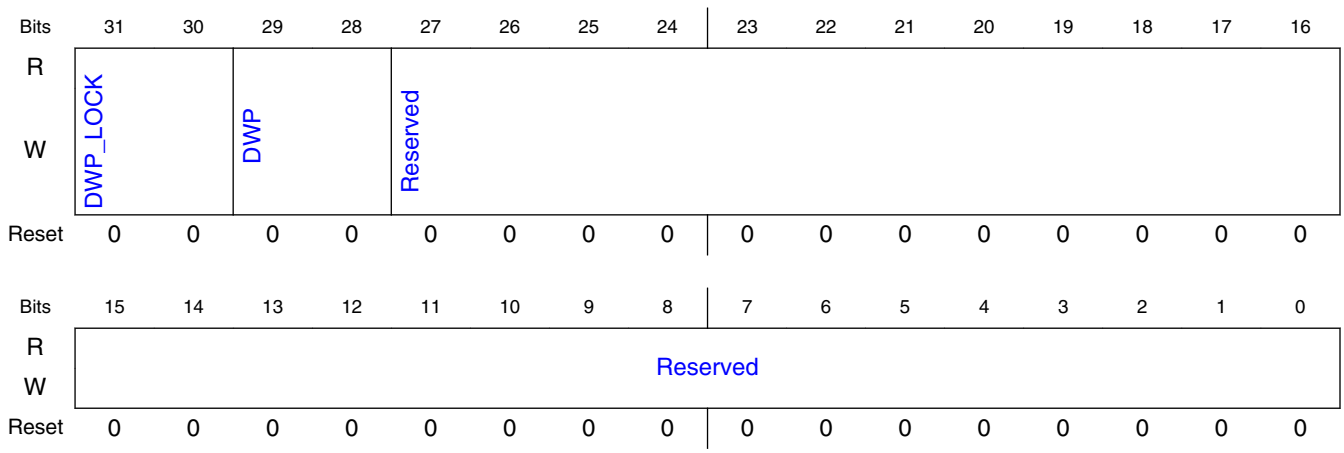
12.4.5.47.1 Offset

Register	Offset
GPR46	B8h

12.4.5.47.2 Function

GPR Register

12.4.5.47.3 Diagram



12.4.5.47.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.48 GPR47 General Purpose Register (GPR47)

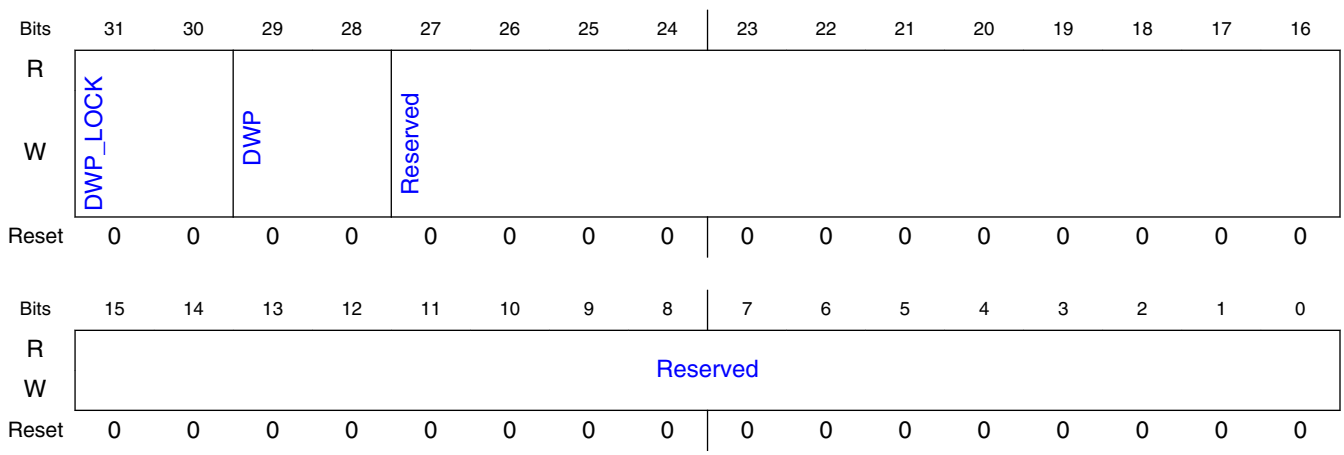
12.4.5.48.1 Offset

Register	Offset
GPR47	BCh

12.4.5.48.2 Function

GPR Register

12.4.5.48.3 Diagram



12.4.5.48.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 —	Reserved

12.4.5.49 GPR48 General Purpose Register (GPR48)

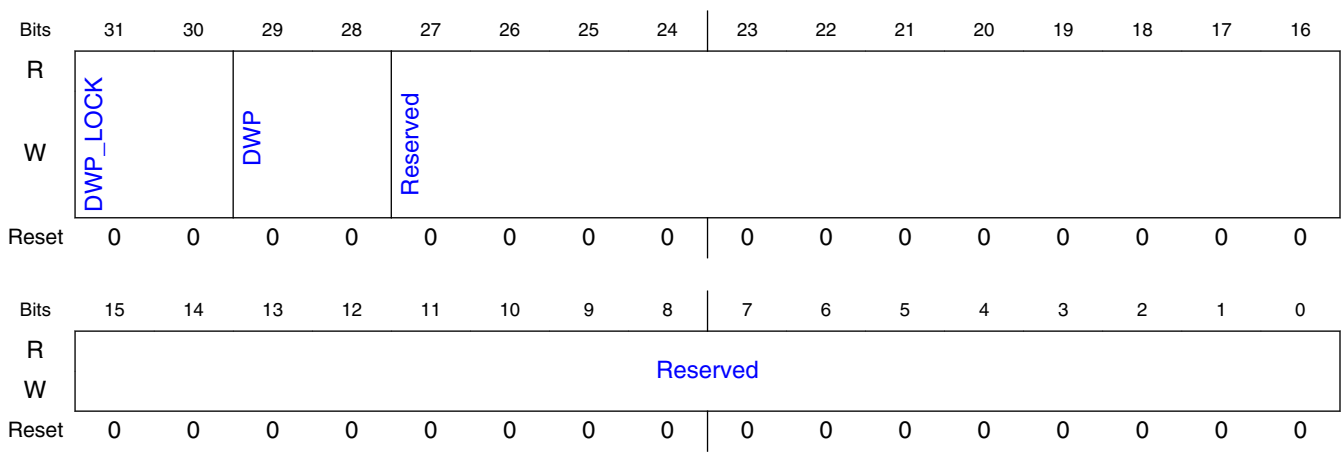
12.4.5.49.1 Offset

Register	Offset
GPR48	C0h

12.4.5.49.2 Function

GPR Register

12.4.5.49.3 Diagram



12.4.5.49.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked

Table continues on the next page...

Field	Function
	11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.50 GPR49 General Purpose Register (GPR49)

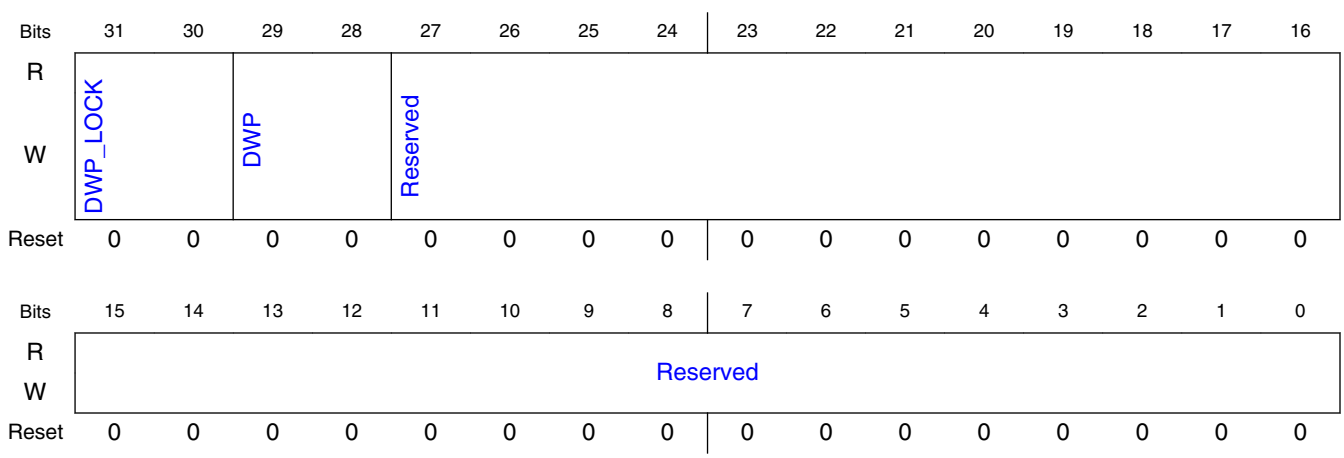
12.4.5.50.1 Offset

Register	Offset
GPR49	C4h

12.4.5.50.2 Function

GPR Register

12.4.5.50.3 Diagram



12.4.5.50.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-0 —	Reserved

12.4.5.51 GPR50 General Purpose Register (GPR50)

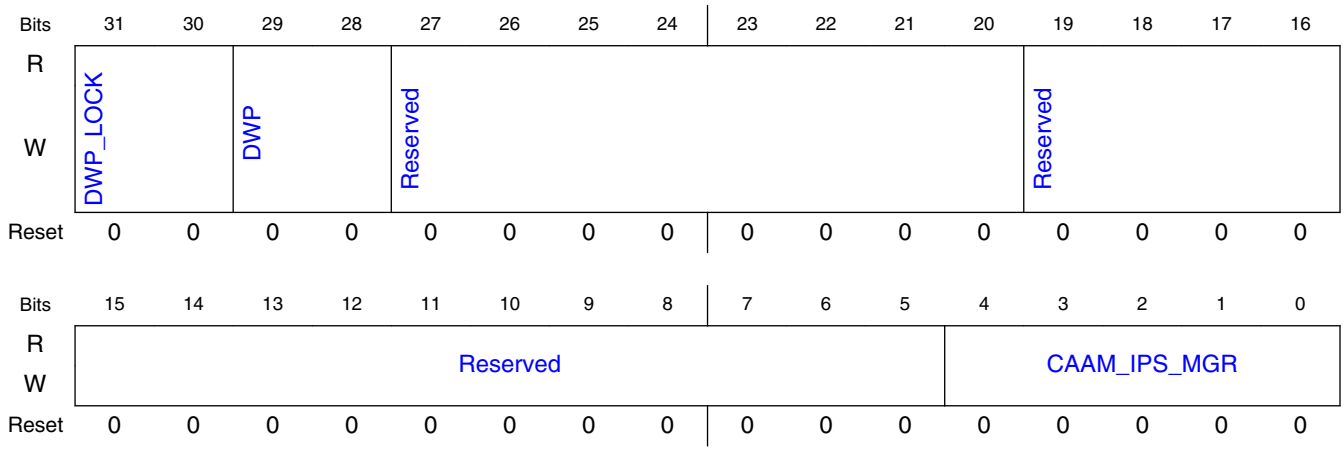
12.4.5.51.1 Offset

Register	Offset
GPR50	C8h

12.4.5.51.2 Function

GPR Register

12.4.5.51.3 Diagram



12.4.5.51.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-20 —	Reserved
19-5 —	Reserved
4-0 CAAM_IPS_MGR	CAAM manager processor identifier This register provides the value to CAAM module's caam_ips_manager[4:0] input signal.

12.4.5.52 GPR51 General Purpose Register (GPR51)

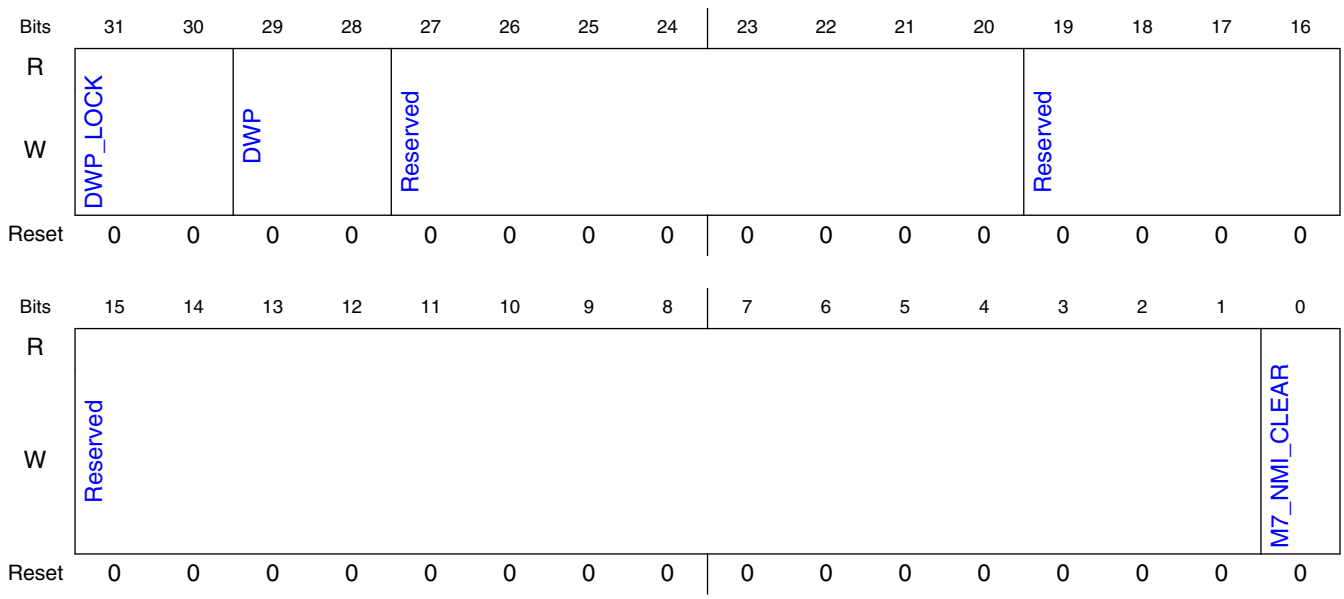
12.4.5.52.1 Offset

Register	Offset
GPR51	CCh

12.4.5.52.2 Function

GPR Register

12.4.5.52.3 Diagram



12.4.5.52.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden

Table continues on the next page...

Field	Function
27-20 —	Reserved
19-1 —	Reserved
0 M7_NMI_CLEAR	Clear CM7 NMI holding register The NMI input from IO wil be held internally until this bit is set to 1. Note that this bit need software write 0 to clear.

12.4.5.53 GPR52 General Purpose Register (GPR52)

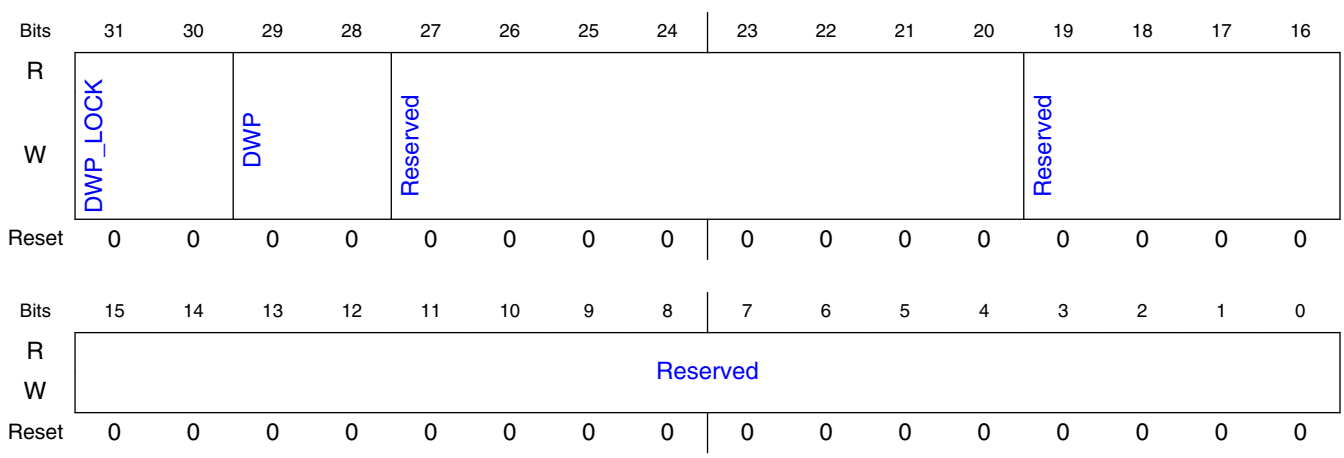
12.4.5.53.1 Offset

Register	Offset
GPR52	D0h

12.4.5.53.2 Function

GPR Register

12.4.5.53.3 Diagram



12.4.5.53.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-20 —	Reserved
19-0 —	Reserved

12.4.5.54 GPR53 General Purpose Register (GPR53)

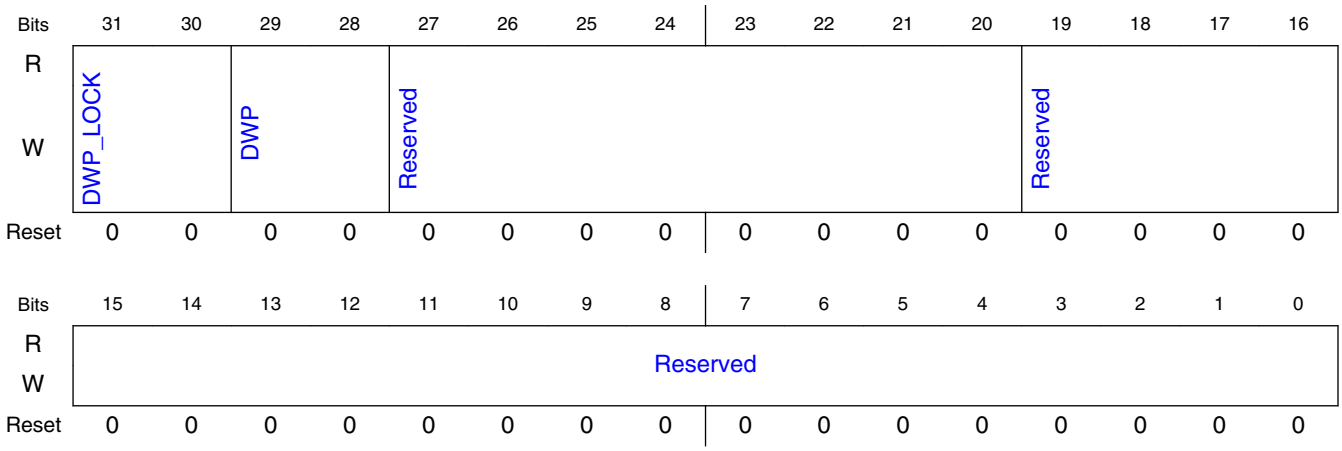
12.4.5.54.1 Offset

Register	Offset
GPR53	D4h

12.4.5.54.2 Function

GPR Register

12.4.5.54.3 Diagram



12.4.5.54.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-20 —	Reserved
19-0 —	Reserved

12.4.5.55 GPR54 General Purpose Register (GPR54)

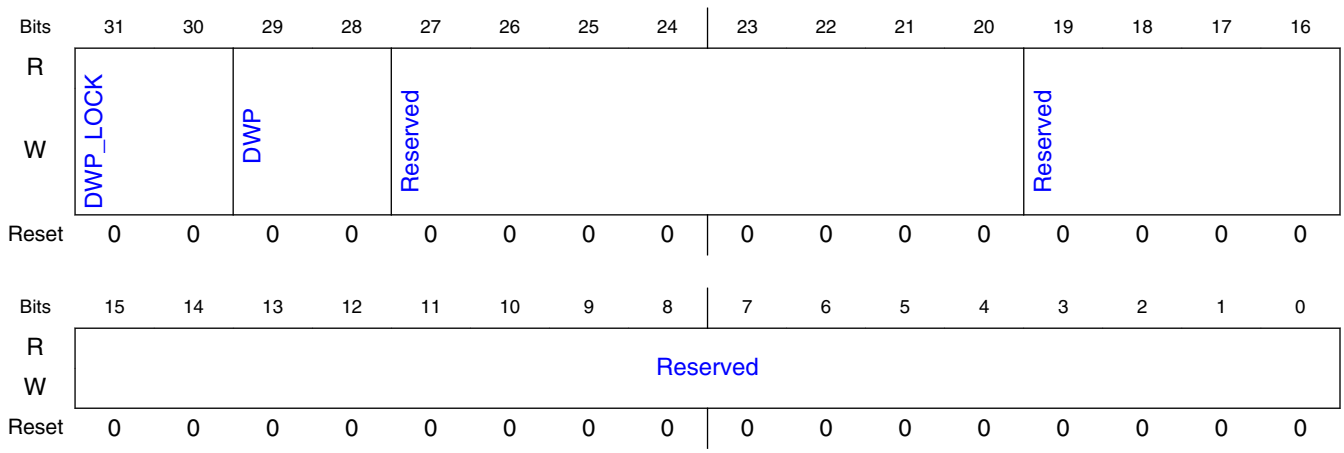
12.4.5.55.1 Offset

Register	Offset
GPR54	D8h

12.4.5.55.2 Function

GPR Register

12.4.5.55.3 Diagram



12.4.5.55.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-20 —	Reserved
19-0 —	Reserved

12.4.5.56 GPR55 General Purpose Register (GPR55)

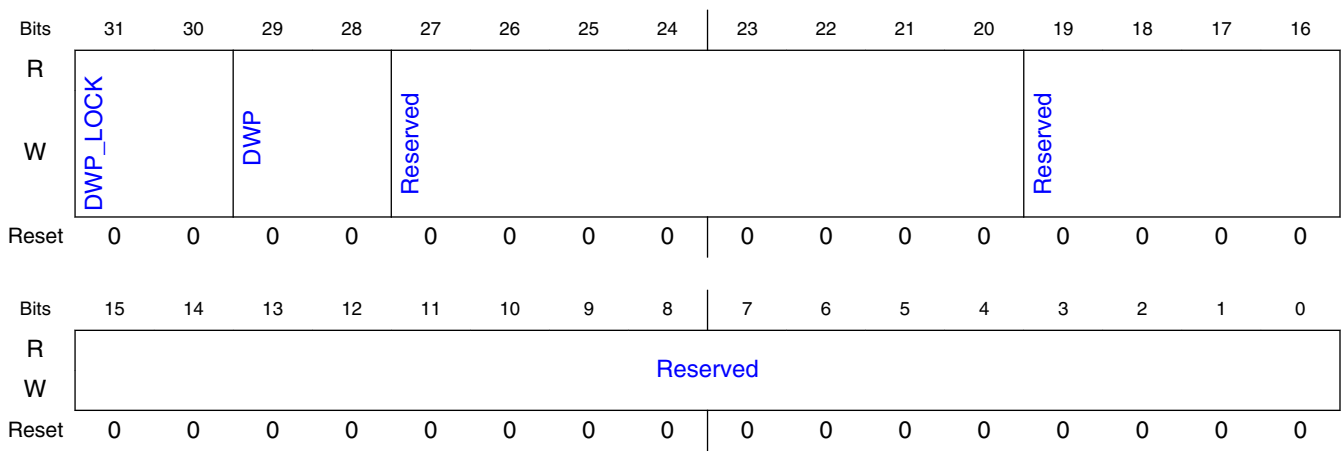
12.4.5.56.1 Offset

Register	Offset
GPR55	DCh

12.4.5.56.2 Function

GPR Register

12.4.5.56.3 Diagram



12.4.5.56.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-20	Reserved

Table continues on the next page...

Field	Function
—	
19-0 —	Reserved

12.4.5.57 GPR59 General Purpose Register (GPR59)

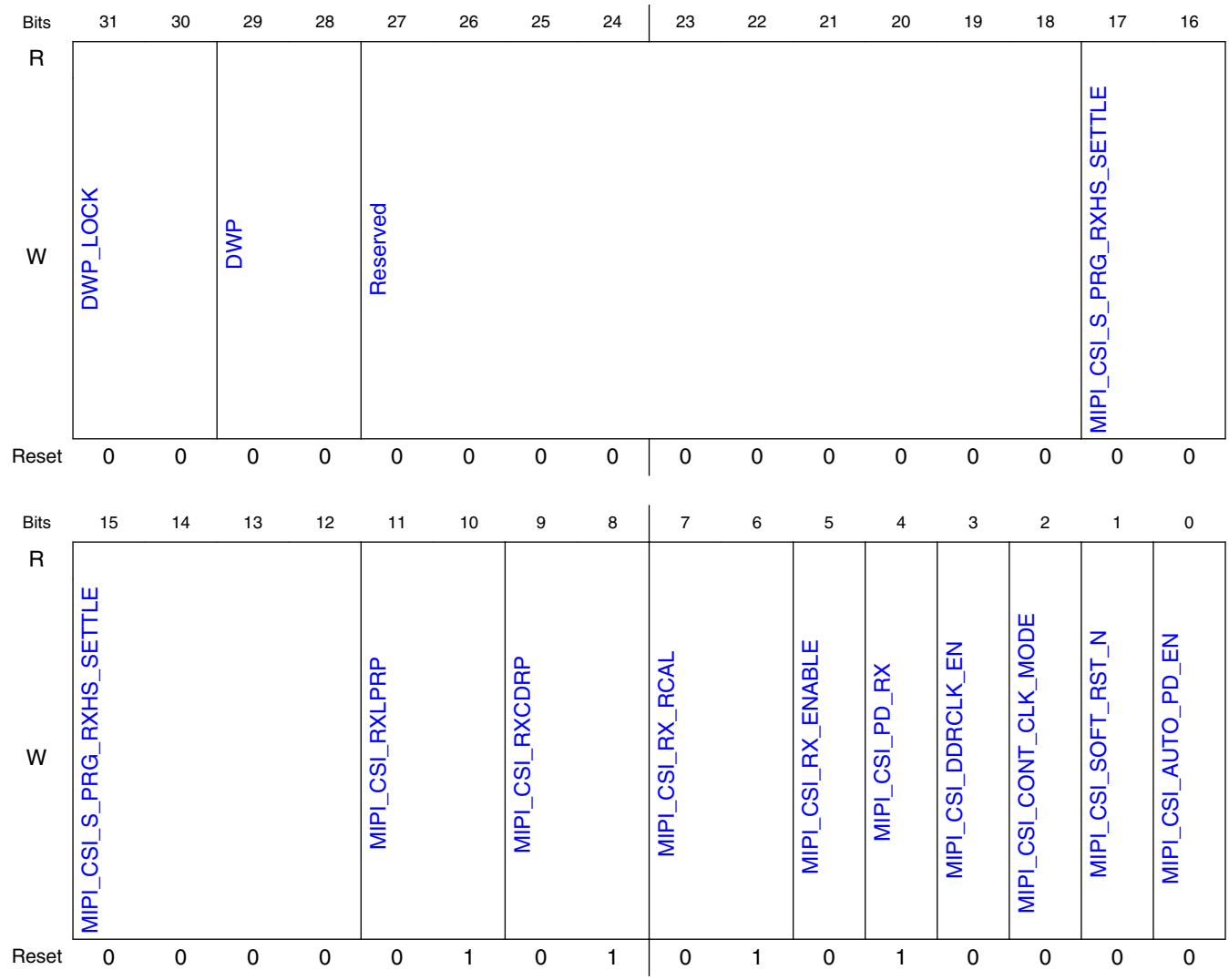
12.4.5.57.1 Offset

Register	Offset
GPR59	ECh

12.4.5.57.2 Function

GPR Register

12.4.5.57.3 Diagram



12.4.5.57.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 17:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>10b - CM4 is forbidden
11b - Both cores are forbidden</td></tr>
<tr><td>27-18
—</td><td>Reserved</td></tr>
<tr><td>17-12
MIPI_CSI_S_PRG_RXHS_SETTLE</td><td>Bits used to program T_HS_SETTLE.
HS-RX waits for Time-out T_HS_SETTLE in order to neglect transition effects.
a. The equation is T_HS_SETTLE= (PRG_RXHS_SETTLE + 1) * (Tperiod of RxClkInEsc)
b. Min value of T_HS_SETTLE = 85n + 6*UI, where UI is the time period of User Clock (clk_ui)
c. Max value of T_HS_SETTLE = 145ns + 10*UI</td></tr>
<tr><td>11-10
MIPI_CSI_RXLPRP</td><td>Programming bits that adjust the threshold voltage of LP-RX, default setting 2’b01
<table><tr><th>Bitfield Value</th><th>High Threshold Voltage</th><th>Low Threshold Voltage</th></tr><tr><td>2'b00</td><td>782mV</td><td>730mV</td></tr><tr><td>2'b01 (Default)</td><td>745mV</td><td>692mV</td></tr><tr><td>2'b10</td><td>708mV</td><td>655mV</td></tr><tr><td>2'b11</td><td>Invalid</td><td>Invalid</td></tr></table></td></tr>
<tr><td>9-8
MIPI_CSI_RXCDRP</td><td>Programming bits that adjust the threshold voltage of LP-CD, default setting 2’b01
00b - 344mV
01b - 325mV (Default)
10b - 307mV
11b - Invalid</td></tr>
<tr><td>7-6
MIPI_CSI_RX_RCAL</td><td>MIPI CSI PHY on-chip termination control bits
On-chip termination control bits for manual calibration.
00: 20% higher than mid range. Highest impedance setting
01 Mid range impedance setting.
10 15% lower than mid range
11 25% lower than mid range. Lowest impedance setting</td></tr>
<tr><td>5
MIPI_CSI_RX_ENABLE</td><td>Assert to enable MIPI CSI Receive Enable
When deasserted, the RX controller will pause data reception at the next packet boundary. Since the CSI-2 protocol does not allow the Receiver to pause data, when rx_enable is deasserted the RX Controller still receives data from the RX DPHY but it does not forward the receive packets to the user interface but instead discards the received data. When rx_enable is asserted, the RX controller will wait for the start of the next packet before allowing data to be sent out over the user interface.</td></tr>
<tr><td>4
MIPI_CSI_PD_RX</td><td>Power Down input for MIPI CSI PHY.
When high, all blocks are powered down.</td></tr>
<tr><td>3
MIPI_CSI_DDRCLK_EN</td><td>When high, enables received DDR clock on CLK_DRXHS</td></tr>
<tr><td>2
MIPI_CSI_CONT_CLK_MODE</td><td>Enables the slave clock lane feature to maintain HS reception state during continuous clock mode operation, despite line glitches.
1: Feature enabled</td></tr>
</table>

Table continues on the next page...

Field	Function
	0: Feature disabled
1 MIPI_CSI_SOFT_RST_N	MIPI CSI APB clock domain and User interface clock domain software reset bit Async reset for all logic in the pclk clock domain. Note this bit is active low and not self-clearing. By default it holds the MIPI CSI in reset state. 0b - Assert reset 1b - De-assert reset
0 MIPI_CSI_AUTO_PD_EN	Powers down inactive lanes reported by CSI2X_CFG_NUM_LANES. 1'b0: inactive lanes are powered up and driving LP11 1'b1: inactive lanes are powered down.

12.4.5.58 GPR62 General Purpose Register (GPR62)

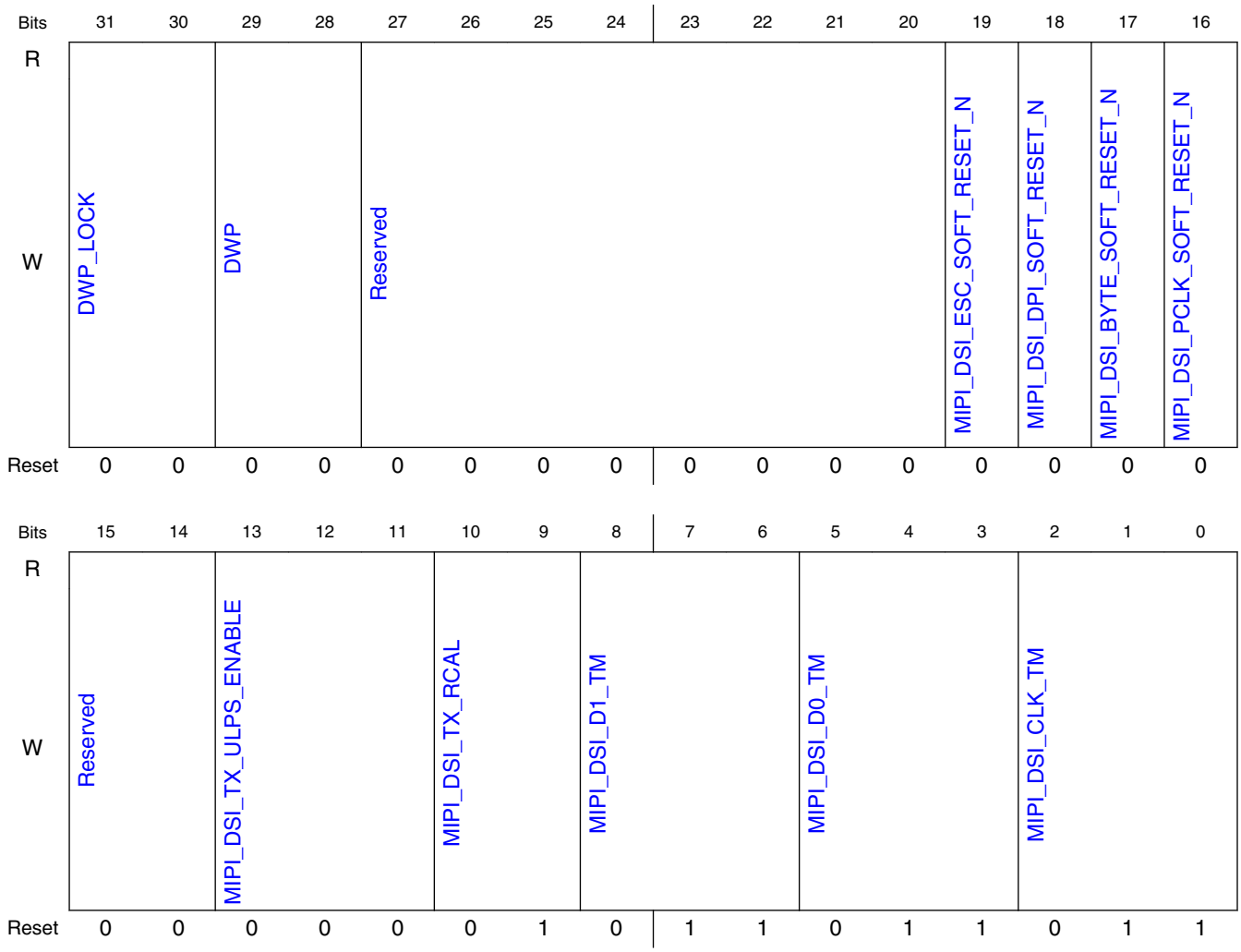
12.4.5.58.1 Offset

Register	Offset
GPR62	F8h

12.4.5.58.2 Function

GPR Register

12.4.5.58.3 Diagram



12.4.5.58.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden

Table continues on the next page...

Field	Function
	11b - Both cores are forbidden
27-20 —	Reserved
19 MIPI_DSI_ESC_SOFT_RESET_N	MIPI DSI Escape clock domain software reset bit Note this bit is active low and not self-clearing. By default it holds the MIPI DSI in reset state 0b - Assert reset 1b - De-assert reset
18 MIPI_DSI_DPI_SOFT_RESET_N	MIPI DSI Pixel clock domain software reset bit Note this bit is active low and not self-clearing. By default it holds the MIPI DSI in reset state. 0b - Assert reset 1b - De-assert reset
17 MIPI_DSI_BYTE_SOFT_RESET_N	MIPI DSI Byte clock domain software reset bit Note this bit is active low and not self-clearing. By default it holds the MIPI DSI in reset state. 0b - Assert reset 1b - De-assert reset
16 MIPI_DSI_PCLK_SOFT_RESET_N	MIPI DSI APB clock domain software reset bit Note this bit is active low and not self-clearing. By default it holds the MIPI DSI in reset state. 0b - Assert reset 1b - De-assert reset
15-14 —	Reserved
13-11 MIPI_DSI_TX_ULPS_ENABLE	DSI transmit ULPS mode enable Each bit represents a data lane and the clock lane. A ‘1’ enables the associated clock lane or data lane ULPS mode. [0] – clock lane [1] – data lane 0 [2] – data lane 1
10-9 MIPI_DSI_TX_RCAL	MIPI DSI PHY on-chip termination control bits On-chip termination control bits for manual calibration of HS-TX. 00: 20% higher than mid-range (Highest impedance setting) 01: Mid-range impedance setting (default) 10: 15% lower than mid-range 11: 25% lower than mid-range (Lowest impedance setting)
8-6 MIPI_DSI_D1_TM	MIPI DSI Data Lane 1 triming bits LPTX VOH voltage level trimming bus for Data Lane 1. Total of 8 trimming levels, voltage adjustment granularity is approximately 34mV. Default setting 3'b011. Voltage levels for each setting are listed below(typical case, no mismatch): 3'b000=1086mV 3'b001=1120mV 3'b010=1155mV 3'b011=1190mV 3'b100=1224mV

Table continues on the next page...

Field	Function
	3'b101=1258mV 3'b110=1293mV 3'b111=1327mV
5-3 MIPI_DSI_D0_TM	MIPI DSI Data Lane 0 triming bits LPTX VOH voltage level trimming bus for Data Lane 0. Feature same as MIPI_DSI_D1_TM.
2-0 MIPI_DSI_CLK_TM	MIPI DSI Clock Lane triming bits LPTX VOH voltage level trimming bus for Clock Lane. Feature same as MIPI_DSI_D1_TM.

12.4.5.59 GPR63 General Purpose Register (GPR63)

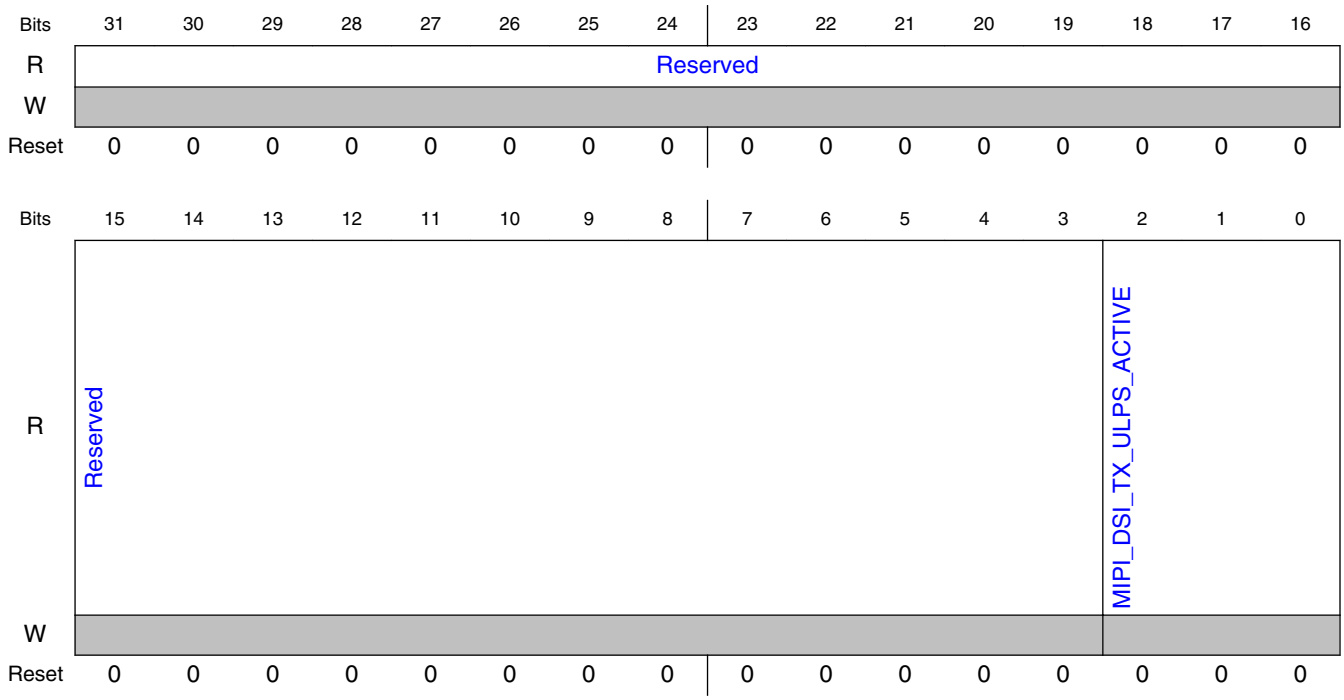
12.4.5.59.1 Offset

Register	Offset
GPR63	FCh

12.4.5.59.2 Function

GPR Register

12.4.5.59.3 Diagram



12.4.5.59.4 Fields

Field	Function
31-3 —	Reserved
2-0 MIPI_DSI_TX_ULPS_ACTIVE	DSI transmit ULPS mode active flag Each bit represents a data lane and the clock lane. A ‘1’ indicates the associated clock lane or data lane is in ULPS mode, ‘0’ indicates not in ULPS mode [0] – clock lane [1] – data lane 0 [2] – data lane 1

12.4.5.60 GPR64 General Purpose Register (GPR64)

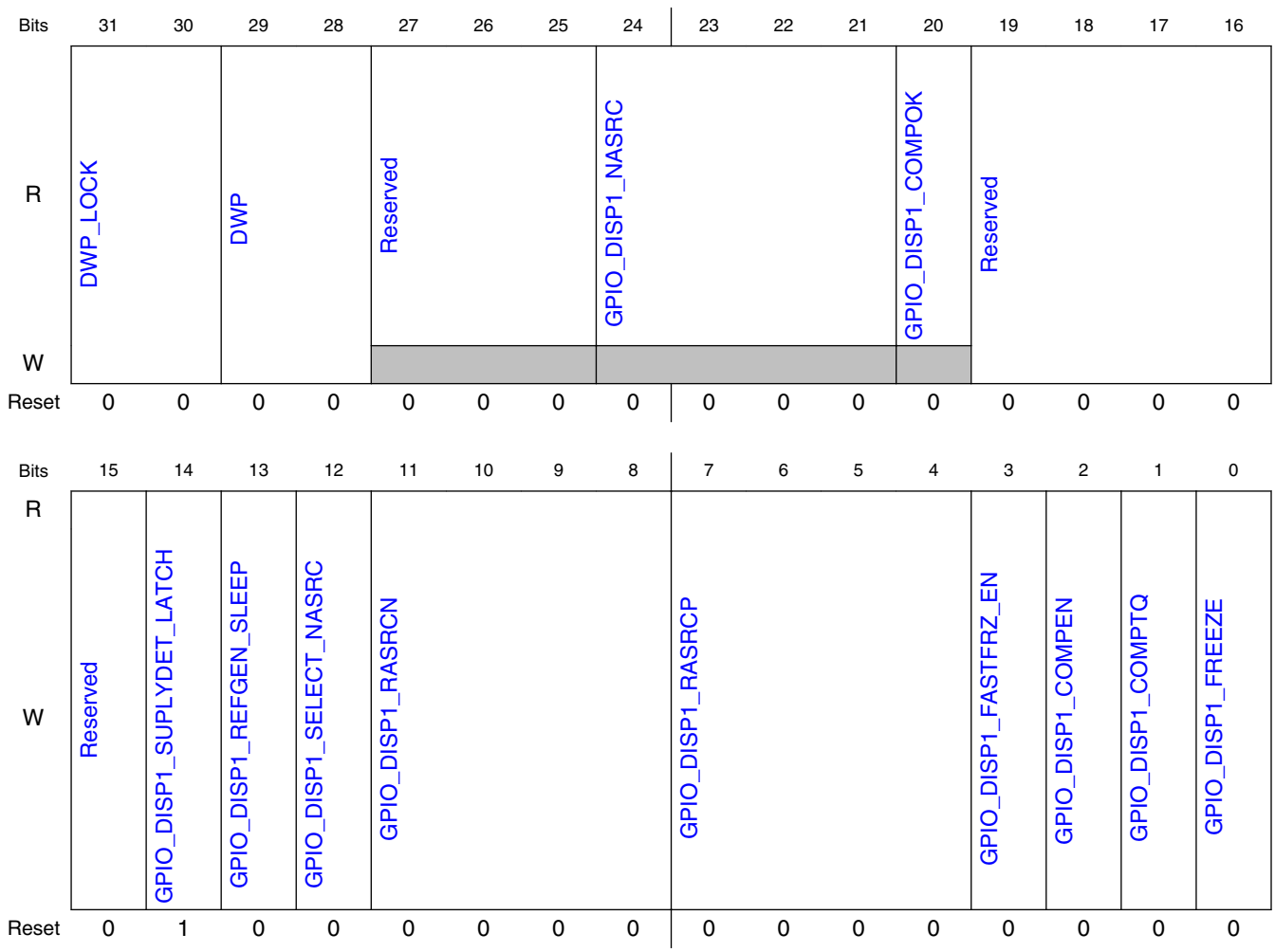
12.4.5.60.1 Offset

Register	Offset
GPR64	100h

12.4.5.60.2 Function

GPR Register

12.4.5.60.3 Diagram



12.4.5.60.4 Fields

Field	Function
31-30	Domain write protection lock

Table continues on the next page...

Field	Function
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-25 —	Reserved
24-21 GPIO_DISP1_NASRC	GPIO_DISP_B1 IO bank compensation codes 4-bit NMOS compensation codes or 4-bit PMOS compensation codes selected by GPIO_DISP1_SELECT_NASRC.
20 GPIO_DISP1_COMPOK	GPIO_DISP_B1 IO bank compensation OK flag It can be high only in the Normal mode and when a new measured code is available.
19-15 —	Reserved
14 GPIO_DISP1_SUPLYDET_LATCH	GPIO_DISP_B1 IO bank power supply mode latch enable Supply detector cell is used to detect the IO supply range and set IO to 3V3 supply mode or 1V8 supply mode according to the detection. Setting this bit will latch the current detect result.
13 GPIO_DISP1_REFGEN_SLEEP	GPIO_DISP_B1 IO bank reference voltage generator cell sleep enable Reference voltage generator cell provides reference voltages to the IO bank. When this bit is high, it puts the reference voltage generator cell in sleep mode to reduce the static power consumption.
12 GPIO_DISP1_SELECT_NASRC	GPIO_DISP1_NASRC selection 0: Show the 4-bit PMOS compensation codes in GPIO_DISP1_NASRC field 1: Show the 4-bit NMOS compensation codes in GPIO_DISP1_NASRC field
11-8 GPIO_DISP1_RASRCN	GPIO_DISP_B1 IO bank's 4-bit NMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
7-4 GPIO_DISP1_RASRCP	GPIO_DISP_B1 IO bank's 4-bit PMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
3 GPIO_DISP1_FASTFRZ_EN	Compensation code fast freeze When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, low, high, compensation cell Fast Freeze mode is enabled. Fast Freeze mode is activated on signal rising edge. The compensation cell delivers a refreshed compensation code, at a comparatively lesser delay after signal falling edge than the delay of Freeze mode.
2 GPIO_DISP1_COMPEN	COMPEN and COMPTQ control the operating modes of the compensation cell

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Normal Mode: In this mode, the macro cell constantly tracks the PVT condition of the chip and generates an 8-bit digital code. This 8-bit code is referred to as f(PVT) and it represents current PVT state. In Normal mode, the internal reference current generators are active and power consumption is higher than that in all the other modes.

Freeze mode: This mode is used where the current consumption is to be kept low. In this mode, all the internal blocks are switched off to reduce the static power consumption. Internal latches keep the code, which is calculated before the cell enters the Freeze mode.

Fast Freeze mode: This mode is used to freeze the digital compensation codes, when the data is transferred from the compensated IOs on a chip to an external device during Burst mode. This ensures signal integrity and eliminates the jitter caused due to modification in IO driver strength. In this mode, the current consumption is comparable to consumption in the Normal mode.

Read Mode: In this mode, it is possible to force the digital codes from the chip core logic. The bandgap, measurement block, and comparator blocks are switched off to reduce static power consumption to minimum value.

Fixed Code mode: The digital code is forced to a fixed value, obtained at typical PVT conditions and represents typical bit patterns. The bandgap, measurement, and other blocks are switched off to reduce static power consumption to minimum value.

<table>
<tr><th>COMPEN</th><th>COMPTQ</th><th>FREEZE</th><th>FASTFRZ_EN</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Normal Mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Freeze Mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fast Freeze Mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Read Mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed Code Mode</td></tr>
</table>
</td></tr>
<tr><td>1

GPIO_DISP1_COMPTQ</td><td>COMPEN and COMPTQ control the operating modes of the compensation cell</td></tr>
<tr><td>0

GPIO_DISP1_FREEZE</td><td>Compensation code freeze

When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, high, low, compensation cell Freeze mode is enabled. On the rising edge of FREEZE, it freezes the compensation code at its running value. The compensation cell delivers refreshed compensation code at a delay after FREEZE falling edge.</td></tr>
</table>

12.4.5.61 GPR65 General Purpose Register (GPR65)

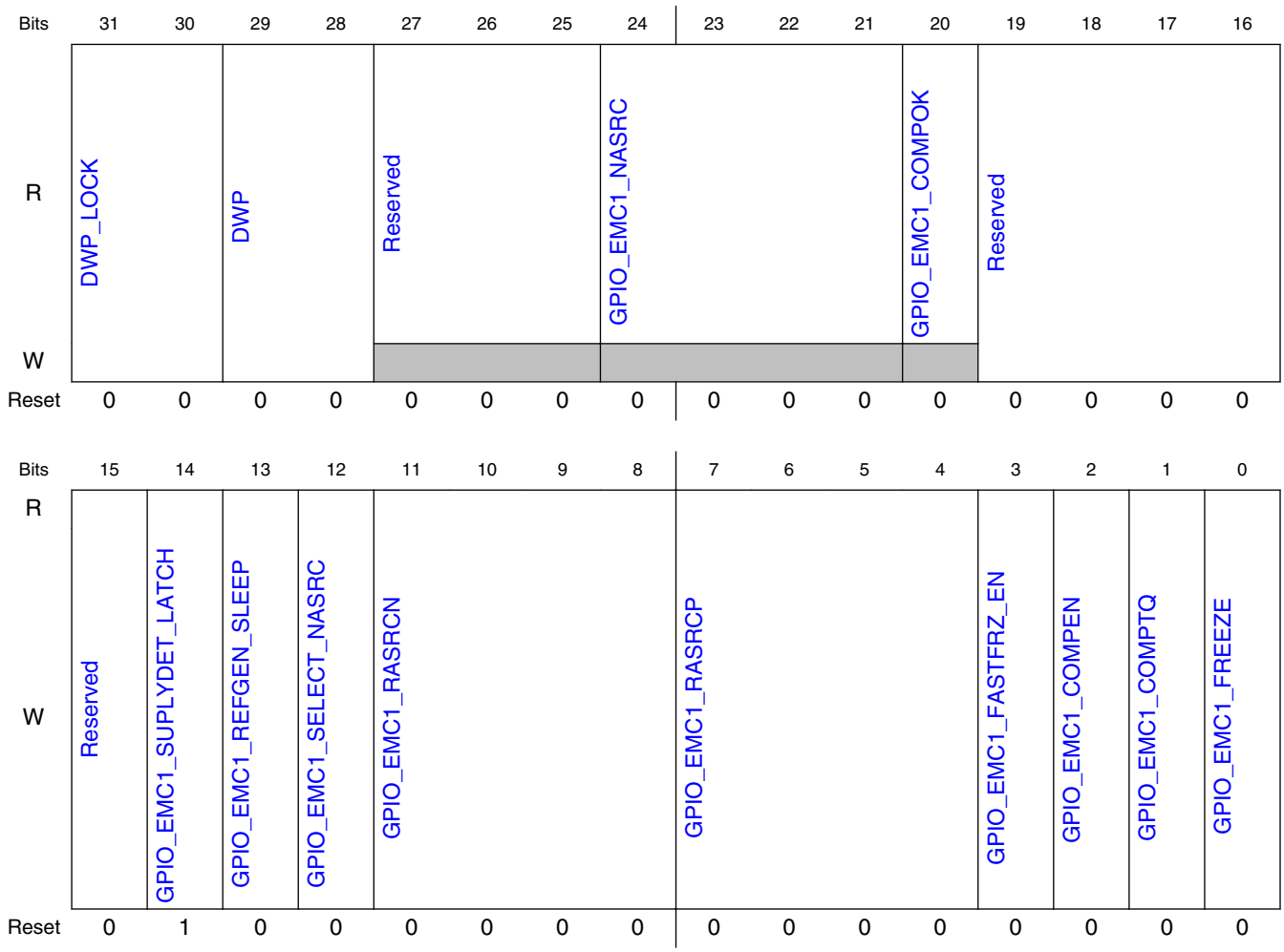
12.4.5.61.1 Offset

Register	Offset
GPR65	104h

12.4.5.61.2 Function

GPR Register

12.4.5.61.3 Diagram



12.4.5.61.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden.

Table continues on the next page...

Field	Function
	00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-25 —	Reserved
24-21 GPIO_EMC1_NASRC	GPIO_EMC_B1 IO bank compensation codes 4-bit NMOS compensation codes or 4-bit PMOS compensation codes selected by GPIO_EMC1_SELECT_NASRC.
20 GPIO_EMC1_COMPOK	GPIO_EMC_B1 IO bank compensation OK flag It can be high only in the Normal mode and when a new measured code is available.
19-15 —	Reserved
14 GPIO_EMC1_SUPLYDET_LATCH	GPIO_EMC_B1 IO bank power supply mode latch enable Supply detector cell is used to detect the IO supply range and set IO to 3V3 supply mode or 1V8 supply mode according to the detection. Setting this bit will latch the current detect result.
13 GPIO_EMC1_REFGEN_SLEEP	GPIO_EMC_B1 IO bank reference voltage generator cell sleep enable Reference voltage generator cell provides reference voltages to the IO bank. When this bit is high, it puts the reference voltage generator cell in sleep mode to reduce the static power consumption.
12 GPIO_EMC1_SELECT_NASRC	GPIO_EMC1_NASRC selection 0: Show the 4-bit PMOS compensation codes in GPIO_EMC1_NASRC field 1: Show the 4-bit NMOS compensation codes in GPIO_EMC1_NASRC field
11-8 GPIO_EMC1_RASRCN	GPIO_EMC_B1 IO bank's 4-bit NMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
7-4 GPIO_EMC1_RASRCP	GPIO_EMC_B1 IO bank's 4-bit PMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
3 GPIO_EMC1_FASTFRZ_EN	Compensation code fast freeze When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, low, high, compensation cell Fast Freeze mode is enabled. Fast Freeze mode is activated on signal rising edge. The compensation cell delivers a refreshed compensation code, at a comparatively lesser delay after signal falling edge than the delay of Freeze mode.
2 GPIO_EMC1_COMPEN	COMPEN and COMPTQ control the operating modes of the compensation cell Normal Mode: In this mode, the macro cell constantly tracks the PVT condition of the chip and generates an 8-bit digital code. This 8-bit code is referred to as f(PVT) and it represents current PVT state. In Normal mode, the internal reference current generators are active and power consumption is higher than that in all the other modes. Freeze mode: This mode is used where the current consumption is to be kept low. In this mode, all the internal blocks are switched off to reduce the static power consumption. Internal latches keep the code, which is calculated before the cell enters the Freeze mode.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Fast Freeze mode: This mode is used to freeze the digital compensation codes, when the data is transferred from the compensated IOs on a chip to an external device during Burst mode. This ensures signal integrity and eliminates the jitter caused due to modification in IO driver strength. In this mode, the current consumption is comparable to consumption in the Normal mode.

Read Mode: In this mode, it is possible to force the digital codes from the chip core logic. The bandgap, measurement block, and comparator blocks are switched off to reduce static power consumption to minimum value.

Fixed Code mode: The digital code is forced to a fixed value, obtained at typical PVT conditions and represents typical bit patterns. The bandgap, measurement, and other blocks are switched off to reduce static power consumption to minimum value.

<table>
<tr><th>COMPEN</th><th>COMPTQ</th><th>FREEZE</th><th>FASTFRZ_EN</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Normal Mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Freeze Mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fast Freeze Mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Read Mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed Code Mode</td></tr>
</table>
</td></tr>
<tr><td>1

GPIO_EMC1_COMPTQ</td><td>COMPEN and COMPTQ control the operating modes of the compensation cell</td></tr>
<tr><td>0

GPIO_EMC1_FREEZE</td><td>Compensation code freeze

When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, high, low, compensation cell Freeze mode is enabled. On the rising edge of FREEZE, it freezes the compensation code at its running value. The compensation cell delivers refreshed compensation code at a delay after FREEZE falling edge.</td></tr>
</table>

12.4.5.62 GPR66 General Purpose Register (GPR66)

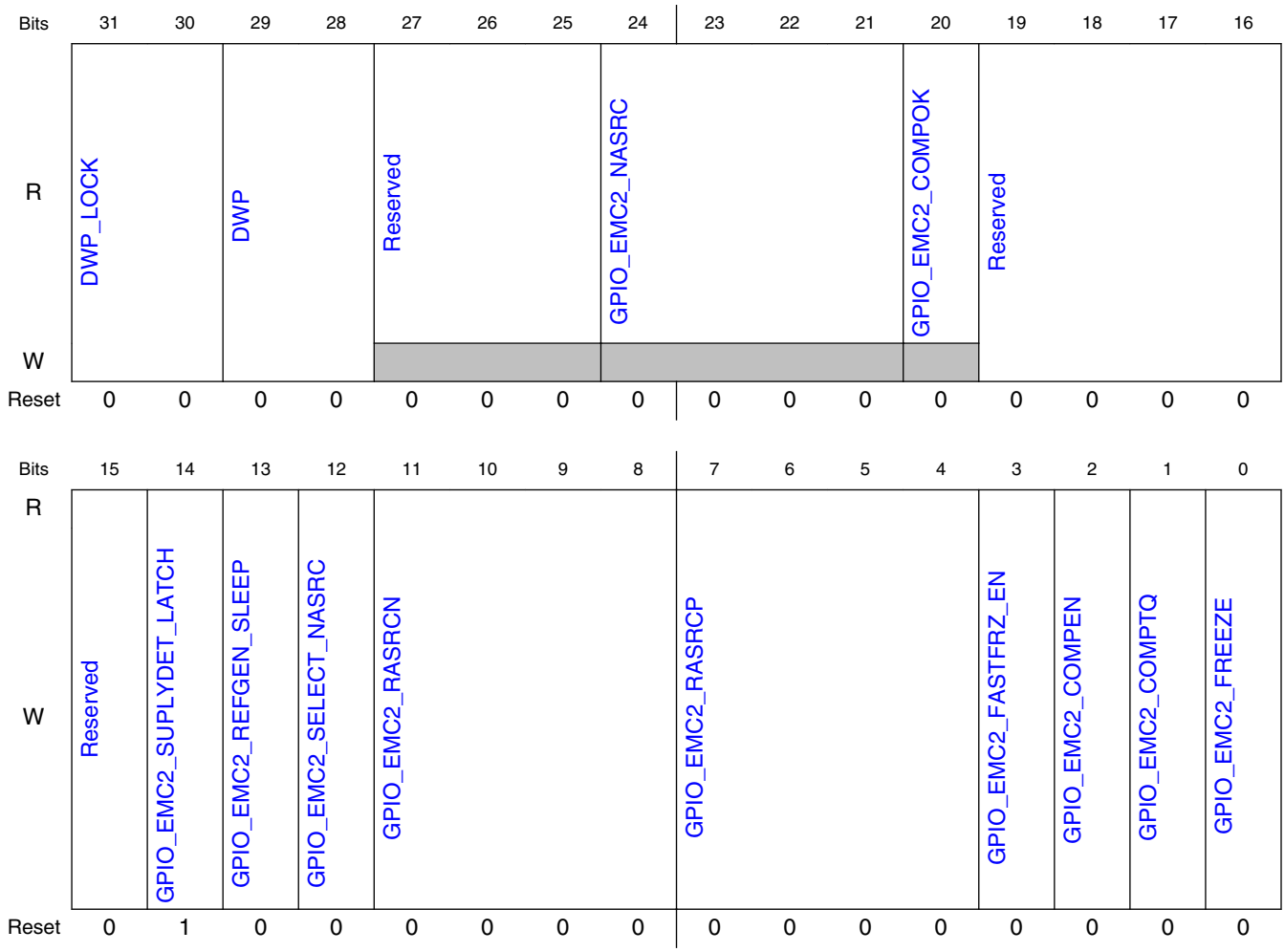
12.4.5.62.1 Offset

Register	Offset
GPR66	108h

12.4.5.62.2 Function

GPR Register

12.4.5.62.3 Diagram



12.4.5.62.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden

Table continues on the next page...

Field	Function
27-25 —	Reserved
24-21 GPIO_EMC2_NASRC	GPIO_EMC_B2 IO bank compensation codes 4-bit NMOS compensation codes or 4-bit PMOS compensation codes selected by GPIO_EMC2_SELECT_NASRC.
20 GPIO_EMC2_COMPOK	GPIO_EMC_B2 IO bank compensation OK flag It can be high only in the Normal mode and when a new measured code is available.
19-15 —	Reserved
14 GPIO_EMC2_SUPLYDET_LATCH	GPIO_EMC_B2 IO bank power supply mode latch enable Supply detector cell is used to detect the IO supply range and set IO to 3V3 supply mode or 1V8 supply mode according to the detection. Setting this bit will latch the current detect result.
13 GPIO_EMC2_REFGEN_SLEEP	GPIO_EMC_B2 IO bank reference voltage generator cell sleep enable Reference voltage generator cell provides reference voltages to the IO bank. When this bit is high, it puts the reference voltage generator cell in sleep mode to reduce the static power consumption.
12 GPIO_EMC2_SELECT_NASRC	GPIO_EMC2_NASRC selection 0: Show the 4-bit PMOS compensation codes in GPIO_EMC2_NASRC field 1: Show the 4-bit NMOS compensation codes in GPIO_EMC2_NASRC field
11-8 GPIO_EMC2_RASRCN	GPIO_EMC_B2 IO bank's 4-bit NMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
7-4 GPIO_EMC2_RASRCP	GPIO_EMC_B2 IO bank's 4-bit PMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
3 GPIO_EMC2_FASTFRZ_EN	Compensation code fast freeze When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, low, high, compensation cell Fast Freeze mode is enabled. Fast Freeze mode is activated on signal rising edge. The compensation cell delivers a refreshed compensation code, at a comparatively lesser delay after signal falling edge than the delay of Freeze mode.
2 GPIO_EMC2_COMPEN	COMPEN and COMPTQ control the operating modes of the compensation cell Normal Mode: In this mode, the macro cell constantly tracks the PVT condition of the chip and generates an 8-bit digital code. This 8-bit code is referred to as f(PVT) and it represents current PVT state. In Normal mode, the internal reference current generators are active and power consumption is higher than that in all the other modes. Freeze mode: This mode is used where the current consumption is to be kept low. In this mode, all the internal blocks are switched off to reduce the static power consumption. Internal latches keep the code, which is calculated before the cell enters the Freeze mode. Fast Freeze mode: This mode is used to freeze the digital compensation codes, when the data is transferred from the compensated IOs on a chip to an external device during Burst mode. This ensures signal integrity and eliminates the jitter caused due to modification in IO driver strength. In this mode, the current consumption is comparable to consumption in the Normal mode. Read Mode: In this mode, it is possible to force the digital codes from the chip core logic. The bandgap, measurement block, and comparator blocks are switched off to reduce static power consumption to minimum value.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Fixed Code mode: The digital code is forced to a fixed value, obtained at typical PVT conditions and represents typical bit patterns. The bandgap, measurement, and other blocks are switched off to reduce static power consumption to minimum value.
<table>
<tr><th>COMPEN</th><th>COMPTQ</th><th>FREEZE</th><th>FASTFRZ_EN</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Normal Mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Freeze Mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fast Freeze Mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Read Mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed Code Mode</td></tr>
</table></td></tr>
<tr><td>1
GPIO_EMC2_COMPTQ</td><td>COMPEN and COMPTQ control the operating modes of the compensation cell</td></tr>
<tr><td>0
GPIO_EMC2_FREEZE</td><td>Compensation code freeze
When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, high, low, compensation cell Freeze mode is enabled. On the rising edge of FREEZE, it freezes the compensation code at its running value. The compensation cell delivers refreshed compensation code at a delay after FREEZE falling edge.</td></tr>
</table>

12.4.5.63 GPR67 General Purpose Register (GPR67)

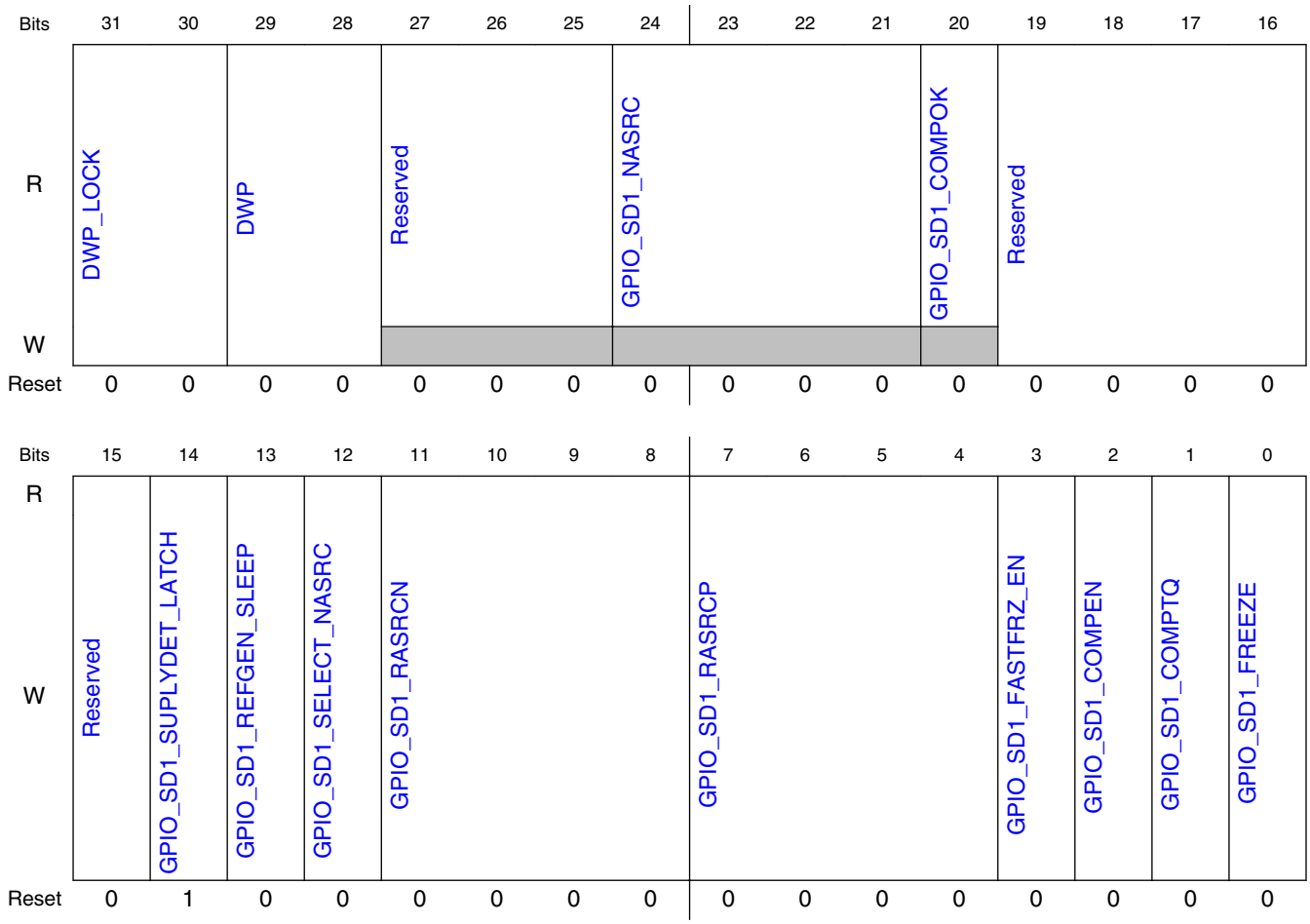
12.4.5.63.1 Offset

Register	Offset
GPR67	10Ch

12.4.5.63.2 Function

GPR Register

12.4.5.63.3 Diagram



12.4.5.63.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-25	Reserved

Table continues on the next page...

Field	Function
—	
24-21 GPIO_SD1_NASRC	GPIO_SD_B1 IO bank compensation codes 4-bit NMOS compensation codes or 4-bit PMOS compensation codes selected by GPIO_SD1_SELECT_NASRC.
20 GPIO_SD1_COMPOK	GPIO_SD_B1 IO bank compensation OK flag It can be high only in the Normal mode and when a new measured code is available.
19-15 —	Reserved
14 GPIO_SD1_SUPLYDET_LATCH	GPIO_SD_B1 IO bank power supply mode latch enable Supply detector cell is used to detect the IO supply range and set IO to 3V3 supply mode or 1V8 supply mode according to the detection. Setting this bit will latch the current detect result.
13 GPIO_SD1_REFGEN_SLEEP	GPIO_SD_B1 IO bank reference voltage generator cell sleep enable Reference voltage generator cell provides reference voltages to the IO bank. When this bit is high, it puts the reference voltage generator cell in sleep mode to reduce the static power consumption.
12 GPIO_SD1_SELECT_NASRC	GPIO_SD1_NASRC selection 0: Show the 4-bit PMOS compensation codes in GPIO_SD1_NASRC field 1: Show the 4-bit NMOS compensation codes in GPIO_SD1_NASRC field
11-8 GPIO_SD1_RASRCN	GPIO_SD_B1 IO bank's 4-bit NMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
7-4 GPIO_SD1_RASRCP	GPIO_SD_B1 IO bank's 4-bit PMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
3 GPIO_SD1_FASTFRZ_EN	Compensation code fast freeze When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, low, high, compensation cell Fast Freeze mode is enabled. Fast Freeze mode is activated on signal rising edge. The compensation cell delivers a refreshed compensation code, at a comparatively lesser delay after signal falling edge than the delay of Freeze mode.
2 GPIO_SD1_COMPEN	COMPEN and COMPTQ control the operating modes of the compensation cell Normal Mode: In this mode, the macro cell constantly tracks the PVT condition of the chip and generates an 8-bit digital code. This 8-bit code is referred to as f(PVT) and it represents current PVT state. In Normal mode, the internal reference current generators are active and power consumption is higher than that in all the other modes. Freeze mode: This mode is used where the current consumption is to be kept low. In this mode, all the internal blocks are switched off to reduce the static power consumption. Internal latches keep the code, which is calculated before the cell enters the Freeze mode. Fast Freeze mode: This mode is used to freeze the digital compensation codes, when the data is transferred from the compensated IOs on a chip to an external device during Burst mode. This ensures signal integrity and eliminates the jitter caused due to modification in IO driver strength. In this mode, the current consumption is comparable to consumption in the Normal mode. Read Mode: In this mode, it is possible to force the digital codes from the chip core logic. The bandgap, measurement block, and comparator blocks are switched off to reduce static power consumption to minimum value.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Fixed Code mode: The digital code is forced to a fixed value, obtained at typical PVT conditions and represents typical bit patterns. The bandgap, measurement, and other blocks are switched off to reduce static power consumption to minimum value.
<table>
<tr><th>COMPEN</th><th>COMPTQ</th><th>FREEZE</th><th>FASTFRZ_EN</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Normal Mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Freeze Mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fast Freeze Mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Read Mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed Code Mode</td></tr>
</table></td></tr>
<tr><td>1
GPIO_SD1_COMPTQ</td><td>COMPEN and COMPTQ control the operating modes of the compensation cell</td></tr>
<tr><td>0
GPIO_SD1_FREEZE</td><td>Compensation code freeze
When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, high, low, compensation cell Freeze mode is enabled. On the rising edge of FREEZE, it freezes the compensation code at its running value. The compensation cell delivers refreshed compensation code at a delay after FREEZE falling edge.</td></tr>
</table>

12.4.5.64 GPR68 General Purpose Register (GPR68)

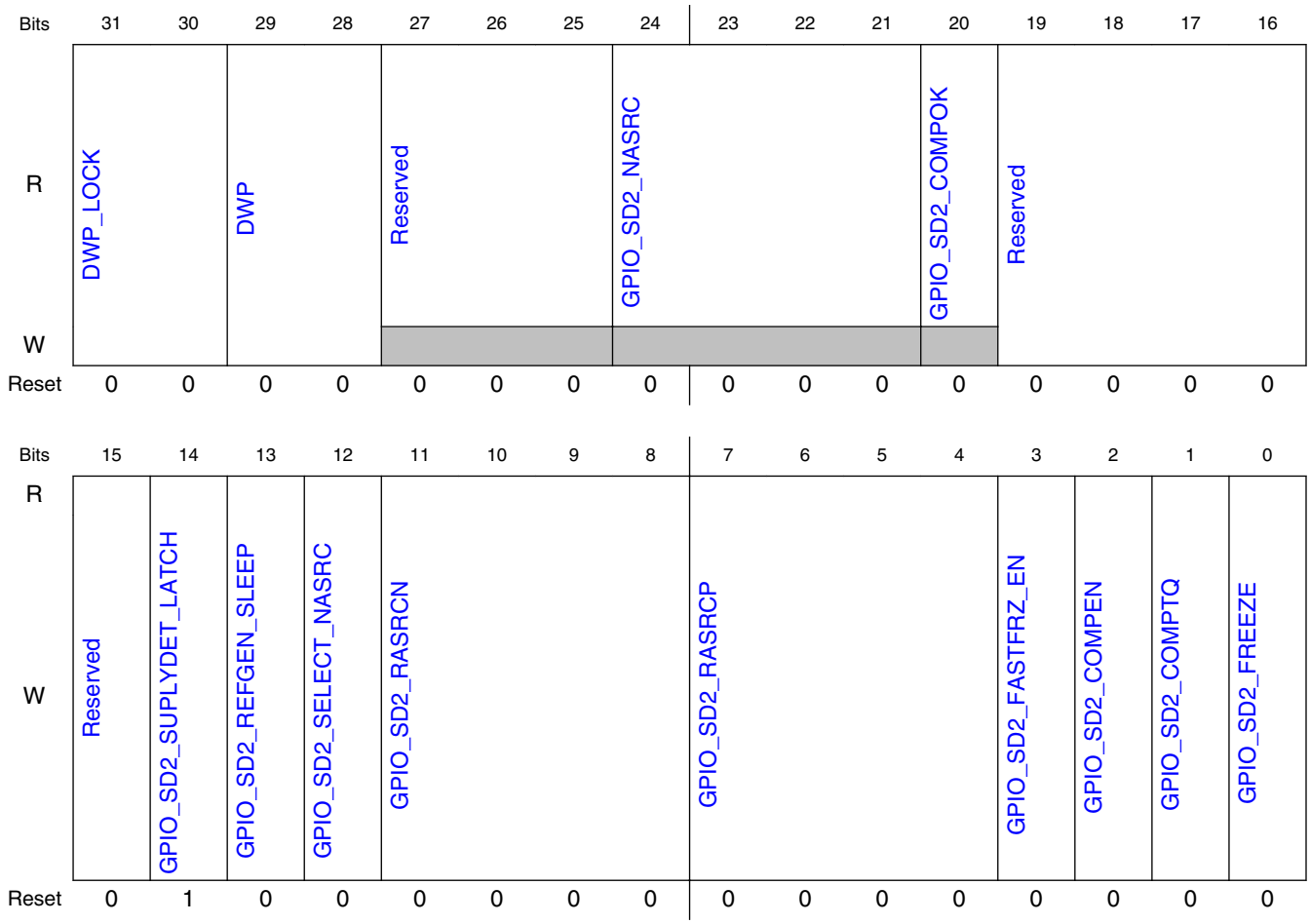
12.4.5.64.1 Offset

Register	Offset
GPR68	110h

12.4.5.64.2 Function

GPR Register

12.4.5.64.3 Diagram



12.4.5.64.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 19:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-25	Reserved

Table continues on the next page...

Field	Function
—	
24-21 GPIO_SD2_NASRC	GPIO_SD_B2 IO bank compensation codes 4-bit NMOS compensation codes or 4-bit PMOS compensation codes selected by GPIO_SD2_SELECT_NASRC.
20 GPIO_SD2_COMPOK	GPIO_SD_B2 IO bank compensation OK flag It can be high only in the Normal mode and when a new measured code is available.
19-15 —	Reserved
14 GPIO_SD2_SUPLYDET_LATCH	GPIO_SD_B2 IO bank power supply mode latch enable Supply detector cell is used to detect the IO supply range and set IO to 3V3 supply mode or 1V8 supply mode according to the detection. Setting this bit will latch the current detect result.
13 GPIO_SD2_REFGEN_SLEEP	GPIO_SD_B2 IO bank reference voltage generator cell sleep enable Reference voltage generator cell provides reference voltages to the IO bank. When this bit is high, it puts the reference voltage generator cell in sleep mode to reduce the static power consumption.
12 GPIO_SD2_SELECT_NASRC	GPIO_SD2_NASRC selection 0: Show the 4-bit PMOS compensation codes in GPIO_SD2_NASRC field 1: Show the 4-bit NMOS compensation codes in GPIO_SD2_NASRC field
11-8 GPIO_SD2_RASRCN	GPIO_SD_B2 IO bank's 4-bit NMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
7-4 GPIO_SD2_RASRCP	GPIO_SD_B2 IO bank's 4-bit PMOS compensation codes from core These codes are used in compensation Read mode when COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is high, high, low, low.
3 GPIO_SD2_FASTFRZ_EN	Compensation code fast freeze When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, low, high, compensation cell Fast Freeze mode is enabled. Fast Freeze mode is activated on signal rising edge. The compensation cell delivers a refreshed compensation code, at a comparatively lesser delay after signal falling edge than the delay of Freeze mode.
2 GPIO_SD2_COMPEN	COMPEN and COMPTQ control the operating modes of the compensation cell Normal Mode: In this mode, the macro cell constantly tracks the PVT condition of the chip and generates an 8-bit digital code. This 8-bit code is referred to as f(PVT) and it represents current PVT state. In Normal mode, the internal reference current generators are active and power consumption is higher than that in all the other modes. Freeze mode: This mode is used where the current consumption is to be kept low. In this mode, all the internal blocks are switched off to reduce the static power consumption. Internal latches keep the code, which is calculated before the cell enters the Freeze mode. Fast Freeze mode: This mode is used to freeze the digital compensation codes, when the data is transferred from the compensated IOs on a chip to an external device during Burst mode. This ensures signal integrity and eliminates the jitter caused due to modification in IO driver strength. In this mode, the current consumption is comparable to consumption in the Normal mode. Read Mode: In this mode, it is possible to force the digital codes from the chip core logic. The bandgap, measurement block, and comparator blocks are switched off to reduce static power consumption to minimum value.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Fixed Code mode: The digital code is forced to a fixed value, obtained at typical PVT conditions and represents typical bit patterns. The bandgap, measurement, and other blocks are switched off to reduce static power consumption to minimum value.
<table>
<tr><th>COMPEN</th><th>COMPTQ</th><th>FREEZE</th><th>FASTFRZ_EN</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Normal Mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Freeze Mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fast Freeze Mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Read Mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed Code Mode</td></tr>
</table></td></tr>
<tr><td>1
GPIO_SD2_COMPTQ</td><td>COMPEN and COMPTQ control the operating modes of the compensation cell</td></tr>
<tr><td>0
GPIO_SD2_FREEZE</td><td>Compensation code freeze
When COMPEN, COMPTQ, FREEZE, FASTFRZ_EN is low, low, high, low, compensation cell Freeze mode is enabled. On the rising edge of FREEZE, it freezes the compensation code at its running value. The compensation cell delivers refreshed compensation code at a delay after FREEZE falling edge.</td></tr>
</table>

12.4.5.65 GPR69 General Purpose Register (GPR69)

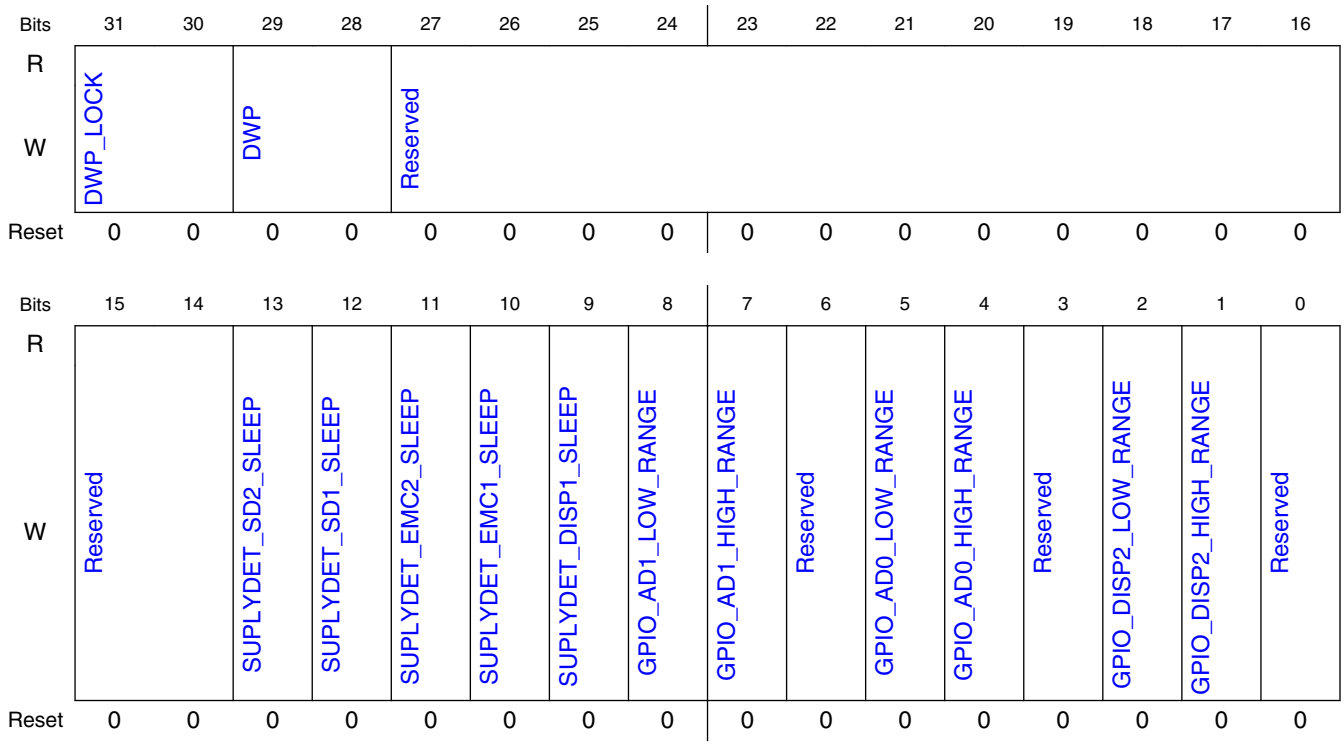
12.4.5.65.1 Offset

Register	Offset
GPR69	114h

12.4.5.65.2 Function

GPR Register

12.4.5.65.3 Diagram



12.4.5.65.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 15:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-16 —	Reserved
15-14 —	Reserved
13	GPIO_SD_B2 IO bank supply voltage detector sleep mode enable

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>SUPLYDET_SD2_SLEEP</td><td>Supply detector cell is used to detect the IO supply range. When this bit is high, it puts the cell in sleep mode to reduce the static power consumption. The detect result can be latched by setting SUPLYDET_LATCH bit before putting the cell in sleep mode.</td></tr>
<tr><td>12
SUPLYDET_SD1_SLEEP</td><td>GPIO_SD_B1 IO bank supply voltage detector sleep mode enable
Supply detector cell is used to detect the IO supply range. When this bit is high, it puts the cell in sleep mode to reduce the static power consumption. The detect result can be latched by setting SUPLYDET_LATCH bit before putting the cell in sleep mode.</td></tr>
<tr><td>11
SUPLYDET_EMC2_SLEEP</td><td>GPIO_EMC_B2 IO bank supply voltage detector sleep mode enable
Supply detector cell is used to detect the IO supply range. When this bit is high, it puts the cell in sleep mode to reduce the static power consumption. The detect result can be latched by setting SUPLYDET_LATCH bit before putting the cell in sleep mode.</td></tr>
<tr><td>10
SUPLYDET_EMC1_SLEEP</td><td>GPIO_EMC_B1 IO bank supply voltage detector sleep mode enable
Supply detector cell is used to detect the IO supply range. When this bit is high, it puts the cell in sleep mode to reduce the static power consumption. The detect result can be latched by setting SUPLYDET_LATCH bit before putting the cell in sleep mode.</td></tr>
<tr><td>9
SUPLYDET_DISP1_SLEEP</td><td>GPIO_DISP_B1 IO bank supply voltage detector sleep mode enable
Supply detector cell is used to detect the IO supply range. When this bit is high, it puts the supply detector cell in sleep mode to reduce the static power consumption. The detect result can be latched by setting SUPLYDET_LATCH bit before putting the cell in sleep mode.</td></tr>
<tr><td>8
GPIO_AD1_LOW_RANGE</td><td>GPIO_LPSR IO bank supply voltage range selection for GPIO_AD_18 to GPIO_AD_35
<table>
<tr><th>GPIO_AD1_HIGH_RANGE</th><th>GPIO_AD1_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_AD1_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_AD1_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_AD1_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
<tr><td>7
GPIO_AD1_HIGH_RANGE</td><td>GPIO_LPSR IO bank supply voltage range selection for GPIO_AD_18 to GPIO_AD_35
<table>
<tr><th>GPIO_AD0_HIGH_RANGE</th><th>GPIO_AD0_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_AD1_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_AD1_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_AD1_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>6

—</td><td>Reserved</td></tr>
<tr><td>5

GPIO_AD0_LOW_RANGE</td><td>GPIO_AD IO bank supply voltage range selection for GPIO_AD_00 to GPIO_AD_17

<table>
<tr><th>GPIO_AD0_HIGH_RANGE</th><th>GPIO_AD0_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_AD0_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_AD0_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_AD0_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
<tr><td>4

GPIO_AD0_HIGH_RANGE</td><td>GPIO_AD IO bank supply voltage range selection for GPIO_AD_00 to GPIO_AD_17

<table>
<tr><th>GPIO_AD0_HIGH_RANGE</th><th>GPIO_AD0_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_AD0_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_AD0_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_AD0_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
<tr><td>3

—</td><td>Reserved</td></tr>
<tr><td>2

GPIO_DISP2_LOW_RANGE</td><td>GPIO_DISP_B2 IO bank supply voltage range selection

<table>
<tr><th>GPIO_DISP_B2_HIGH_RANGE</th><th>GPIO_DISP_B2_LOW_RANGE</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>GPIO_DISP_B2_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr>
<tr><td>0</td><td>1</td><td>GPIO_DISP_B2_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr>
<tr><td>1</td><td>0</td><td>GPIO_DISP_B2_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr>
<tr><td>1</td><td>1</td><td>Not allowed</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>1
GPIO_DISP2_HIGH_RANGE</td><td>GPIO_DISP_B2 IO bank supply voltage range selection
<table><tr><th>GPIO_DISP_B2_HIGH_RANGE</th><th>GPIO_DISP_B2_LOW_RANGE</th><th>Mode</th></tr><tr><td>0</td><td>0</td><td>GPIO_DISP_B2_xx IO will work in continuous range mode with supply voltage in 1.71v-3.6v</td></tr><tr><td>0</td><td>1</td><td>GPIO_DISP_B2_xx IO will work in low range mode with supply voltage in 1.71v-1.98v</td></tr><tr><td>1</td><td>0</td><td>GPIO_DISP_B2_xx IO will work in high range mode with supply voltage in 3v-3.6v</td></tr><tr><td>1</td><td>1</td><td>Not allowed</td></tr></table></td></tr>
<tr><td>0
—</td><td>Reserved</td></tr>
</table>

12.4.5.66 GPR70 General Purpose Register (GPR70)

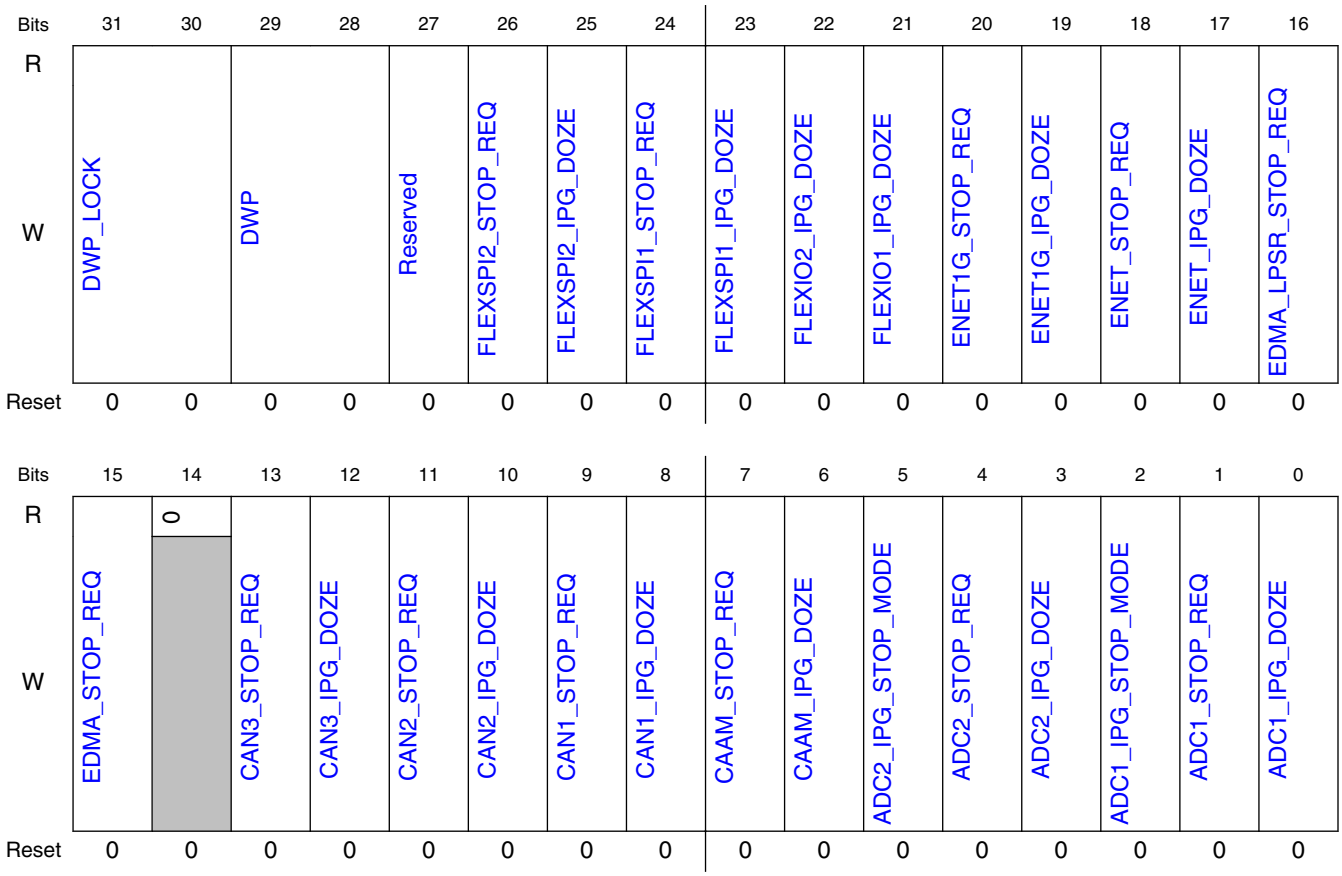
12.4.5.66.1 Offset

Register	Offset
GPR70	118h

12.4.5.66.2 Function

GPR Register

12.4.5.66.3 Diagram



12.4.5.66.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved
26	FLEXSPI2 stop request

Table continues on the next page...

Field	Function
FLEXSPI2_STOP_REQ	0: stop request off 1: stop request on
25 FLEXSPI2_IPG_DOZE	FLEXSPI2 doze mode 0: not in doze mode 1: in doze mode
24 FLEXSPI1_STOP_REQ	FLEXSPI1 stop request 0: stop request off 1: stop request on
23 FLEXSPI1_IPG_DOZE	FLEXSPI1 doze mode 0: not in doze mode 1: in doze mode
22 FLEXIO2_IPG_DOZE	FLEXIO2 doze mode 0: not in doze mode 1: in doze mode
21 FLEXIO1_IPG_DOZE	FLEXIO2 doze mode 0: not in doze mode 1: in doze mode
20 ENET1G_STOP_REQ	ENET1G stop request 0: stop request off 1: stop request on
19 ENET1G_IPG_DOZE	ENET1G doze mode 0: not in doze mode 1: in doze mode
18 ENET_STOP_REQ	ENET stop request 0: stop request off 1: stop request on
17 ENET_IPG_DOZE	ENET doze mode 0: not in doze mode 1: in doze mode
16 EDMA_LPSR_STOP_REQ	EDMA_LPSR stop request 0: stop request off 1: stop request on
15 EDMA_STOP_REQ	EDMA stop request 0: stop request off 1: stop request on
14 —	Reserved
13 CAN3_STOP_REQ	CAN3 stop request 0: stop request off

Table continues on the next page...

Field	Function
	1: stop request on
12 CAN3_IPG_DOZE	CAN3 doze mode 0: not in doze mode 1: in doze mode
11 CAN2_STOP_REQ	CAN2 stop request 0: stop request off 1: stop request on
10 CAN2_IPG_DOZE	CAN2 doze mode 0: not in doze mode 1: in doze mode
9 CAN1_STOP_REQ	CAN1 stop request 0: stop request off 1: stop request on
8 CAN1_IPG_DOZE	CAN1 doze mode 0: not in doze mode 1: in doze mode
7 CAAM_STOP_REQ	CAAM stop request 0: stop request off 1: stop request on
6 CAAM_IPG_DOZE	CAN3 doze mode 0: not in doze mode 1: in doze mode
5 ADC2_IPG_STOP_MODE	ADC2 stop mode selection, cannot change when ADC2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 ADC2_STOP_REQ	ADC2 stop request 0: stop request off 1: stop request on
3 ADC2_IPG_DOZE	ADC2 doze mode 0: not in doze mode 1: in doze mode
2 ADC1_IPG_STOP_MODE	ADC1 stop mode selection, cannot change when ADC1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1 ADC1_STOP_REQ	ADC1 stop request 0: stop request off 1: stop request on
0 ADC1_IPG_DOZE	ADC1 doze mode 0: not in doze mode

Field	Function
	1: in doze mode

12.4.5.67 GPR71 General Purpose Register (GPR71)

12.4.5.67.1 Offset

Register	Offset
GPR71	11Ch

12.4.5.67.2 Function

GPR Register

12.4.5.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DWP_LOCK		DWP		Reserved	LPSPI1_IPG_STOP_MODE	LPSPI1_STOP_REQ	LPSPI1_IPG_DOZE	LPI2C6_IPG_STOP_MODE	LPI2C6_STOP_REQ	LPI2C6_IPG_DOZE	LPI2C5_IPG_STOP_MODE	LPI2C5_STOP_REQ	LPI2C5_IPG_DOZE	LPI2C4_IPG_STOP_MODE	LPI2C4_STOP_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	LPI2C4_IPG_DOZE	LPI2C3_IPG_STOP_MODE	LPI2C3_STOP_REQ	LPI2C3_IPG_DOZE	LPI2C2_IPG_STOP_MODE	LPI2C2_STOP_REQ	LPI2C2_IPG_DOZE	LPI2C1_IPG_STOP_MODE	LPI2C1_STOP_REQ	LPI2C1_IPG_DOZE	GPT6_IPG_DOZE	GPT5_IPG_DOZE	GPT4_IPG_DOZE	GPT3_IPG_DOZE	GPT2_IPG_DOZE	GPT1_IPG_DOZE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.5.67.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved
26 LPSPI1_IPG_STOP_MODE	LPSPI1 stop mode selection, cannot change when LPSPI1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 LPSPI1_STOP_REQ	LPSPI1 stop request 0: stop request off 1: stop request on
24 LPSPI1_IPG_DOZE	LPSPI1 doze mode 0: not in doze mode 1: in doze mode
23 LPI2C6_IPG_STOP_MODE	LPI2C6 stop mode selection, cannot change when LPI2C6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPI2C6_STOP_REQ	LPI2C6 stop request 0: stop request off 1: stop request on
21 LPI2C6_IPG_DOZE	LPI2C6 doze mode 0: not in doze mode 1: in doze mode
20 LPI2C5_IPG_STOP_MODE	LPI2C5 stop mode selection, cannot change when LPI2C5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPI2C5_STOP_REQ	LPI2C5 stop request 0: stop request off 1: stop request on

Table continues on the next page...

Field	Function
18 LPI2C5_IPG_DOZE	LPI2C5 doze mode 0: not in doze mode 1: in doze mode
17 LPI2C4_IPG_STOP_MODE	LPI2C4 stop mode selection, cannot change when LPI2C4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPI2C4_STOP_REQ	LPI2C4 stop request 0: stop request off 1: stop request on
15 LPI2C4_IPG_DOZE	LPI2C4 doze mode 0: not in doze mode 1: in doze mode
14 LPI2C3_IPG_STOP_MODE	LPI2C3 stop mode selection, cannot change when LPI2C3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
13 LPI2C3_STOP_REQ	LPI2C3 stop request 0: stop request off 1: stop request on
12 LPI2C3_IPG_DOZE	LPI2C3 doze mode 0: not in doze mode 1: in doze mode
11 LPI2C2_IPG_STOP_MODE	LPI2C2 stop mode selection, cannot change when LPI2C2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPI2C2_STOP_REQ	LPI2C2 stop request 0: stop request off 1: stop request on
9 LPI2C2_IPG_DOZE	LPI2C2 doze mode 0: not in doze mode 1: in doze mode
8 LPI2C1_IPG_STOP_MODE	LPI2C1 stop mode selection, cannot change when LPI2C1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPI2C1_STOP_REQ	LPI2C1 stop request 0: stop request off 1: stop request on
6 LPI2C1_IPG_DOZE	LPI2C1 doze mode 0: not in doze mode 1: in doze mode

Table continues on the next page...

Field	Function
5 GPT6_IPG_DOZE	GPT6 doze mode 0: not in doze mode 1: in doze mode
4 GPT5_IPG_DOZE	GPT5 doze mode 0: not in doze mode 1: in doze mode
3 GPT4_IPG_DOZE	GPT4 doze mode 0: not in doze mode 1: in doze mode
2 GPT3_IPG_DOZE	GPT3 doze mode 0: not in doze mode 1: in doze mode
1 GPT2_IPG_DOZE	GPT2 doze mode 0: not in doze mode 1: in doze mode
0 GPT1_IPG_DOZE	GPT1 doze mode 0: not in doze mode 1: in doze mode

12.4.5.68 GPR72 General Purpose Register (GPR72)

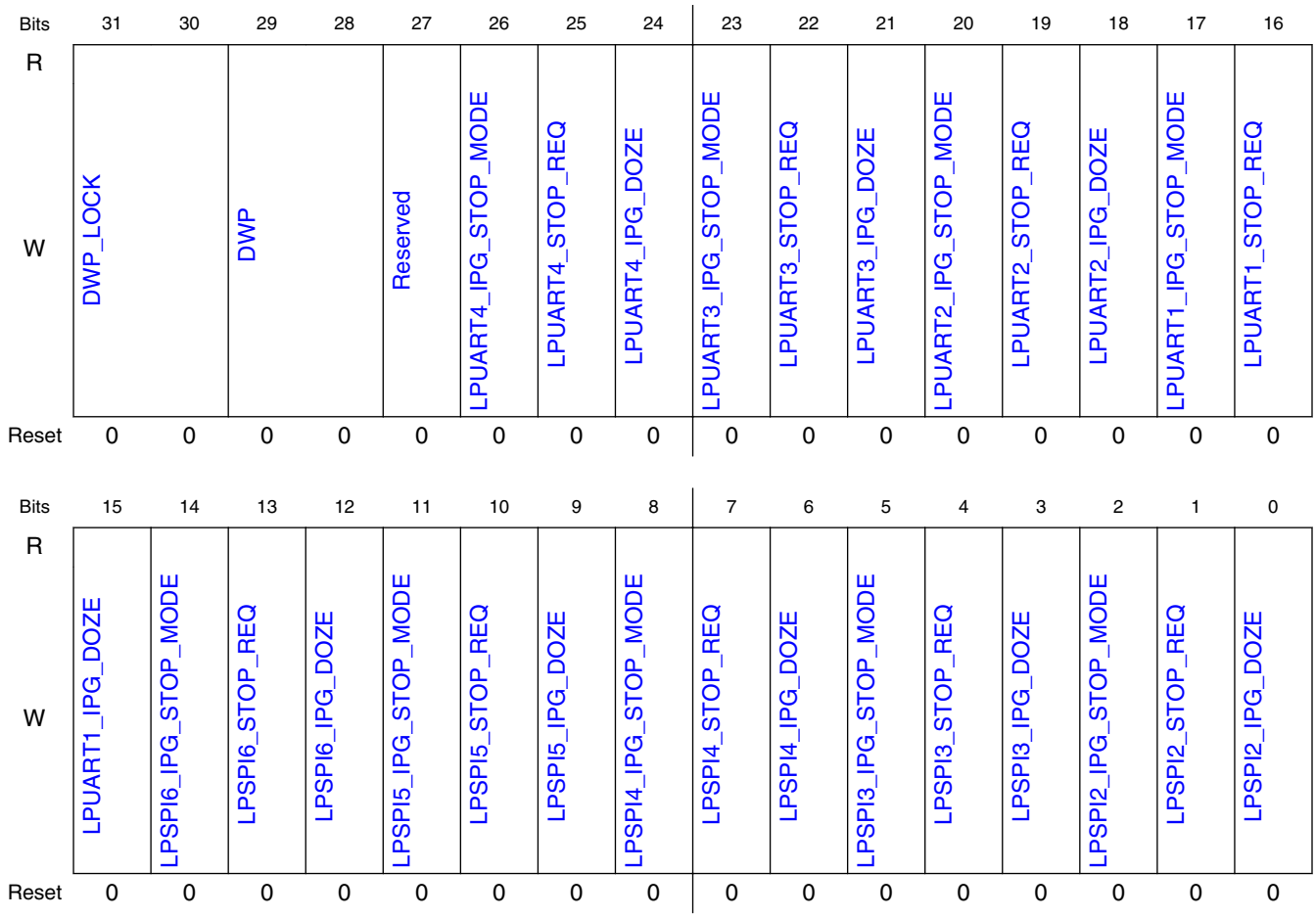
12.4.5.68.1 Offset

Register	Offset
GPR72	120h

12.4.5.68.2 Function

GPR Register

12.4.5.68.3 Diagram



12.4.5.68.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27 —	Reserved

Table continues on the next page...

Field	Function
26 LPUART4_IPG_STOP_MODE	LPUART4 stop mode selection, cannot change when LPUART4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 LPUART4_STOP_REQ	LPUART4 stop request 0: stop request off 1: stop request on
24 LPUART4_IPG_DOZE	LPUART4 doze mode 0: not in doze mode 1: in doze mode
23 LPUART3_IPG_STOP_MODE	LPUART3 stop mode selection, cannot change when LPUART3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPUART3_STOP_REQ	LPUART3 stop request 0: stop request off 1: stop request on
21 LPUART3_IPG_DOZE	LPUART3 doze mode 0: not in doze mode 1: in doze mode
20 LPUART2_IPG_STOP_MODE	LPUART2 stop mode selection, cannot change when LPUART2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPUART2_STOP_REQ	LPUART2 stop request 0: stop request off 1: stop request on
18 LPUART2_IPG_DOZE	LPUART2 doze mode 0: not in doze mode 1: in doze mode
17 LPUART1_IPG_STOP_MODE	LPUART1 stop mode selection, cannot change when LPUART1_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPUART1_STOP_REQ	LPUART1 stop request 0: stop request off 1: stop request on
15 LPUART1_IPG_DOZE	LPUART1 doze mode 0: not in doze mode 1: in doze mode
14 LPSPI6_IPG_STOP_MODE	LPSPI6 stop mode selection, cannot change when LPSPI6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.

Table continues on the next page...

Field	Function
13 LPSPI6_STOP_REQ	LPSPI6 stop request 0: stop request off 1: stop request on
12 LPSPI6_IPG_DOZE	LPSPI6 doze mode 0: not in doze mode 1: in doze mode
11 LPSPI5_IPG_STOP_MODE	LPSPI5 stop mode selection, cannot change when LPSPI5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPSPI5_STOP_REQ	LPSPI5 stop request 0: stop request off 1: stop request on
9 LPSPI5_IPG_DOZE	LPSPI5 doze mode 0: not in doze mode 1: in doze mode
8 LPSPI4_IPG_STOP_MODE	LPSPI4 stop mode selection, cannot change when LPSPI4_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPSPI4_STOP_REQ	LPSPI4 stop request 0: stop request off 1: stop request on
6 LPSPI4_IPG_DOZE	LPSPI4 doze mode 0: not in doze mode 1: in doze mode
5 LPSPI3_IPG_STOP_MODE	LPSPI3 stop mode selection, cannot change when LPSPI3_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 LPSPI3_STOP_REQ	LPSPI3 stop request 0: stop request off 1: stop request on
3 LPSPI3_IPG_DOZE	LPSPI3 doze mode 0: not in doze mode 1: in doze mode
2 LPSPI2_IPG_STOP_MODE	LPSPI2 stop mode selection, cannot change when LPSPI2_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1 LPSPI2_STOP_REQ	LPSPI2 stop request 0: stop request off 1: stop request on

Table continues on the next page...

Field	Function
0 LPSPI2_IPG_DOZE	LPSPI2 doze mode 0: not in doze mode 1: in doze mode

12.4.5.69 GPR73 General Purpose Register (GPR73)

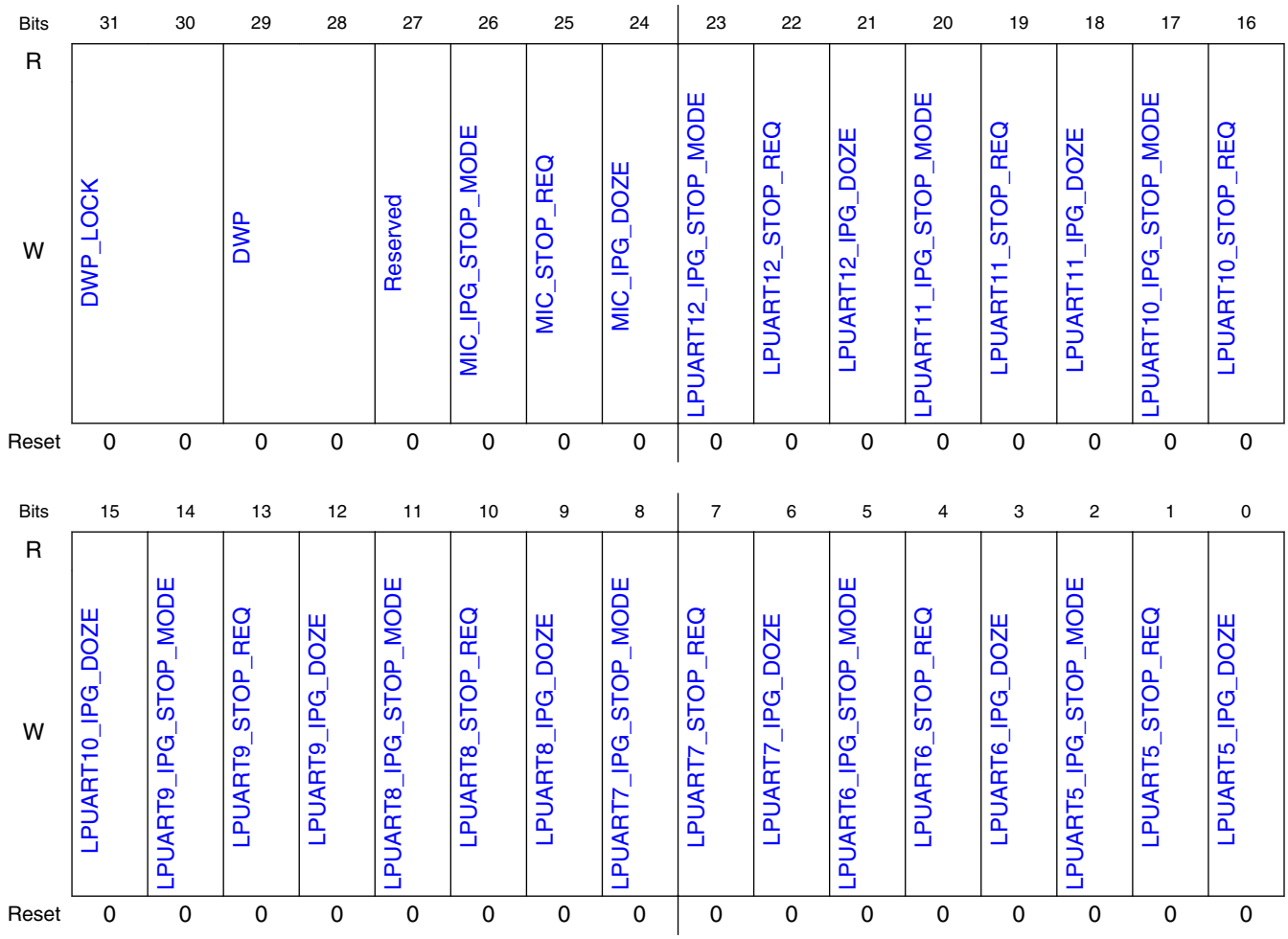
12.4.5.69.1 Offset

Register	Offset
GPR73	124h

12.4.5.69.2 Function

GPR Register

12.4.5.69.3 Diagram



12.4.5.69.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27	Reserved

Table continues on the next page...

Field	Function
—	
26 MIC_IPG_STOP_MODE	MIC stop mode selection, cannot change when MIC_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
25 MIC_STOP_REQ	MIC stop request 0: stop request off 1: stop request on
24 MIC_IPG_DOZE	MIC doze mode 0: not in doze mode 1: in doze mode
23 LPUART12_IPG_STOP_MODE	LPUART12 stop mode selection, cannot change when LPUART12_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
22 LPUART12_STOP_REQ	LPUART12 stop request 0: stop request off 1: stop request on
21 LPUART12_IPG_DOZE	LPUART12 doze mode 0: not in doze mode 1: in doze mode
20 LPUART11_IPG_STOP_MODE	LPUART11 stop mode selection, cannot change when LPUART11_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
19 LPUART11_STOP_REQ	LPUART11 stop request 0: stop request off 1: stop request on
18 LPUART11_IPG_DOZE	LPUART11 doze mode 0: not in doze mode 1: in doze mode
17 LPUART10_IPG_STOP_MODE	LPUART10 stop mode selection, cannot change when LPUART10_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
16 LPUART10_STOP_REQ	LPUART10 stop request 0: stop request off 1: stop request on
15 LPUART10_IPG_DOZE	LPUART10 doze mode 0: not in doze mode 1: in doze mode
14 LPUART9_IPG_STOP_MODE	LPUART9 stop mode selection, cannot change when LPUART9_STOP_REQ is asserted. 0b - This module is functional in Stop Mode

Table continues on the next page...

Field	Function
	1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
13 LPUART9_STOP_REQ	LPUART9 stop request 0: stop request off 1: stop request on
12 LPUART9_IPG_DOZE	LPUART9 doze mode 0: not in doze mode 1: in doze mode
11 LPUART8_IPG_STOP_MODE	LPUART8 stop mode selection, cannot change when LPUART8_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
10 LPUART8_STOP_REQ	LPUART8 stop request 0: stop request off 1: stop request on
9 LPUART8_IPG_DOZE	LPUART8 doze mode 0: not in doze mode 1: in doze mode
8 LPUART7_IPG_STOP_MODE	LPUART7 stop mode selection, cannot change when LPUART7_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
7 LPUART7_STOP_REQ	LPUART7 stop request 0: stop request off 1: stop request on
6 LPUART7_IPG_DOZE	LPUART7 doze mode 0: not in doze mode 1: in doze mode
5 LPUART6_IPG_STOP_MODE	LPUART6 stop mode selection, cannot change when LPUART6_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
4 LPUART6_STOP_REQ	LPUART6 stop request 0: stop request off 1: stop request on
3 LPUART6_IPG_DOZE	LPUART6 doze mode 0: not in doze mode 1: in doze mode
2 LPUART5_IPG_STOP_MODE	LPUART5 stop mode selection, cannot change when LPUART5_STOP_REQ is asserted. 0b - This module is functional in Stop Mode 1b - This module is not functional in Stop Mode and the corresponding x_STOP_REQ field is set to '1'.
1	LPUART5 stop request 0: stop request off

Table continues on the next page...

Field	Function
LPUART5_STOP_REQ	1: stop request on
0 LPUART5_IPG_DOZE	LPUART5 doze mode 0: not in doze mode 1: in doze mode

12.4.5.70 GPR74 General Purpose Register (GPR74)

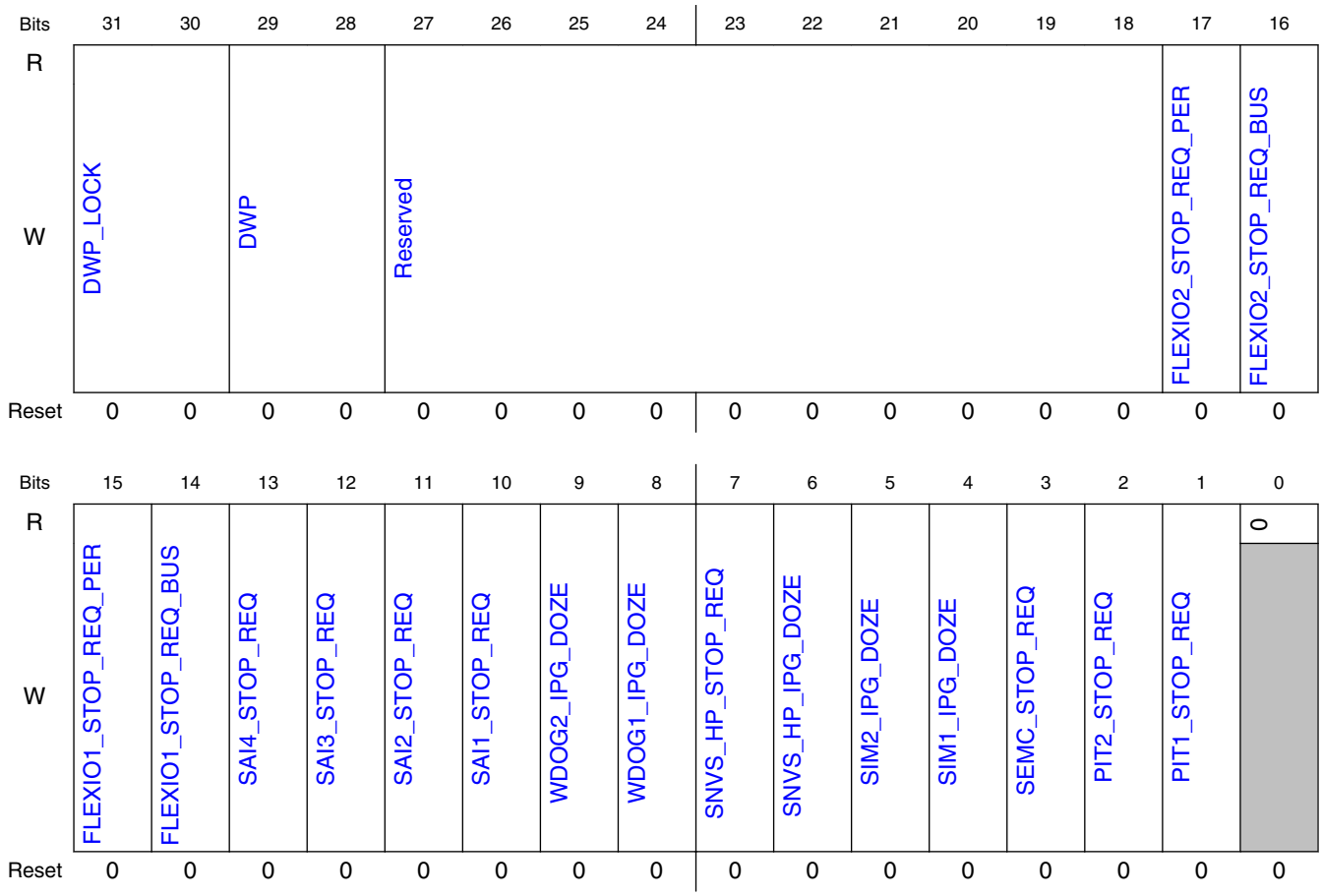
12.4.5.70.1 Offset

Register	Offset
GPR74	128h

12.4.5.70.2 Function

GPR Register

12.4.5.70.3 Diagram



12.4.5.70.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 27:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-18 —	Reserved

Table continues on the next page...

Field	Function
17 FLEXIO2_STOP_REQ_PER	FLEXIO2 peripheral clock domain stop request 0: stop request off 1: stop request on
16 FLEXIO2_STOP_REQ_BUS	FLEXIO2 bus clock domain stop request 0: stop request off 1: stop request on
15 FLEXIO1_STOP_REQ_PER	FLEXIO1 peripheral clock domain stop request 0: stop request off 1: stop request on
14 FLEXIO1_STOP_REQ_BUS	FLEXIO1 bus clock domain stop request 0: stop request off 1: stop request on
13 SAI4_STOP_REQ	SAI4 stop request 0: stop request off 1: stop request on
12 SAI3_STOP_REQ	SAI3 stop request 0: stop request off 1: stop request on
11 SAI2_STOP_REQ	SAI2 stop request 0: stop request off 1: stop request on
10 SAI1_STOP_REQ	SAI1 stop request 0: stop request off 1: stop request on
9 WDOG2_IPG_DOZE	WDOG2 doze mode 0: not in doze mode 1: in doze mode
8 WDOG1_IPG_DOZE	WDOG1 doze mode 0: not in doze mode 1: in doze mode
7 SNVS_HP_STOP_REQ	SNVS_HP stop request 0: stop request off 1: stop request on
6 SNVS_HP_IPG_DOZE	SNVS_HP doze mode 0: not in doze mode 1: in doze mode
5 SIM2_IPG_DOZE	SIM2 doze mode 0: not in doze mode 1: in doze mode

Table continues on the next page...

Field	Function
4 SIM1_IPG_DOZE	SIM1 doze mode 0: not in doze mode 1: in doze mode
3 SEMC_STOP_REQ	SEMC stop request 0: stop request off 1: stop request on
2 PIT2_STOP_REQ	PIT2 stop request 0: stop request off 1: stop request on
1 PIT1_STOP_REQ	PIT1 stop request 0: stop request off 1: stop request on
0 —	Reserved

12.4.5.71 GPR75 General Purpose Register (GPR75)

12.4.5.71.1 Offset

Register	Offset
GPR75	12Ch

12.4.5.71.2 Function

GPR Register

12.4.5.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LPUART8_STOP_ACK	LPUART7_STOP_ACK	LPUART6_STOP_ACK	LPUART5_STOP_ACK	LPUART4_STOP_ACK	LPUART3_STOP_ACK	LPUART2_STOP_ACK	LPUART1_STOP_ACK	LPSPI6_STOP_ACK	LPSPI5_STOP_ACK	LPSPI4_STOP_ACK	LPSPI3_STOP_ACK	LPSPI2_STOP_ACK	LPSPI1_STOP_ACK	LPI2C6_STOP_ACK	LPI2C5_STOP_ACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPI2C4_STOP_ACK	LPI2C3_STOP_ACK	LPI2C2_STOP_ACK	LPI2C1_STOP_ACK	FLEXSPI2_STOP_ACK	FLEXSPI1_STOP_ACK	ENET1G_STOP_ACK	ENET_STOP_ACK	EDMA_LPSR_STOP_ACK	EDMA_STOP_ACK	CAN3_STOP_ACK	CAN2_STOP_ACK	CAN1_STOP_ACK	CAAM_STOP_ACK	ADC2_STOP_ACK	ADC1_STOP_ACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.5.71.4 Fields

Field	Function
31 LPUART8_STOP_ACK	LPUART8 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
30 LPUART7_STOP_ACK	LPUART7 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
29 LPUART6_STOP_ACK	LPUART6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
28 LPUART5_STOP_ACK	LPUART5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
27 LPUART4_STOP_ACK	LPUART4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
26 LPUART3_STOP_ACK	LPUART3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
25 LPUART2_STOP_ACK	LPUART2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
24 LPUART1_STOP_ACK	LPUART1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
23 LPSPI6_STOP_ACK	LPSPI6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
22 LPSPI5_STOP_ACK	LPSPI5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
21 LPSPI4_STOP_ACK	LPSPI4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
20 LPSPI3_STOP_ACK	LPSPI3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
19 LPSPI2_STOP_ACK	LPSPI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
18 LPSPI1_STOP_ACK	LPSPI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
17 LPI2C6_STOP_ACK	LPI2C6 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
16 LPI2C5_STOP_ACK	LPI2C5 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
15 LPI2C4_STOP_ACK	LPI2C4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
14 LPI2C3_STOP_ACK	LPI2C3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
13 LPI2C2_STOP_ACK	LPI2C2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
12 LPI2C1_STOP_ACK	LPI2C1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
11 FLEXSPI2_STOP_ACK	FLEXSPI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
10 FLEXSPI1_STOP_ACK	FLEXSPI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
9 ENET1G_STOP_ACK	ENET1G stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
8 ENET_STOP_ACK	ENET stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
7 EDMA_LPSR_STOP_ACK	EDMA_LPSR stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
6 EDMA_STOP_ACK	EDMA stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
5 CAN3_STOP_ACK	CAN3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
4 CAN2_STOP_ACK	CAN2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
3 CAN1_STOP_ACK	CAN1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
2 CAAM_STOP_ACK	CAAM stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
1 ADC2_STOP_ACK	ADC2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
0 ADC1_STOP_ACK	ADC1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

12.4.5.72 GPR76 General Purpose Register (GPR76)

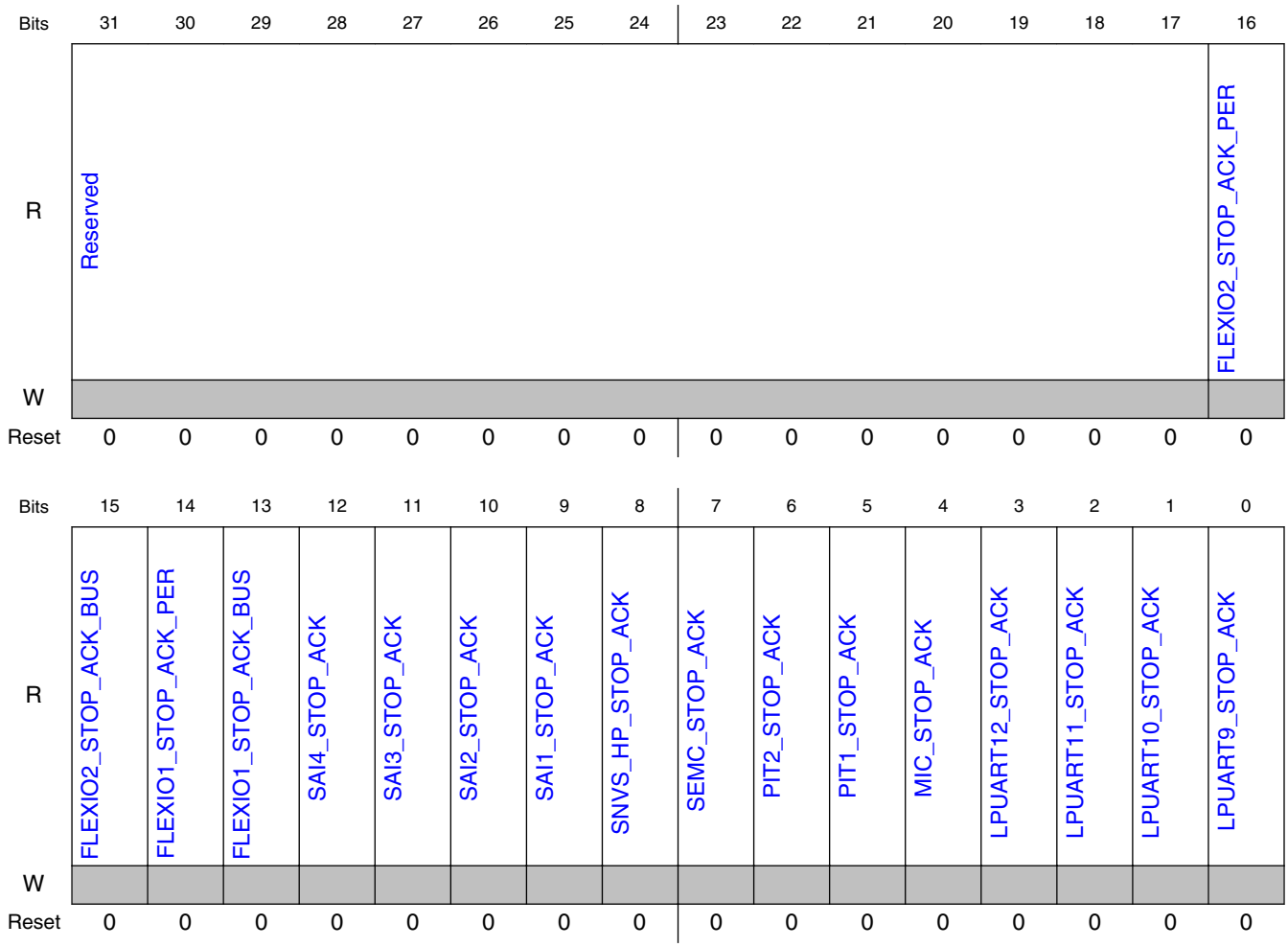
12.4.5.72.1 Offset

Register	Offset
GPR76	130h

12.4.5.72.2 Function

GPR Register

12.4.5.72.3 Diagram



12.4.5.72.4 Fields

Field	Function
31-17 —	Reserved
16 FLEXIO2_STOP_ACK_PER	FLEXIO2 stop acknowledge of peripheral clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
15 FLEXIO2_STOP_ACK_BUS	FLEXIO2 stop acknowledge of bus clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
14 FLEXIO1_STOP_ACK_PER	FLEXIO1 stop acknowledge of peripheral clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
13 FLEXIO1_STOP_ACK_BUS	FLEXIO1 stop acknowledge of bus clock domain 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
12 SAI4_STOP_ACK	SAI4 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
11 SAI3_STOP_ACK	SAI3 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
10 SAI2_STOP_ACK	SAI2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
9 SAI1_STOP_ACK	SAI1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
8 SNVS_HP_STOP_ACK	SNVS_HP stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
7 SEMC_STOP_ACK	SEMC stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
6 PIT2_STOP_ACK	PIT2 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
5 PIT1_STOP_ACK	PIT1 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
4 MIC_STOP_ACK	MIC stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
3 LPUART12_STOP_ACK	LPUART12 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
2 LPUART11_STOP_ACK	LPUART11 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)
1 LPUART10_STOP_ACK	LPUART10 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

Table continues on the next page...

Field	Function
0 LPUART9_STOP_ACK	LPUART9 stop acknowledge 0: stop acknowledge is not asserted 1: stop acknowledge is asserted (the module is in Stop mode)

12.4.6 IOMUXC register descriptions

12.4.6.1 iomuxc memory map

IOMUXC base address: 400E_8000h

Offset	Register	Width (In bits)	Access	Reset value
10h	SW_MUX_CTL_PAD_GPIO_EMC_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_00)	32	RW	0000_0005h
14h	SW_MUX_CTL_PAD_GPIO_EMC_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_01)	32	RW	0000_0005h
18h	SW_MUX_CTL_PAD_GPIO_EMC_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_02)	32	RW	0000_0005h
1Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_03)	32	RW	0000_0005h
20h	SW_MUX_CTL_PAD_GPIO_EMC_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_04)	32	RW	0000_0005h
24h	SW_MUX_CTL_PAD_GPIO_EMC_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_05)	32	RW	0000_0005h
28h	SW_MUX_CTL_PAD_GPIO_EMC_B1_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_06)	32	RW	0000_0005h
2Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_07)	32	RW	0000_0005h
30h	SW_MUX_CTL_PAD_GPIO_EMC_B1_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_08)	32	RW	0000_0005h
34h	SW_MUX_CTL_PAD_GPIO_EMC_B1_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_09)	32	RW	0000_0005h
38h	SW_MUX_CTL_PAD_GPIO_EMC_B1_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_10)	32	RW	0000_0005h
3Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_11)	32	RW	0000_0005h
40h	SW_MUX_CTL_PAD_GPIO_EMC_B1_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_12)	32	RW	0000_0005h
44h	SW_MUX_CTL_PAD_GPIO_EMC_B1_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_13)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
48h	SW_MUX_CTL_PAD_GPIO_EMC_B1_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_14)	32	RW	0000_0005h
4Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_15)	32	RW	0000_0005h
50h	SW_MUX_CTL_PAD_GPIO_EMC_B1_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_16)	32	RW	0000_0005h
54h	SW_MUX_CTL_PAD_GPIO_EMC_B1_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_17)	32	RW	0000_0005h
58h	SW_MUX_CTL_PAD_GPIO_EMC_B1_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_18)	32	RW	0000_0005h
5Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_19)	32	RW	0000_0005h
60h	SW_MUX_CTL_PAD_GPIO_EMC_B1_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_20)	32	RW	0000_0005h
64h	SW_MUX_CTL_PAD_GPIO_EMC_B1_21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_21)	32	RW	0000_0005h
68h	SW_MUX_CTL_PAD_GPIO_EMC_B1_22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_22)	32	RW	0000_0005h
6Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_23)	32	RW	0000_0005h
70h	SW_MUX_CTL_PAD_GPIO_EMC_B1_24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_24)	32	RW	0000_0005h
74h	SW_MUX_CTL_PAD_GPIO_EMC_B1_25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_25)	32	RW	0000_0005h
78h	SW_MUX_CTL_PAD_GPIO_EMC_B1_26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_26)	32	RW	0000_0005h
7Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_27)	32	RW	0000_0005h
80h	SW_MUX_CTL_PAD_GPIO_EMC_B1_28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_28)	32	RW	0000_0005h
84h	SW_MUX_CTL_PAD_GPIO_EMC_B1_29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_29)	32	RW	0000_0005h
88h	SW_MUX_CTL_PAD_GPIO_EMC_B1_30 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_30)	32	RW	0000_0005h
8Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_31 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_31)	32	RW	0000_0005h
90h	SW_MUX_CTL_PAD_GPIO_EMC_B1_32 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_32)	32	RW	0000_0005h
94h	SW_MUX_CTL_PAD_GPIO_EMC_B1_33 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_33)	32	RW	0000_0005h
98h	SW_MUX_CTL_PAD_GPIO_EMC_B1_34 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_34)	32	RW	0000_0005h
9Ch	SW_MUX_CTL_PAD_GPIO_EMC_B1_35 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_35)	32	RW	0000_0005h
A0h	SW_MUX_CTL_PAD_GPIO_EMC_B1_36 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_36)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
A4h	SW_MUX_CTL_PAD_GPIO_EMC_B1_37 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_37)	32	RW	0000_0005h
A8h	SW_MUX_CTL_PAD_GPIO_EMC_B1_38 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_38)	32	RW	0000_0005h
ACh	SW_MUX_CTL_PAD_GPIO_EMC_B1_39 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_39)	32	RW	0000_0005h
B0h	SW_MUX_CTL_PAD_GPIO_EMC_B1_40 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_40)	32	RW	0000_0005h
B4h	SW_MUX_CTL_PAD_GPIO_EMC_B1_41 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_41)	32	RW	0000_0005h
B8h	SW_MUX_CTL_PAD_GPIO_EMC_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_00)	32	RW	0000_0005h
BCh	SW_MUX_CTL_PAD_GPIO_EMC_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_01)	32	RW	0000_0005h
C0h	SW_MUX_CTL_PAD_GPIO_EMC_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_02)	32	RW	0000_0005h
C4h	SW_MUX_CTL_PAD_GPIO_EMC_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_03)	32	RW	0000_0005h
C8h	SW_MUX_CTL_PAD_GPIO_EMC_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_04)	32	RW	0000_0005h
CCh	SW_MUX_CTL_PAD_GPIO_EMC_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_05)	32	RW	0000_0005h
D0h	SW_MUX_CTL_PAD_GPIO_EMC_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_06)	32	RW	0000_0005h
D4h	SW_MUX_CTL_PAD_GPIO_EMC_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_07)	32	RW	0000_0005h
D8h	SW_MUX_CTL_PAD_GPIO_EMC_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_08)	32	RW	0000_0005h
DCh	SW_MUX_CTL_PAD_GPIO_EMC_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_09)	32	RW	0000_0005h
E0h	SW_MUX_CTL_PAD_GPIO_EMC_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_10)	32	RW	0000_0005h
E4h	SW_MUX_CTL_PAD_GPIO_EMC_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_11)	32	RW	0000_0005h
E8h	SW_MUX_CTL_PAD_GPIO_EMC_B2_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_12)	32	RW	0000_0005h
ECh	SW_MUX_CTL_PAD_GPIO_EMC_B2_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_13)	32	RW	0000_0005h
F0h	SW_MUX_CTL_PAD_GPIO_EMC_B2_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_14)	32	RW	0000_0005h
F4h	SW_MUX_CTL_PAD_GPIO_EMC_B2_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_15)	32	RW	0000_0005h
F8h	SW_MUX_CTL_PAD_GPIO_EMC_B2_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_16)	32	RW	0000_0005h
FCh	SW_MUX_CTL_PAD_GPIO_EMC_B2_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_17)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
100h	SW_MUX_CTL_PAD_GPIO_EMC_B2_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_18)	32	RW	0000_0005h
104h	SW_MUX_CTL_PAD_GPIO_EMC_B2_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_19)	32	RW	0000_0005h
108h	SW_MUX_CTL_PAD_GPIO_EMC_B2_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_20)	32	RW	0000_0005h
10Ch	SW_MUX_CTL_PAD_GPIO_AD_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_00)	32	RW	0000_0005h
110h	SW_MUX_CTL_PAD_GPIO_AD_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_01)	32	RW	0000_0005h
114h	SW_MUX_CTL_PAD_GPIO_AD_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_02)	32	RW	0000_0005h
118h	SW_MUX_CTL_PAD_GPIO_AD_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_03)	32	RW	0000_0005h
11Ch	SW_MUX_CTL_PAD_GPIO_AD_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_04)	32	RW	0000_0005h
120h	SW_MUX_CTL_PAD_GPIO_AD_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_05)	32	RW	0000_0005h
124h	SW_MUX_CTL_PAD_GPIO_AD_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_06)	32	RW	0000_0005h
128h	SW_MUX_CTL_PAD_GPIO_AD_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_07)	32	RW	0000_0005h
12Ch	SW_MUX_CTL_PAD_GPIO_AD_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_08)	32	RW	0000_0005h
130h	SW_MUX_CTL_PAD_GPIO_AD_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_09)	32	RW	0000_0005h
134h	SW_MUX_CTL_PAD_GPIO_AD_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_10)	32	RW	0000_0005h
138h	SW_MUX_CTL_PAD_GPIO_AD_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_11)	32	RW	0000_0005h
13Ch	SW_MUX_CTL_PAD_GPIO_AD_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_12)	32	RW	0000_0005h
140h	SW_MUX_CTL_PAD_GPIO_AD_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_13)	32	RW	0000_0005h
144h	SW_MUX_CTL_PAD_GPIO_AD_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_14)	32	RW	0000_0005h
148h	SW_MUX_CTL_PAD_GPIO_AD_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_15)	32	RW	0000_0005h
14Ch	SW_MUX_CTL_PAD_GPIO_AD_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_16)	32	RW	0000_0005h
150h	SW_MUX_CTL_PAD_GPIO_AD_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_17)	32	RW	0000_0005h
154h	SW_MUX_CTL_PAD_GPIO_AD_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_18)	32	RW	0000_0005h
158h	SW_MUX_CTL_PAD_GPIO_AD_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_19)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
15Ch	SW_MUX_CTL_PAD_GPIO_AD_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_20)	32	RW	0000_0005h
160h	SW_MUX_CTL_PAD_GPIO_AD_21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_21)	32	RW	0000_0005h
164h	SW_MUX_CTL_PAD_GPIO_AD_22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_22)	32	RW	0000_0005h
168h	SW_MUX_CTL_PAD_GPIO_AD_23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_23)	32	RW	0000_0005h
16Ch	SW_MUX_CTL_PAD_GPIO_AD_24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_24)	32	RW	0000_0005h
170h	SW_MUX_CTL_PAD_GPIO_AD_25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_25)	32	RW	0000_0005h
174h	SW_MUX_CTL_PAD_GPIO_AD_26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_26)	32	RW	0000_0005h
178h	SW_MUX_CTL_PAD_GPIO_AD_27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_27)	32	RW	0000_0005h
17Ch	SW_MUX_CTL_PAD_GPIO_AD_28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_28)	32	RW	0000_0005h
180h	SW_MUX_CTL_PAD_GPIO_AD_29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_29)	32	RW	0000_0005h
184h	SW_MUX_CTL_PAD_GPIO_AD_30 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_30)	32	RW	0000_0005h
188h	SW_MUX_CTL_PAD_GPIO_AD_31 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_31)	32	RW	0000_0005h
18Ch	SW_MUX_CTL_PAD_GPIO_AD_32 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_32)	32	RW	0000_0005h
190h	SW_MUX_CTL_PAD_GPIO_AD_33 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_33)	32	RW	0000_0005h
194h	SW_MUX_CTL_PAD_GPIO_AD_34 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_34)	32	RW	0000_0005h
198h	SW_MUX_CTL_PAD_GPIO_AD_35 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_35)	32	RW	0000_0005h
19Ch	SW_MUX_CTL_PAD_GPIO_SD_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_00)	32	RW	0000_0005h
1A0h	SW_MUX_CTL_PAD_GPIO_SD_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_01)	32	RW	0000_0005h
1A4h	SW_MUX_CTL_PAD_GPIO_SD_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_02)	32	RW	0000_0005h
1A8h	SW_MUX_CTL_PAD_GPIO_SD_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_03)	32	RW	0000_0005h
1ACh	SW_MUX_CTL_PAD_GPIO_SD_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_04)	32	RW	0000_0005h
1B0h	SW_MUX_CTL_PAD_GPIO_SD_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_05)	32	RW	0000_0005h
1B4h	SW_MUX_CTL_PAD_GPIO_SD_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_00)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1B8h	SW_MUX_CTL_PAD_GPIO_SD_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_01)	32	RW	0000_0005h
1BCh	SW_MUX_CTL_PAD_GPIO_SD_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_02)	32	RW	0000_0005h
1C0h	SW_MUX_CTL_PAD_GPIO_SD_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_03)	32	RW	0000_0005h
1C4h	SW_MUX_CTL_PAD_GPIO_SD_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_04)	32	RW	0000_0005h
1C8h	SW_MUX_CTL_PAD_GPIO_SD_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_05)	32	RW	0000_0005h
1CCh	SW_MUX_CTL_PAD_GPIO_SD_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_06)	32	RW	0000_0005h
1D0h	SW_MUX_CTL_PAD_GPIO_SD_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_07)	32	RW	0000_0005h
1D4h	SW_MUX_CTL_PAD_GPIO_SD_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_08)	32	RW	0000_0005h
1D8h	SW_MUX_CTL_PAD_GPIO_SD_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_09)	32	RW	0000_0005h
1DCh	SW_MUX_CTL_PAD_GPIO_SD_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_10)	32	RW	0000_0005h
1E0h	SW_MUX_CTL_PAD_GPIO_SD_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_11)	32	RW	0000_0005h
1E4h	SW_MUX_CTL_PAD_GPIO_DISP_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_00)	32	RW	0000_0005h
1E8h	SW_MUX_CTL_PAD_GPIO_DISP_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_01)	32	RW	0000_0005h
1ECh	SW_MUX_CTL_PAD_GPIO_DISP_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_02)	32	RW	0000_0005h
1F0h	SW_MUX_CTL_PAD_GPIO_DISP_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_03)	32	RW	0000_0005h
1F4h	SW_MUX_CTL_PAD_GPIO_DISP_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_04)	32	RW	0000_0005h
1F8h	SW_MUX_CTL_PAD_GPIO_DISP_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_05)	32	RW	0000_0005h
1FCh	SW_MUX_CTL_PAD_GPIO_DISP_B1_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_06)	32	RW	0000_0005h
200h	SW_MUX_CTL_PAD_GPIO_DISP_B1_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_07)	32	RW	0000_0005h
204h	SW_MUX_CTL_PAD_GPIO_DISP_B1_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_08)	32	RW	0000_0005h
208h	SW_MUX_CTL_PAD_GPIO_DISP_B1_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_09)	32	RW	0000_0005h
20Ch	SW_MUX_CTL_PAD_GPIO_DISP_B1_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_10)	32	RW	0000_0005h
210h	SW_MUX_CTL_PAD_GPIO_DISP_B1_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_11)	32	RW	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
214h	SW_MUX_CTL_PAD_GPIO_DISP_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_00)	32	RW	0000_0005h
218h	SW_MUX_CTL_PAD_GPIO_DISP_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_01)	32	RW	0000_0005h
21Ch	SW_MUX_CTL_PAD_GPIO_DISP_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_02)	32	RW	0000_0005h
220h	SW_MUX_CTL_PAD_GPIO_DISP_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_03)	32	RW	0000_0005h
224h	SW_MUX_CTL_PAD_GPIO_DISP_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_04)	32	RW	0000_0005h
228h	SW_MUX_CTL_PAD_GPIO_DISP_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_05)	32	RW	0000_0005h
22Ch	SW_MUX_CTL_PAD_GPIO_DISP_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_06)	32	RW	0000_0005h
230h	SW_MUX_CTL_PAD_GPIO_DISP_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_07)	32	RW	0000_0005h
234h	SW_MUX_CTL_PAD_GPIO_DISP_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_08)	32	RW	0000_0005h
238h	SW_MUX_CTL_PAD_GPIO_DISP_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_09)	32	RW	0000_0005h
23Ch	SW_MUX_CTL_PAD_GPIO_DISP_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_10)	32	RW	0000_0005h
240h	SW_MUX_CTL_PAD_GPIO_DISP_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_11)	32	RW	0000_0005h
244h	SW_MUX_CTL_PAD_GPIO_DISP_B2_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_12)	32	RW	0000_0005h
248h	SW_MUX_CTL_PAD_GPIO_DISP_B2_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_13)	32	RW	0000_0005h
24Ch	SW_MUX_CTL_PAD_GPIO_DISP_B2_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_14)	32	RW	0000_0005h
250h	SW_MUX_CTL_PAD_GPIO_DISP_B2_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_15)	32	RW	0000_0005h
254h	SW_PAD_CTL_PAD_GPIO_EMC_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_00)	32	RW	0000_0008h
258h	SW_PAD_CTL_PAD_GPIO_EMC_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_01)	32	RW	0000_0008h
25Ch	SW_PAD_CTL_PAD_GPIO_EMC_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_02)	32	RW	0000_0008h
260h	SW_PAD_CTL_PAD_GPIO_EMC_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_03)	32	RW	0000_0008h
264h	SW_PAD_CTL_PAD_GPIO_EMC_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_04)	32	RW	0000_0008h
268h	SW_PAD_CTL_PAD_GPIO_EMC_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_05)	32	RW	0000_0008h
26Ch	SW_PAD_CTL_PAD_GPIO_EMC_B1_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_06)	32	RW	0000_0008h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
270h	SW_PAD_CTL_PAD_GPIO_EMC_B1_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_07)	32	RW	0000_0008h
274h	SW_PAD_CTL_PAD_GPIO_EMC_B1_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_08)	32	RW	0000_0008h
278h	SW_PAD_CTL_PAD_GPIO_EMC_B1_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_09)	32	RW	0000_0008h
27Ch	SW_PAD_CTL_PAD_GPIO_EMC_B1_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_10)	32	RW	0000_0008h
280h	SW_PAD_CTL_PAD_GPIO_EMC_B1_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_11)	32	RW	0000_0008h
284h	SW_PAD_CTL_PAD_GPIO_EMC_B1_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_12)	32	RW	0000_0008h
288h	SW_PAD_CTL_PAD_GPIO_EMC_B1_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_13)	32	RW	0000_0008h
28Ch	SW_PAD_CTL_PAD_GPIO_EMC_B1_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_14)	32	RW	0000_0008h
290h	SW_PAD_CTL_PAD_GPIO_EMC_B1_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_15)	32	RW	0000_0008h
294h	SW_PAD_CTL_PAD_GPIO_EMC_B1_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_16)	32	RW	0000_0008h
298h	SW_PAD_CTL_PAD_GPIO_EMC_B1_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_17)	32	RW	0000_0008h
29Ch	SW_PAD_CTL_PAD_GPIO_EMC_B1_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_18)	32	RW	0000_0008h
2A0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_19)	32	RW	0000_0008h
2A4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_20)	32	RW	0000_0008h
2A8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_21)	32	RW	0000_0008h
2ACh	SW_PAD_CTL_PAD_GPIO_EMC_B1_22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_22)	32	RW	0000_0008h
2B0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_23)	32	RW	0000_0008h
2B4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_24)	32	RW	0000_0008h
2B8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_25)	32	RW	0000_0008h
2BCh	SW_PAD_CTL_PAD_GPIO_EMC_B1_26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_26)	32	RW	0000_0008h
2C0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_27)	32	RW	0000_0008h
2C4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_28)	32	RW	0000_0008h
2C8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_29)	32	RW	0000_0008h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2CCh	SW_PAD_CTL_PAD_GPIO_EMC_B1_30 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_30)	32	RW	0000_0008h
2D0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_31 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_31)	32	RW	0000_0008h
2D4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_32 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_32)	32	RW	0000_0008h
2D8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_33 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_33)	32	RW	0000_0008h
2DCh	SW_PAD_CTL_PAD_GPIO_EMC_B1_34 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_34)	32	RW	0000_0008h
2E0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_35 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_35)	32	RW	0000_0008h
2E4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_36 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_36)	32	RW	0000_0008h
2E8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_37 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_37)	32	RW	0000_0008h
2ECh	SW_PAD_CTL_PAD_GPIO_EMC_B1_38 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_38)	32	RW	0000_0008h
2F0h	SW_PAD_CTL_PAD_GPIO_EMC_B1_39 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_39)	32	RW	0000_0008h
2F4h	SW_PAD_CTL_PAD_GPIO_EMC_B1_40 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_40)	32	RW	0000_0008h
2F8h	SW_PAD_CTL_PAD_GPIO_EMC_B1_41 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_41)	32	RW	0000_0008h
2FCh	SW_PAD_CTL_PAD_GPIO_EMC_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_00)	32	RW	0000_0008h
300h	SW_PAD_CTL_PAD_GPIO_EMC_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_01)	32	RW	0000_0008h
304h	SW_PAD_CTL_PAD_GPIO_EMC_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_02)	32	RW	0000_0008h
308h	SW_PAD_CTL_PAD_GPIO_EMC_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_03)	32	RW	0000_0008h
30Ch	SW_PAD_CTL_PAD_GPIO_EMC_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_04)	32	RW	0000_0008h
310h	SW_PAD_CTL_PAD_GPIO_EMC_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_05)	32	RW	0000_0008h
314h	SW_PAD_CTL_PAD_GPIO_EMC_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_06)	32	RW	0000_0008h
318h	SW_PAD_CTL_PAD_GPIO_EMC_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_07)	32	RW	0000_0008h
31Ch	SW_PAD_CTL_PAD_GPIO_EMC_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_08)	32	RW	0000_0004h
320h	SW_PAD_CTL_PAD_GPIO_EMC_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_09)	32	RW	0000_0008h
324h	SW_PAD_CTL_PAD_GPIO_EMC_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_10)	32	RW	0000_0008h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
328h	SW_PAD_CTL_PAD_GPIO_EMC_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_11)	32	RW	0000_0004h
32Ch	SW_PAD_CTL_PAD_GPIO_EMC_B2_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_12)	32	RW	0000_0008h
330h	SW_PAD_CTL_PAD_GPIO_EMC_B2_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_13)	32	RW	0000_0008h
334h	SW_PAD_CTL_PAD_GPIO_EMC_B2_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_14)	32	RW	0000_0008h
338h	SW_PAD_CTL_PAD_GPIO_EMC_B2_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_15)	32	RW	0000_0008h
33Ch	SW_PAD_CTL_PAD_GPIO_EMC_B2_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_16)	32	RW	0000_0008h
340h	SW_PAD_CTL_PAD_GPIO_EMC_B2_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_17)	32	RW	0000_0008h
344h	SW_PAD_CTL_PAD_GPIO_EMC_B2_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_18)	32	RW	0000_0008h
348h	SW_PAD_CTL_PAD_GPIO_EMC_B2_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_19)	32	RW	0000_0008h
34Ch	SW_PAD_CTL_PAD_GPIO_EMC_B2_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_20)	32	RW	0000_0008h
350h	SW_PAD_CTL_PAD_GPIO_AD_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_00)	32	RW	0000_000Eh
354h	SW_PAD_CTL_PAD_GPIO_AD_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_01)	32	RW	0000_000Eh
358h	SW_PAD_CTL_PAD_GPIO_AD_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_02)	32	RW	0000_0006h
35Ch	SW_PAD_CTL_PAD_GPIO_AD_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_03)	32	RW	0000_0006h
360h	SW_PAD_CTL_PAD_GPIO_AD_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_04)	32	RW	0000_0006h
364h	SW_PAD_CTL_PAD_GPIO_AD_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_05)	32	RW	0000_0006h
368h	SW_PAD_CTL_PAD_GPIO_AD_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_06)	32	RW	0000_0006h
36Ch	SW_PAD_CTL_PAD_GPIO_AD_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_07)	32	RW	0000_0006h
370h	SW_PAD_CTL_PAD_GPIO_AD_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_08)	32	RW	0000_0006h
374h	SW_PAD_CTL_PAD_GPIO_AD_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_09)	32	RW	0000_0006h
378h	SW_PAD_CTL_PAD_GPIO_AD_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_10)	32	RW	0000_0006h
37Ch	SW_PAD_CTL_PAD_GPIO_AD_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_11)	32	RW	0000_0006h
380h	SW_PAD_CTL_PAD_GPIO_AD_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_12)	32	RW	0000_0006h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
384h	SW_PAD_CTL_PAD_GPIO_AD_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_13)	32	RW	0000_0006h
388h	SW_PAD_CTL_PAD_GPIO_AD_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_14)	32	RW	0000_0006h
38Ch	SW_PAD_CTL_PAD_GPIO_AD_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_15)	32	RW	0000_0006h
390h	SW_PAD_CTL_PAD_GPIO_AD_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_16)	32	RW	0000_0006h
394h	SW_PAD_CTL_PAD_GPIO_AD_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_17)	32	RW	0000_0006h
398h	SW_PAD_CTL_PAD_GPIO_AD_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_18)	32	RW	0000_000Eh
39Ch	SW_PAD_CTL_PAD_GPIO_AD_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_19)	32	RW	0000_0006h
3A0h	SW_PAD_CTL_PAD_GPIO_AD_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_20)	32	RW	0000_0006h
3A4h	SW_PAD_CTL_PAD_GPIO_AD_21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_21)	32	RW	0000_0006h
3A8h	SW_PAD_CTL_PAD_GPIO_AD_22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_22)	32	RW	0000_0006h
3ACh	SW_PAD_CTL_PAD_GPIO_AD_23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_23)	32	RW	0000_0006h
3B0h	SW_PAD_CTL_PAD_GPIO_AD_24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_24)	32	RW	0000_0006h
3B4h	SW_PAD_CTL_PAD_GPIO_AD_25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_25)	32	RW	0000_0006h
3B8h	SW_PAD_CTL_PAD_GPIO_AD_26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_26)	32	RW	0000_000Eh
3BCh	SW_PAD_CTL_PAD_GPIO_AD_27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_27)	32	RW	0000_000Eh
3C0h	SW_PAD_CTL_PAD_GPIO_AD_28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_28)	32	RW	0000_0006h
3C4h	SW_PAD_CTL_PAD_GPIO_AD_29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_29)	32	RW	0000_0006h
3C8h	SW_PAD_CTL_PAD_GPIO_AD_30 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_30)	32	RW	0000_0006h
3CCh	SW_PAD_CTL_PAD_GPIO_AD_31 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_31)	32	RW	0000_0006h
3D0h	SW_PAD_CTL_PAD_GPIO_AD_32 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_32)	32	RW	0000_0006h
3D4h	SW_PAD_CTL_PAD_GPIO_AD_33 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_33)	32	RW	0000_0006h
3D8h	SW_PAD_CTL_PAD_GPIO_AD_34 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_34)	32	RW	0000_0006h
3DCh	SW_PAD_CTL_PAD_GPIO_AD_35 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_35)	32	RW	0000_000Eh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3E0h	SW_PAD_CTL_PAD_GPIO_SD_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_00)	32	RW	0000_0004h
3E4h	SW_PAD_CTL_PAD_GPIO_SD_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_01)	32	RW	0000_0008h
3E8h	SW_PAD_CTL_PAD_GPIO_SD_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_02)	32	RW	0000_0004h
3ECh	SW_PAD_CTL_PAD_GPIO_SD_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_03)	32	RW	0000_0004h
3F0h	SW_PAD_CTL_PAD_GPIO_SD_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_04)	32	RW	0000_0004h
3F4h	SW_PAD_CTL_PAD_GPIO_SD_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_05)	32	RW	0000_0008h
3F8h	SW_PAD_CTL_PAD_GPIO_SD_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_00)	32	RW	0000_0008h
3FCh	SW_PAD_CTL_PAD_GPIO_SD_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_01)	32	RW	0000_0008h
400h	SW_PAD_CTL_PAD_GPIO_SD_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_02)	32	RW	0000_0008h
404h	SW_PAD_CTL_PAD_GPIO_SD_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_03)	32	RW	0000_0008h
408h	SW_PAD_CTL_PAD_GPIO_SD_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_04)	32	RW	0000_0004h
40Ch	SW_PAD_CTL_PAD_GPIO_SD_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_05)	32	RW	0000_0004h
410h	SW_PAD_CTL_PAD_GPIO_SD_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_06)	32	RW	0000_0004h
414h	SW_PAD_CTL_PAD_GPIO_SD_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_07)	32	RW	0000_0008h
418h	SW_PAD_CTL_PAD_GPIO_SD_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_08)	32	RW	0000_0008h
41Ch	SW_PAD_CTL_PAD_GPIO_SD_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_09)	32	RW	0000_0008h
420h	SW_PAD_CTL_PAD_GPIO_SD_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_10)	32	RW	0000_0008h
424h	SW_PAD_CTL_PAD_GPIO_SD_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_11)	32	RW	0000_0008h
428h	SW_PAD_CTL_PAD_GPIO_DISP_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_00)	32	RW	0000_0008h
42Ch	SW_PAD_CTL_PAD_GPIO_DISP_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_01)	32	RW	0000_0008h
430h	SW_PAD_CTL_PAD_GPIO_DISP_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_02)	32	RW	0000_0008h
434h	SW_PAD_CTL_PAD_GPIO_DISP_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_03)	32	RW	0000_0008h
438h	SW_PAD_CTL_PAD_GPIO_DISP_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_04)	32	RW	0000_0008h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
43Ch	SW_PAD_CTL_PAD_GPIO_DISP_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_05)	32	RW	0000_0008h
440h	SW_PAD_CTL_PAD_GPIO_DISP_B1_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_06)	32	RW	0000_000Ch
444h	SW_PAD_CTL_PAD_GPIO_DISP_B1_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_07)	32	RW	0000_000Ch
448h	SW_PAD_CTL_PAD_GPIO_DISP_B1_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_08)	32	RW	0000_000Ch
44Ch	SW_PAD_CTL_PAD_GPIO_DISP_B1_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_09)	32	RW	0000_000Ch
450h	SW_PAD_CTL_PAD_GPIO_DISP_B1_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_10)	32	RW	0000_000Ch
454h	SW_PAD_CTL_PAD_GPIO_DISP_B1_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_11)	32	RW	0000_000Ch
458h	SW_PAD_CTL_PAD_GPIO_DISP_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_00)	32	RW	0000_0002h
45Ch	SW_PAD_CTL_PAD_GPIO_DISP_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_01)	32	RW	0000_0002h
460h	SW_PAD_CTL_PAD_GPIO_DISP_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_02)	32	RW	0000_0002h
464h	SW_PAD_CTL_PAD_GPIO_DISP_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_03)	32	RW	0000_0002h
468h	SW_PAD_CTL_PAD_GPIO_DISP_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_04)	32	RW	0000_0002h
46Ch	SW_PAD_CTL_PAD_GPIO_DISP_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_05)	32	RW	0000_0002h
470h	SW_PAD_CTL_PAD_GPIO_DISP_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_06)	32	RW	0000_0006h
474h	SW_PAD_CTL_PAD_GPIO_DISP_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_07)	32	RW	0000_0006h
478h	SW_PAD_CTL_PAD_GPIO_DISP_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_08)	32	RW	0000_0006h
47Ch	SW_PAD_CTL_PAD_GPIO_DISP_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_09)	32	RW	0000_0006h
480h	SW_PAD_CTL_PAD_GPIO_DISP_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_10)	32	RW	0000_0006h
484h	SW_PAD_CTL_PAD_GPIO_DISP_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_11)	32	RW	0000_0006h
488h	SW_PAD_CTL_PAD_GPIO_DISP_B2_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_12)	32	RW	0000_0006h
48Ch	SW_PAD_CTL_PAD_GPIO_DISP_B2_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_13)	32	RW	0000_0006h
490h	SW_PAD_CTL_PAD_GPIO_DISP_B2_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_14)	32	RW	0000_0006h
494h	SW_PAD_CTL_PAD_GPIO_DISP_B2_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_15)	32	RW	0000_000Eh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
498h	FLEXCAN1_RX_SELECT_INPUT DAISY Register (FLEXCAN1_RX_SELECT_INPUT)	32	RW	0000_0000h
49Ch	FLEXCAN2_RX_SELECT_INPUT DAISY Register (FLEXCAN2_RX_SELECT_INPUT)	32	RW	0000_0000h
4A0h	CCM_ENET_QOS_REF_CLK_SELECT_INPUT DAISY Register (CCM_ENET_QOS_REF_CLK_SELECT_INPUT)	32	RW	0000_0000h
4A4h	CCM_ENET_QOS_TX_CLK_SELECT_INPUT DAISY Register (CCM_ENET_QOS_TX_CLK_SELECT_INPUT)	32	RW	0000_0000h
4A8h	ENET_IPG_CLK_RMII_SELECT_INPUT DAISY Register (ENET_IPG_CLK_RMII_SELECT_INPUT)	32	RW	0000_0000h
4ACh	ENET_MAC0_MDIO_SELECT_INPUT DAISY Register (ENET_MAC0_MDIO_SELECT_INPUT)	32	RW	0000_0000h
4B0h	ENET_MAC0_RXDATA_SELECT_INPUT_0 DAISY Register (ENET_MAC0_RXDATA_SELECT_INPUT_0)	32	RW	0000_0000h
4B4h	ENET_MAC0_RXDATA_SELECT_INPUT_1 DAISY Register (ENET_MAC0_RXDATA_SELECT_INPUT_1)	32	RW	0000_0000h
4B8h	ENET_MAC0_RXEN_SELECT_INPUT DAISY Register (ENET_MAC0_RXEN_SELECT_INPUT)	32	RW	0000_0000h
4BCh	ENET_MAC0_RXERR_SELECT_INPUT DAISY Register (ENET_MAC0_RXERR_SELECT_INPUT)	32	RW	0000_0000h
4C0h	ENET_MAC0_TXCLK_SELECT_INPUT DAISY Register (ENET_MAC0_TXCLK_SELECT_INPUT)	32	RW	0000_0000h
4C4h	ENET_1G_IPG_CLK_RMII_SELECT_INPUT DAISY Register (ENET_1G_IPG_CLK_RMII_SELECT_INPUT)	32	RW	0000_0000h
4C8h	ENET_1G_MAC0_MDIO_SELECT_INPUT DAISY Register (ENET_1G_MAC0_MDIO_SELECT_INPUT)	32	RW	0000_0000h
4CCh	ENET_1G_MAC0_RXCLK_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXCLK_SELECT_INPUT)	32	RW	0000_0000h
4D0h	ENET_1G_MAC0_RXDATA_0_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_0_SELECT_INPUT)	32	RW	0000_0000h
4D4h	ENET_1G_MAC0_RXDATA_1_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_1_SELECT_INPUT)	32	RW	0000_0000h
4D8h	ENET_1G_MAC0_RXDATA_2_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_2_SELECT_INPUT)	32	RW	0000_0000h
4DCh	ENET_1G_MAC0_RXDATA_3_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_3_SELECT_INPUT)	32	RW	0000_0000h
4E0h	ENET_1G_MAC0_RXEN_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXEN_SELECT_INPUT)	32	RW	0000_0000h
4E4h	ENET_1G_MAC0_RXERR_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXERR_SELECT_INPUT)	32	RW	0000_0000h
4E8h	ENET_1G_MAC0_TXCLK_SELECT_INPUT DAISY Register (ENET_1G_MAC0_TXCLK_SELECT_INPUT)	32	RW	0000_0000h
4ECh	ENET_QOS_GMII_MDI_I_SELECT_INPUT DAISY Register (ENET_QOS_GMII_MDI_I_SELECT_INPUT)	32	RW	0000_0000h
4F0h	ENET_QOS_PHY_RXD_I_SELECT_INPUT_0 DAISY Register (ENET_QOS_PHY_RXD_I_SELECT_INPUT_0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4F4h	ENET_QOS_PHY_RXD_I_SELECT_INPUT_1 DAISY Register (ENET_QOS_PHY_RXD_I_SELECT_INPUT_1)	32	RW	0000_0000h
4F8h	ENET_QOS_PHY_RXDV_I_SELECT_INPUT DAISY Register (ENET_QOS_PHY_RXDV_I_SELECT_INPUT)	32	RW	0000_0000h
4FCh	ENET_QOS_PHY_RXER_I_SELECT_INPUT DAISY Register (ENET_QOS_PHY_RXER_I_SELECT_INPUT)	32	RW	0000_0000h
500h	FLEXPWM1_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_0)	32	RW	0000_0000h
504h	FLEXPWM1_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_1)	32	RW	0000_0000h
508h	FLEXPWM1_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_2)	32	RW	0000_0000h
50Ch	FLEXPWM1_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_0)	32	RW	0000_0000h
510h	FLEXPWM1_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_1)	32	RW	0000_0000h
514h	FLEXPWM1_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_2)	32	RW	0000_0000h
518h	FLEXPWM2_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_0)	32	RW	0000_0000h
51Ch	FLEXPWM2_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_1)	32	RW	0000_0000h
520h	FLEXPWM2_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_2)	32	RW	0000_0000h
524h	FLEXPWM2_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_0)	32	RW	0000_0000h
528h	FLEXPWM2_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_1)	32	RW	0000_0000h
52Ch	FLEXPWM2_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_2)	32	RW	0000_0000h
530h	FLEXPWM3_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_0)	32	RW	0000_0000h
534h	FLEXPWM3_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_1)	32	RW	0000_0000h
538h	FLEXPWM3_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_2)	32	RW	0000_0000h
53Ch	FLEXPWM3_PWMA_SELECT_INPUT_3 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_3)	32	RW	0000_0000h
540h	FLEXPWM3_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_0)	32	RW	0000_0000h
544h	FLEXPWM3_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_1)	32	RW	0000_0000h
548h	FLEXPWM3_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_2)	32	RW	0000_0000h
54Ch	FLEXPWM3_PWMB_SELECT_INPUT_3 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_3)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
550h	FLEXSPI1_I_DQS_FA_SELECT_INPUT DAISY Register (FLEXSPI1_I_DQS_FA_SELECT_INPUT)	32	RW	0000_0000h
554h	FLEXSPI1_I_IO_FA_SELECT_INPUT_0 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_0)	32	RW	0000_0000h
558h	FLEXSPI1_I_IO_FA_SELECT_INPUT_1 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_1)	32	RW	0000_0000h
55Ch	FLEXSPI1_I_IO_FA_SELECT_INPUT_2 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_2)	32	RW	0000_0000h
560h	FLEXSPI1_I_IO_FA_SELECT_INPUT_3 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_3)	32	RW	0000_0000h
564h	FLEXSPI1_I_IO_FB_SELECT_INPUT_0 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_0)	32	RW	0000_0000h
568h	FLEXSPI1_I_IO_FB_SELECT_INPUT_1 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_1)	32	RW	0000_0000h
56Ch	FLEXSPI1_I_IO_FB_SELECT_INPUT_2 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_2)	32	RW	0000_0000h
570h	FLEXSPI1_I_IO_FB_SELECT_INPUT_3 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_3)	32	RW	0000_0000h
574h	FLEXSPI1_I_SCK_FA_SELECT_INPUT DAISY Register (FLEXSPI1_I_SCK_FA_SELECT_INPUT)	32	RW	0000_0000h
578h	FLEXSPI1_I_SCK_FB_SELECT_INPUT DAISY Register (FLEXSPI1_I_SCK_FB_SELECT_INPUT)	32	RW	0000_0000h
57Ch	FLEXSPI2_I_IO_FA_SELECT_INPUT_0 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_0)	32	RW	0000_0000h
580h	FLEXSPI2_I_IO_FA_SELECT_INPUT_1 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_1)	32	RW	0000_0000h
584h	FLEXSPI2_I_IO_FA_SELECT_INPUT_2 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_2)	32	RW	0000_0000h
588h	FLEXSPI2_I_IO_FA_SELECT_INPUT_3 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_3)	32	RW	0000_0000h
58Ch	FLEXSPI2_I_SCK_FA_SELECT_INPUT DAISY Register (FLEXSPI2_I_SCK_FA_SELECT_INPUT)	32	RW	0000_0000h
590h	GPT3_CAPIN1_SELECT_INPUT DAISY Register (GPT3_CAPIN1_SELECT_INPUT)	32	RW	0000_0000h
594h	GPT3_CAPIN2_SELECT_INPUT DAISY Register (GPT3_CAPIN2_SELECT_INPUT)	32	RW	0000_0000h
598h	GPT3_CLKIN_SELECT_INPUT DAISY Register (GPT3_CLKIN_SELECT_INPUT)	32	RW	0000_0000h
59Ch	KPP_COL_SELECT_INPUT_6 DAISY Register (KPP_COL_SELECT_INPUT_6)	32	RW	0000_0000h
5A0h	KPP_COL_SELECT_INPUT_7 DAISY Register (KPP_COL_SELECT_INPUT_7)	32	RW	0000_0000h
5A4h	KPP_ROW_SELECT_INPUT_6 DAISY Register (KPP_ROW_SELECT_INPUT_6)	32	RW	0000_0000h
5A8h	KPP_ROW_SELECT_INPUT_7 DAISY Register (KPP_ROW_SELECT_INPUT_7)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5ACh	LPI2C1_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C1_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
5B0h	LPI2C1_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C1_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
5B4h	LPI2C2_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C2_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
5B8h	LPI2C2_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C2_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
5BCh	LPI2C3_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C3_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
5C0h	LPI2C3_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C3_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
5C4h	LPI2C4_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C4_LPI2C_SCL_SELECT_INPUT)	32	RW	0000_0000h
5C8h	LPI2C4_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C4_LPI2C_SDA_SELECT_INPUT)	32	RW	0000_0000h
5CCh	LPSPI1_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI1_LPSPI_PCS_SELECT_INPUT_0)	32	RW	0000_0000h
5D0h	LPSPI1_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SCK_SELECT_INPUT)	32	RW	0000_0000h
5D4h	LPSPI1_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SDI_SELECT_INPUT)	32	RW	0000_0000h
5D8h	LPSPI1_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SDO_SELECT_INPUT)	32	RW	0000_0000h
5DCh	LPSPI2_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI2_LPSPI_PCS_SELECT_INPUT_0)	32	RW	0000_0000h
5E0h	LPSPI2_LPSPI_PCS_SELECT_INPUT_1 DAISY Register (LPSPI2_LPSPI_PCS_SELECT_INPUT_1)	32	RW	0000_0000h
5E4h	LPSPI2_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SCK_SELECT_INPUT)	32	RW	0000_0000h
5E8h	LPSPI2_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SDI_SELECT_INPUT)	32	RW	0000_0000h
5ECh	LPSPI2_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SDO_SELECT_INPUT)	32	RW	0000_0000h
5F0h	LPSPI3_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_0)	32	RW	0000_0000h
5F4h	LPSPI3_LPSPI_PCS_SELECT_INPUT_1 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_1)	32	RW	0000_0000h
5F8h	LPSPI3_LPSPI_PCS_SELECT_INPUT_2 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_2)	32	RW	0000_0000h
5FCh	LPSPI3_LPSPI_PCS_SELECT_INPUT_3 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_3)	32	RW	0000_0000h
600h	LPSPI3_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SCK_SELECT_INPUT)	32	RW	0000_0000h
604h	LPSPI3_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SDI_SELECT_INPUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
608h	LPSPI3_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SDO_SELECT_INPUT)	32	RW	0000_0000h
60Ch	LPSPI4_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI4_LPSPI_PCS_SELECT_INPUT_0)	32	RW	0000_0000h
610h	LPSPI4_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SCK_SELECT_INPUT)	32	RW	0000_0000h
614h	LPSPI4_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SDI_SELECT_INPUT)	32	RW	0000_0000h
618h	LPSPI4_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SDO_SELECT_INPUT)	32	RW	0000_0000h
61Ch	LPUART1_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART1_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
620h	LPUART1_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART1_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
624h	LPUART10_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART10_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
628h	LPUART10_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART10_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
62Ch	LPUART7_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART7_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
630h	LPUART7_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART7_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
634h	LPUART8_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART8_LPUART_RXD_SELECT_INPUT)	32	RW	0000_0000h
638h	LPUART8_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART8_LPUART_TXD_SELECT_INPUT)	32	RW	0000_0000h
63Ch	QTIMER1_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR0_INPUT_SELECT_INPUT)	32	RW	0000_0000h
640h	QTIMER1_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR1_INPUT_SELECT_INPUT)	32	RW	0000_0000h
644h	QTIMER1_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR2_INPUT_SELECT_INPUT)	32	RW	0000_0000h
648h	QTIMER2_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR0_INPUT_SELECT_INPUT)	32	RW	0000_0000h
64Ch	QTIMER2_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR1_INPUT_SELECT_INPUT)	32	RW	0000_0000h
650h	QTIMER2_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR2_INPUT_SELECT_INPUT)	32	RW	0000_0000h
654h	QTIMER3_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR0_INPUT_SELECT_INPUT)	32	RW	0000_0000h
658h	QTIMER3_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR1_INPUT_SELECT_INPUT)	32	RW	0000_0000h
65Ch	QTIMER3_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR2_INPUT_SELECT_INPUT)	32	RW	0000_0000h
660h	QTIMER4_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR0_INPUT_SELECT_INPUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
664h	QTIMER4_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR1_INPUT_SELECT_INPUT)	32	RW	0000_0000h
668h	QTIMER4_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR2_INPUT_SELECT_INPUT)	32	RW	0000_0000h
66Ch	SAI1_IPG_CLK_SAI_MCLK_SELECT_INPUT DAISY Register (SAI1_IPG_CLK_SAI_MCLK_SELECT_INPUT)	32	RW	0000_0000h
670h	SAI1_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI1_SAI_RXBCLK_SELECT_INPUT)	32	RW	0000_0000h
674h	SAI1_SAI_RXDATA_SELECT_INPUT_0 DAISY Register (SAI1_SAI_RXDATA_SELECT_INPUT_0)	32	RW	0000_0000h
678h	SAI1_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI1_SAI_RXSYNC_SELECT_INPUT)	32	RW	0000_0000h
67Ch	SAI1_SAI_TXBCLK_SELECT_INPUT DAISY Register (SAI1_SAI_TXBCLK_SELECT_INPUT)	32	RW	0000_0000h
680h	SAI1_SAI_TXSYNC_SELECT_INPUT DAISY Register (SAI1_SAI_TXSYNC_SELECT_INPUT)	32	RW	0000_0000h
69Ch	EMVSIM1_SIO_SELECT_INPUT DAISY Register (EMVSIM1_SIO_SELECT_INPUT)	32	RW	0000_0000h
6A0h	EMVSIM1_IPP_SIMPD_SELECT_INPUT DAISY Register (EMVSIM1_IPP_SIMPD_SELECT_INPUT)	32	RW	0000_0000h
6A4h	EMVSIM1_POWER_FAIL_SELECT_INPUT DAISY Register (EMVSIM1_POWER_FAIL_SELECT_INPUT)	32	RW	0000_0000h
6A8h	EMVSIM2_SIO_SELECT_INPUT DAISY Register (EMVSIM2_SIO_SELECT_INPUT)	32	RW	0000_0000h
6ACh	EMVSIM2_IPP_SIMPD_SELECT_INPUT DAISY Register (EMVSIM2_IPP_SIMPD_SELECT_INPUT)	32	RW	0000_0000h
6B0h	EMVSIM2_POWER_FAIL_SELECT_INPUT DAISY Register (EMVSIM2_POWER_FAIL_SELECT_INPUT)	32	RW	0000_0000h
6B4h	SPDIF_SPDIF_IN1_SELECT_INPUT DAISY Register (SPDIF_SPDIF_IN1_SELECT_INPUT)	32	RW	0000_0000h
6B8h	USB_OTG2_OC_SELECT_INPUT DAISY Register (USB_OTG2_OC_SELECT_INPUT)	32	RW	0000_0000h
6BCh	USB_OTG_OC_SELECT_INPUT DAISY Register (USB_OTG_OC_SELECT_INPUT)	32	RW	0000_0000h
6C0h	USBPHY1_USB_ID_SELECT_INPUT DAISY Register (USBPHY1_USB_ID_SELECT_INPUT)	32	RW	0000_0000h
6C4h	USBPHY2_USB_ID_SELECT_INPUT DAISY Register (USBPHY2_USB_ID_SELECT_INPUT)	32	RW	0000_0000h
6C8h	USDHC1_IPP_CARD_DET_SELECT_INPUT DAISY Register (USDHC1_IPP_CARD_DET_SELECT_INPUT)	32	RW	0000_0000h
6CCh	USDHC1_IPP_WP_ON_SELECT_INPUT DAISY Register (USDHC1_IPP_WP_ON_SELECT_INPUT)	32	RW	0000_0000h
6D0h	USDHC2_IPP_CARD_DET_SELECT_INPUT DAISY Register (USDHC2_IPP_CARD_DET_SELECT_INPUT)	32	RW	0000_0000h
6D4h	USDHC2_IPP_WP_ON_SELECT_INPUT DAISY Register (USDHC2_IPP_WP_ON_SELECT_INPUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
6D8h	XBAR1_IN_SELECT_INPUT_20 DAISY Register (XBAR1_IN_SELECT_INPUT_20)	32	RW	0000_0000h
6DCh	XBAR1_IN_SELECT_INPUT_21 DAISY Register (XBAR1_IN_SELECT_INPUT_21)	32	RW	0000_0000h
6E0h	XBAR1_IN_SELECT_INPUT_22 DAISY Register (XBAR1_IN_SELECT_INPUT_22)	32	RW	0000_0000h
6E4h	XBAR1_IN_SELECT_INPUT_23 DAISY Register (XBAR1_IN_SELECT_INPUT_23)	32	RW	0000_0000h
6E8h	XBAR1_IN_SELECT_INPUT_24 DAISY Register (XBAR1_IN_SELECT_INPUT_24)	32	RW	0000_0000h
6ECh	XBAR1_IN_SELECT_INPUT_25 DAISY Register (XBAR1_IN_SELECT_INPUT_25)	32	RW	0000_0000h
6F0h	XBAR1_IN_SELECT_INPUT_26 DAISY Register (XBAR1_IN_SELECT_INPUT_26)	32	RW	0000_0000h
6F4h	XBAR1_IN_SELECT_INPUT_27 DAISY Register (XBAR1_IN_SELECT_INPUT_27)	32	RW	0000_0000h
6F8h	XBAR1_IN_SELECT_INPUT_28 DAISY Register (XBAR1_IN_SELECT_INPUT_28)	32	RW	0000_0000h
6FCh	XBAR1_IN_SELECT_INPUT_29 DAISY Register (XBAR1_IN_SELECT_INPUT_29)	32	RW	0000_0000h
700h	XBAR1_IN_SELECT_INPUT_30 DAISY Register (XBAR1_IN_SELECT_INPUT_30)	32	RW	0000_0000h
704h	XBAR1_IN_SELECT_INPUT_31 DAISY Register (XBAR1_IN_SELECT_INPUT_31)	32	RW	0000_0000h
708h	XBAR1_IN_SELECT_INPUT_32 DAISY Register (XBAR1_IN_SELECT_INPUT_32)	32	RW	0000_0000h
70Ch	XBAR1_IN_SELECT_INPUT_33 DAISY Register (XBAR1_IN_SELECT_INPUT_33)	32	RW	0000_0000h
710h	XBAR1_IN_SELECT_INPUT_34 DAISY Register (XBAR1_IN_SELECT_INPUT_34)	32	RW	0000_0000h
714h	XBAR1_IN_SELECT_INPUT_35 DAISY Register (XBAR1_IN_SELECT_INPUT_35)	32	RW	0000_0000h

12.4.6.2 SW_MUX_CTL_PAD_GPIO_EMC_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_00)

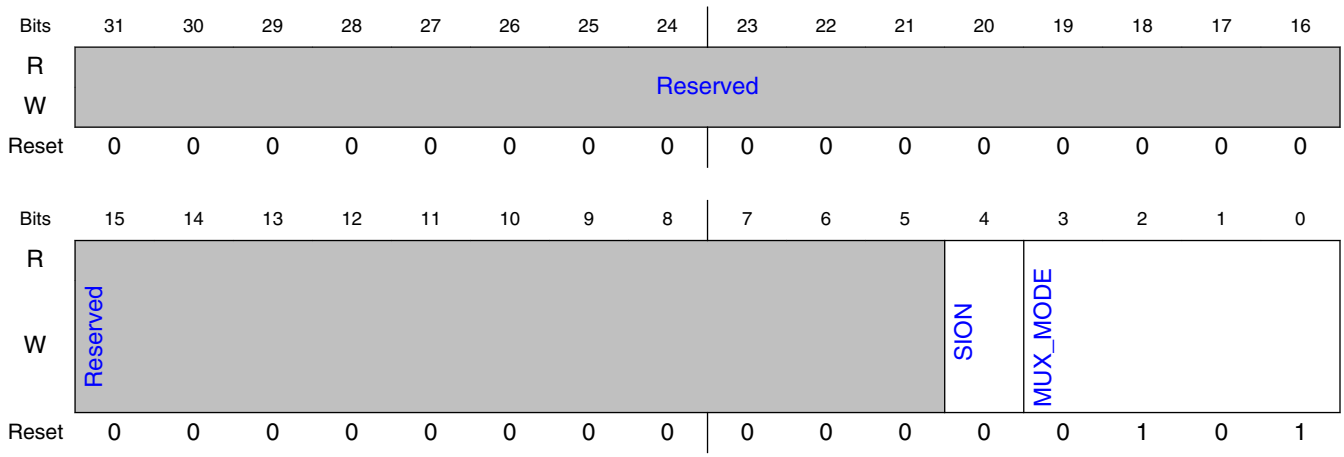
12.4.6.2.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_00	10h

12.4.6.2.2 Function

SW_MUX_CTL Register

12.4.6.2.3 Diagram



12.4.6.2.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_00. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA00 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM0_A of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO00 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D00 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO00 of instance: GPIO7

12.4.6.3 SW_MUX_CTL_PAD_GPIO_EMC_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_01)

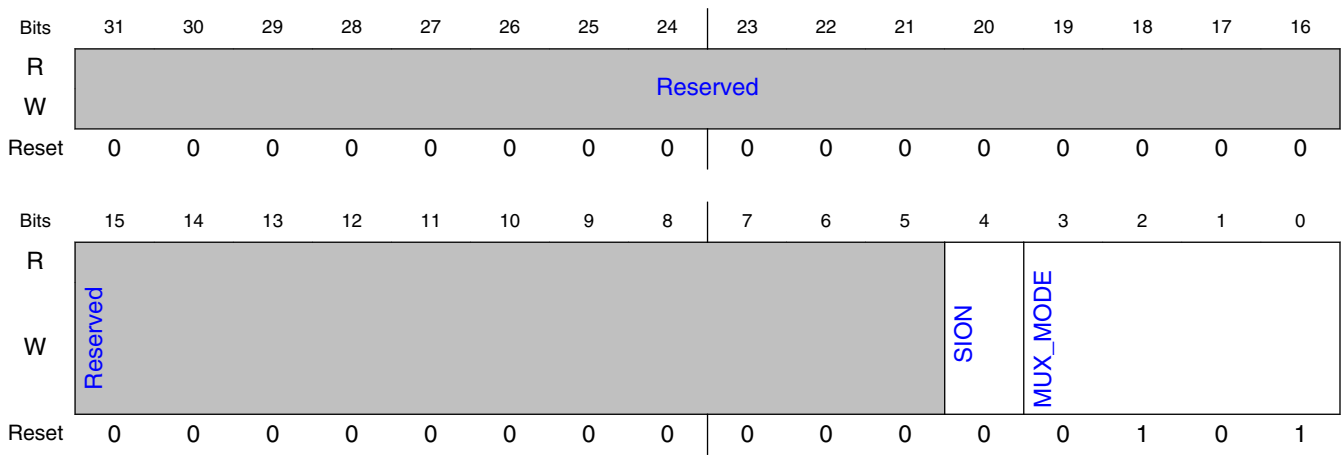
12.4.6.3.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_01	14h

12.4.6.3.2 Function

SW_MUX_CTL Register

12.4.6.3.3 Diagram



12.4.6.3.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_01. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA01 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM0_B of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO01 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D01 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO01 of instance: GPIO7

12.4.6.4 SW_MUX_CTL_PAD_GPIO_EMC_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_02)

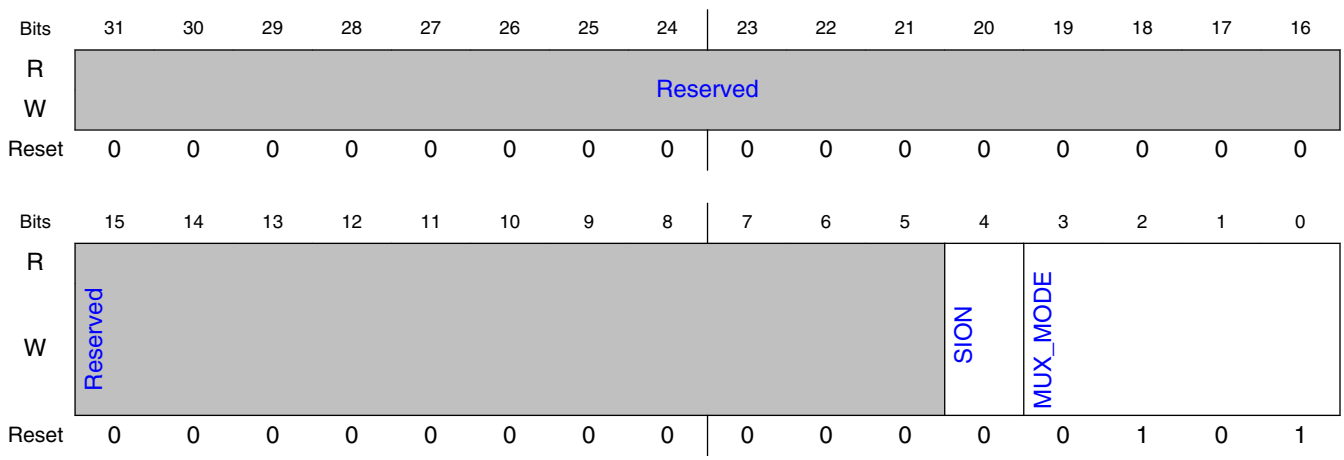
12.4.6.4.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_02	18h

12.4.6.4.2 Function

SW_MUX_CTL Register

12.4.6.4.3 Diagram



12.4.6.4.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_02.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_DATA02 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM1_A of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO02 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D02 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO02 of instance: GPIO7

12.4.6.5 SW_MUX_CTL_PAD_GPIO_EMC_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_03)

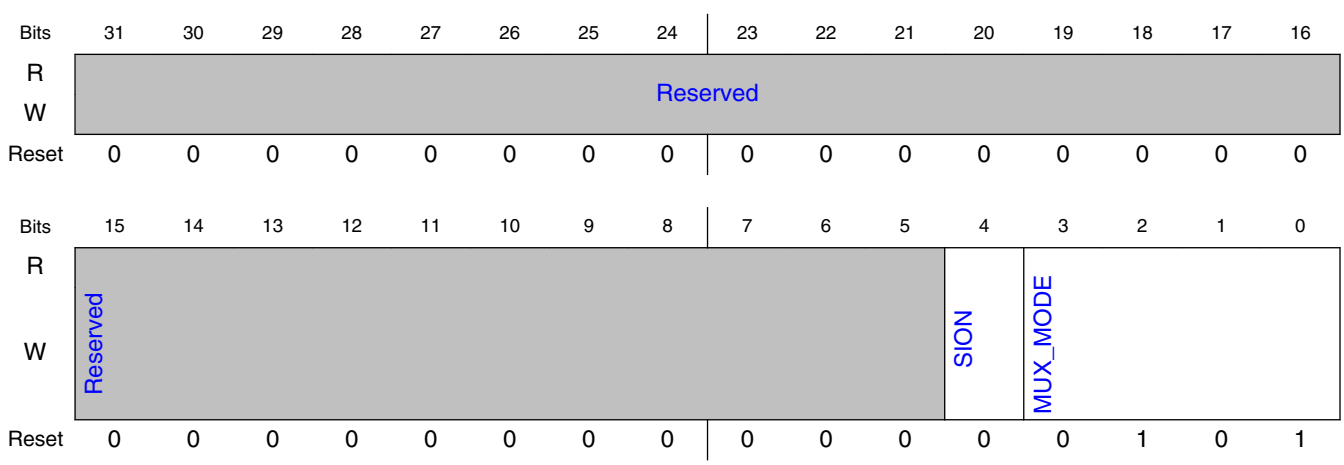
12.4.6.5.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_03	1Ch

12.4.6.5.2 Function

SW_MUX_CTL Register

12.4.6.5.3 Diagram



12.4.6.5.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_03. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA03 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM1_B of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO03 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D03 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO03 of instance: GPIO7

12.4.6.6 SW_MUX_CTL_PAD_GPIO_EMC_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_04)

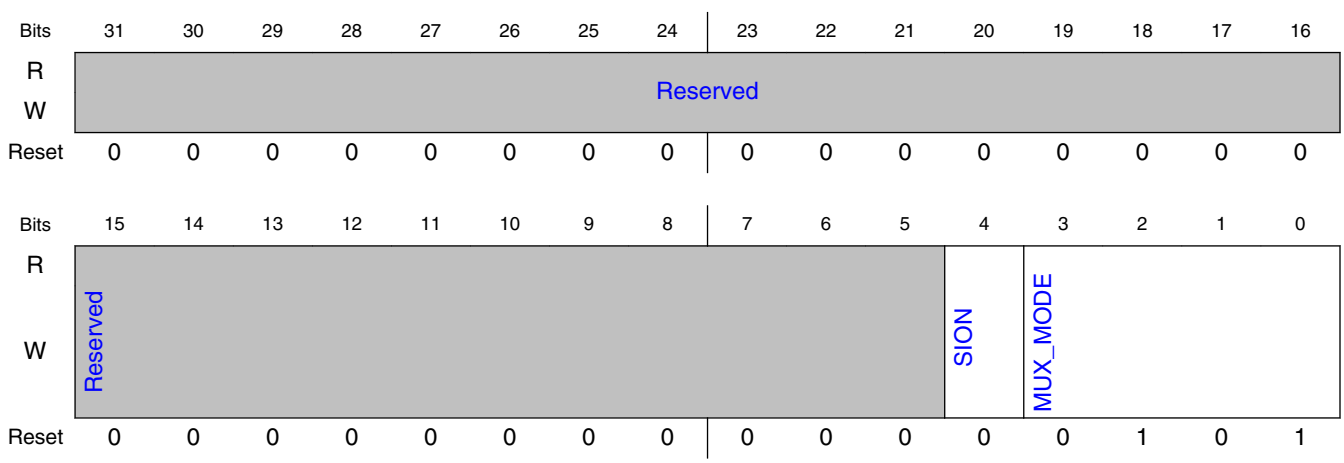
12.4.6.6.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_04	20h

12.4.6.6.2 Function

SW_MUX_CTL Register

12.4.6.6.3 Diagram



12.4.6.6.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_04. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA04 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM2_A of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO04 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D04 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO04 of instance: GPIO7

12.4.6.7 SW_MUX_CTL_PAD_GPIO_EMC_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_05)

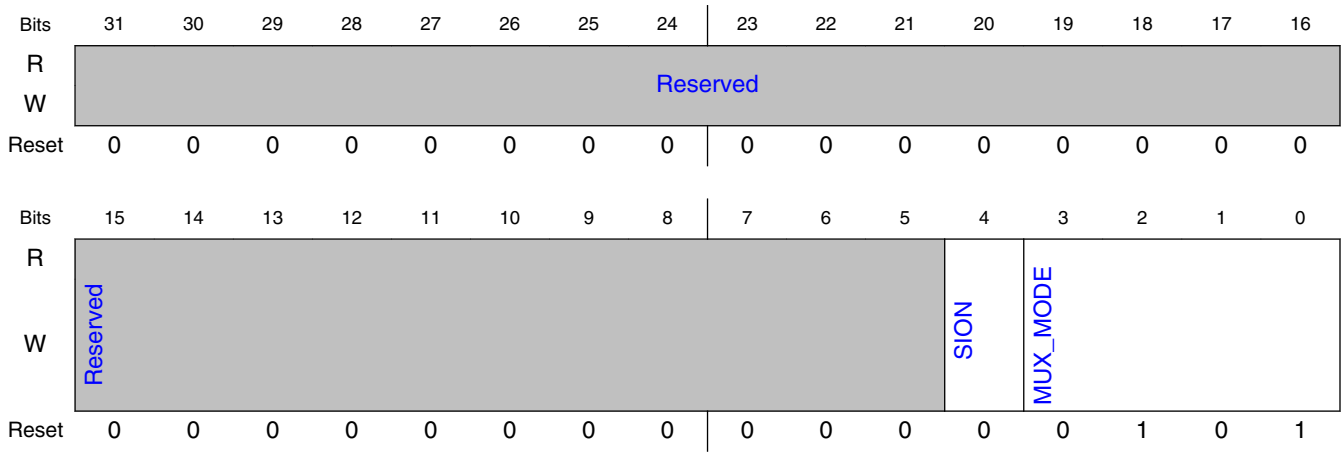
12.4.6.7.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_05	24h

12.4.6.7.2 Function

SW_MUX_CTL Register

12.4.6.7.3 Diagram



12.4.6.7.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_05. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA05 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM2_B of instance: FLEXPWM4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO05 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D05 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO05 of instance: GPIO7

12.4.6.8 SW_MUX_CTL_PAD_GPIO_EMC_B1_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_06)

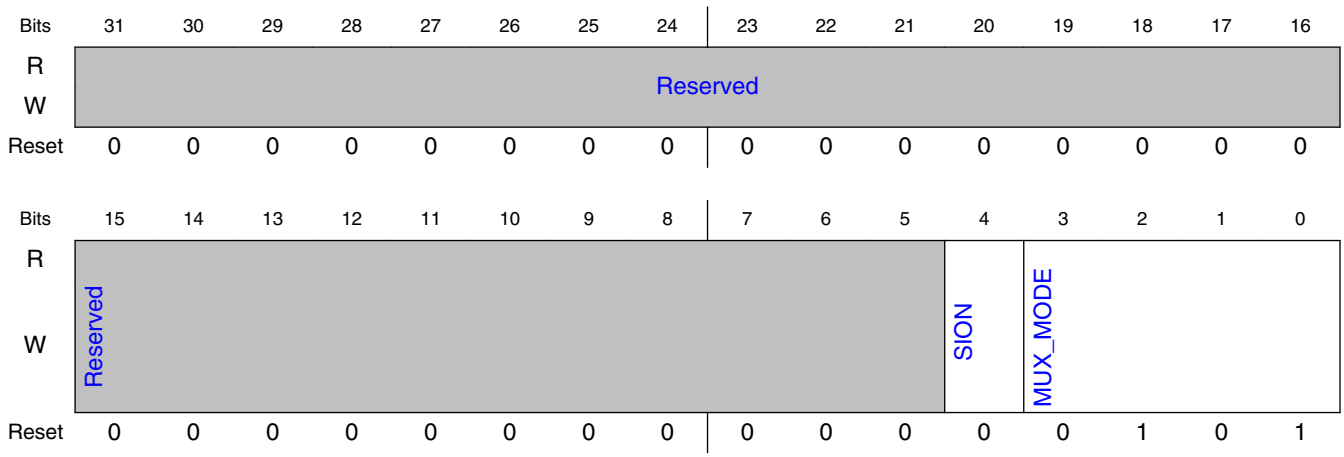
12.4.6.8.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_06	28h

12.4.6.8.2 Function

SW_MUX_CTL Register

12.4.6.8.3 Diagram



12.4.6.8.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_06. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA06 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM0_A of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO06 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D06 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO06 of instance: GPIO7

12.4.6.9 SW_MUX_CTL_PAD_GPIO_EMC_B1_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_07)

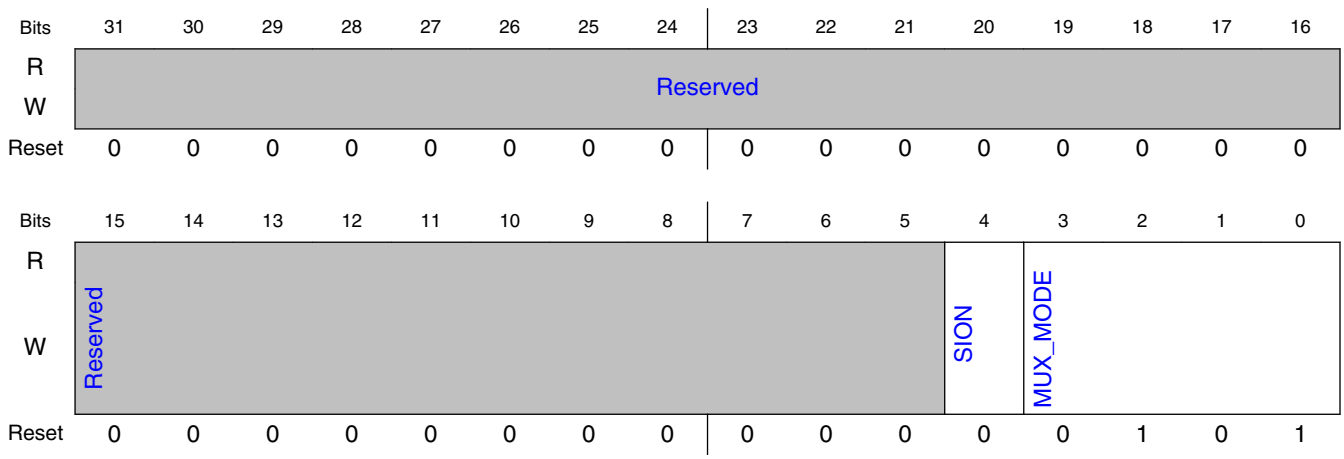
12.4.6.9.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_07	2Ch

12.4.6.9.2 Function

SW_MUX_CTL Register

12.4.6.9.3 Diagram



12.4.6.9.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_07. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA07 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM0_B of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO07 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D07 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO07 of instance: GPIO7

12.4.6.10 SW_MUX_CTL_PAD_GPIO_EMC_B1_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_08)

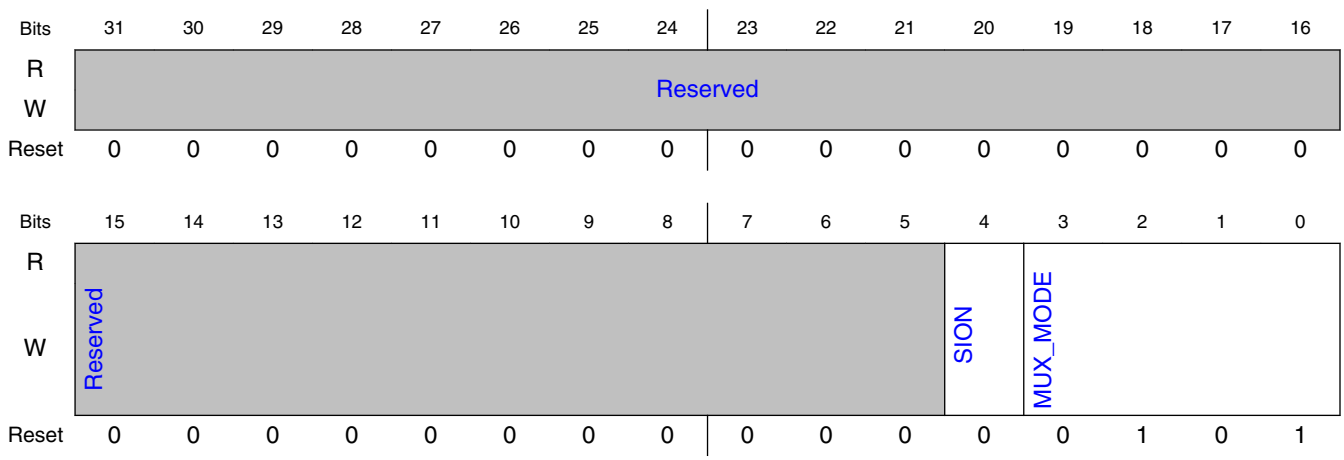
12.4.6.10.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_08	30h

12.4.6.10.2 Function

SW_MUX_CTL Register

12.4.6.10.3 Diagram



12.4.6.10.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_08.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_DM00 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM1_A of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO08 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D08 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO08 of instance: GPIO7

12.4.6.11 SW_MUX_CTL_PAD_GPIO_EMC_B1_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_09)

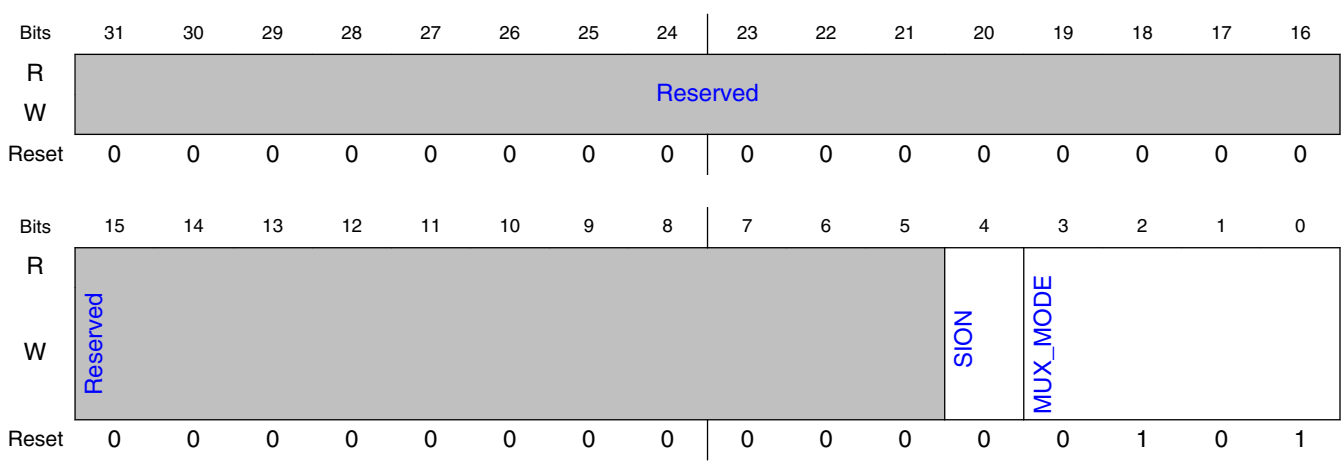
12.4.6.11.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_09	34h

12.4.6.11.2 Function

SW_MUX_CTL Register

12.4.6.11.3 Diagram



12.4.6.11.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_09. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR00 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM1_B of instance: FLEXPWM2 0010b - Select mux mode: ALT2 mux port: GPT5_CAPTURE1 of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO09 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D09 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO09 of instance: GPIO7

12.4.6.12 SW_MUX_CTL_PAD_GPIO_EMC_B1_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_10)

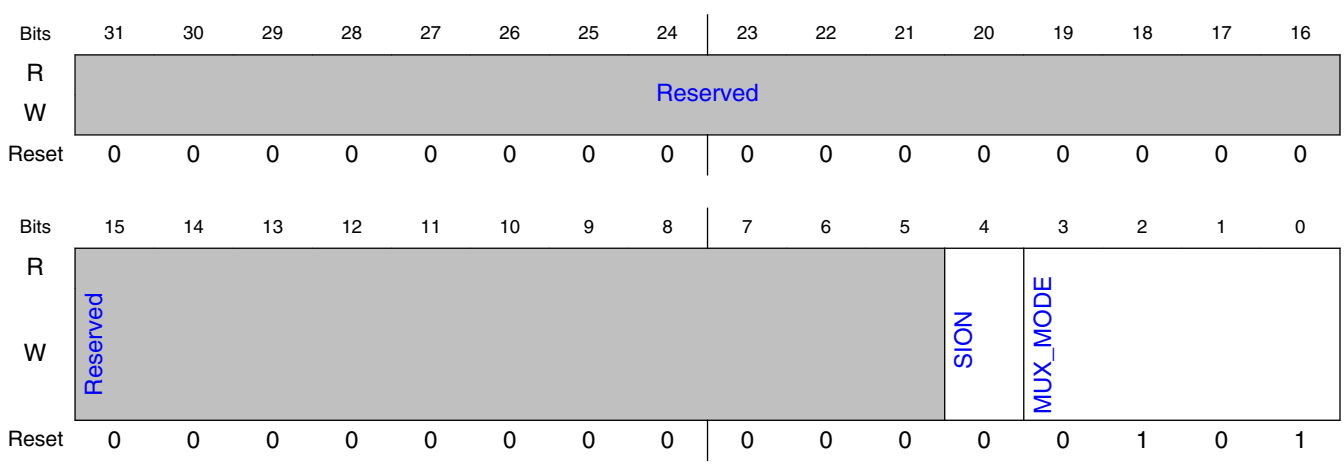
12.4.6.12.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_10	38h

12.4.6.12.2 Function

SW_MUX_CTL Register

12.4.6.12.3 Diagram



12.4.6.12.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_10. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR01 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM2_A of instance: FLEXPWM2 0010b - Select mux mode: ALT2 mux port: GPT5_CAPTURE2 of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO10 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D10 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO10 of instance: GPIO7

12.4.6.13 SW_MUX_CTL_PAD_GPIO_EMC_B1_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_11)

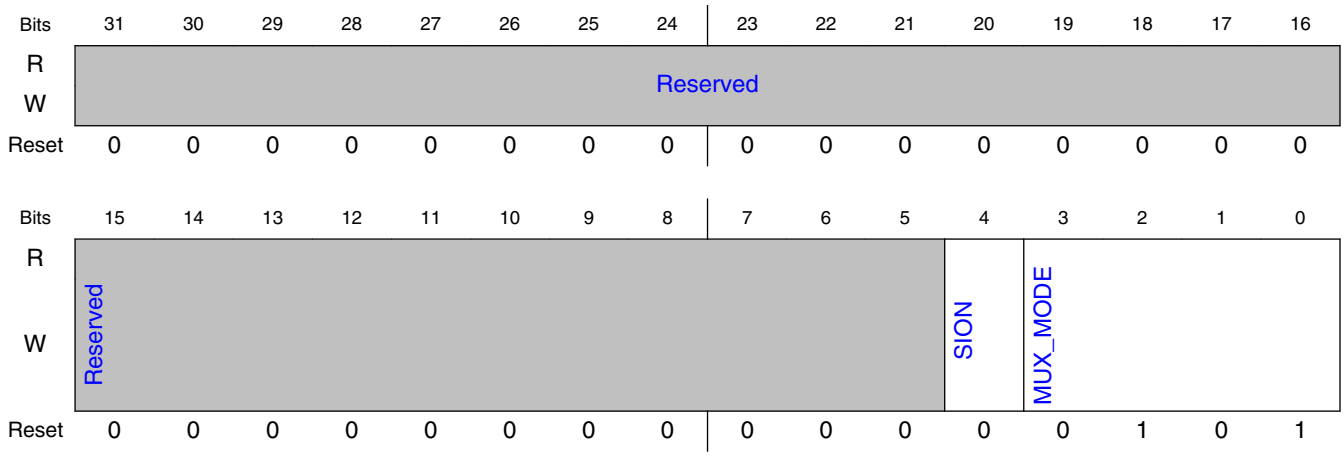
12.4.6.13.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_11	3Ch

12.4.6.13.2 Function

SW_MUX_CTL Register

12.4.6.13.3 Diagram



12.4.6.13.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_11. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR02 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM2_B of instance: FLEXPWM2 0010b - Select mux mode: ALT2 mux port: GPT5_COMPARE1 of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO11 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D11 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO11 of instance: GPIO7

12.4.6.14 SW_MUX_CTL_PAD_GPIO_EMC_B1_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_12)

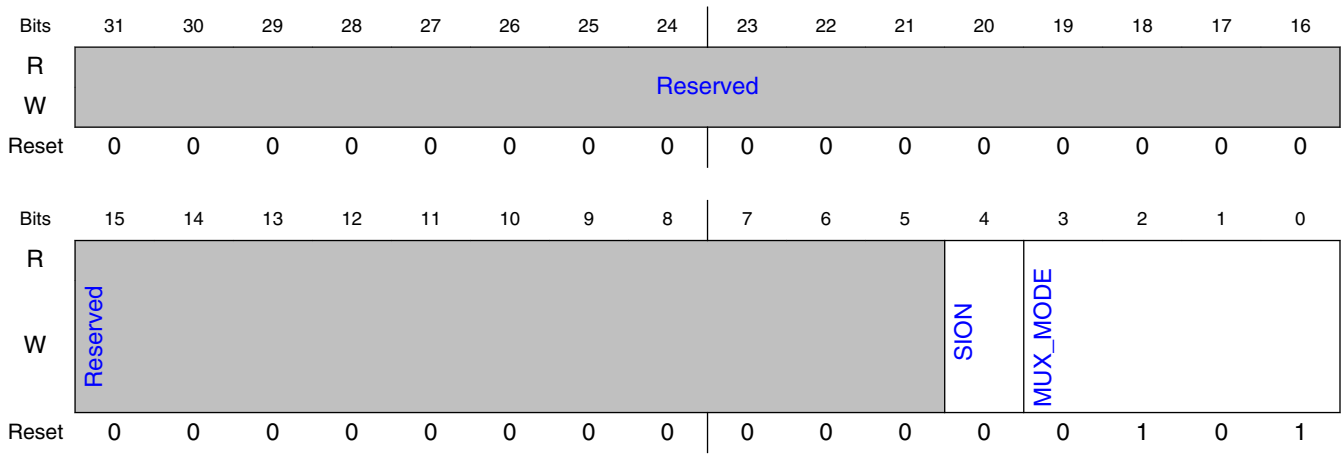
12.4.6.14.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_12	40h

12.4.6.14.2 Function

SW_MUX_CTL Register

12.4.6.14.3 Diagram



12.4.6.14.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_12
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_12. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR03 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT04 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: GPT5_COMPARE2 of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO12 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D12 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO12 of instance: GPIO7

12.4.6.15 SW_MUX_CTL_PAD_GPIO_EMC_B1_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_13)

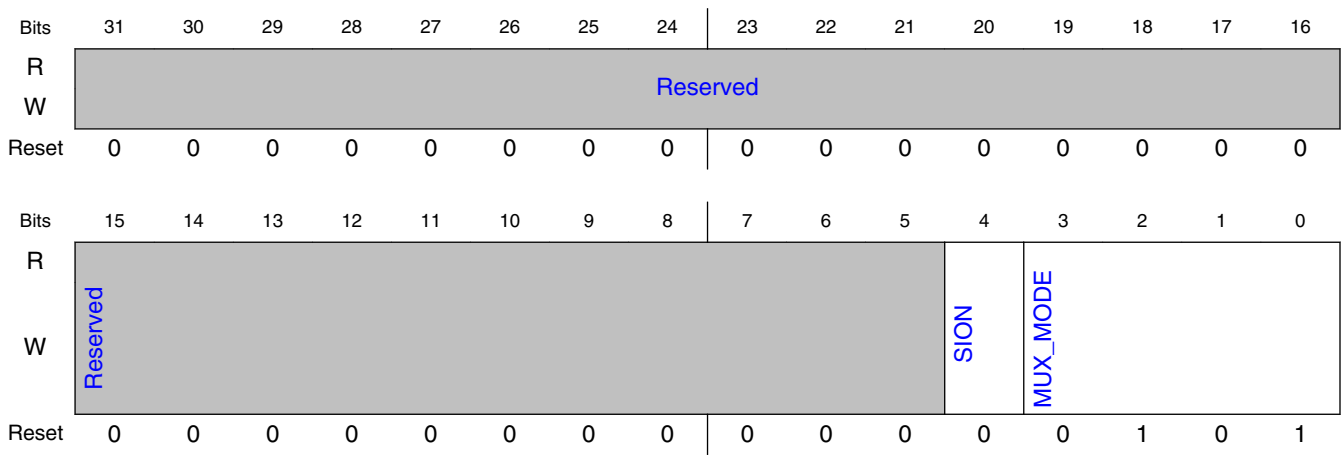
12.4.6.15.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_13	44h

12.4.6.15.2 Function

SW_MUX_CTL Register

12.4.6.15.3 Diagram



12.4.6.15.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_13
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_13. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR04 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT05 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: GPT5_COMPARE3 of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO13 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D13 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO13 of instance: GPIO7

12.4.6.16 SW_MUX_CTL_PAD_GPIO_EMC_B1_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_14)

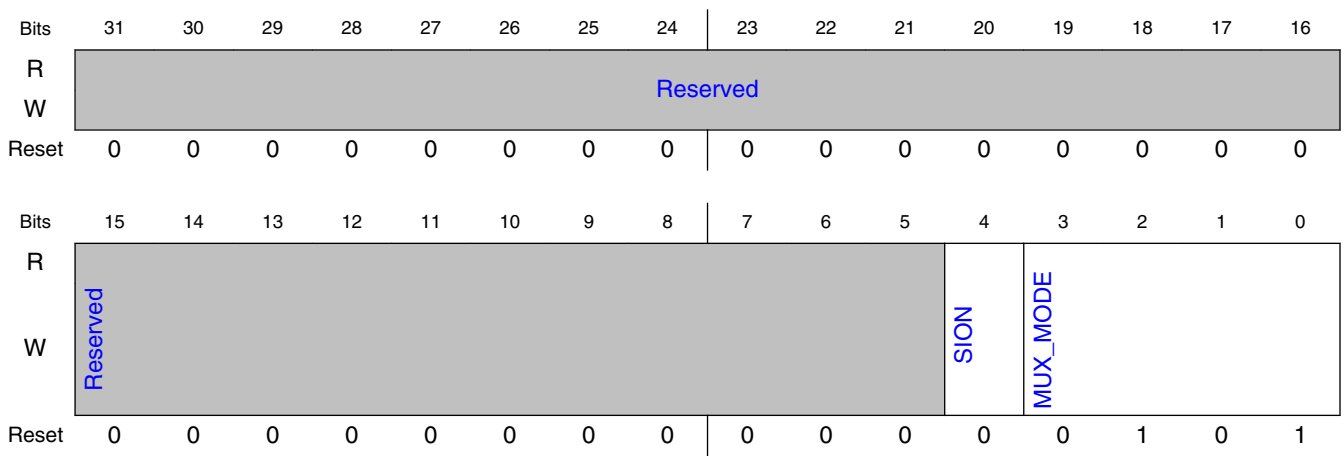
12.4.6.16.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_14	48h

12.4.6.16.2 Function

SW_MUX_CTL Register

12.4.6.16.3 Diagram



12.4.6.16.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_14
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_14.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_ADDR05 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT06 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: GPT5_CLK of instance: GPT5 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO14 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D14 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO14 of instance: GPIO7

12.4.6.17 SW_MUX_CTL_PAD_GPIO_EMC_B1_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_15)

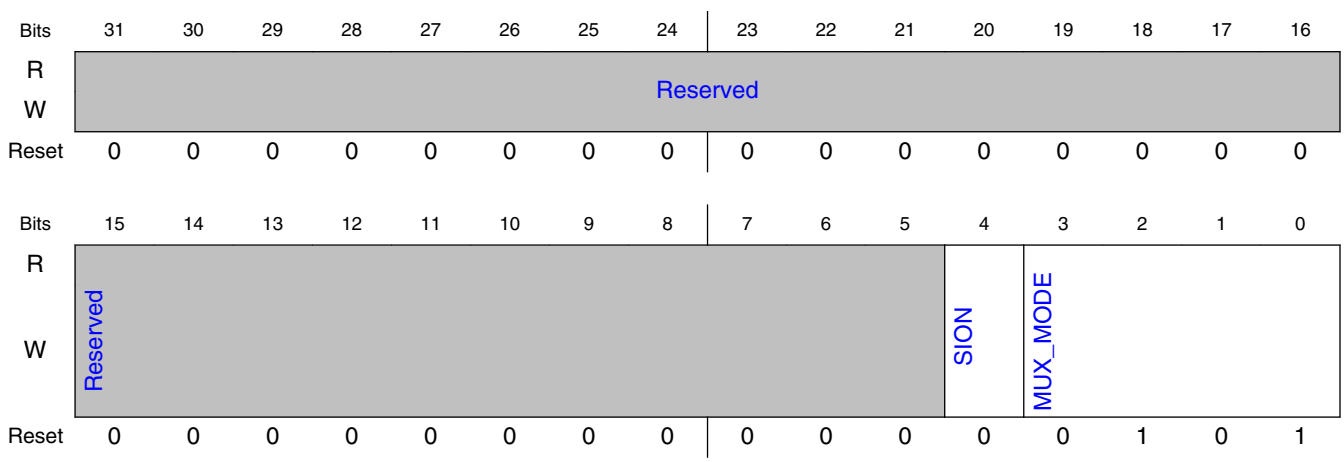
12.4.6.17.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_15	4Ch

12.4.6.17.2 Function

SW_MUX_CTL Register

12.4.6.17.3 Diagram



12.4.6.17.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_15
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_15. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR06 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT07 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO15 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D15 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO15 of instance: GPIO7

12.4.6.18 SW_MUX_CTL_PAD_GPIO_EMC_B1_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_16)

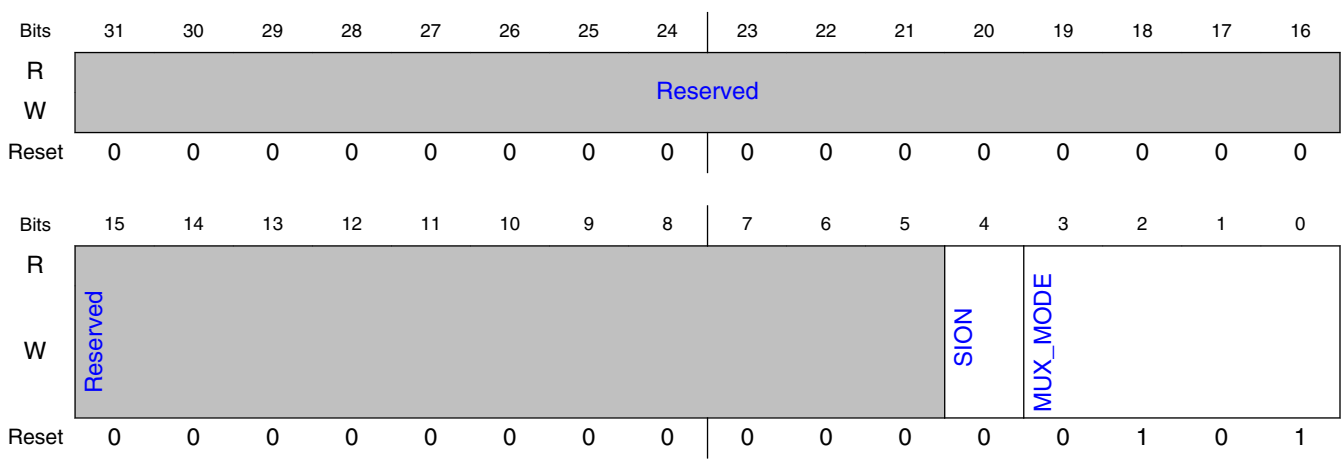
12.4.6.18.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_16	50h

12.4.6.18.2 Function

SW_MUX_CTL Register

12.4.6.18.3 Diagram



12.4.6.18.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_16
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_16. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR07 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT08 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO16 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D16 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO16 of instance: GPIO7

12.4.6.19 SW_MUX_CTL_PAD_GPIO_EMC_B1_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_17)

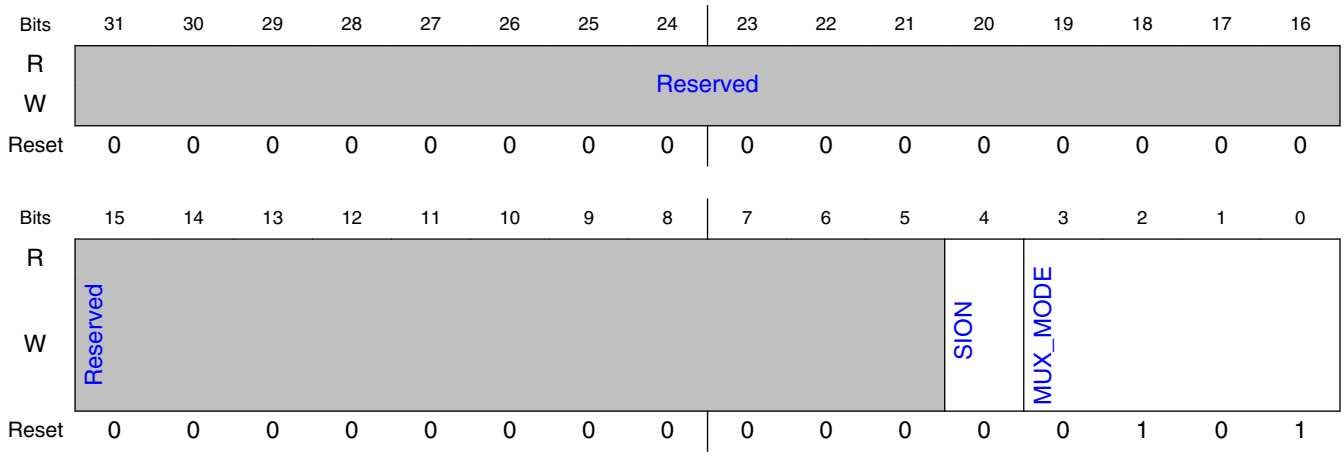
12.4.6.19.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_17	54h

12.4.6.19.2 Function

SW_MUX_CTL Register

12.4.6.19.3 Diagram



12.4.6.19.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_17
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: GPIO_EMC_B1_17. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR08 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM3_A of instance: FLEXPWM4 0010b - Select mux mode: ALT2 mux port: TMR1_TIMER0 of instance: TMR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO17 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D17 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO17 of instance: GPIO7

12.4.6.20 SW_MUX_CTL_PAD_GPIO_EMC_B1_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_18)

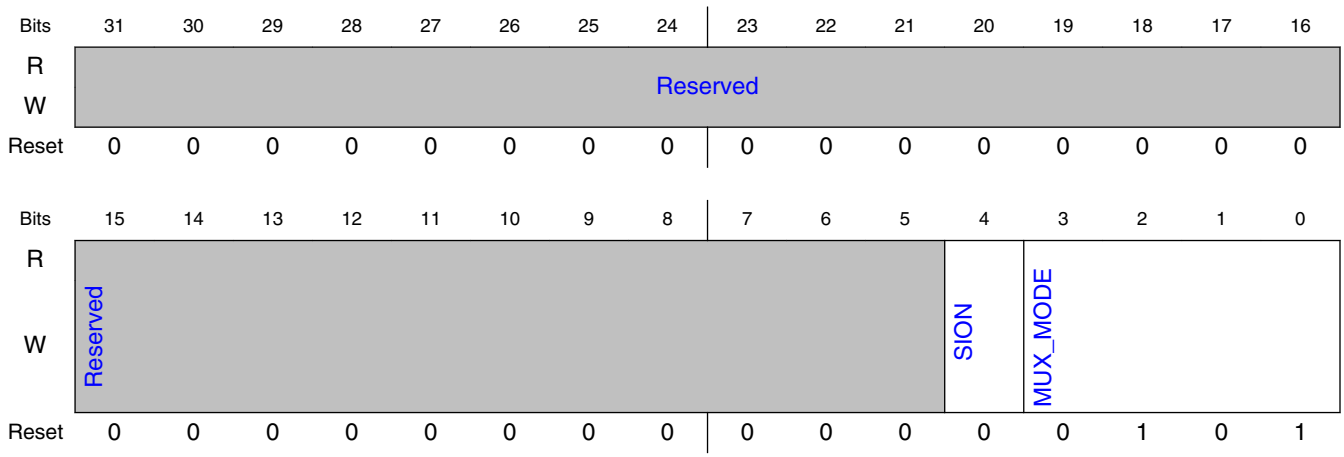
12.4.6.20.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_18	58h

12.4.6.20.2 Function

SW_MUX_CTL Register

12.4.6.20.3 Diagram



12.4.6.20.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_18
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_18. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR09 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM4_PWM3_B of instance: FLEXPWM4 0010b - Select mux mode: ALT2 mux port: TMR2_TIMER0 of instance: TMR2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO18 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D18 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO18 of instance: GPIO7

12.4.6.21 SW_MUX_CTL_PAD_GPIO_EMC_B1_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_19)

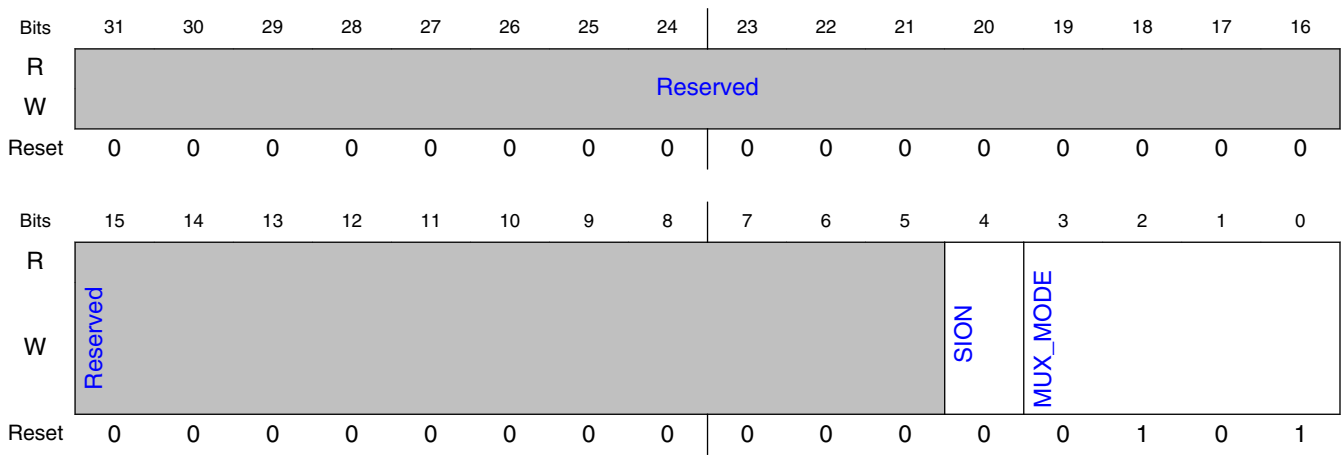
12.4.6.21.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_19	5Ch

12.4.6.21.2 Function

SW_MUX_CTL Register

12.4.6.21.3 Diagram



12.4.6.21.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_19
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_19. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR11 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM3_A of instance: FLEXPWM2 0010b - Select mux mode: ALT2 mux port: TMR3_TIMER0 of instance: TMR3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO19 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D19 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO19 of instance: GPIO7

12.4.6.22 SW_MUX_CTL_PAD_GPIO_EMC_B1_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_20)

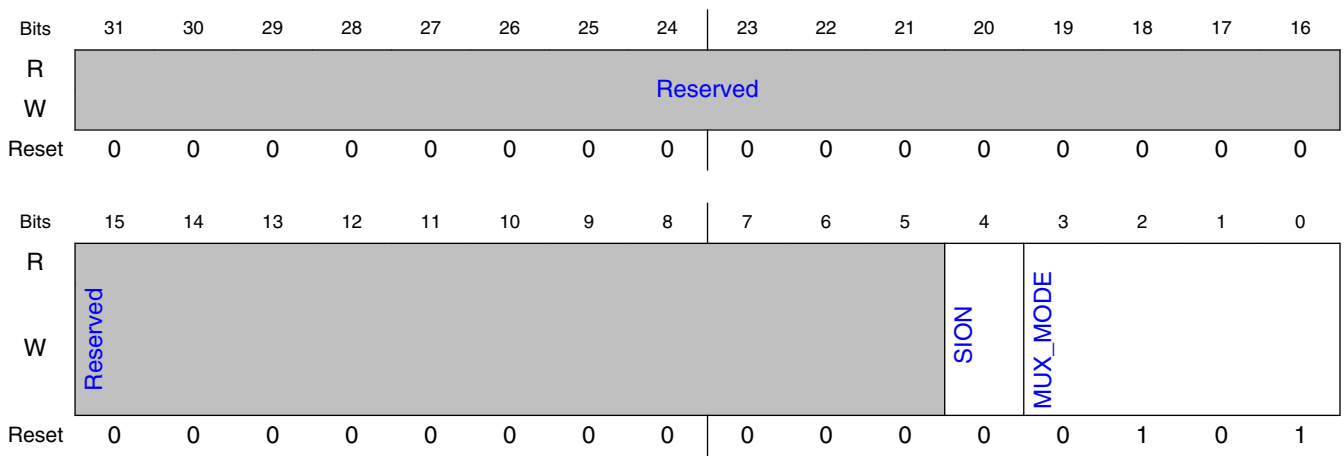
12.4.6.22.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_20	60h

12.4.6.22.2 Function

SW_MUX_CTL Register

12.4.6.22.3 Diagram



12.4.6.22.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_20
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_20.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_ADDR12 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM2_PWM3_B of instance: FLEXPWM2 0010b - Select mux mode: ALT2 mux port: TMR4_TIMER0 of instance: TMR4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO20 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D20 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO20 of instance: GPIO7

12.4.6.23 SW_MUX_CTL_PAD_GPIO_EMC_B1_21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_21)

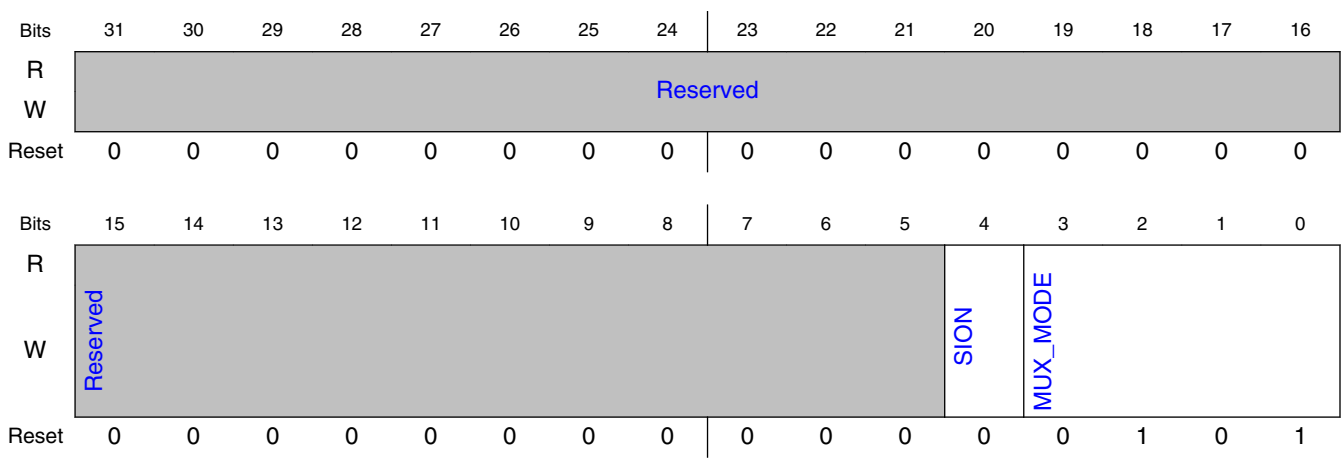
12.4.6.23.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_21	64h

12.4.6.23.2 Function

SW_MUX_CTL Register

12.4.6.23.3 Diagram



12.4.6.23.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_21
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_21. 0000b - Select mux mode: ALT0 mux port: SEMC_BA0 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM3_A of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO21 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D21 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO21 of instance: GPIO7

12.4.6.24 SW_MUX_CTL_PAD_GPIO_EMC_B1_22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_22)

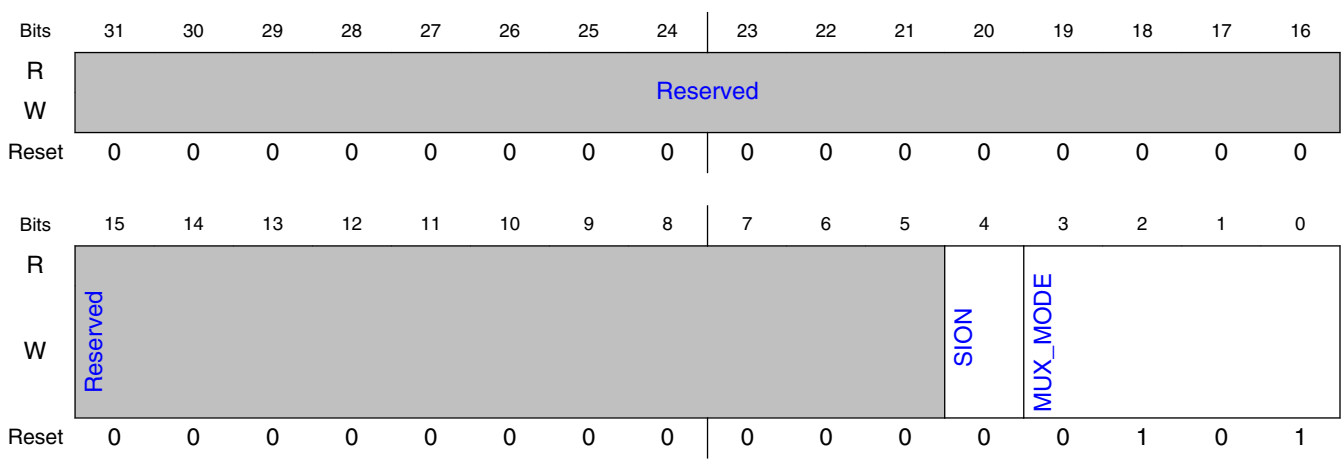
12.4.6.24.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_22	68h

12.4.6.24.2 Function

SW_MUX_CTL Register

12.4.6.24.3 Diagram



12.4.6.24.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_22
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_22. 0000b - Select mux mode: ALT0 mux port: SEMC_BA1 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM3_B of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO22 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D22 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO22 of instance: GPIO7

12.4.6.25 SW_MUX_CTL_PAD_GPIO_EMC_B1_23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_23)

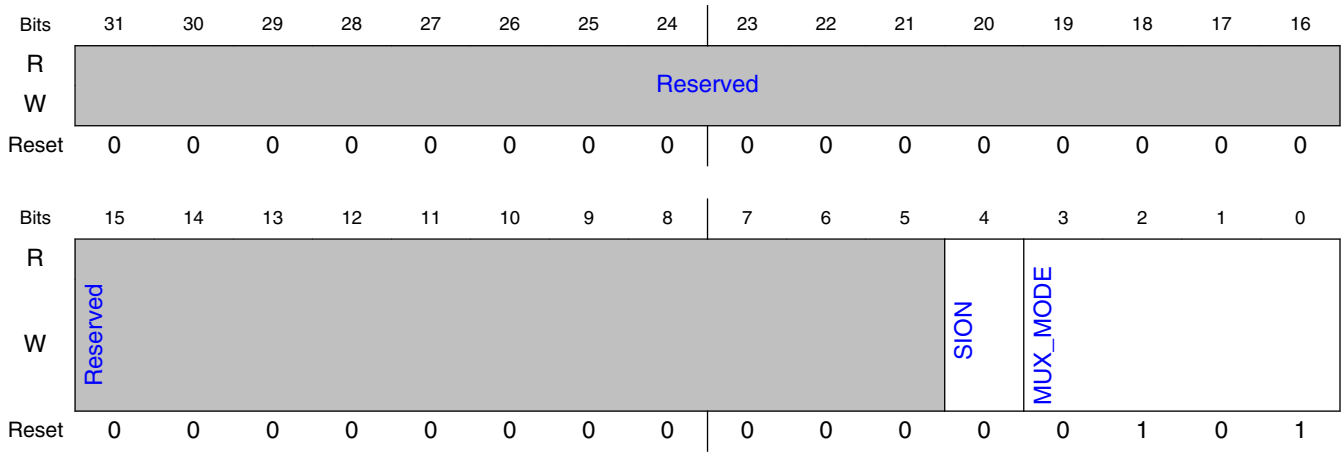
12.4.6.25.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_23	6Ch

12.4.6.25.2 Function

SW_MUX_CTL Register

12.4.6.25.3 Diagram



12.4.6.25.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_23
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_23. 0000b - Select mux mode: ALT0 mux port: SEMC_ADDR10 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM0_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO23 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D23 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO23 of instance: GPIO7

12.4.6.26 SW_MUX_CTL_PAD_GPIO_EMC_B1_24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_24)

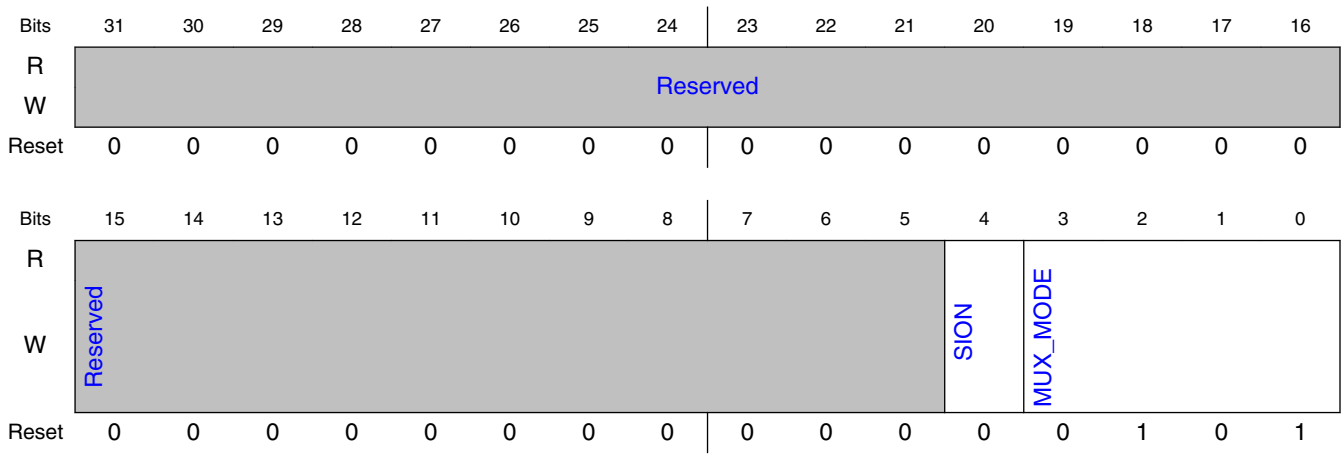
12.4.6.26.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_24	70h

12.4.6.26.2 Function

SW_MUX_CTL Register

12.4.6.26.3 Diagram



12.4.6.26.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_24
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_24. 0000b - Select mux mode: ALT0 mux port: SEMC_CAS of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM0_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO24 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D24 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO24 of instance: GPIO7

12.4.6.27 SW_MUX_CTL_PAD_GPIO_EMC_B1_25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_25)

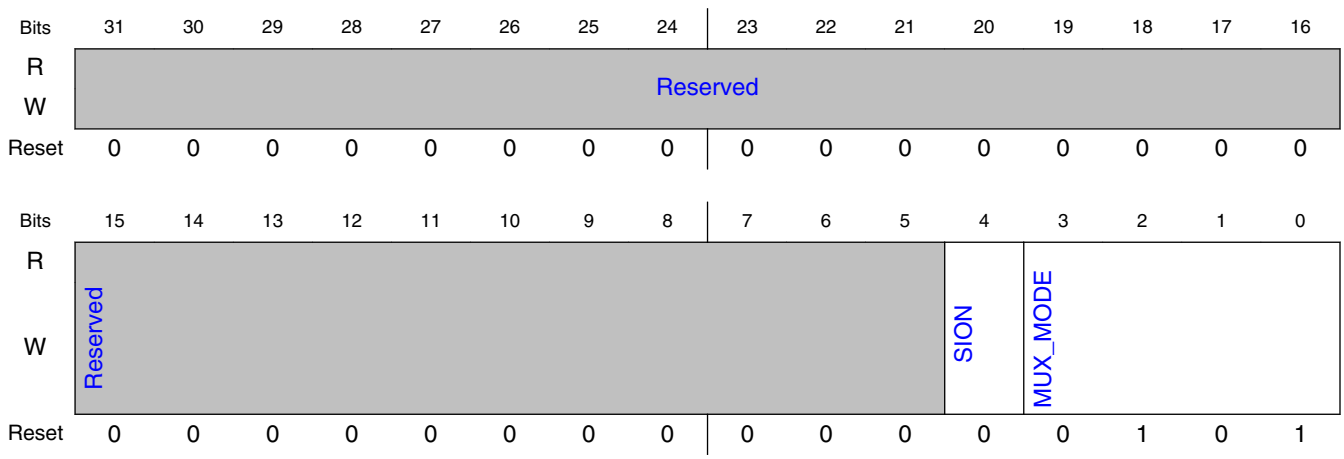
12.4.6.27.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_25	74h

12.4.6.27.2 Function

SW_MUX_CTL Register

12.4.6.27.3 Diagram



12.4.6.27.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_25
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_25. 0000b - Select mux mode: ALT0 mux port: SEMC_RAS of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM1_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO25 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D25 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO25 of instance: GPIO7

12.4.6.28 SW_MUX_CTL_PAD_GPIO_EMC_B1_26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_26)

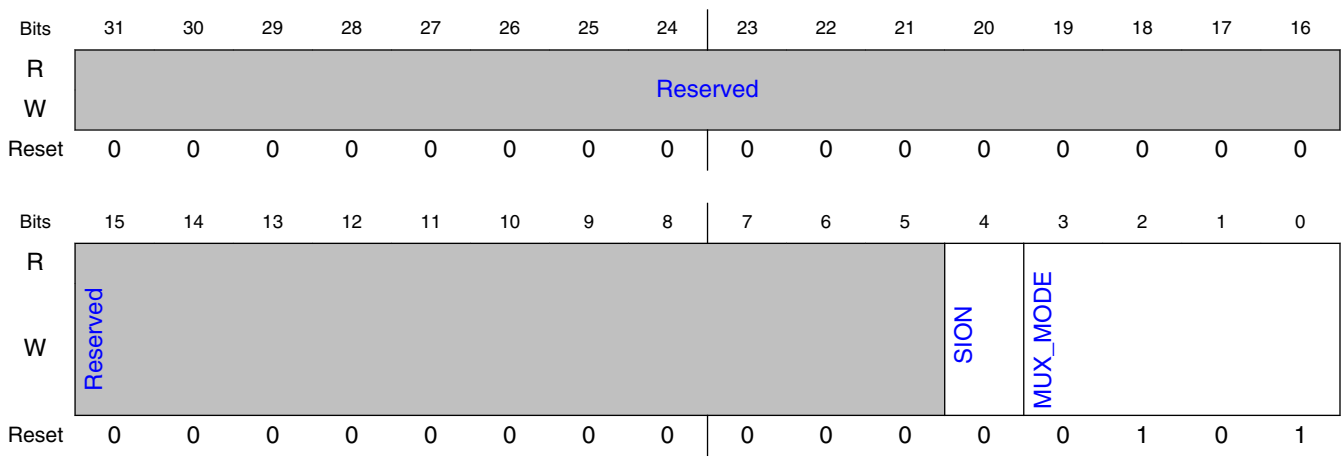
12.4.6.28.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_26	78h

12.4.6.28.2 Function

SW_MUX_CTL Register

12.4.6.28.3 Diagram



12.4.6.28.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_26
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_26.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_CLK of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM1_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO26 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D26 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO26 of instance: GPIO7

12.4.6.29 SW_MUX_CTL_PAD_GPIO_EMC_B1_27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_27)

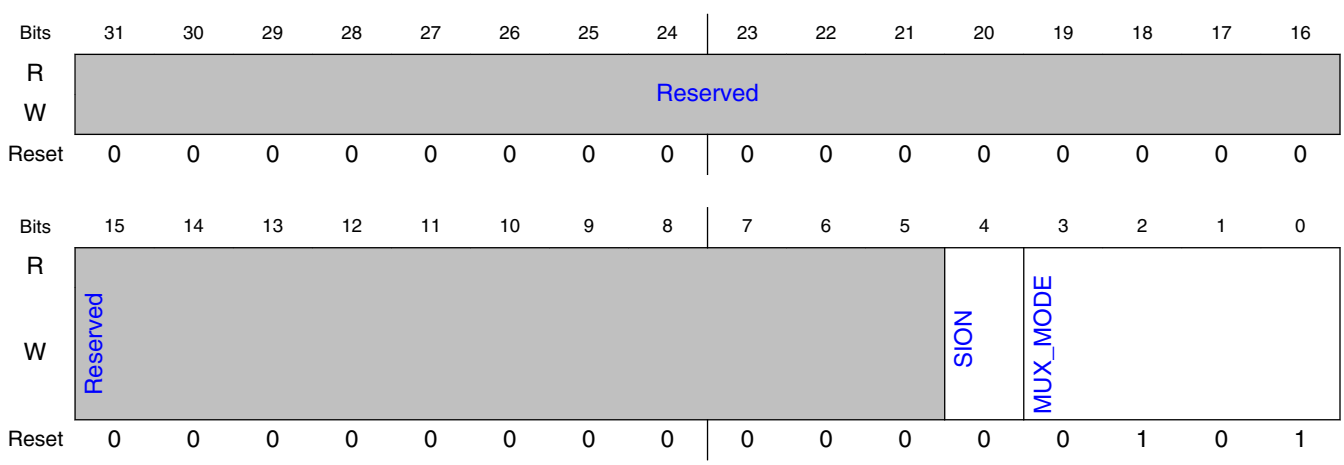
12.4.6.29.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_27	7Ch

12.4.6.29.2 Function

SW_MUX_CTL Register

12.4.6.29.3 Diagram



12.4.6.29.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_27
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_27. 0000b - Select mux mode: ALT0 mux port: SEMC_CKE of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM2_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO27 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D27 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO27 of instance: GPIO7

12.4.6.30 SW_MUX_CTL_PAD_GPIO_EMC_B1_28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_28)

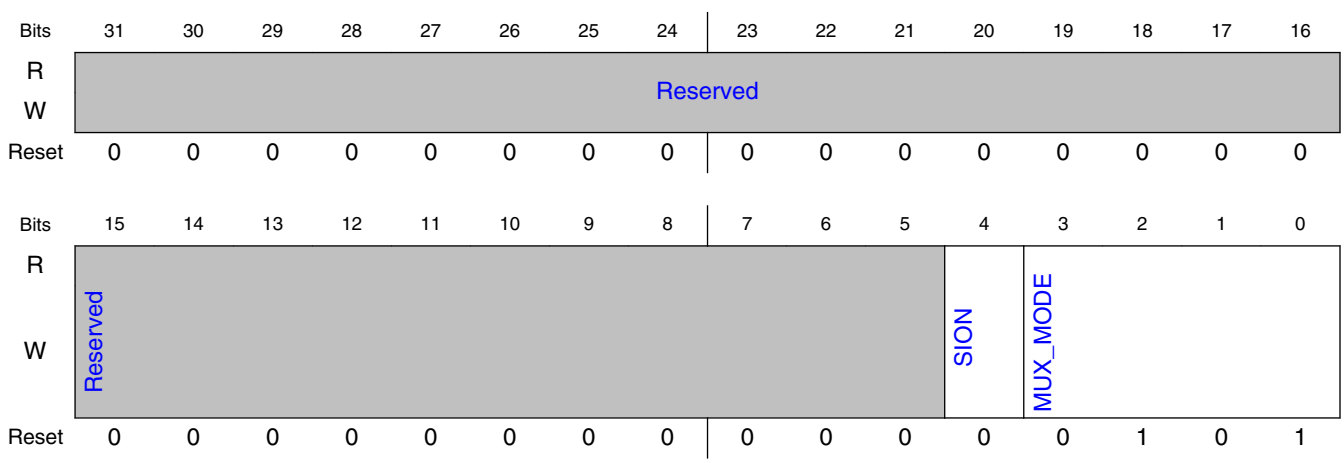
12.4.6.30.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_28	80h

12.4.6.30.2 Function

SW_MUX_CTL Register

12.4.6.30.3 Diagram



12.4.6.30.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_28
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_28. 0000b - Select mux mode: ALT0 mux port: SEMC_WE of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM2_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO28 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D28 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO28 of instance: GPIO7

12.4.6.31 SW_MUX_CTL_PAD_GPIO_EMC_B1_29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_29)

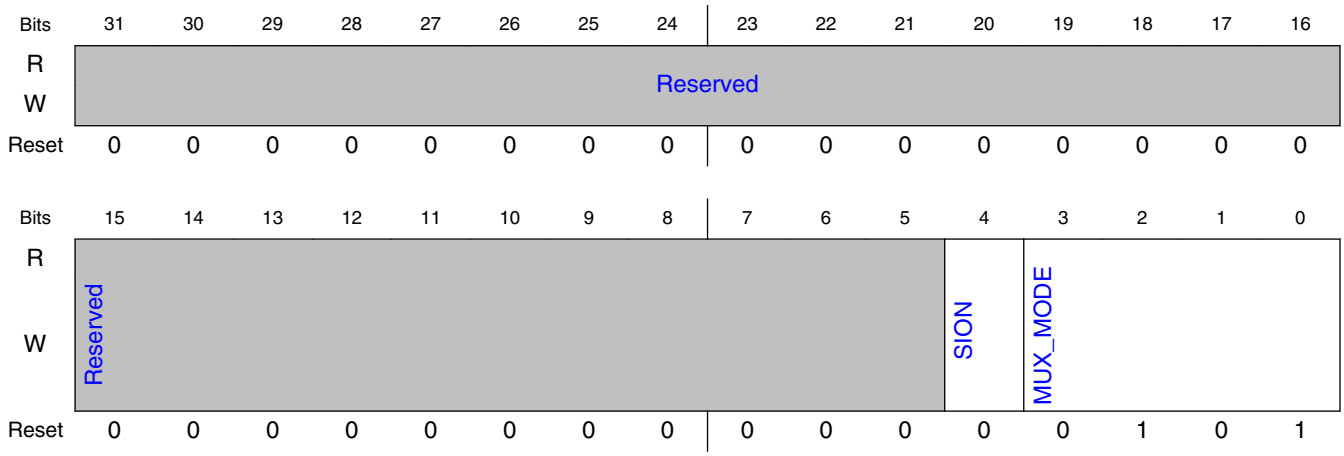
12.4.6.31.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_29	84h

12.4.6.31.2 Function

SW_MUX_CTL Register

12.4.6.31.3 Diagram



12.4.6.31.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_29
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_29. 0000b - Select mux mode: ALT0 mux port: SEMC_CS0 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM0_A of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO29 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D29 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO29 of instance: GPIO7

12.4.6.32 SW_MUX_CTL_PAD_GPIO_EMC_B1_30 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_30)

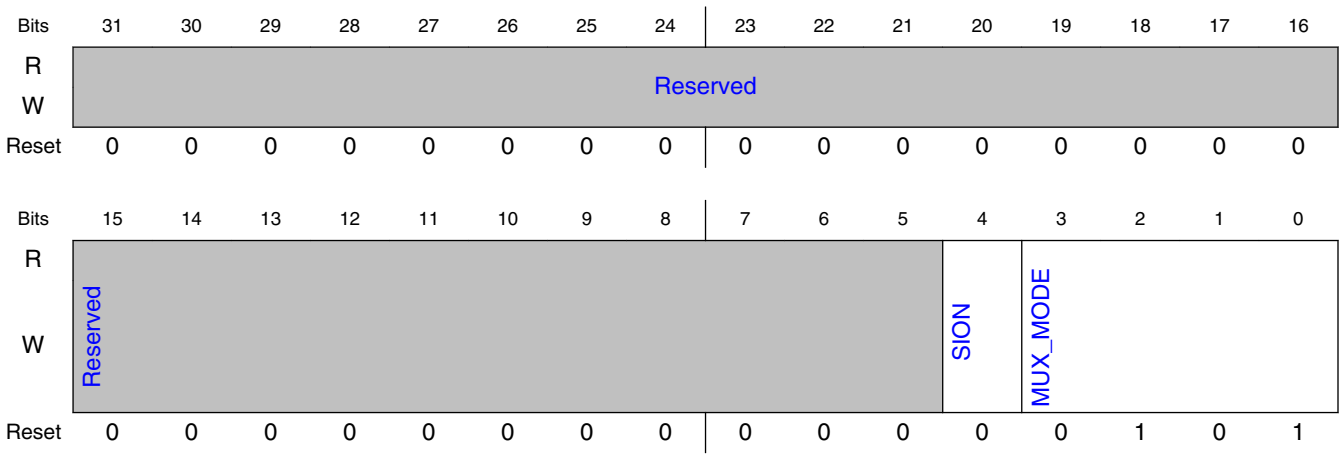
12.4.6.32.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_30	88h

12.4.6.32.2 Function

SW_MUX_CTL Register

12.4.6.32.3 Diagram



12.4.6.32.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_30
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_30. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA08 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM0_B of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO30 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D30 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO30 of instance: GPIO7

12.4.6.33 SW_MUX_CTL_PAD_GPIO_EMC_B1_31 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_31)

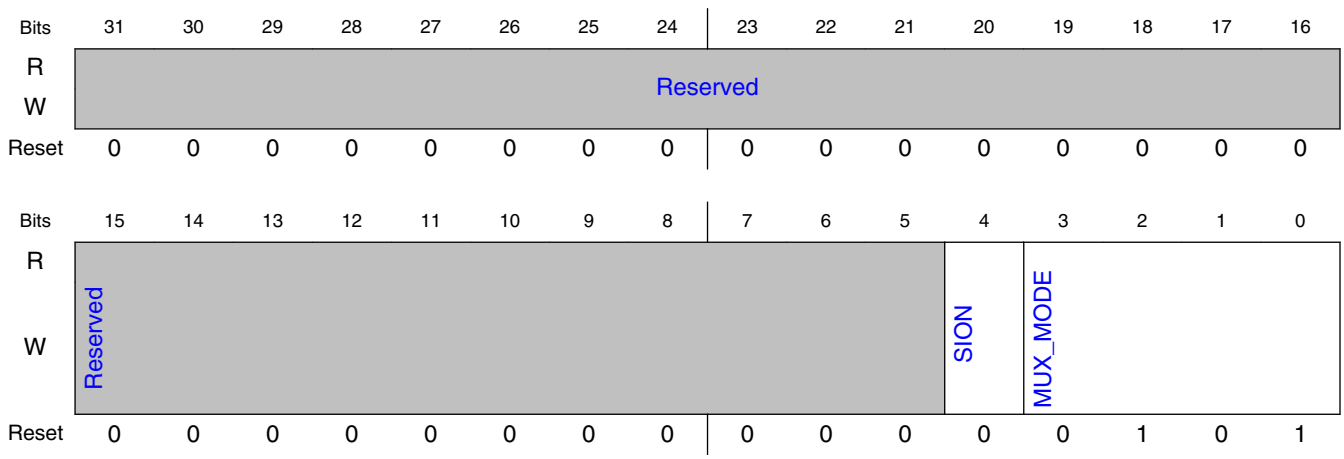
12.4.6.33.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_31	8Ch

12.4.6.33.2 Function

SW_MUX_CTL Register

12.4.6.33.3 Diagram



12.4.6.33.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_31
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_31. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA09 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM1_A of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX1_IO31 of instance: GPIO_MUX1 1000b - Select mux mode: ALT8 mux port: FLEXIO1_D31 of instance: FLEXIO1 1010b - Select mux mode: ALT10 mux port: GPIO7_IO31 of instance: GPIO7

12.4.6.34 SW_MUX_CTL_PAD_GPIO_EMC_B1_32 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_32)

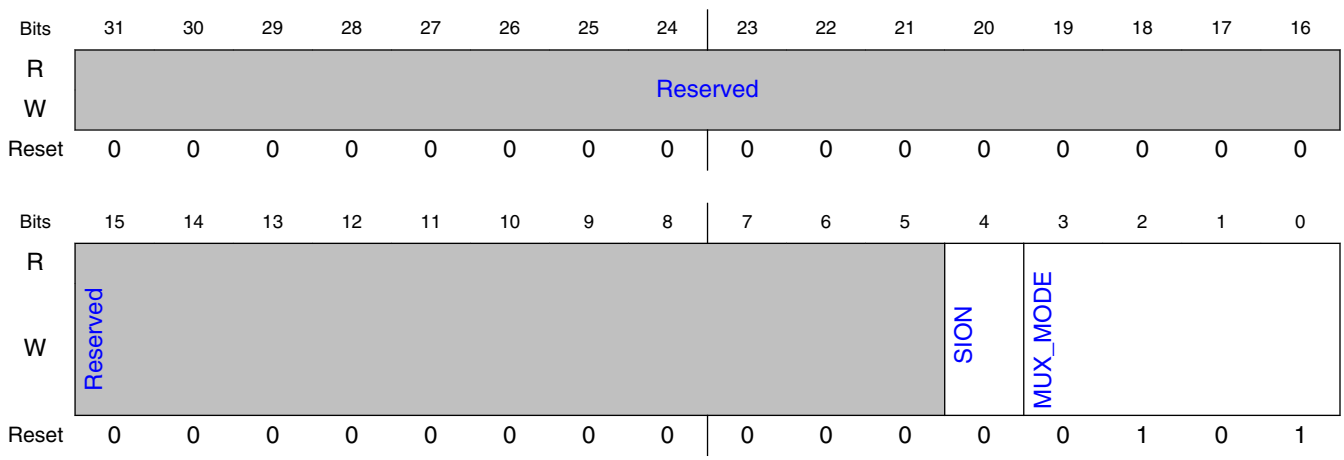
12.4.6.34.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_32	90h

12.4.6.34.2 Function

SW_MUX_CTL Register

12.4.6.34.3 Diagram



12.4.6.34.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_32
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_EMC_B1_32.

Field	Function
	0000b - Select mux mode: ALT0 mux port: SEMC_DATA10 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM1_B of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO00 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO00 of instance: GPIO8

12.4.6.35 SW_MUX_CTL_PAD_GPIO_EMC_B1_33 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_33)

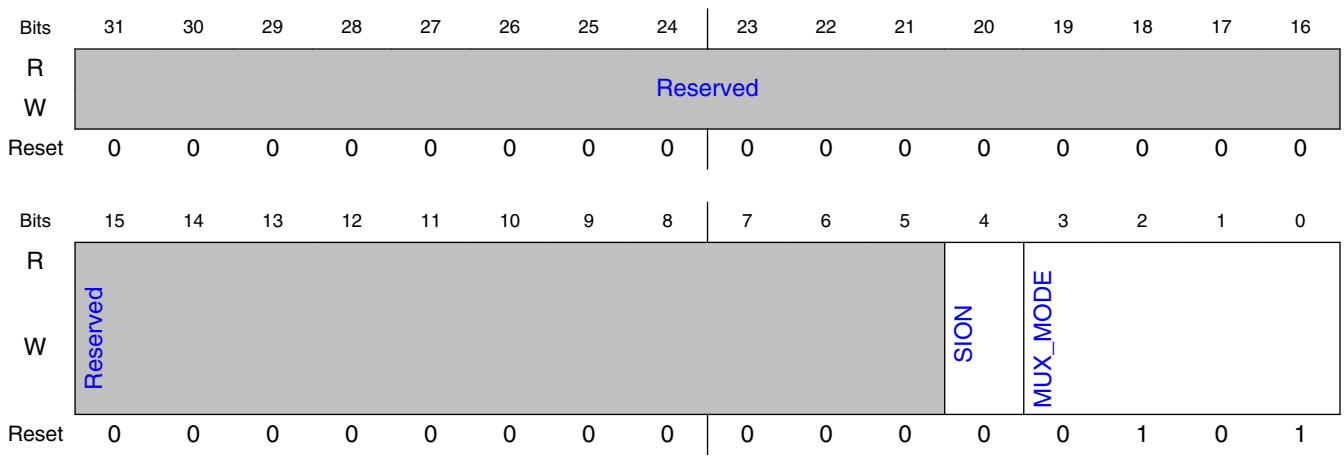
12.4.6.35.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_33	94h

12.4.6.35.2 Function

SW_MUX_CTL Register

12.4.6.35.3 Diagram



12.4.6.35.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_33
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_EMC_B1_33. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA11 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM2_A of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO01 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO01 of instance: GPIO8

12.4.6.36 SW_MUX_CTL_PAD_GPIO_EMC_B1_34 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_34)

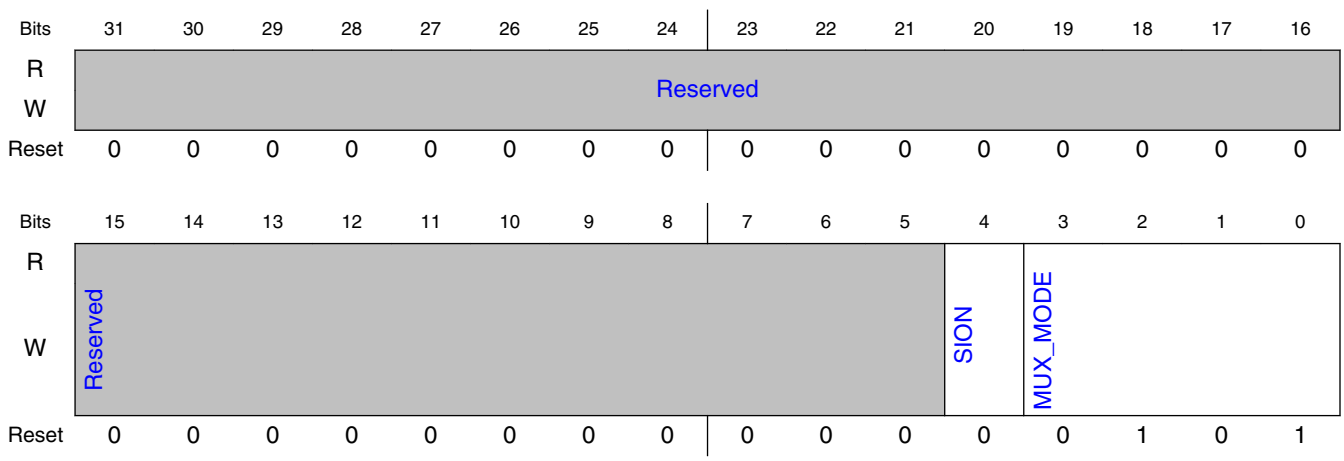
12.4.6.36.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_34	98h

12.4.6.36.2 Function

SW_MUX_CTL Register

12.4.6.36.3 Diagram



12.4.6.36.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_34
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_EMC_B1_34. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA12 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM3_PWM2_B of instance: FLEXPWM3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO02 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO02 of instance: GPIO8

12.4.6.37 SW_MUX_CTL_PAD_GPIO_EMC_B1_35 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_35)

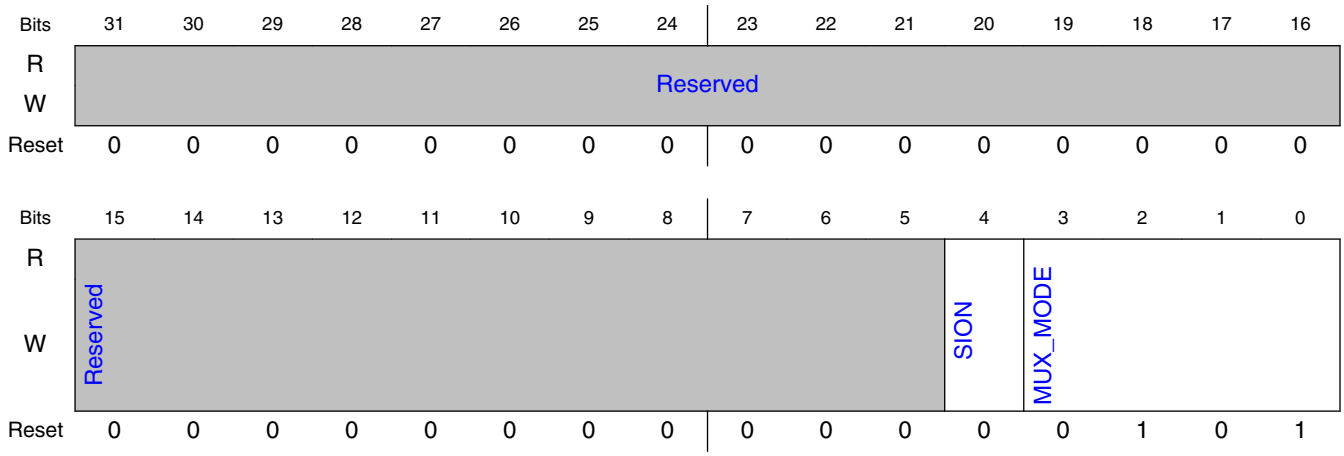
12.4.6.37.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_35	9Ch

12.4.6.37.2 Function

SW_MUX_CTL Register

12.4.6.37.3 Diagram



12.4.6.37.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_35
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_EMC_B1_35. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA13 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT09 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO03 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO03 of instance: GPIO8

12.4.6.38 SW_MUX_CTL_PAD_GPIO_EMC_B1_36 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_36)

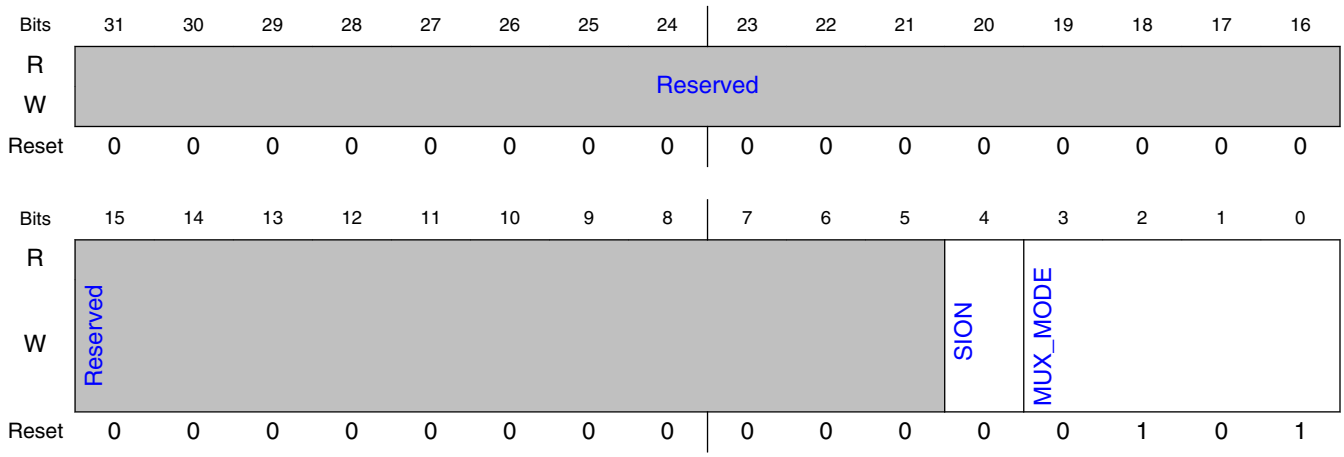
12.4.6.38.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_36	A0h

12.4.6.38.2 Function

SW_MUX_CTL Register

12.4.6.38.3 Diagram



12.4.6.38.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_36
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: GPIO_EMC_B1_36. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA14 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT10 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO04 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO04 of instance: GPIO8

12.4.6.39 SW_MUX_CTL_PAD_GPIO_EMC_B1_37 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_37)

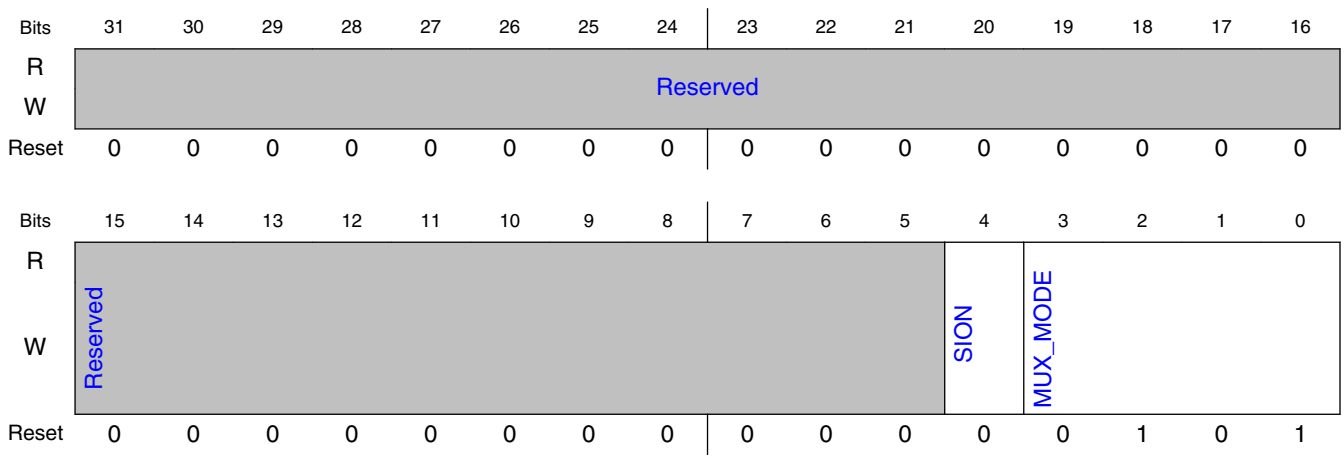
12.4.6.39.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_37	A4h

12.4.6.39.2 Function

SW_MUX_CTL Register

12.4.6.39.3 Diagram



12.4.6.39.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_37
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: GPIO_EMC_B1_37. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA15 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT11 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO05 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO05 of instance: GPIO8

12.4.6.40 SW_MUX_CTL_PAD_GPIO_EMC_B1_38 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_38)

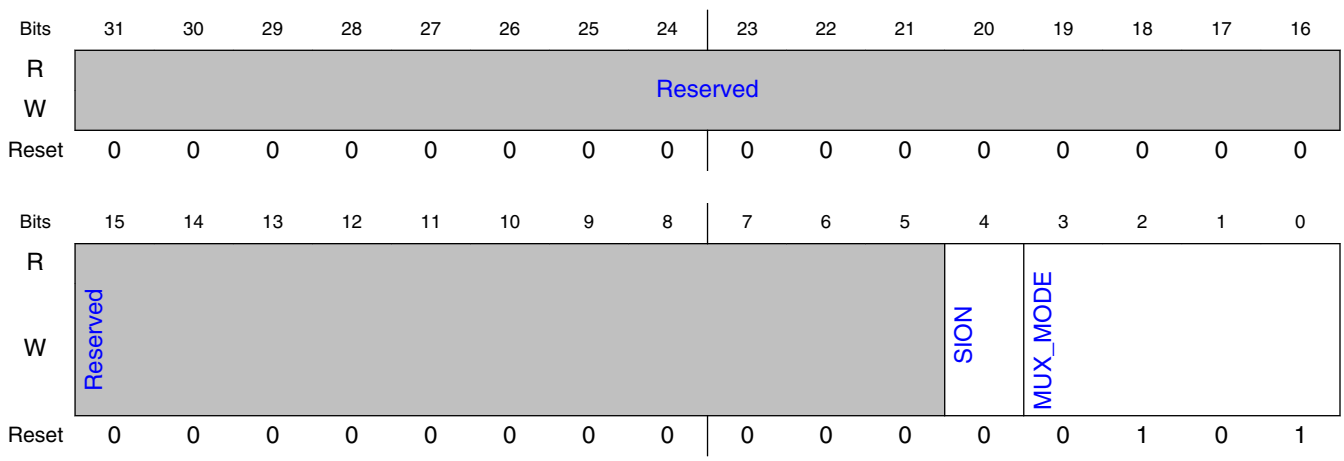
12.4.6.40.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_38	A8h

12.4.6.40.2 Function

SW_MUX_CTL Register

12.4.6.40.3 Diagram



12.4.6.40.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_38
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_38. 0000b - Select mux mode: ALT0 mux port: SEMC_DM01 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM3_A of instance: FLEXPWM1 0010b - Select mux mode: ALT2 mux port: TMR1_TIMER1 of instance: TMR1

Field	Function
	0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO06 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO06 of instance: GPIO8

12.4.6.41 SW_MUX_CTL_PAD_GPIO_EMC_B1_39 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_39)

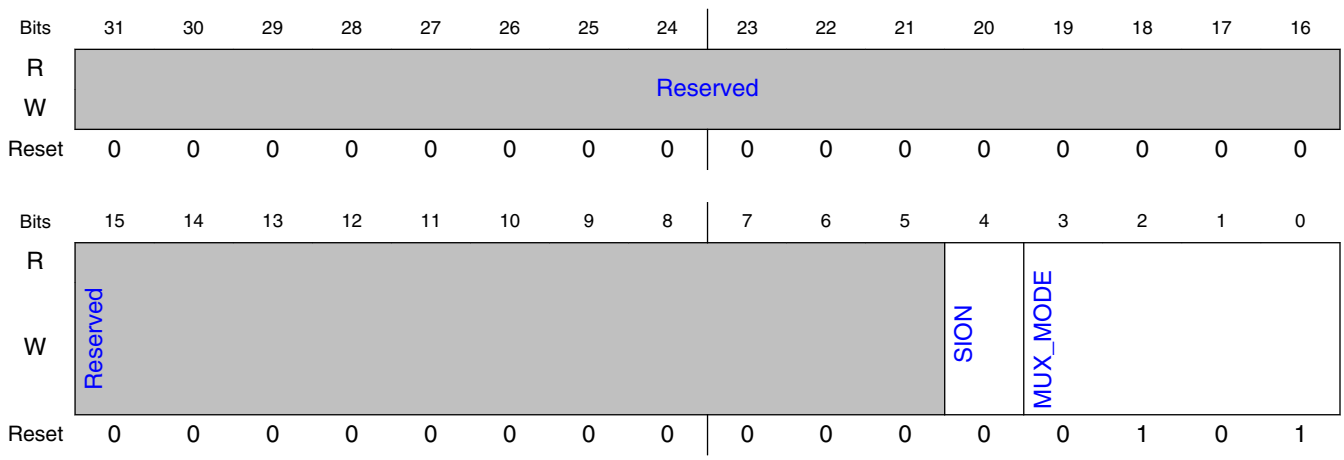
12.4.6.41.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_39	ACh

12.4.6.41.2 Function

SW_MUX_CTL Register

12.4.6.41.3 Diagram



12.4.6.41.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_39
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: GPIO_EMC_B1_39. 0000b - Select mux mode: ALT0 mux port: SEMC_DQS of instance: SEMC 0001b - Select mux mode: ALT1 mux port: FLEXPWM1_PWM3_B of instance: FLEXPWM1 0010b - Select mux mode: ALT2 mux port: TMR2_TIMER1 of instance: TMR2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO07 of instance: GPIO_MUX2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO07 of instance: GPIO8

12.4.6.42 SW_MUX_CTL_PAD_GPIO_EMC_B1_40 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_40)

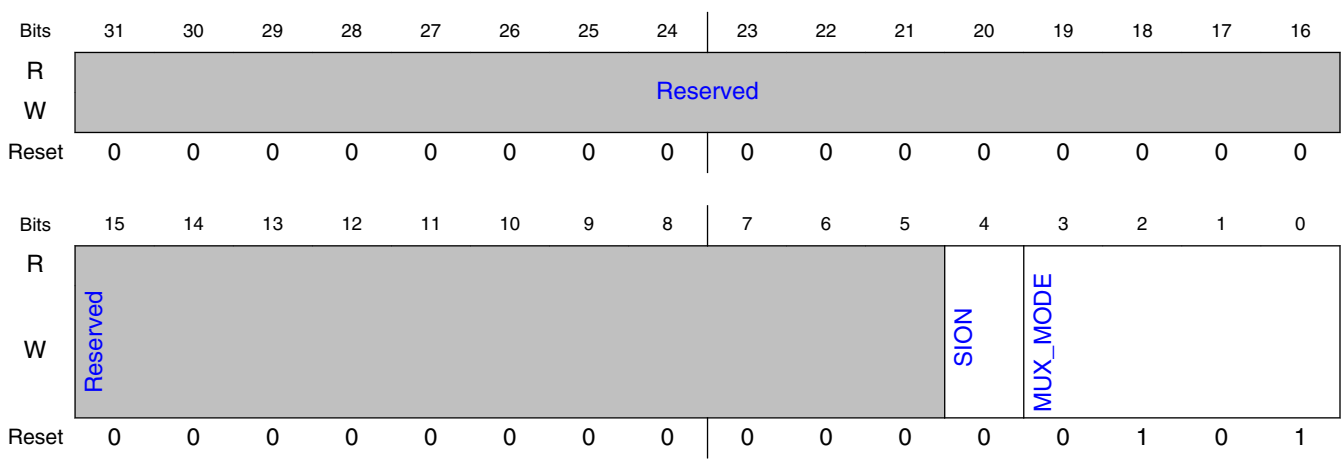
12.4.6.42.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_40	B0h

12.4.6.42.2 Function

SW_MUX_CTL Register

12.4.6.42.3 Diagram



12.4.6.42.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_40
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_EMC_B1_40. 0000b - Select mux mode: ALT0 mux port: SEMC_RDY of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT12 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: MQS_RIGHT of instance: MQS 0011b - Select mux mode: ALT3 mux port: LPUART6_TXD of instance: LPUART6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO08 of instance: GPIO_MUX2 0111b - Select mux mode: ALT7 mux port: ENET_1G_MDC of instance: ENET_1G 1001b - Select mux mode: ALT9 mux port: CCM_CLKO1 of instance: CCM 1010b - Select mux mode: ALT10 mux port: GPIO8_IO08 of instance: GPIO8

12.4.6.43 SW_MUX_CTL_PAD_GPIO_EMC_B1_41 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B1_41)

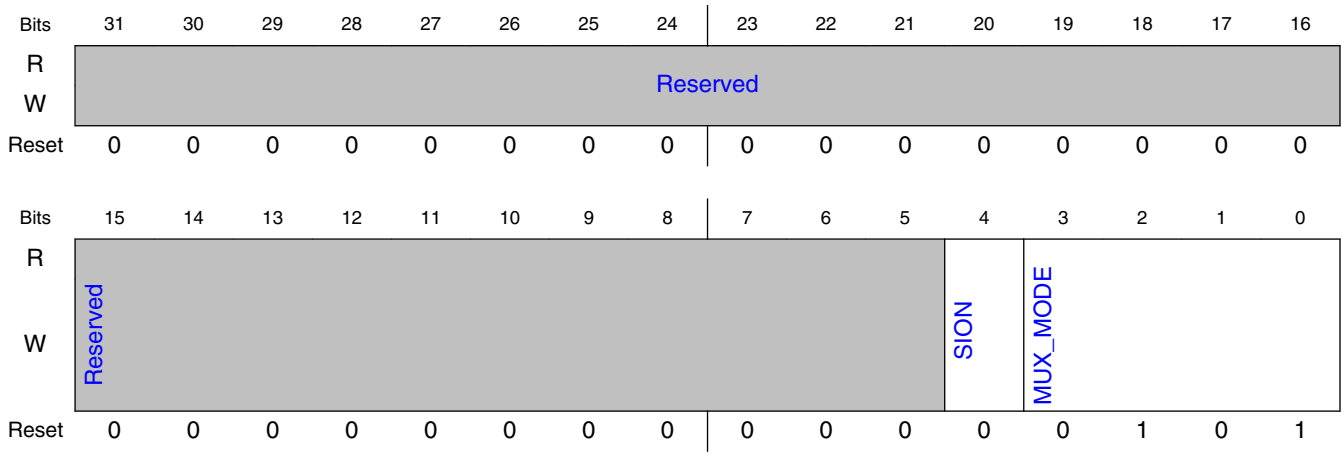
12.4.6.43.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B1_41	B4h

12.4.6.43.2 Function

SW_MUX_CTL Register

12.4.6.43.3 Diagram



12.4.6.43.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B1_41
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_EMC_B1_41. 0000b - Select mux mode: ALT0 mux port: SEMC_CSX00 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT13 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: MQS_LEFT of instance: MQS 0011b - Select mux mode: ALT3 mux port: LPUART6_RXD of instance: LPUART6 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA07 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO09 of instance: GPIO_MUX2 0111b - Select mux mode: ALT7 mux port: ENET_1G_MDIO of instance: ENET_1G 1001b - Select mux mode: ALT9 mux port: CCM_CLKO2 of instance: CCM 1010b - Select mux mode: ALT10 mux port: GPIO8_IO09 of instance: GPIO8

12.4.6.44 SW_MUX_CTL_PAD_GPIO_EMC_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_00)

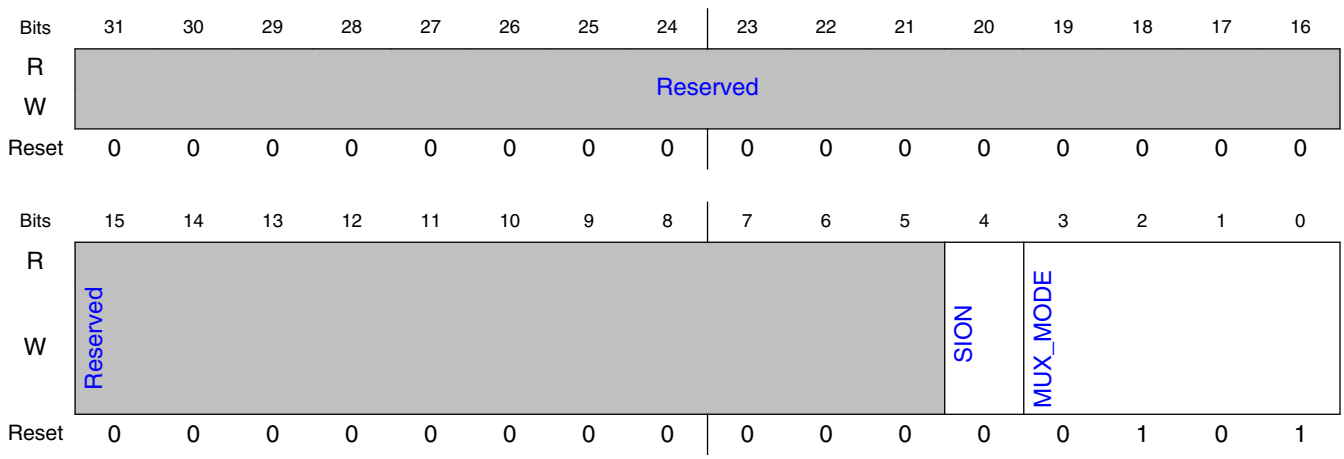
12.4.6.44.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_00	B8h

12.4.6.44.2 Function

SW_MUX_CTL Register

12.4.6.44.3 Diagram



12.4.6.44.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_00. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA16 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: CCM_ENET_REF_CLK_25M of instance: CCM 0010b - Select mux mode: ALT2 mux port: TMR3_TIMER1 of instance: TMR3 0011b - Select mux mode: ALT3 mux port: LPUART6_CTS_B of instance: LPUART6 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA06 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO10 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT20 of instance: XBAR1

Field	Function
	0111b - Select mux mode: ALT7 mux port: ENET_QOS_1588_EVENT1_OUT of instance: ENET_QOS 1000b - Select mux mode: ALT8 mux port: LPSPI1_SCK of instance: LPSPI1 1001b - Select mux mode: ALT9 mux port: LPI2C2_SCL of instance: LPI2C2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO10 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM0_A of instance: FLEXPWM3

12.4.6.45 SW_MUX_CTL_PAD_GPIO_EMC_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_01)

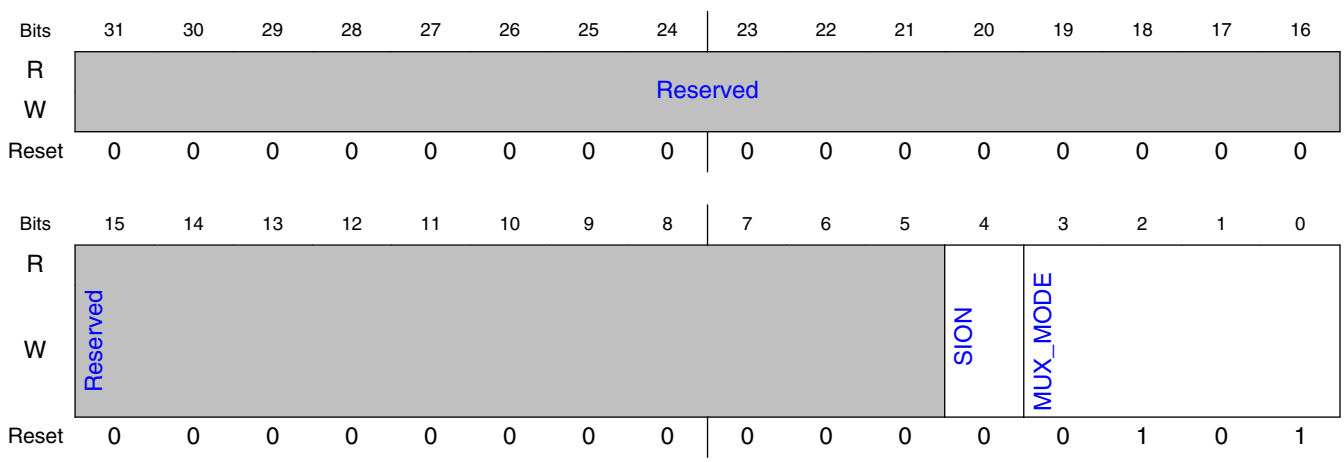
12.4.6.45.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_01	BCh

12.4.6.45.2 Function

SW_MUX_CTL Register

12.4.6.45.3 Diagram



12.4.6.45.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_01. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA17 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: USDHC2_CD_B of instance: USDHC2 0010b - Select mux mode: ALT2 mux port: TMR4_TIMER1 of instance: TMR4 0011b - Select mux mode: ALT3 mux port: LPUART6_RTS_B of instance: LPUART6 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA05 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO11 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT21 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_QOS_1588_EVENT1_IN of instance: ENET_QOS 1000b - Select mux mode: ALT8 mux port: LPSPI1_PCS0 of instance: LPSPI1 1001b - Select mux mode: ALT9 mux port: LPI2C2_SDA of instance: LPI2C2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO11 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM0_B of instance: FLEXPWM3

12.4.6.46 SW_MUX_CTL_PAD_GPIO_EMC_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_02)

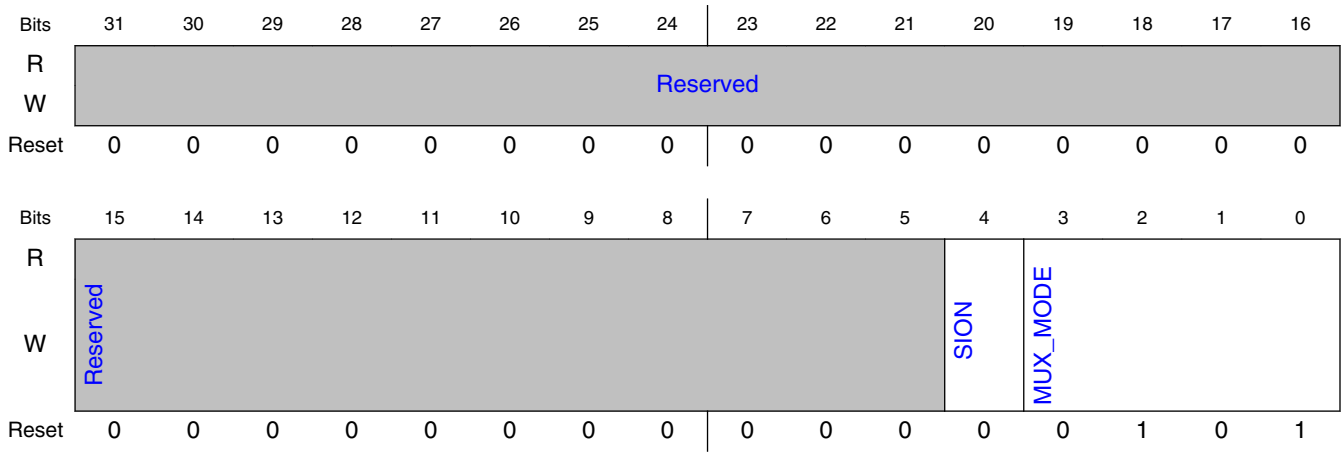
12.4.6.46.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_02	C0h

12.4.6.46.2 Function

SW_MUX_CTL Register

12.4.6.46.3 Diagram



12.4.6.46.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_02. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA18 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: USDHC2_WP of instance: USDHC2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA23 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA04 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO12 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT22 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_QOS_1588_EVENT1_AUX_IN of instance: ENET_QOS 1000b - Select mux mode: ALT8 mux port: LPSPI1_SOUT of instance: LPSPI1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO12 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM1_A of instance: FLEXPWM3

12.4.6.47 SW_MUX_CTL_PAD_GPIO_EMC_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_03)

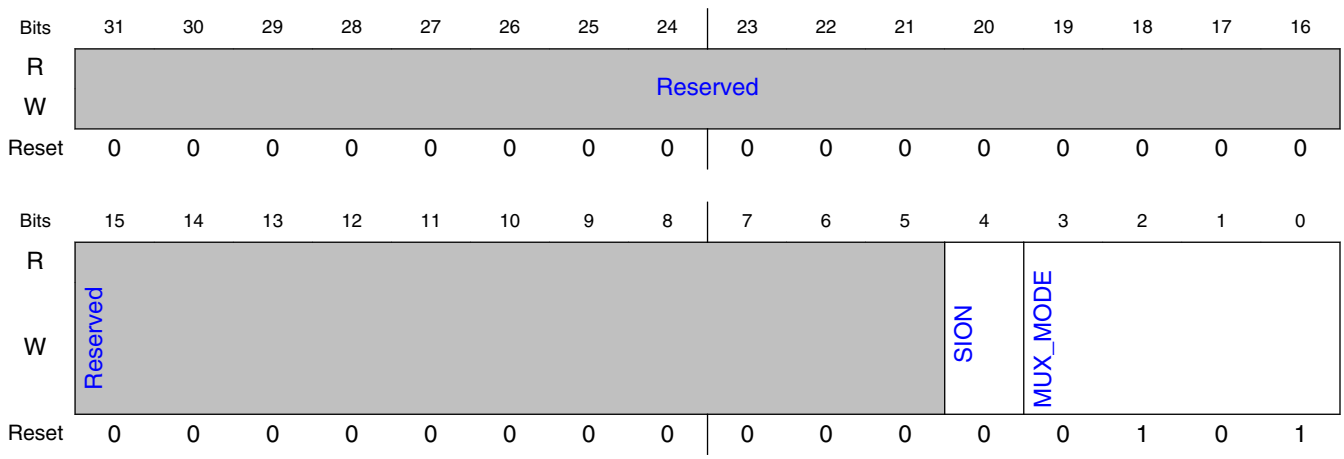
12.4.6.47.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_03	C4h

12.4.6.47.2 Function

SW_MUX_CTL Register

12.4.6.47.3 Diagram



12.4.6.47.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_03. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA19 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: USDHC2_VSELECT of instance: USDHC2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA22 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA03 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO13 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT23 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_TX_DATA03 of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI1_SIN of instance: LPSPI1

Field	Function
	1010b - Select mux mode: ALT10 mux port: GPIO8_IO13 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM1_B of instance: FLEXPWM3

12.4.6.48 SW_MUX_CTL_PAD_GPIO_EMC_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_04)

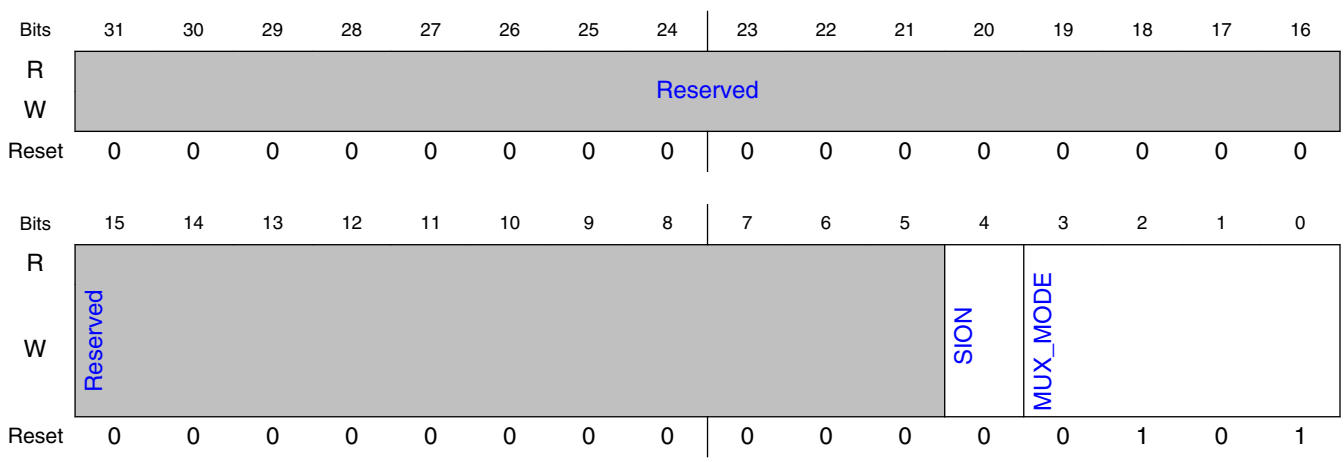
12.4.6.48.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_04	C8h

12.4.6.48.2 Function

SW_MUX_CTL Register

12.4.6.48.3 Diagram



12.4.6.48.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_04. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA20 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: USDHC2_RESET_B of instance: USDHC2 0010b - Select mux mode: ALT2 mux port: SAI2_MCLK of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA21 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA02 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO14 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT24 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_TX_DATA02 of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI3_SCK of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO8_IO14 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM2_A of instance: FLEXPWM3

12.4.6.49 SW_MUX_CTL_PAD_GPIO_EMC_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_05)

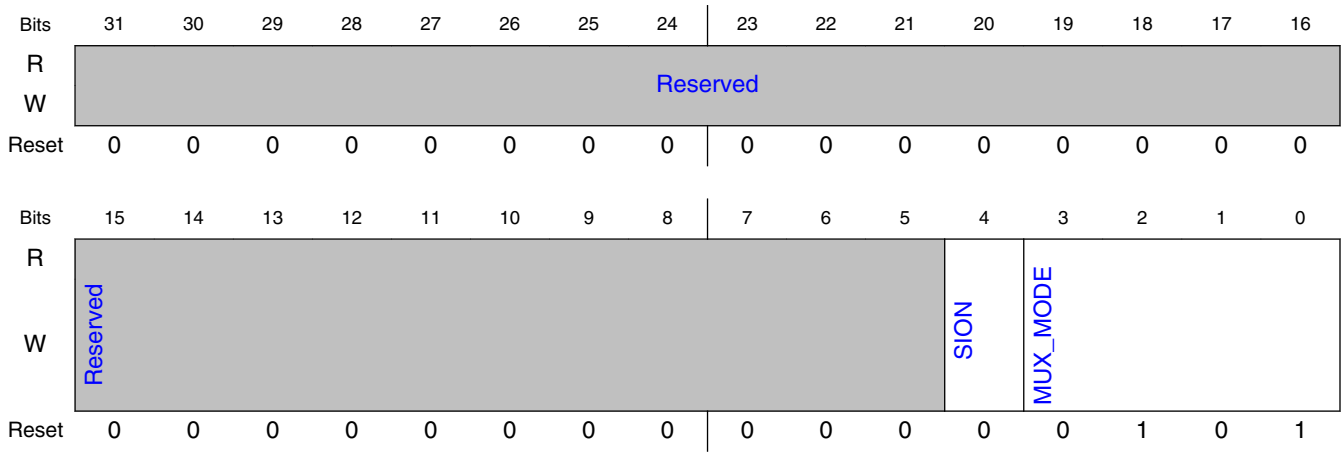
12.4.6.49.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_05	CCh

12.4.6.49.2 Function

SW_MUX_CTL Register

12.4.6.49.3 Diagram



12.4.6.49.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_05. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA21 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_CLK of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_RX_SYNC of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA20 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA01 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO15 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT25 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_RX_CLK of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI3_PCS0 of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: PIT1_TRIGGER0 of instance: PIT1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO15 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM2_B of instance: FLEXPWM3

12.4.6.50 SW_MUX_CTL_PAD_GPIO_EMC_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_06)

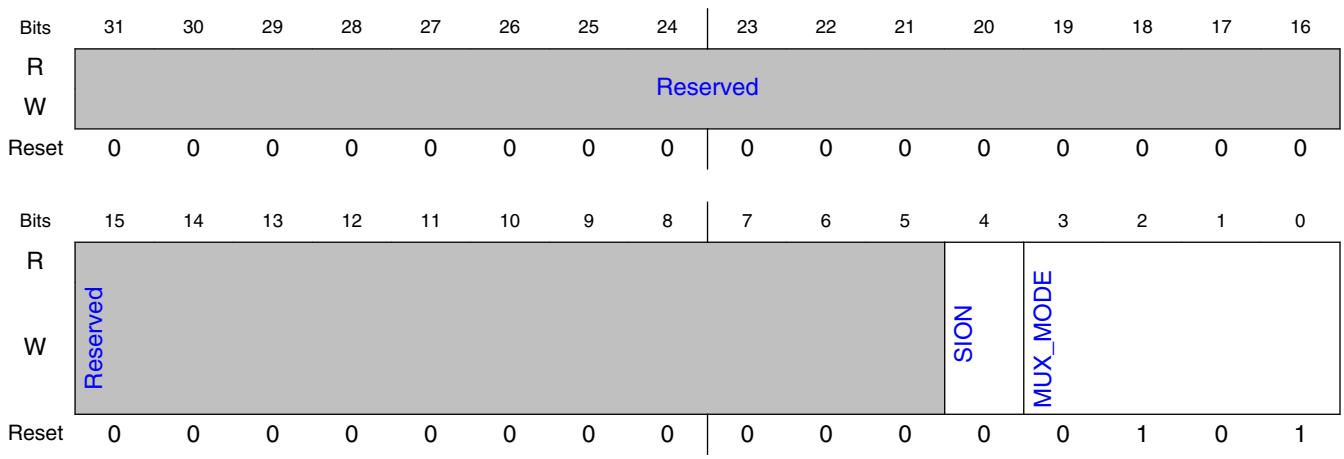
12.4.6.50.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_06	D0h

12.4.6.50.2 Function

SW_MUX_CTL Register

12.4.6.50.3 Diagram



12.4.6.50.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_06. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA22 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_CAPTURE1 of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_RX_BCLK of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA19 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DATA00 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO16 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT26 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_TX_ER of instance: ENET_1G

Field	Function
	1000b - Select mux mode: ALT8 mux port: LPSPI3_SOUT of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: PIT1_TRIGGER1 of instance: PIT1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO16 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM3_A of instance: FLEXPWM3

12.4.6.51 SW_MUX_CTL_PAD_GPIO_EMC_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_07)

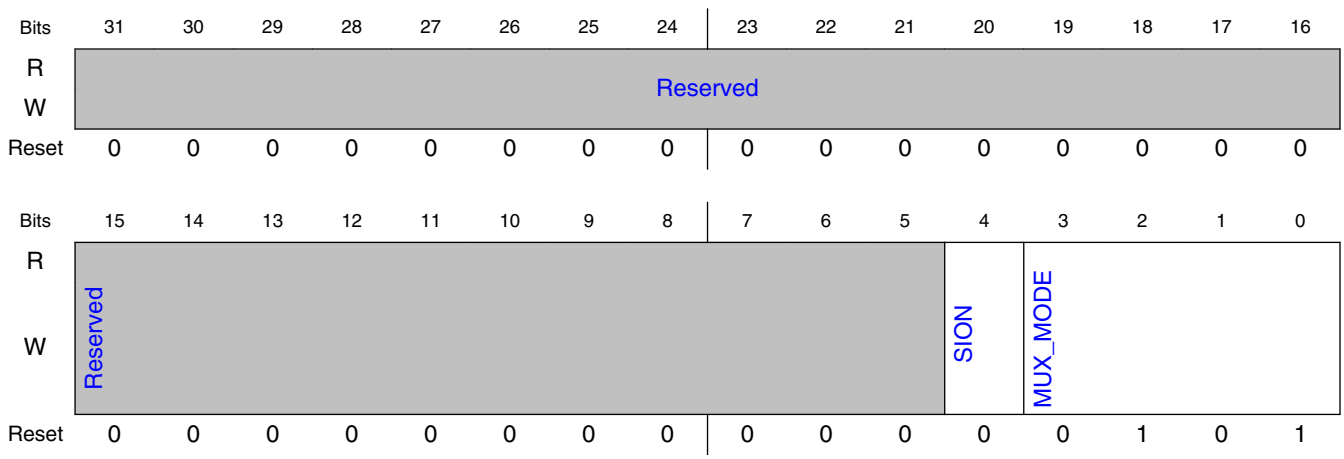
12.4.6.51.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_07	D4h

12.4.6.51.2 Function

SW_MUX_CTL Register

12.4.6.51.3 Diagram



12.4.6.51.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_07. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA23 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_CAPTURE2 of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_RX_DATA of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA18 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_DQS of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO17 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT27 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_RX_DATA03 of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI3_SIN of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: PIT1_TRIGGER2 of instance: PIT1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO17 of instance: GPIO8 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM3_B of instance: FLEXPWM3

12.4.6.52 SW_MUX_CTL_PAD_GPIO_EMC_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_08)

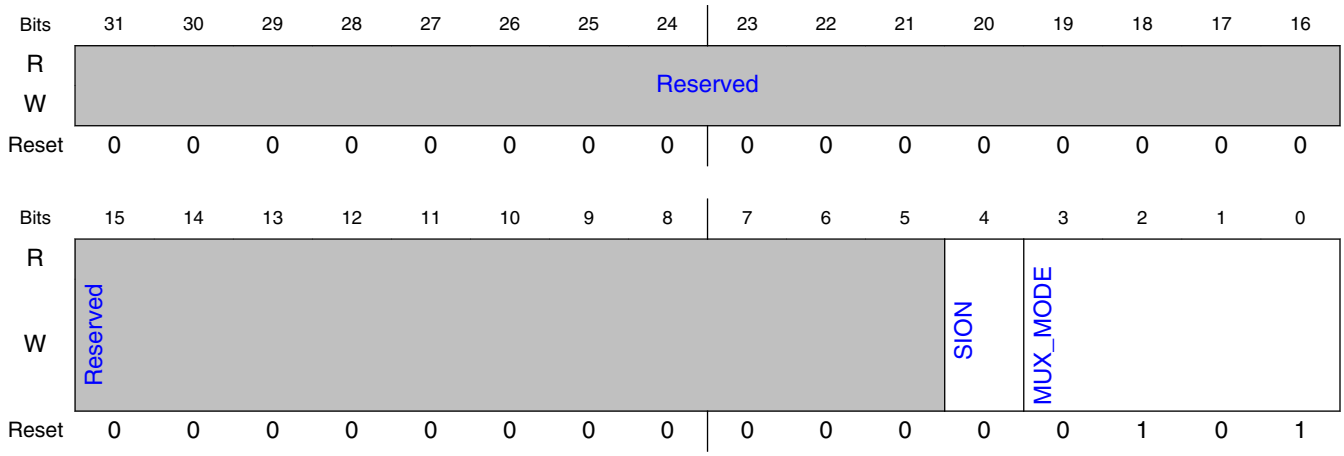
12.4.6.52.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_08	D8h

12.4.6.52.2 Function

SW_MUX_CTL Register

12.4.6.52.3 Diagram



12.4.6.52.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_08. 0000b - Select mux mode: ALT0 mux port: SEMC_DM02 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_COMPARE1 of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_TX_DATA of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA17 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_SS0_B of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO18 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT28 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_RX_DATA02 of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI3_PCS1 of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: PIT1_TRIGGER3 of instance: PIT1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO18 of instance: GPIO8

12.4.6.53 SW_MUX_CTL_PAD_GPIO_EMC_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_09)

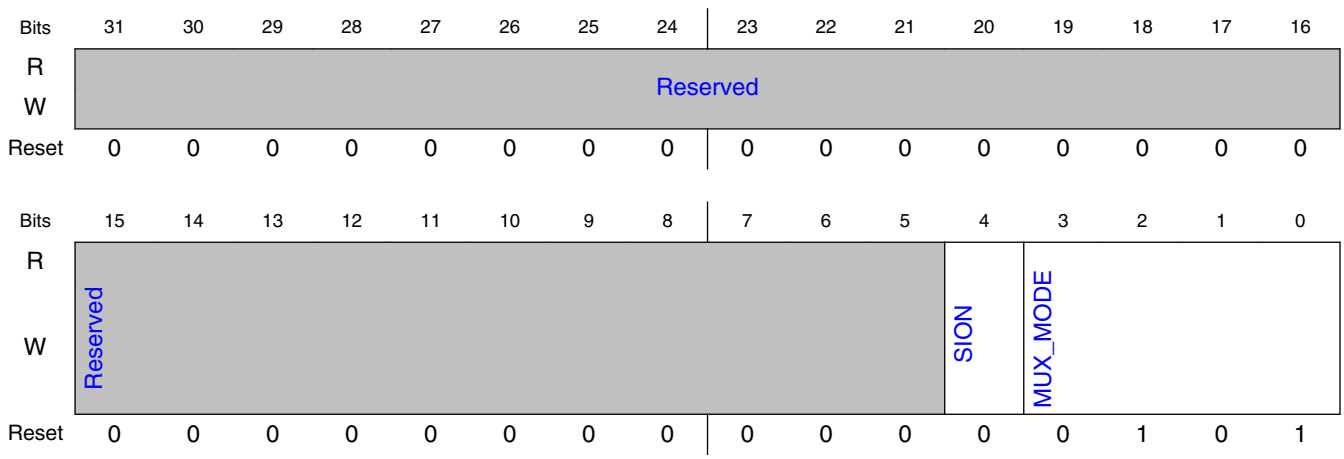
12.4.6.53.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_09	DCh

12.4.6.53.2 Function

SW_MUX_CTL Register

12.4.6.53.3 Diagram



12.4.6.53.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_09. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA24 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_COMPARE2 of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_TX_BCLK of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_DATA16 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_B_SCLK of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO19 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT29 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_CRS of instance: ENET_1G

Field	Function
	1000b - Select mux mode: ALT8 mux port: LPSPI3_PCS2 of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: TMR1_TIMER0 of instance: TMR1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO19 of instance: GPIO8

12.4.6.54 SW_MUX_CTL_PAD_GPIO_EMC_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_10)

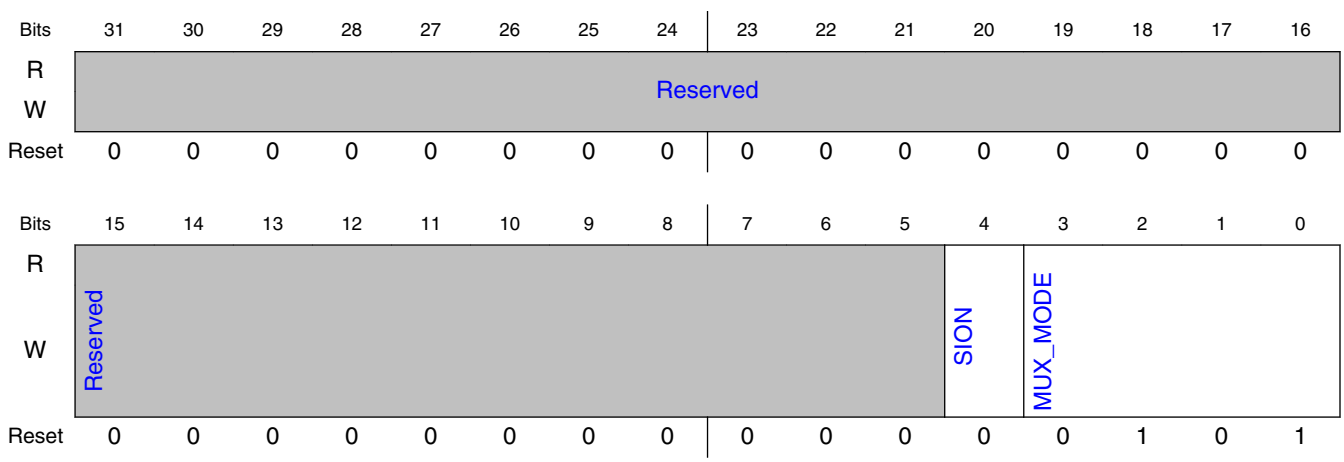
12.4.6.54.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_10	E0h

12.4.6.54.2 Function

SW_MUX_CTL Register

12.4.6.54.3 Diagram



12.4.6.54.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_10. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA25 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: GPT3_COMPARE3 of instance: GPT3 0010b - Select mux mode: ALT2 mux port: SAI2_TX_SYNC of instance: SAI2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_CSI_FIELD of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_SCLK of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO20 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT30 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: ENET_1G_COL of instance: ENET_1G 1000b - Select mux mode: ALT8 mux port: LPSPI3_PCS3 of instance: LPSPI3 1001b - Select mux mode: ALT9 mux port: TMR1_TIMER1 of instance: TMR1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO20 of instance: GPIO8

12.4.6.55 SW_MUX_CTL_PAD_GPIO_EMC_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_11)

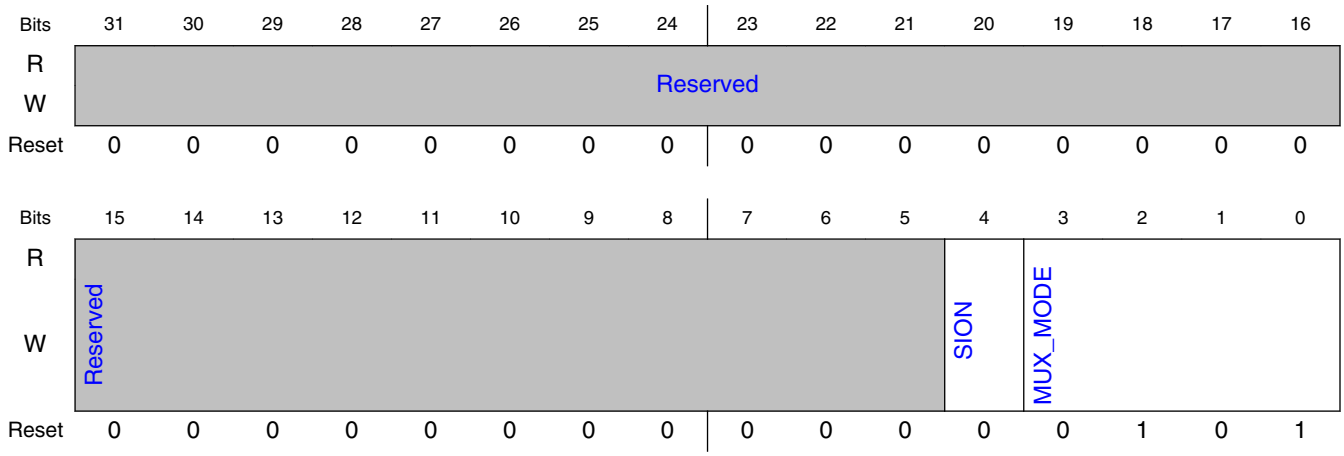
12.4.6.55.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_11	E4h

12.4.6.55.2 Function

SW_MUX_CTL Register

12.4.6.55.3 Diagram



12.4.6.55.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_11. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA26 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: SPDIF_IN of instance: SPDIF 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA00 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_RX_SYNC of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_SS0_B of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO21 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT31 of instance: XBAR1 1000b - Select mux mode: ALT8 mux port: EMVSIM1_IO of instance: EMVSIM1 1001b - Select mux mode: ALT9 mux port: TMR1_TIMER2 of instance: TMR1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO21 of instance: GPIO8

12.4.6.56 SW_MUX_CTL_PAD_GPIO_EMC_B2_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_12)

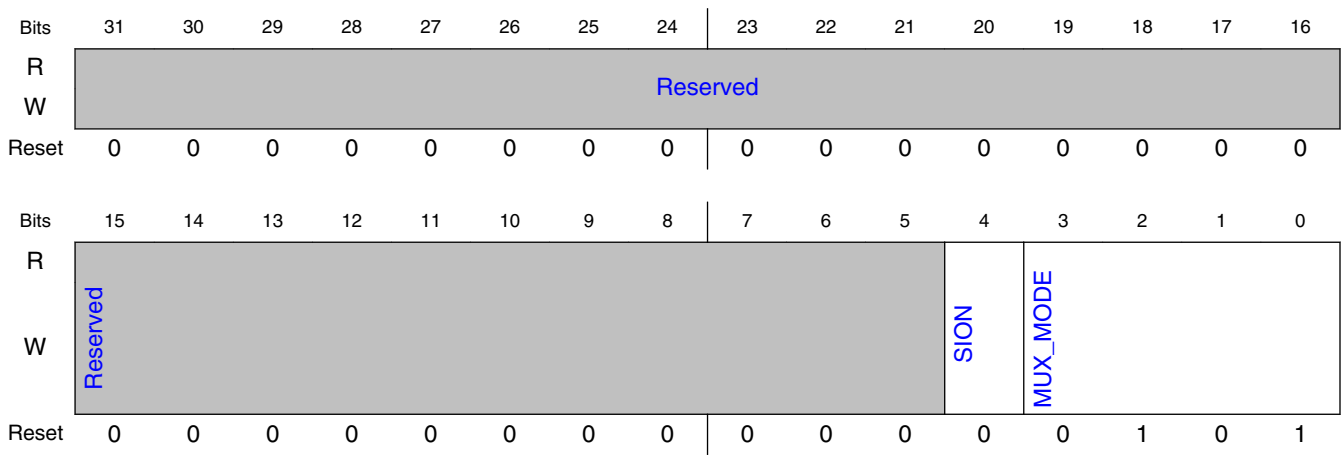
12.4.6.56.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_12	E8h

12.4.6.56.2 Function

SW_MUX_CTL Register

12.4.6.56.3 Diagram



12.4.6.56.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_12
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_12. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA27 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: SPDIF_OUT of instance: SPDIF 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA01 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_RX_BCLK of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DQS of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO22 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT32 of instance: XBAR1 1000b - Select mux mode: ALT8 mux port: EMVSIM1_CLK of instance: EMVSIM1

Field	Function
	1001b - Select mux mode: ALT9 mux port: TMR1_TIMER3 of instance: TMR1 1010b - Select mux mode: ALT10 mux port: GPIO8_IO22 of instance: GPIO8

12.4.6.57 SW_MUX_CTL_PAD_GPIO_EMC_B2_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_13)

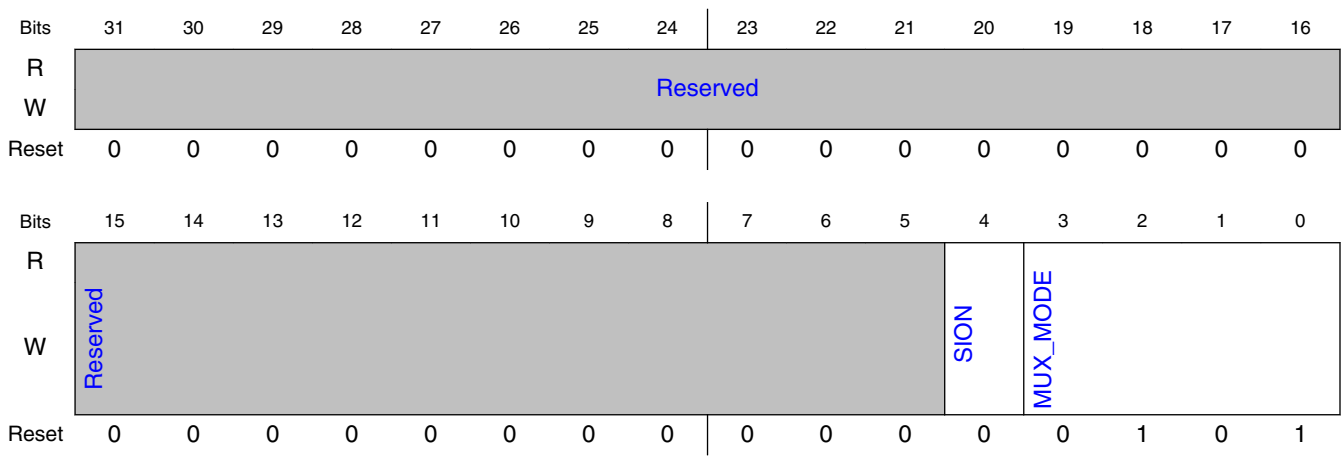
12.4.6.57.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_13	ECh

12.4.6.57.2 Function

SW_MUX_CTL Register

12.4.6.57.3 Diagram



12.4.6.57.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_13
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_13. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA28 of instance: SEMC 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_EN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_RX_DATA of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA00 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO23 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT33 of instance: XBAR1 1000b - Select mux mode: ALT8 mux port: EMVSIM1_RST of instance: EMVSIM1 1001b - Select mux mode: ALT9 mux port: TMR2_TIMER0 of instance: TMR2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO23 of instance: GPIO8

12.4.6.58 SW_MUX_CTL_PAD_GPIO_EMC_B2_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_14)

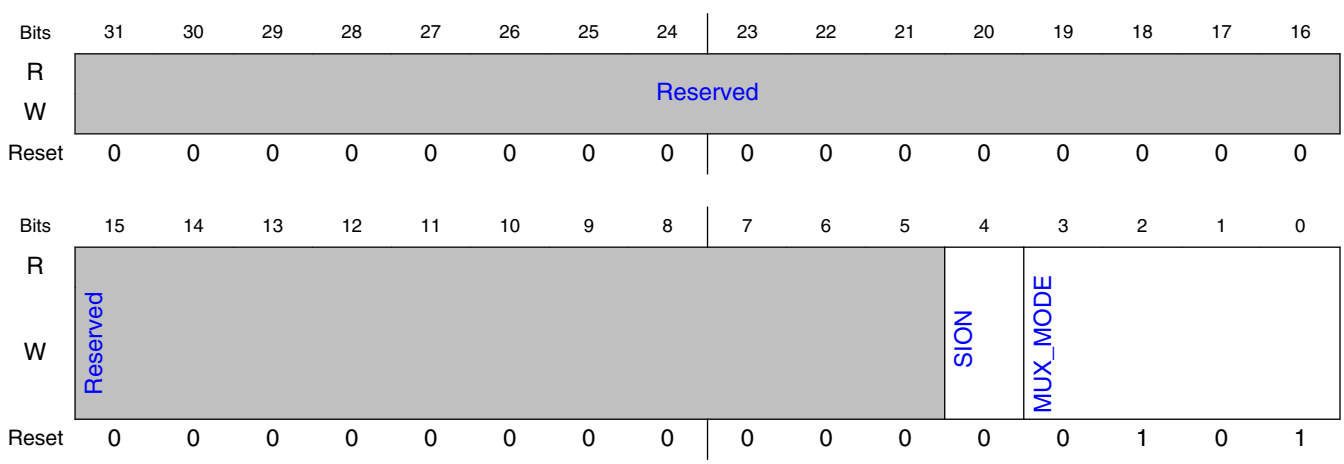
12.4.6.58.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_14	F0h

12.4.6.58.2 Function

SW_MUX_CTL Register

12.4.6.58.3 Diagram



12.4.6.58.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_14
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_14. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA29 of instance: SEMC 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_CLK_IO of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_TX_DATA of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA01 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO24 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT34 of instance: XBAR1 0111b - Select mux mode: ALT7 mux port: SFA_ipp_do_atx_clk_under_test of instance: sfa 1000b - Select mux mode: ALT8 mux port: EMVSIM1_SVEN of instance: EMVSIM1 1001b - Select mux mode: ALT9 mux port: TMR2_TIMER1 of instance: TMR2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO24 of instance: GPIO8

12.4.6.59 SW_MUX_CTL_PAD_GPIO_EMC_B2_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_15)

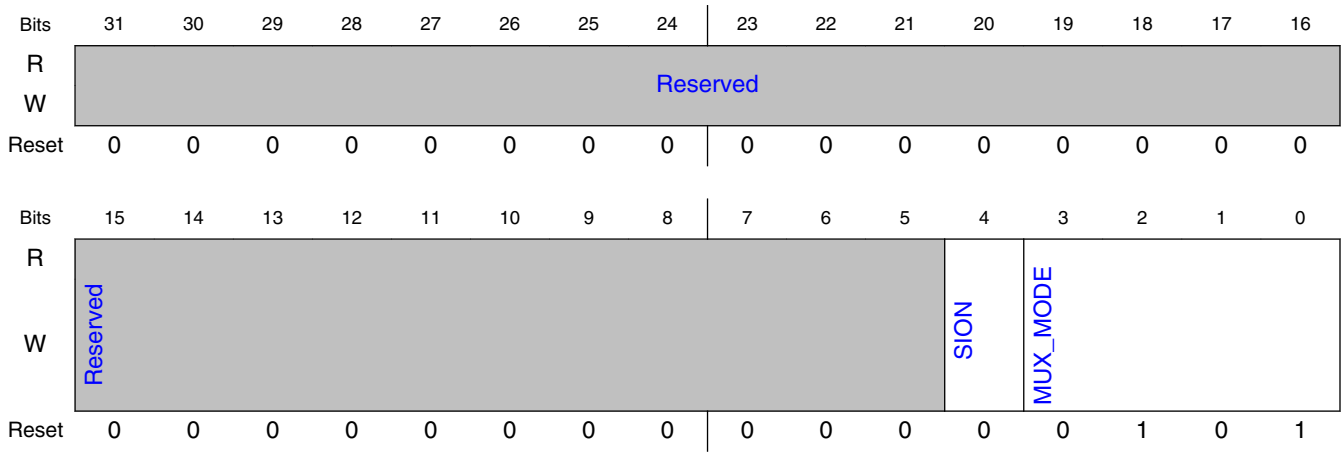
12.4.6.59.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_15	F4h

12.4.6.59.2 Function

SW_MUX_CTL Register

12.4.6.59.3 Diagram



12.4.6.59.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_15
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_15. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA30 of instance: SEMC 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA00 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_TX_BCLK of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA02 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO25 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: XBAR1_INOUT35 of instance: XBAR1 1000b - Select mux mode: ALT8 mux port: EMVSIM1_PD of instance: EMVSIM1 1001b - Select mux mode: ALT9 mux port: TMR2_TIMER2 of instance: TMR2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO25 of instance: GPIO8

12.4.6.60 SW_MUX_CTL_PAD_GPIO_EMC_B2_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_16)

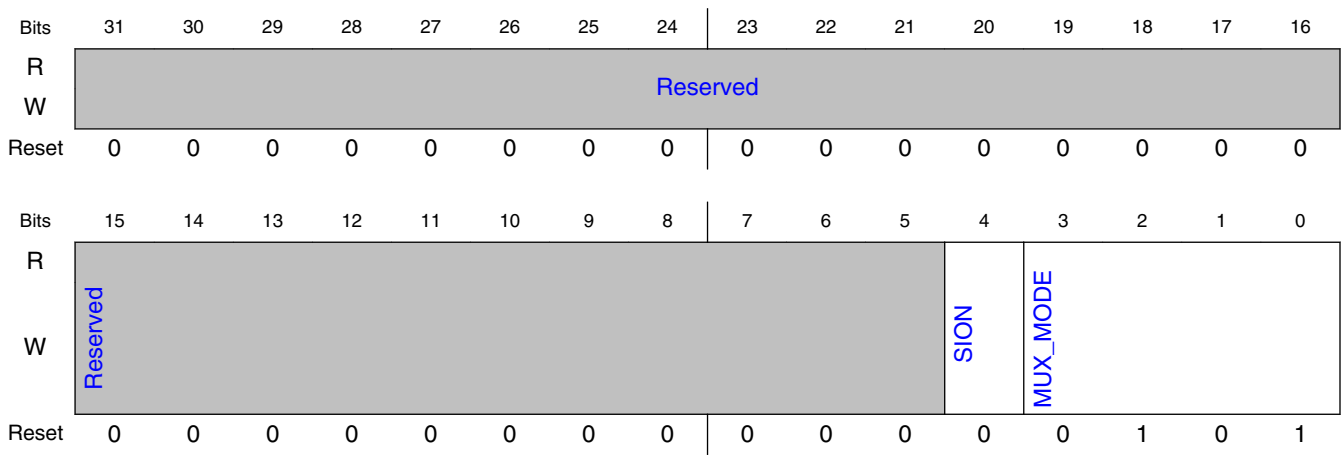
12.4.6.60.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_16	F8h

12.4.6.60.2 Function

SW_MUX_CTL Register

12.4.6.60.3 Diagram



12.4.6.60.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_16
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_16. 0000b - Select mux mode: ALT0 mux port: SEMC_DATA31 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT14 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA01 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_TX_SYNC of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA03 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO26 of instance: GPIO_MUX2 1000b - Select mux mode: ALT8 mux port: EMVSIM1_POWER_FAIL of instance: EMVSIM1 1001b - Select mux mode: ALT9 mux port: TMR2_TIMER3 of instance: TMR2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO26 of instance: GPIO8

12.4.6.61 SW_MUX_CTL_PAD_GPIO_EMC_B2_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_17)

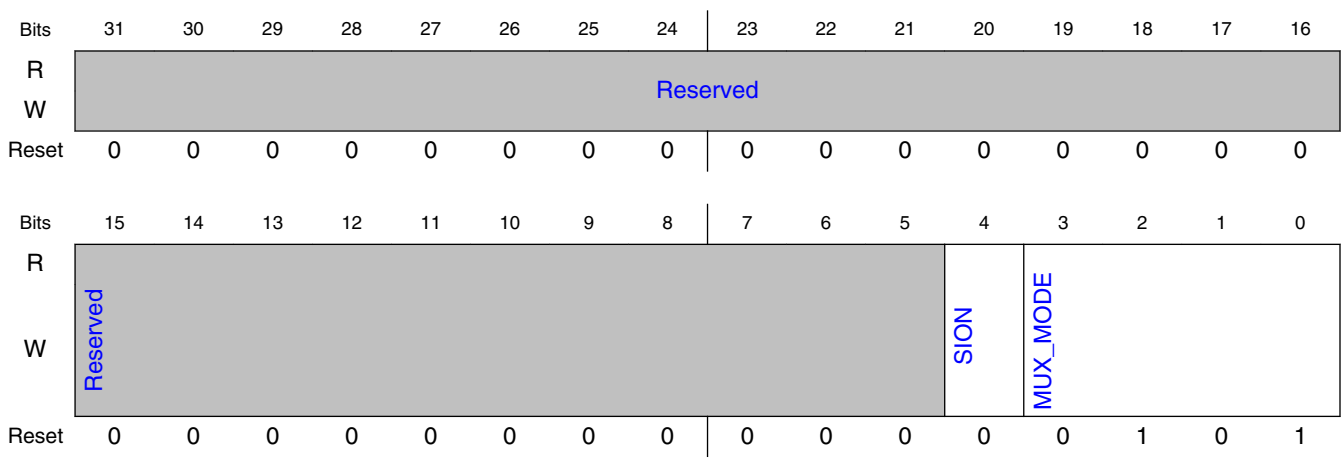
12.4.6.61.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_17	FCh

12.4.6.61.2 Function

SW_MUX_CTL Register

12.4.6.61.3 Diagram



12.4.6.61.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_17
3-0	MUX Mode Select Field.

Field	Function
MUX_MODE	Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_17. 0000b - Select mux mode: ALT0 mux port: SEMC_DM03 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT15 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_EN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: SAI3_MCLK of instance: SAI3 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA04 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO27 of instance: GPIO_MUX2 1000b - Select mux mode: ALT8 mux port: WDOG1_ANY of instance: WDOG1 1001b - Select mux mode: ALT9 mux port: TMR3_TIMER0 of instance: TMR3 1010b - Select mux mode: ALT10 mux port: GPIO8_IO27 of instance: GPIO8

12.4.6.62 SW_MUX_CTL_PAD_GPIO_EMC_B2_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_18)

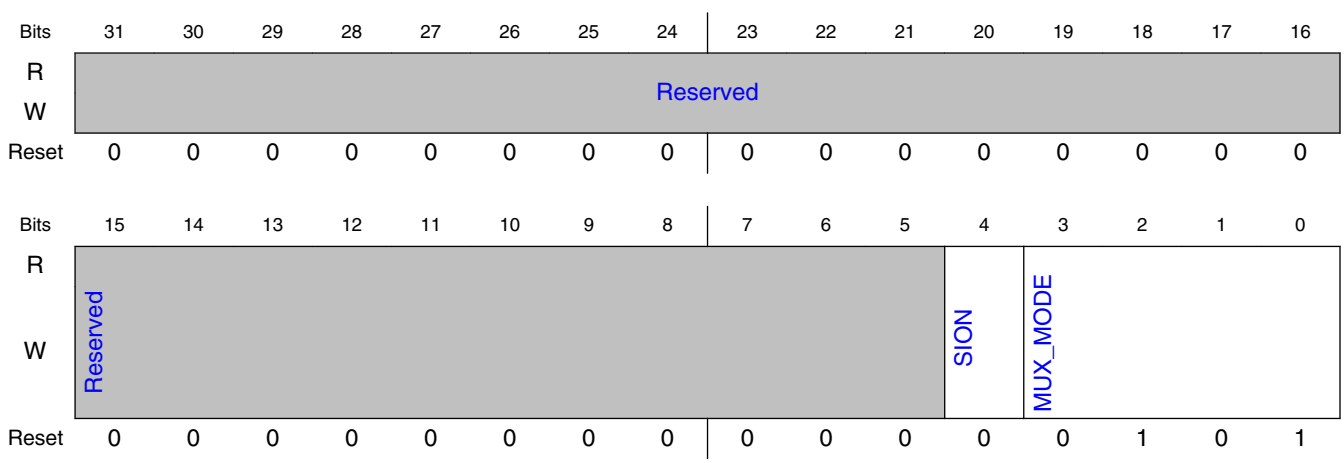
12.4.6.62.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_18	100h

12.4.6.62.2 Function

SW_MUX_CTL Register

12.4.6.62.3 Diagram



12.4.6.62.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_18
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_EMC_B2_18. 0000b - Select mux mode: ALT0 mux port: SEMC_DQS4 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: XBAR1_INOUT16 of instance: XBAR1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_ER of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: EWM_OUT_B of instance: EWM 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA05 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO28 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: FLEXSPI1_A_DQS of instance: FLEXSPI1 1000b - Select mux mode: ALT8 mux port: WDOG1_B of instance: WDOG1 1001b - Select mux mode: ALT9 mux port: TMR3_TIMER1 of instance: TMR3 1010b - Select mux mode: ALT10 mux port: GPIO8_IO28 of instance: GPIO8

12.4.6.63 SW_MUX_CTL_PAD_GPIO_EMC_B2_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_19)

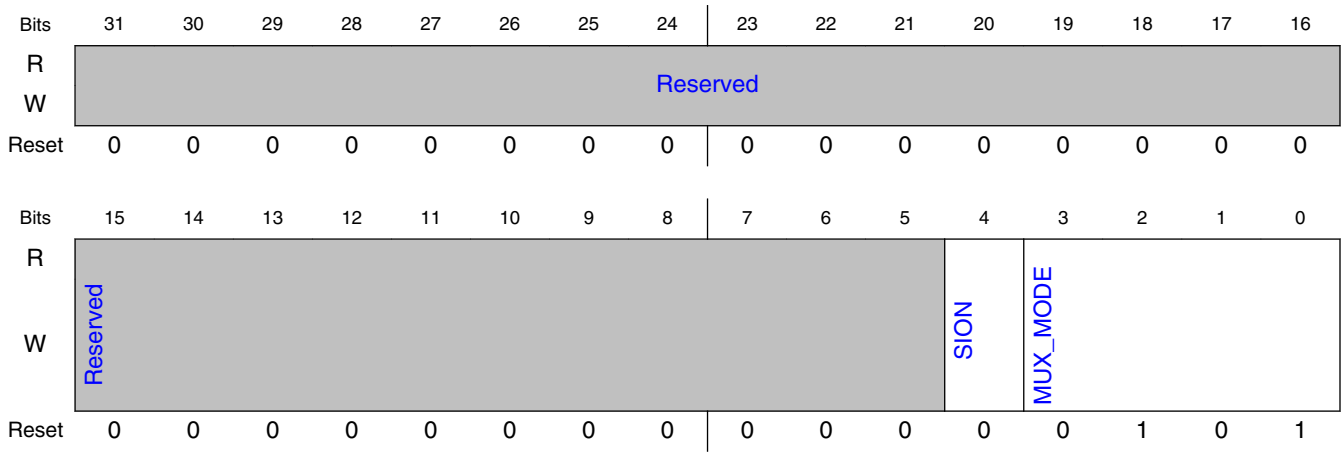
12.4.6.63.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_19	104h

12.4.6.63.2 Function

SW_MUX_CTL Register

12.4.6.63.3 Diagram



12.4.6.63.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_19
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_19. 0000b - Select mux mode: ALT0 mux port: SEMC_CLKX00 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: ENET_MDC of instance: ENET 0010b - Select mux mode: ALT2 mux port: ENET_1G_MDC of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: ENET_1G_REF_CLK of instance: ENET_1G 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA06 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO29 of instance: GPIO_MUX2 1000b - Select mux mode: ALT8 mux port: ENET_QOS_MDC of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: TMR3_TIMER2 of instance: TMR3 1010b - Select mux mode: ALT10 mux port: GPIO8_IO29 of instance: GPIO8

12.4.6.64 SW_MUX_CTL_PAD_GPIO_EMC_B2_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_EMC_B2_20)

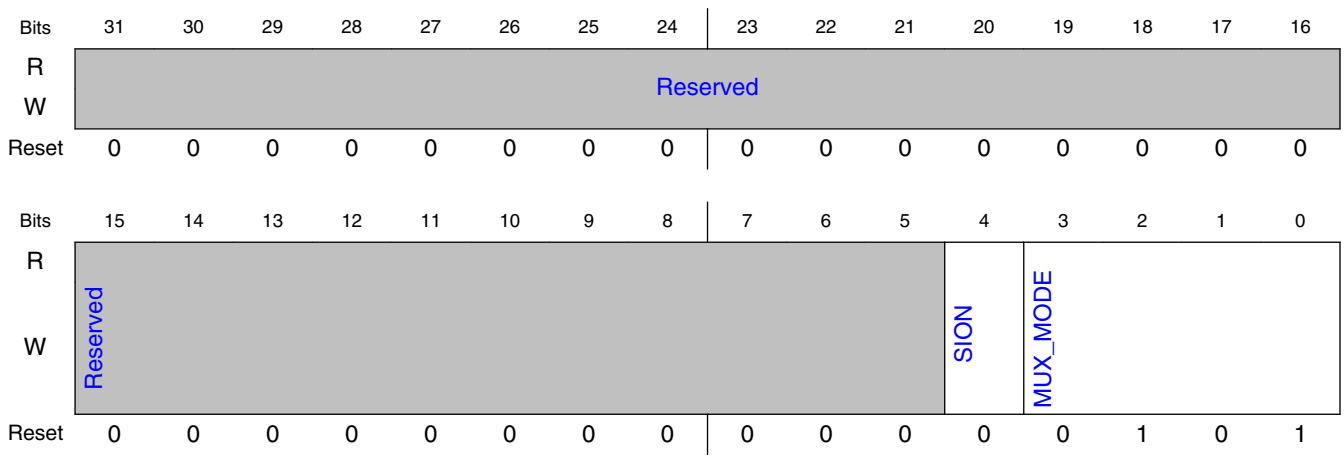
12.4.6.64.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_EMC_B2_20	108h

12.4.6.64.2 Function

SW_MUX_CTL Register

12.4.6.64.3 Diagram



12.4.6.64.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_EMC_B2_20
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_EMC_B2_20. 0000b - Select mux mode: ALT0 mux port: SEMC_CLKX01 of instance: SEMC 0001b - Select mux mode: ALT1 mux port: ENET_MDIO of instance: ENET 0010b - Select mux mode: ALT2 mux port: ENET_1G_MDIO of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: ENET_QOS_REF_CLK of instance: ENET_QOS 0100b - Select mux mode: ALT4 mux port: FLEXSPI2_A_DATA07 of instance: FLEXSPI2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO30 of instance: GPIO_MUX2 1000b - Select mux mode: ALT8 mux port: ENET_QOS_MDIO of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: TMR3_TIMER3 of instance: TMR3 1010b - Select mux mode: ALT10 mux port: GPIO8_IO30 of instance: GPIO8

12.4.6.65 SW_MUX_CTL_PAD_GPIO_AD_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_00)

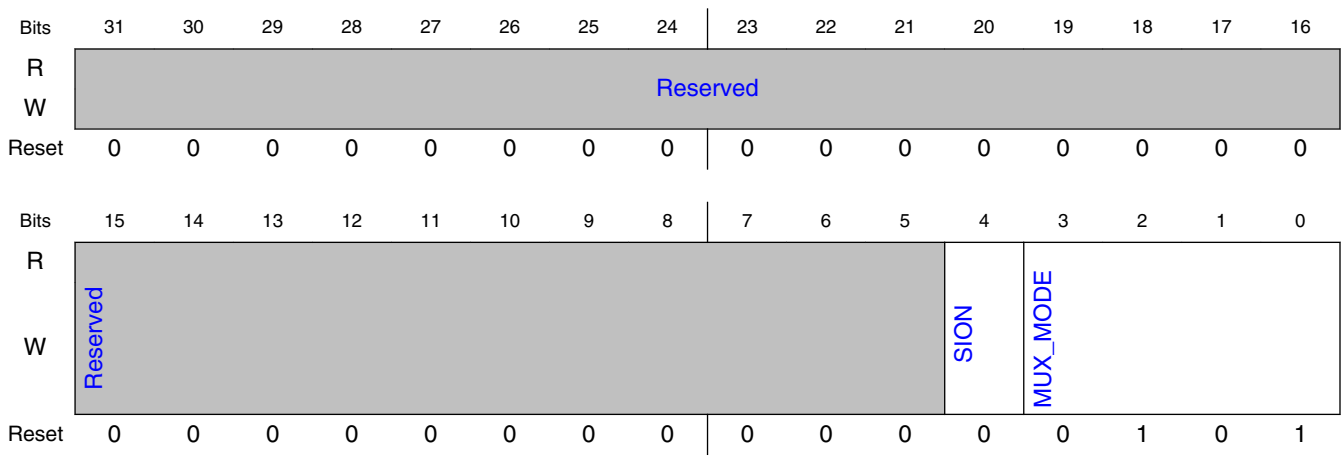
12.4.6.65.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_00	10Ch

12.4.6.65.2 Function

SW_MUX_CTL Register

12.4.6.65.3 Diagram



12.4.6.65.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_00
3-0	MUX Mode Select Field.

Field	Function
MUX_MODE	Select 1 of 11 iomux modes to be used for pad: GPIO_AD_00. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_IO of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: FLEXCAN2_TX of instance: FLEXCAN2 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT1_IN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_CAPTURE1 of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM0_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX2_IO31 of instance: GPIO_MUX2 0110b - Select mux mode: ALT6 mux port: LPUART7_TXD of instance: LPUART7 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D00 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: FLEXSPI2_B_SS1_B of instance: FLEXSPI2 1010b - Select mux mode: ALT10 mux port: GPIO8_IO31 of instance: GPIO8

12.4.6.66 SW_MUX_CTL_PAD_GPIO_AD_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_01)

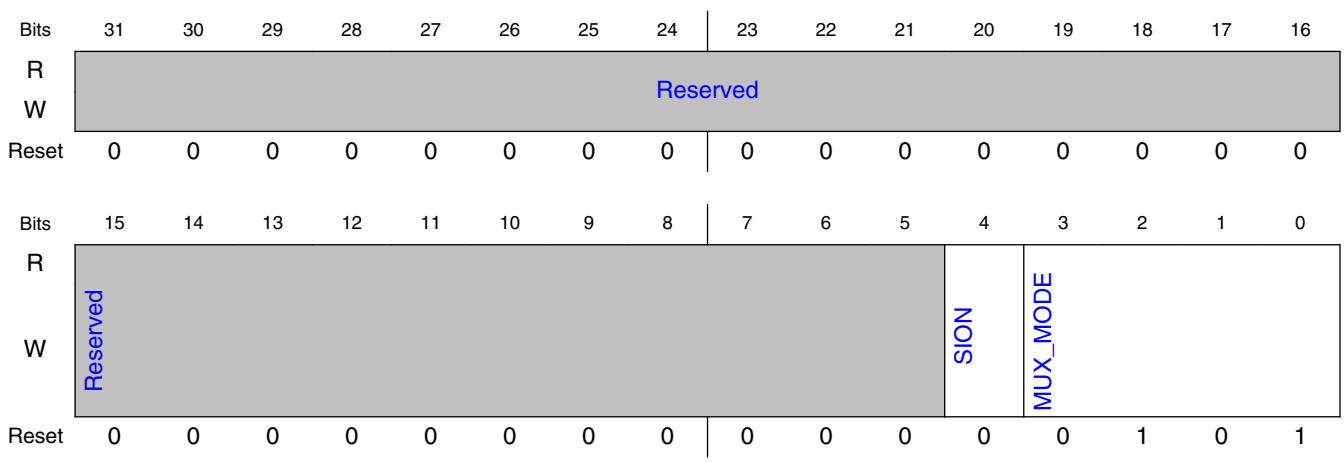
12.4.6.66.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_01	110h

12.4.6.66.2 Function

SW_MUX_CTL Register

12.4.6.66.3 Diagram



12.4.6.66.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_01. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_CLK of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: FLEXCAN2_RX of instance: FLEXCAN2 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT1_OUT of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_CAPTURE2 of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM0_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO00 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: LPUART7_RXD of instance: LPUART7 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D01 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: FLEXSPI2_A_SS1_B of instance: FLEXSPI2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO00 of instance: GPIO9

12.4.6.67 SW_MUX_CTL_PAD_GPIO_AD_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_02)

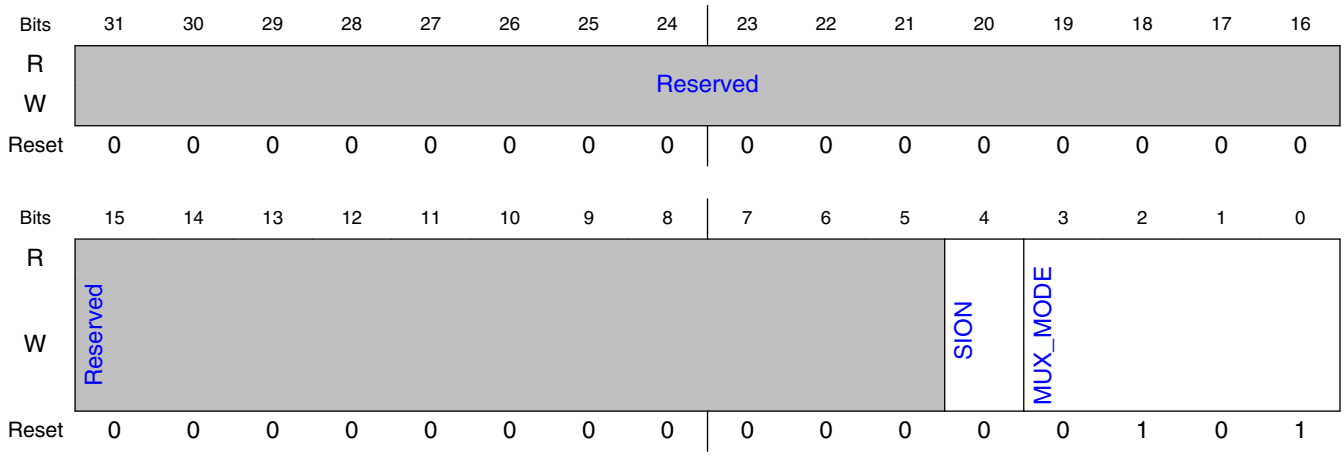
12.4.6.67.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_02	114h

12.4.6.67.2 Function

SW_MUX_CTL Register

12.4.6.67.3 Diagram



12.4.6.67.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_02. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_RST of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: LPUART7_CTS_B of instance: LPUART7 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT2_IN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_COMPARE1 of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM1_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO01 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: LPUART8_TXD of instance: LPUART8 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D02 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: VIDEO_MUX_EXT_DCIC1 of instance: VIDEO_MUX 1010b - Select mux mode: ALT10 mux port: GPIO9_IO01 of instance: GPIO9

12.4.6.68 SW_MUX_CTL_PAD_GPIO_AD_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_03)

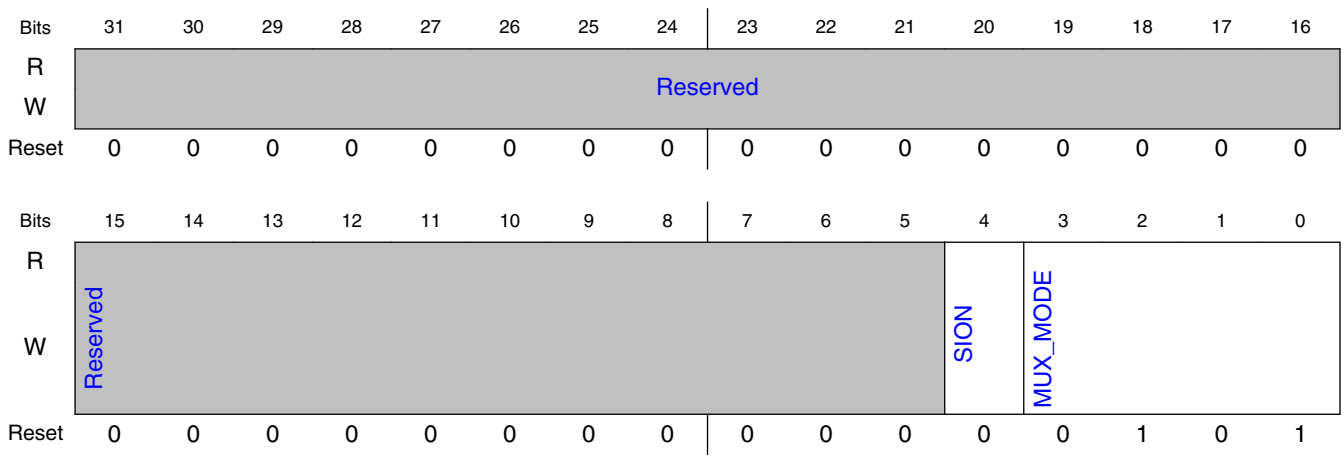
12.4.6.68.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_03	118h

12.4.6.68.2 Function

SW_MUX_CTL Register

12.4.6.68.3 Diagram



12.4.6.68.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_03. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_SVEN of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: LPUART7_RTS_B of instance: LPUART7 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT2_OUT of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_COMPARE2 of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM1_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO02 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: LPUART8_RXD of instance: LPUART8 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D03 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: VIDEO_MUX_EXT_DCIC2 of instance: VIDEO_MUX 1010b - Select mux mode: ALT10 mux port: GPIO9_IO02 of instance: GPIO9

12.4.6.69 SW_MUX_CTL_PAD_GPIO_AD_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_04)

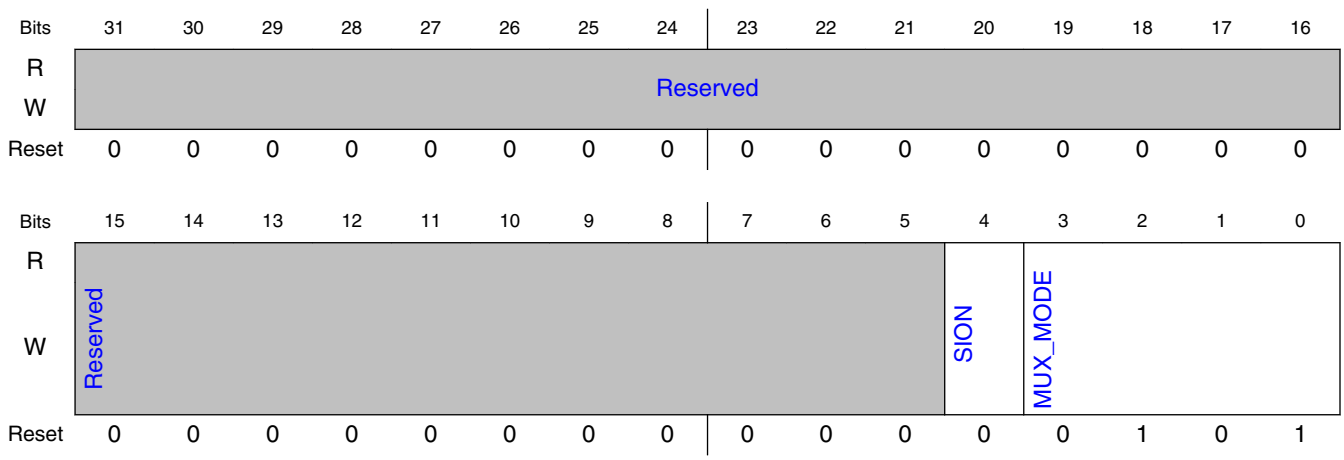
12.4.6.69.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_04	11Ch

12.4.6.69.2 Function

SW_MUX_CTL Register

12.4.6.69.3 Diagram



12.4.6.69.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_04. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_PD of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: LPUART8_CTS_B of instance: LPUART8 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT3_IN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_COMPARE3 of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM2_A of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO03 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: WDOG1_B of instance: WDOG1 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D04 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: TMR4_TIMER0 of instance: TMR4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO03 of instance: GPIO9

12.4.6.70 SW_MUX_CTL_PAD_GPIO_AD_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_05)

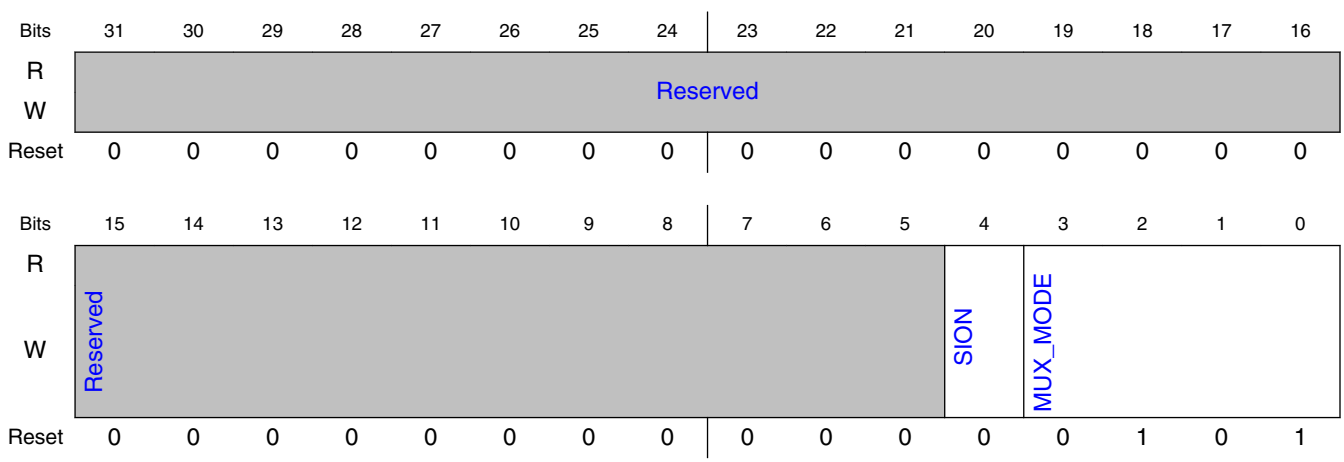
12.4.6.70.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_05	120h

12.4.6.70.2 Function

SW_MUX_CTL Register

12.4.6.70.3 Diagram



12.4.6.70.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_05. 0000b - Select mux mode: ALT0 mux port: EMVSIM1_POWER_FAIL of instance: EMVSIM1 0001b - Select mux mode: ALT1 mux port: LPUART8_RTS_B of instance: LPUART8 0010b - Select mux mode: ALT2 mux port: ENET_1G_1588_EVENT3_OUT of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: GPT2_CLK of instance: GPT2 0100b - Select mux mode: ALT4 mux port: FLEXPWM1_PWM2_B of instance: FLEXPWM1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO04 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: WDOG2_B of instance: WDOG2 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D05 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: TMR4_TIMER1 of instance: TMR4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO04 of instance: GPIO9

12.4.6.71 SW_MUX_CTL_PAD_GPIO_AD_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_06)

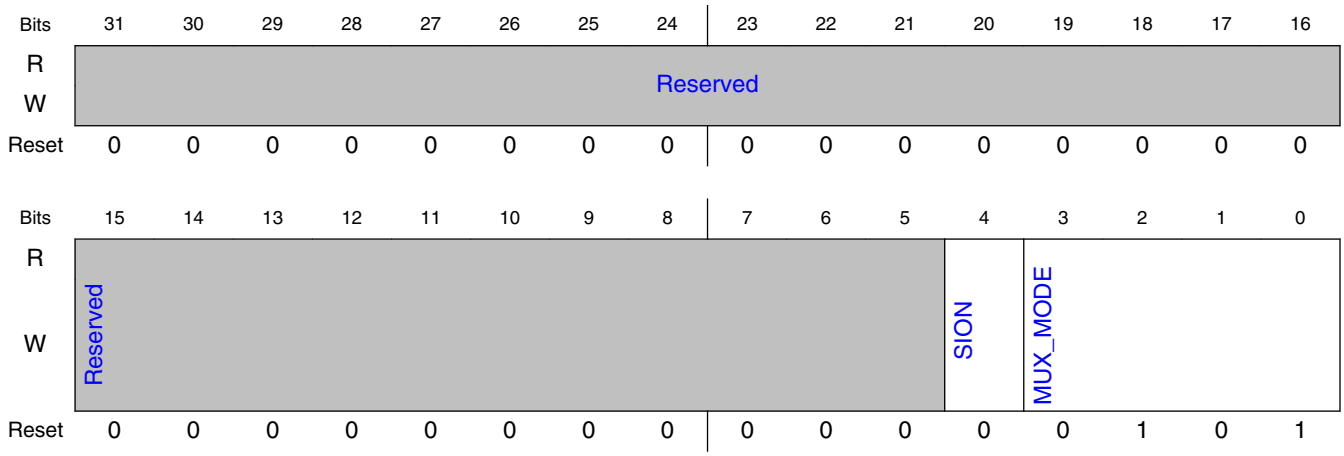
12.4.6.71.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_06	124h

12.4.6.71.2 Function

SW_MUX_CTL Register

12.4.6.71.3 Diagram



12.4.6.71.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_06. 0000b - Select mux mode: ALT0 mux port: USB_OTG2_OC of instance: USB 0001b - Select mux mode: ALT1 mux port: FLEXCAN1_TX of instance: FLEXCAN1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_IO of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_CAPTURE1 of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA15 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO05 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT1_IN of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D06 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: TMR4_TIMER2 of instance: TMR4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO05 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM1_PWM0_X of instance: FLEXPWM1

12.4.6.72 SW_MUX_CTL_PAD_GPIO_AD_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_07)

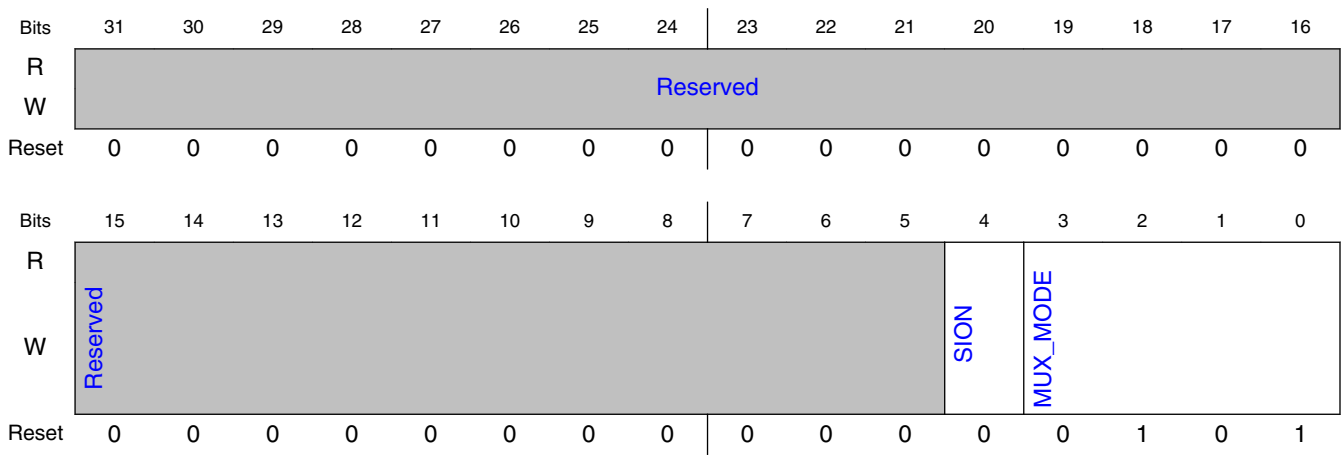
12.4.6.72.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_07	128h

12.4.6.72.2 Function

SW_MUX_CTL Register

12.4.6.72.3 Diagram



12.4.6.72.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_07. 0000b - Select mux mode: ALT0 mux port: USB_OTG2_PWR of instance: USB 0001b - Select mux mode: ALT1 mux port: FLEXCAN1_RX of instance: FLEXCAN1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_CLK of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_CAPTURE2 of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA14 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO06 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT1_OUT of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D07 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: TMR4_TIMER3 of instance: TMR4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO06 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM1_PWM1_X of instance: FLEXPWM1

12.4.6.73 SW_MUX_CTL_PAD_GPIO_AD_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_08)

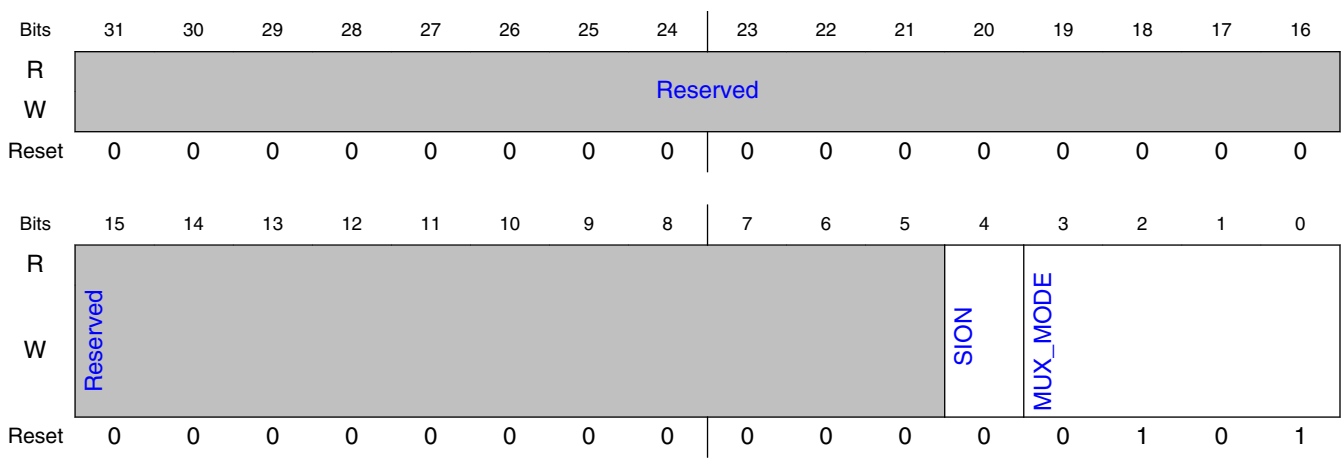
12.4.6.73.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_08	12Ch

12.4.6.73.2 Function

SW_MUX_CTL Register

12.4.6.73.3 Diagram



12.4.6.73.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_08. 0000b - Select mux mode: ALT0 mux port: USBPHY2_OTG_ID of instance: USBPHY2 0001b - Select mux mode: ALT1 mux port: LPI2C1_SCL of instance: LPI2C1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_RST of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_COMPARE1 of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA13 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO07 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT2_IN of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D08 of instance: FLEXIO2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO07 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM1_PWM2_X of instance: FLEXPWM1

12.4.6.74 SW_MUX_CTL_PAD_GPIO_AD_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_09)

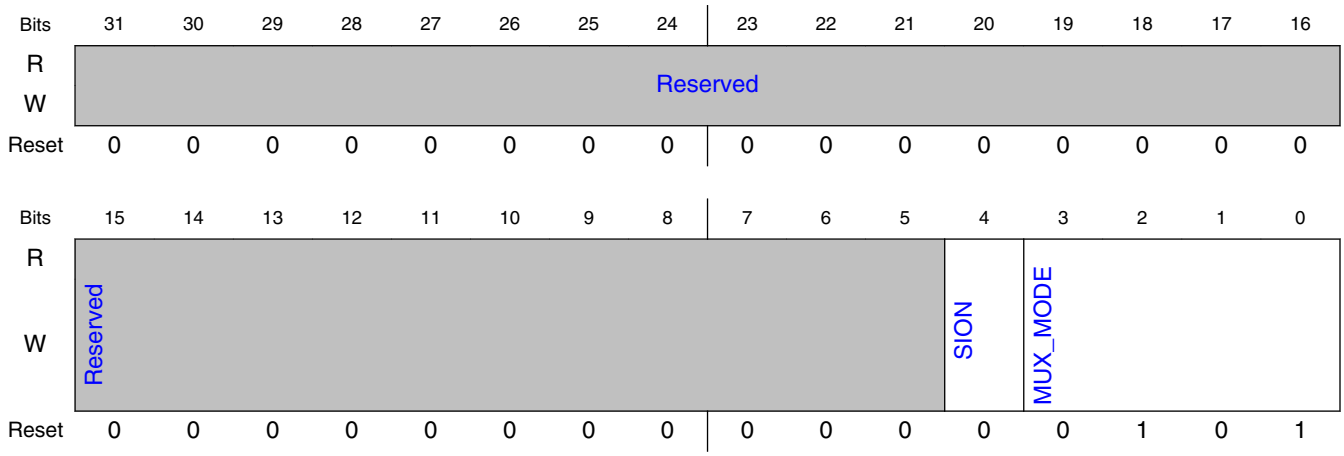
12.4.6.74.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_09	130h

12.4.6.74.2 Function

SW_MUX_CTL Register

12.4.6.74.3 Diagram



12.4.6.74.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_09. 0000b - Select mux mode: ALT0 mux port: USBPHY1_OTG_ID of instance: USBPHY1 0001b - Select mux mode: ALT1 mux port: LPI2C1_SDA of instance: LPI2C1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_SVEN of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_COMPARE2 of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA12 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO08 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT2_OUT of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D09 of instance: FLEXIO2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO08 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM1_PWM3_X of instance: FLEXPWM1

12.4.6.75 SW_MUX_CTL_PAD_GPIO_AD_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_10)

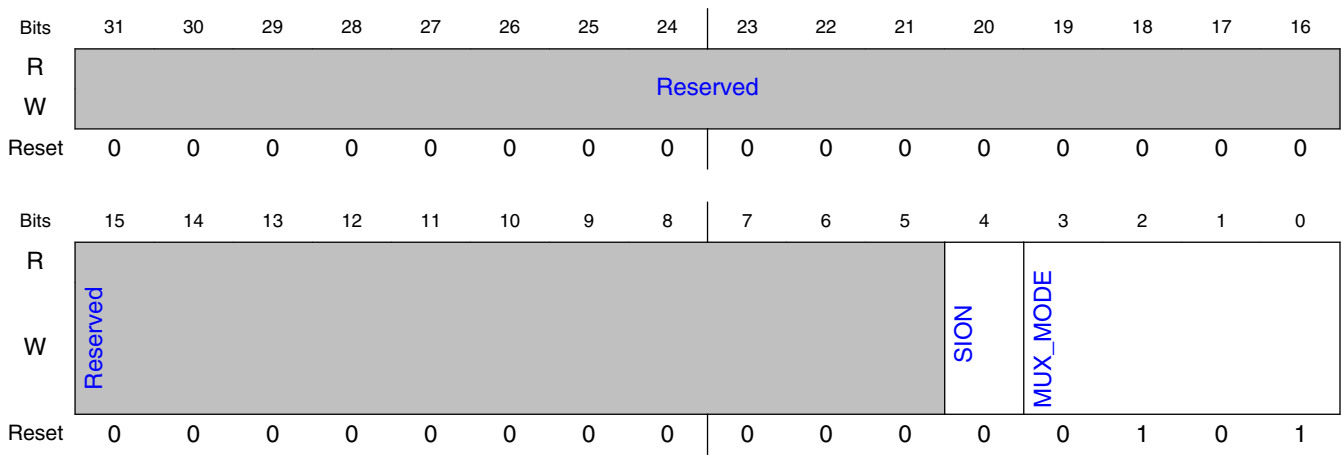
12.4.6.75.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_10	134h

12.4.6.75.2 Function

SW_MUX_CTL Register

12.4.6.75.3 Diagram



12.4.6.75.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_10. 0000b - Select mux mode: ALT0 mux port: USB_OTG1_PWR of instance: USB 0001b - Select mux mode: ALT1 mux port: LPI2C1_SCLS of instance: LPI2C1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_PD of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_COMPARE3 of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA11 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO09 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT3_IN of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D10 of instance: FLEXIO2

Field	Function
	1010b - Select mux mode: ALT10 mux port: GPIO9_IO09 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM2_PWM0_X of instance: FLEXPWM2

12.4.6.76 SW_MUX_CTL_PAD_GPIO_AD_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_11)

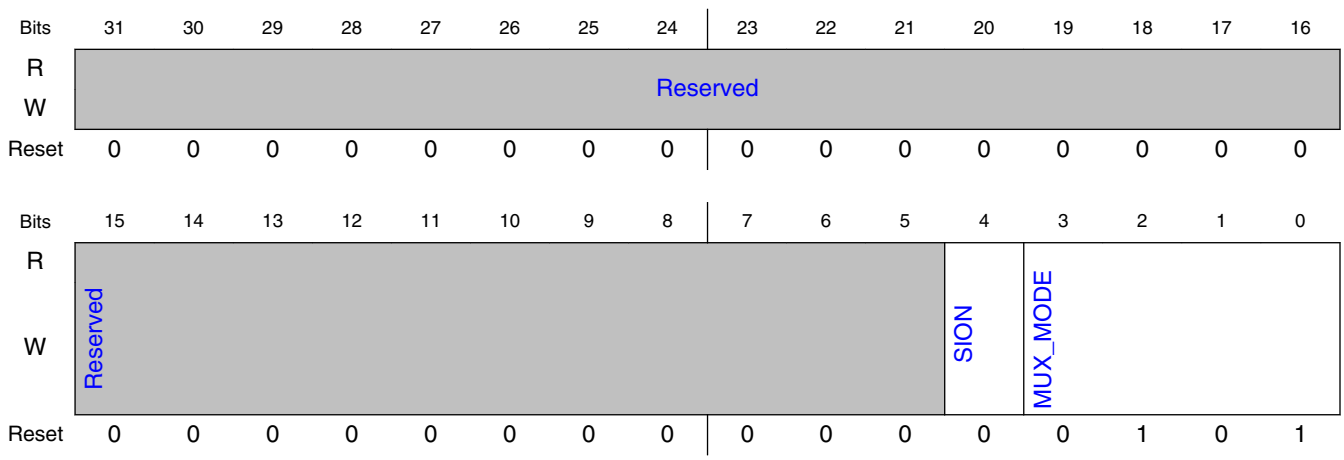
12.4.6.76.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_11	138h

12.4.6.76.2 Function

SW_MUX_CTL Register

12.4.6.76.3 Diagram



12.4.6.76.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_11. 0000b - Select mux mode: ALT0 mux port: USB_OTG1_OC of instance: USB 0001b - Select mux mode: ALT1 mux port: LPI2C1_SDAS of instance: LPI2C1 0010b - Select mux mode: ALT2 mux port: EMVSIM2_POWER_FAIL of instance: EMVSIM2 0011b - Select mux mode: ALT3 mux port: GPT3_CLK of instance: GPT3 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA10 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO10 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_1588_EVENT3_OUT of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D11 of instance: FLEXIO2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO10 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM2_PWM1_X of instance: FLEXPWM2

12.4.6.77 SW_MUX_CTL_PAD_GPIO_AD_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_12)

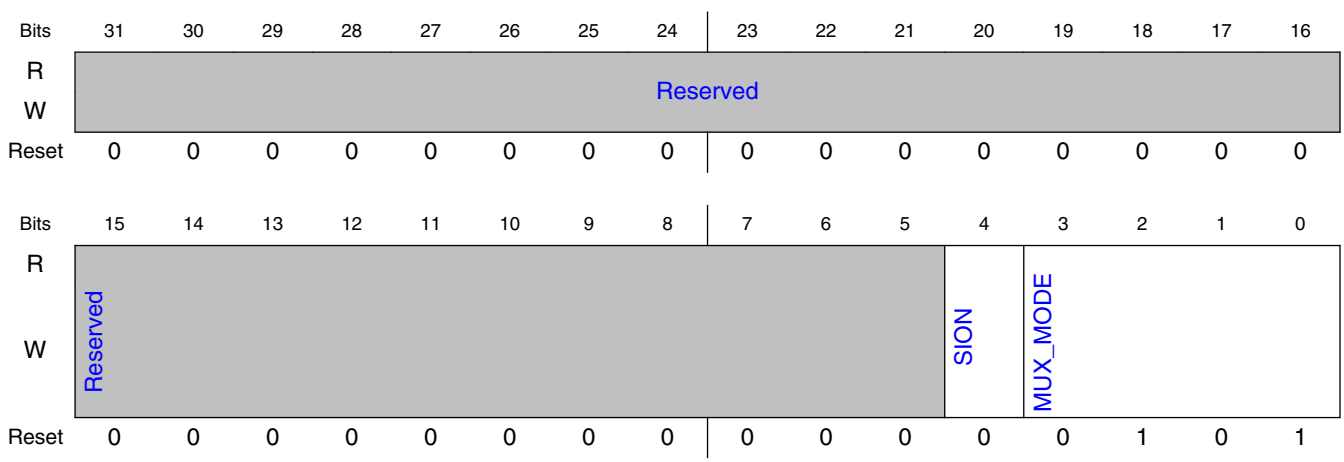
12.4.6.77.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_12	13Ch

12.4.6.77.2 Function

SW_MUX_CTL Register

12.4.6.77.3 Diagram



12.4.6.77.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_12
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_12. 0000b - Select mux mode: ALT0 mux port: SPDIF_LOCK of instance: SPDIF 0001b - Select mux mode: ALT1 mux port: LPI2C1_HREQ of instance: LPI2C1 0010b - Select mux mode: ALT2 mux port: GPT1_CAPTURE1 of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_DATA03 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_PIXCLK of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO11 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_TX_DATA03 of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D12 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: EWM_OUT_B of instance: EWM 1010b - Select mux mode: ALT10 mux port: GPIO9_IO11 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM2_PWM2_X of instance: FLEXPWM2

12.4.6.78 SW_MUX_CTL_PAD_GPIO_AD_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_13)

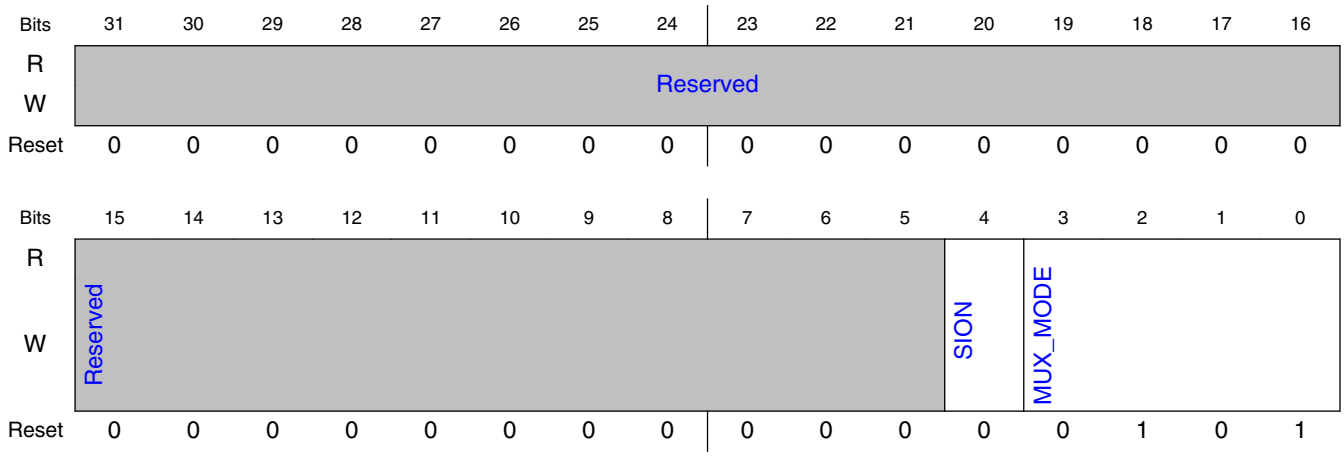
12.4.6.78.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_13	140h

12.4.6.78.2 Function

SW_MUX_CTL Register

12.4.6.78.3 Diagram



12.4.6.78.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_13
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_13. 0000b - Select mux mode: ALT0 mux port: SPDIF_SR_CLK of instance: SPDIF 0001b - Select mux mode: ALT1 mux port: PIT1_TRIGGER0 of instance: PIT1 0010b - Select mux mode: ALT2 mux port: GPT1_CAPTURE2 of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_DATA02 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_MCLK of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO12 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_TX_DATA02 of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D13 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: REF_CLK_32K of instance: XTAL OSC 1010b - Select mux mode: ALT10 mux port: GPIO9_IO12 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM2_PWM3_X of instance: FLEXPWM2

12.4.6.79 SW_MUX_CTL_PAD_GPIO_AD_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_14)

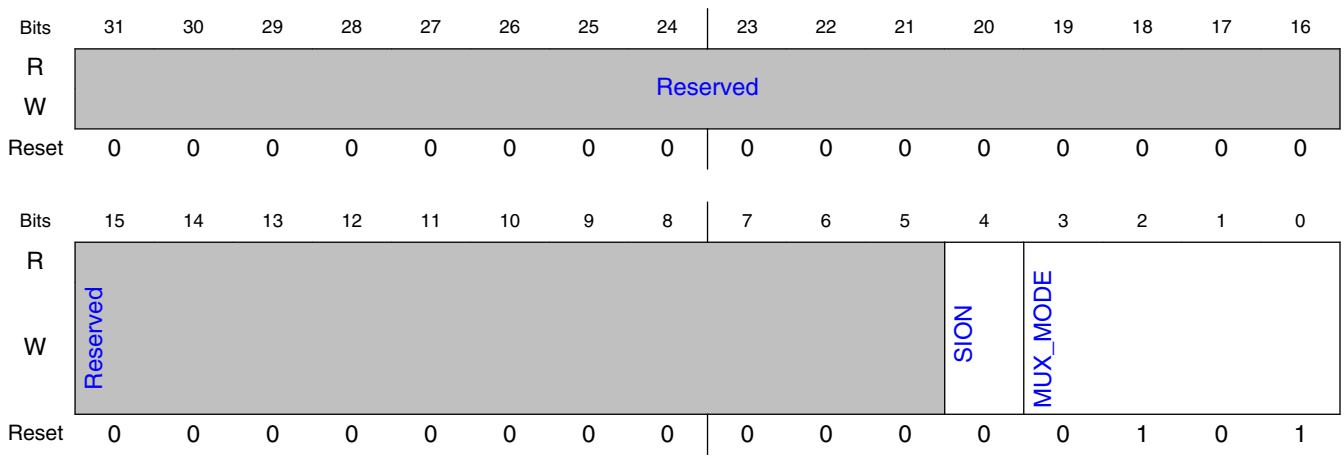
12.4.6.79.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_14	144h

12.4.6.79.2 Function

SW_MUX_CTL Register

12.4.6.79.3 Diagram



12.4.6.79.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_14
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_14. 0000b - Select mux mode: ALT0 mux port: SPDIF_EXT_CLK of instance: SPDIF 0001b - Select mux mode: ALT1 mux port: REF_CLK_24M of instance: XTAL OSC 0010b - Select mux mode: ALT2 mux port: GPT1_COMPARE1 of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_DATA01 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_VSYNC of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO13 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_RX_CLK of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D14 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: CCM_ENET_REF_CLK_25M of instance: CCM 1010b - Select mux mode: ALT10 mux port: GPIO9_IO13 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM0_X of instance: FLEXPWM3

12.4.6.80 SW_MUX_CTL_PAD_GPIO_AD_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_15)

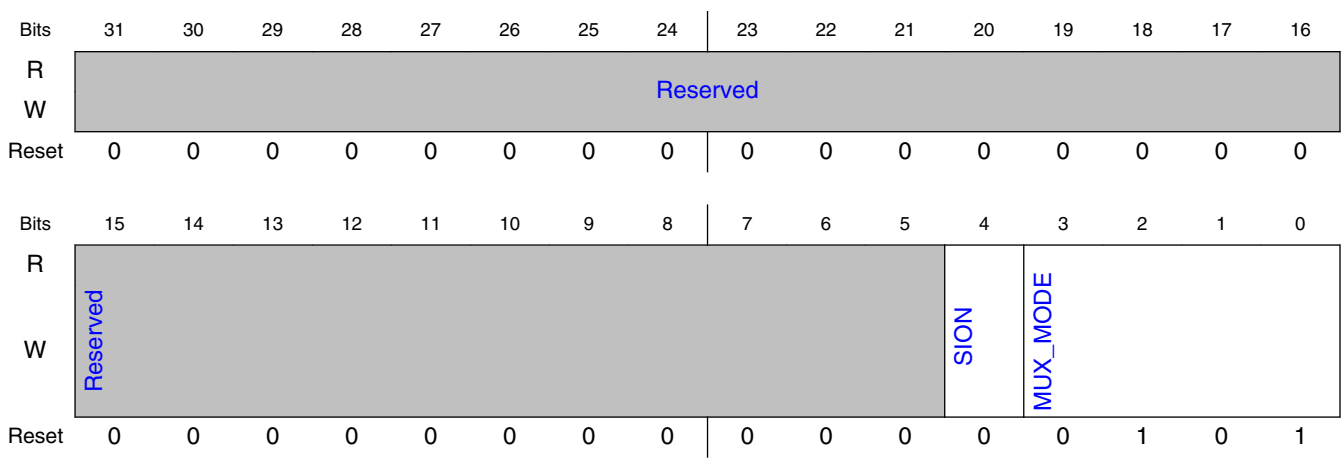
12.4.6.80.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_15	148h

12.4.6.80.2 Function

SW_MUX_CTL Register

12.4.6.80.3 Diagram



12.4.6.80.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_15
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_15. 0000b - Select mux mode: ALT0 mux port: SPDIF_IN of instance: SPDIF 0001b - Select mux mode: ALT1 mux port: LPUART10_TXD of instance: LPUART10 0010b - Select mux mode: ALT2 mux port: GPT1_COMPARE2 of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_DATA00 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_HSYNC of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO14 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_TX_ER of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D15 of instance: FLEXIO2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO14 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM1_X of instance: FLEXPWM3

12.4.6.81 SW_MUX_CTL_PAD_GPIO_AD_16 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_16)

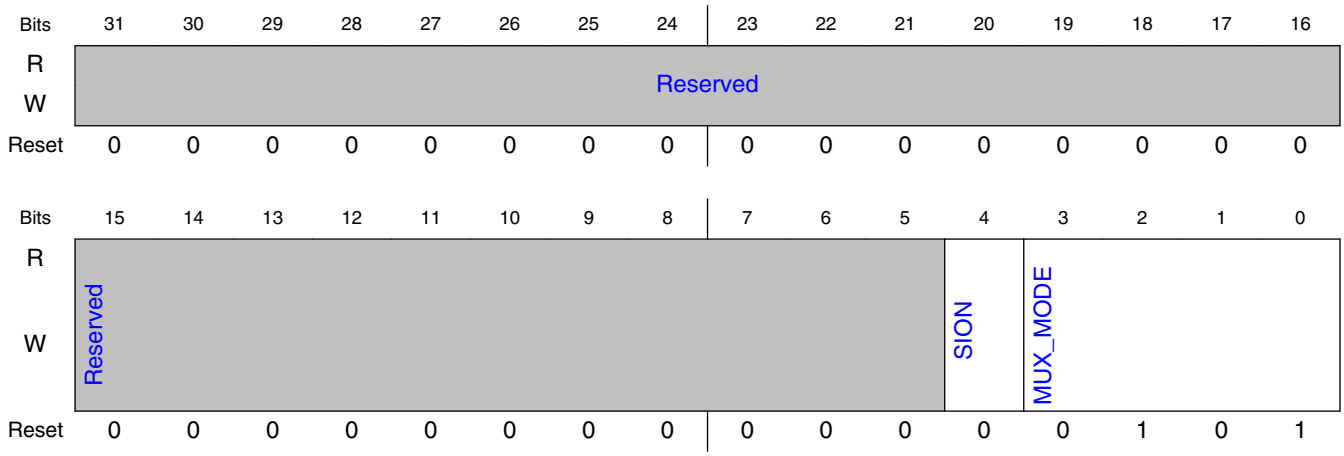
12.4.6.81.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_16	14Ch

12.4.6.81.2 Function

SW_MUX_CTL Register

12.4.6.81.3 Diagram



12.4.6.81.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_16
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_16. 0000b - Select mux mode: ALT0 mux port: SPDIF_OUT of instance: SPDIF 0001b - Select mux mode: ALT1 mux port: LPUART10_RXD of instance: LPUART10 0010b - Select mux mode: ALT2 mux port: GPT1_COMPARE3 of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_SCLK of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA09 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO15 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_RX_DATA03 of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D16 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: ENET_1G_MDC of instance: ENET_1G 1010b - Select mux mode: ALT10 mux port: GPIO9_IO15 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM2_X of instance: FLEXPWM3

12.4.6.82 SW_MUX_CTL_PAD_GPIO_AD_17 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_17)

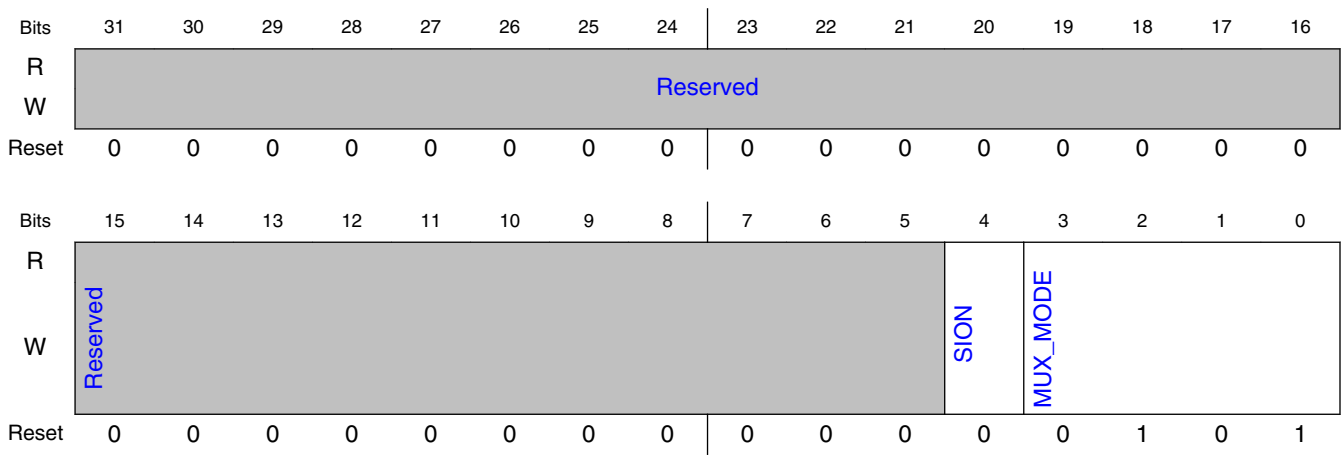
12.4.6.82.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_17	150h

12.4.6.82.2 Function

SW_MUX_CTL Register

12.4.6.82.3 Diagram



12.4.6.82.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_17
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_17. 0000b - Select mux mode: ALT0 mux port: SAI1_MCLK of instance: SAI1 0001b - Select mux mode: ALT1 mux port: ACMP1_OUT of instance: ACMP1 0010b - Select mux mode: ALT2 mux port: GPT1_CLK of instance: GPT1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_DQS of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA08 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO16 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_RX_DATA02 of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D17 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_1G_MDIO of instance: ENET_1G 1010b - Select mux mode: ALT10 mux port: GPIO9_IO16 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM3_PWM3_X of instance: FLEXPWM3

12.4.6.83 SW_MUX_CTL_PAD_GPIO_AD_18 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_18)

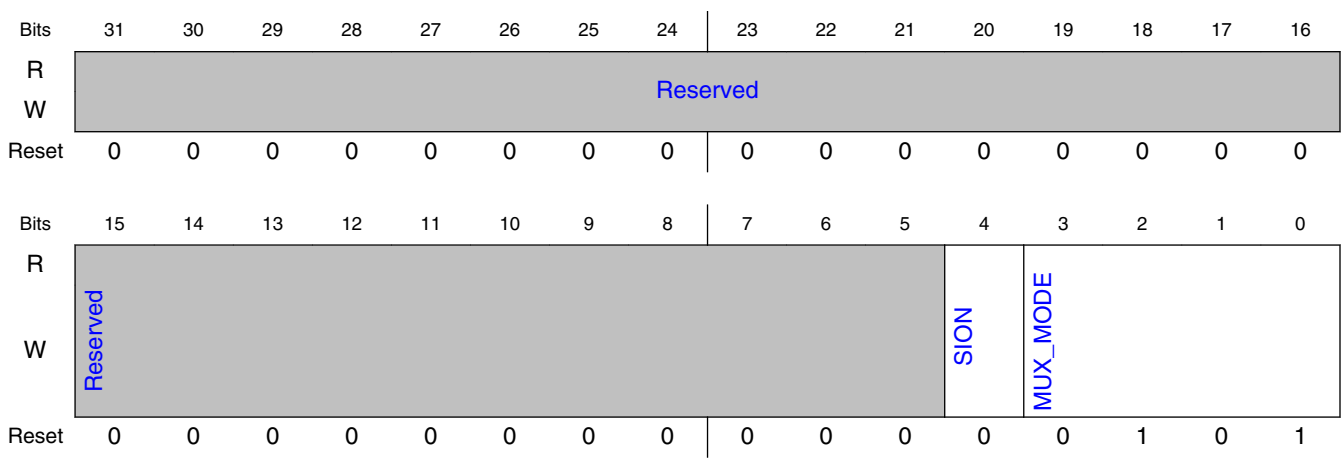
12.4.6.83.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_18	154h

12.4.6.83.2 Function

SW_MUX_CTL Register

12.4.6.83.3 Diagram



12.4.6.83.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_18
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_18. 0000b - Select mux mode: ALT0 mux port: SAI1_RX_SYNC of instance: SAI1 0001b - Select mux mode: ALT1 mux port: ACMP2_OUT of instance: ACMP2 0010b - Select mux mode: ALT2 mux port: LPSPI1_PCS1 of instance: LPSPI1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_SS0_B of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA07 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO17 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_CRS of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D18 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: LPI2C2_SCL of instance: LPI2C2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO17 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM4_PWM0_X of instance: FLEXPWM4

12.4.6.84 SW_MUX_CTL_PAD_GPIO_AD_19 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_19)

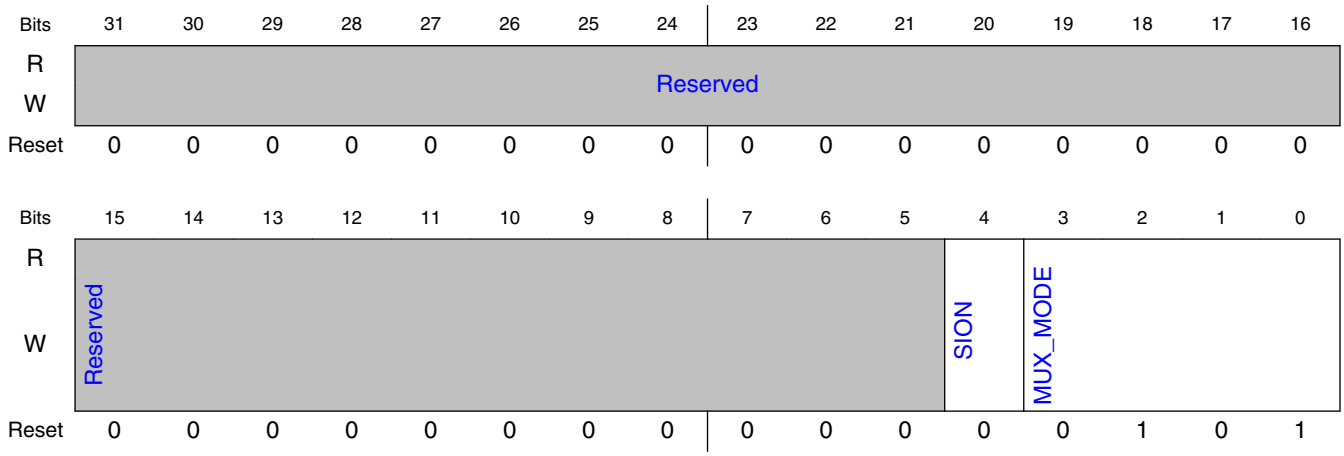
12.4.6.84.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_19	158h

12.4.6.84.2 Function

SW_MUX_CTL Register

12.4.6.84.3 Diagram



12.4.6.84.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_19
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_19. 0000b - Select mux mode: ALT0 mux port: SAI1_RX_BCLK of instance: SAI1 0001b - Select mux mode: ALT1 mux port: ACMP3_OUT of instance: ACMP3 0010b - Select mux mode: ALT2 mux port: LPSPI1_PCS2 of instance: LPSPI1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_SCLK of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA06 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO18 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: ENET_COL of instance: ENET 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D19 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: LPI2C2_SDA of instance: LPI2C2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO18 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM4_PWM1_X of instance: FLEXPWM4

12.4.6.85 SW_MUX_CTL_PAD_GPIO_AD_20 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_20)

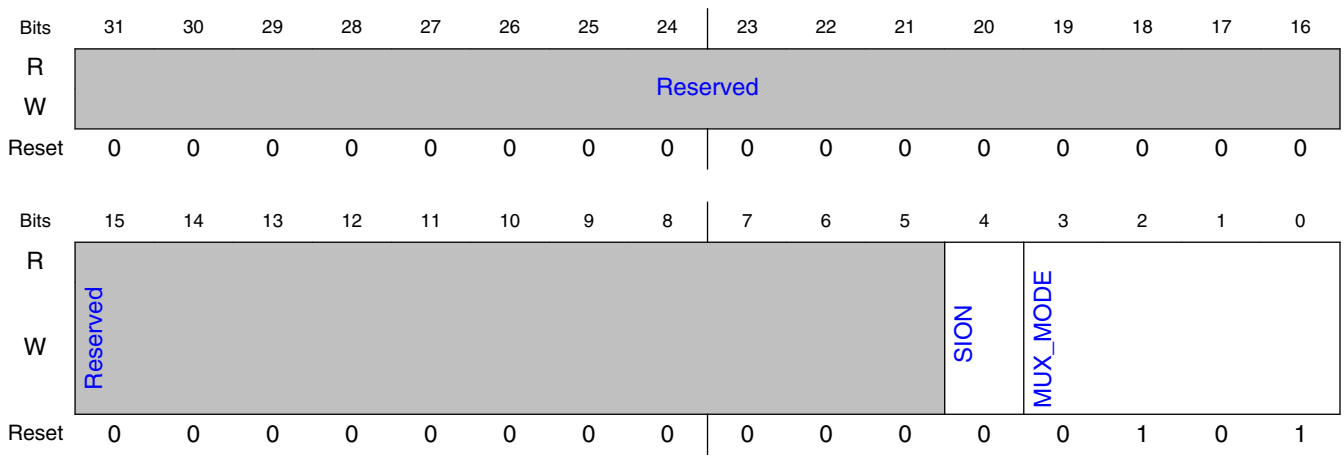
12.4.6.85.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_20	15Ch

12.4.6.85.2 Function

SW_MUX_CTL Register

12.4.6.85.3 Diagram



12.4.6.85.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_20
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_20. 0000b - Select mux mode: ALT0 mux port: SAI1_RX_DATA00 of instance: SAI1 0001b - Select mux mode: ALT1 mux port: ACMP4_OUT of instance: ACMP4 0010b - Select mux mode: ALT2 mux port: LPSPI1_PCS3 of instance: LPSPI1 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_DATA00 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA05 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO19 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW07 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D20 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT2_OUT of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO19 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM4_PWM2_X of instance: FLEXPWM4

12.4.6.86 SW_MUX_CTL_PAD_GPIO_AD_21 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_21)

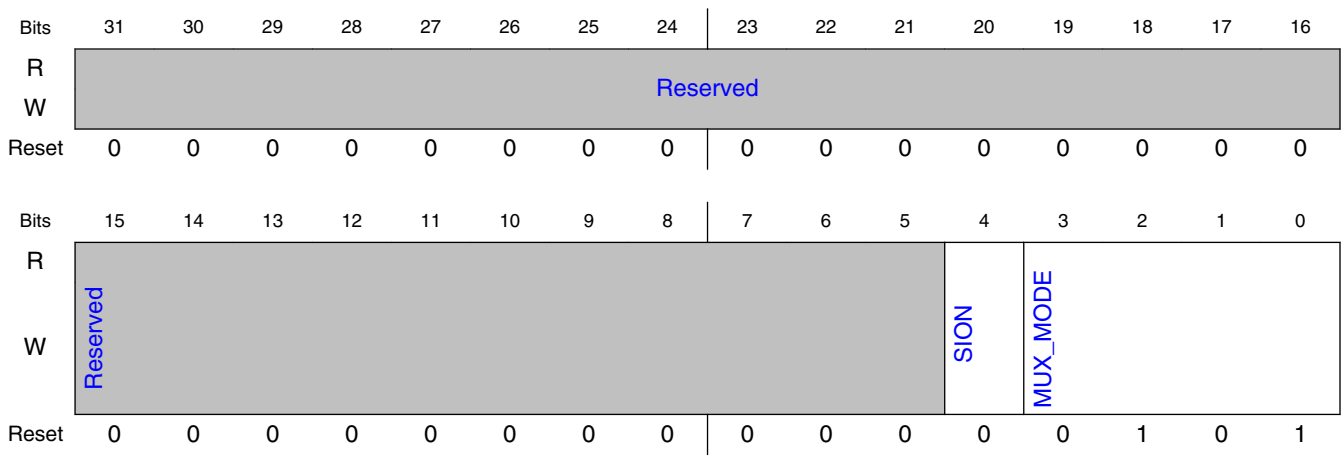
12.4.6.86.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_21	160h

12.4.6.86.2 Function

SW_MUX_CTL Register

12.4.6.86.3 Diagram



12.4.6.86.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_21
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_21. 0000b - Select mux mode: ALT0 mux port: SAI1_TX_DATA00 of instance: SAI1 0010b - Select mux mode: ALT2 mux port: LPSPI2_PCS1 of instance: LPSPI2 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_DATA01 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA04 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO20 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL07 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D21 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT2_IN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO20 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: FLEXPWM4_PWM3_X of instance: FLEXPWM4

12.4.6.87 SW_MUX_CTL_PAD_GPIO_AD_22 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_22)

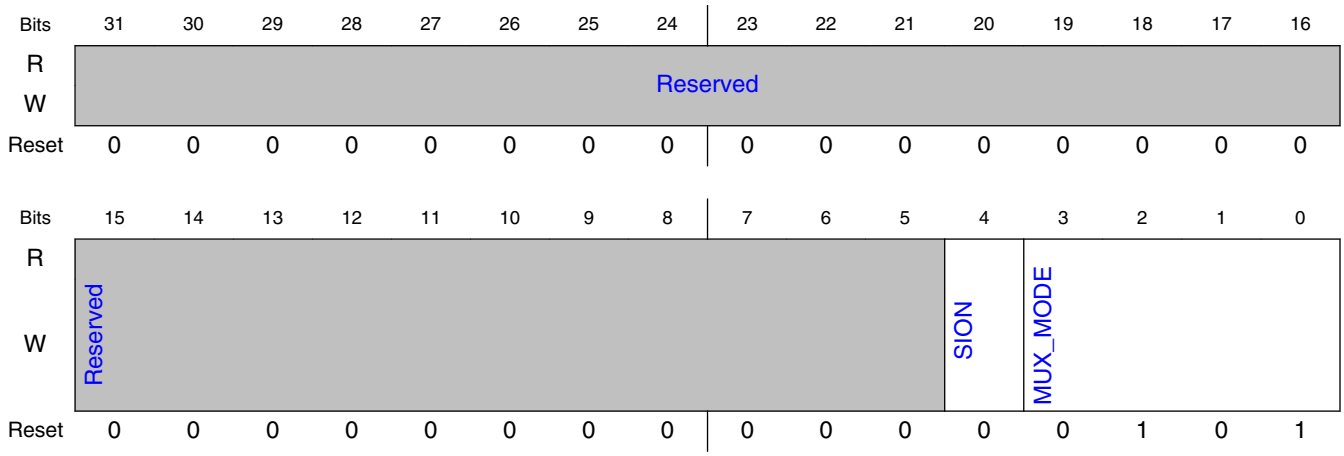
12.4.6.87.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_22	164h

12.4.6.87.2 Function

SW_MUX_CTL Register

12.4.6.87.3 Diagram



12.4.6.87.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_22
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_AD_22. 0000b - Select mux mode: ALT0 mux port: SAI1_TX_BCLK of instance: SAI1 0010b - Select mux mode: ALT2 mux port: LPSPI2_PCS2 of instance: LPSPI2 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_DATA02 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA03 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO21 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW06 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D22 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT3_OUT of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO21 of instance: GPIO9

12.4.6.88 SW_MUX_CTL_PAD_GPIO_AD_23 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_23)

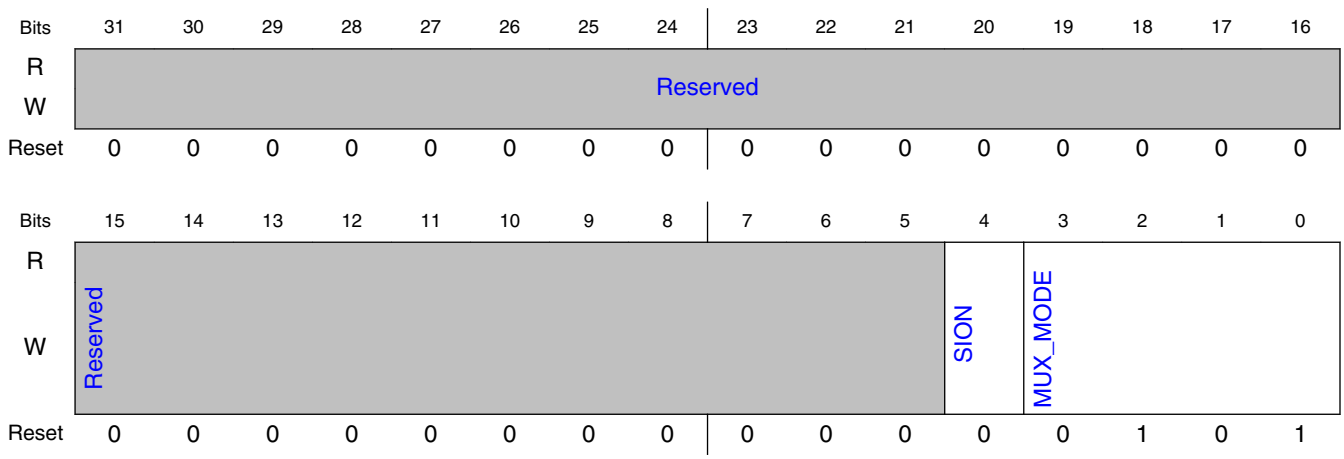
12.4.6.88.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_23	168h

12.4.6.88.2 Function

SW_MUX_CTL Register

12.4.6.88.3 Diagram



12.4.6.88.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_23
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_AD_23. 0000b - Select mux mode: ALT0 mux port: SAI1_TX_SYNC of instance: SAI1 0010b - Select mux mode: ALT2 mux port: LPSPI2_PCS3 of instance: LPSPI2 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_DATA03 of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: VIDEO_MUX_CSI_DATA02 of instance: VIDEO_MUX 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO22 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL06 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D23 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT3_IN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO22 of instance: GPIO9

12.4.6.89 SW_MUX_CTL_PAD_GPIO_AD_24 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_24)

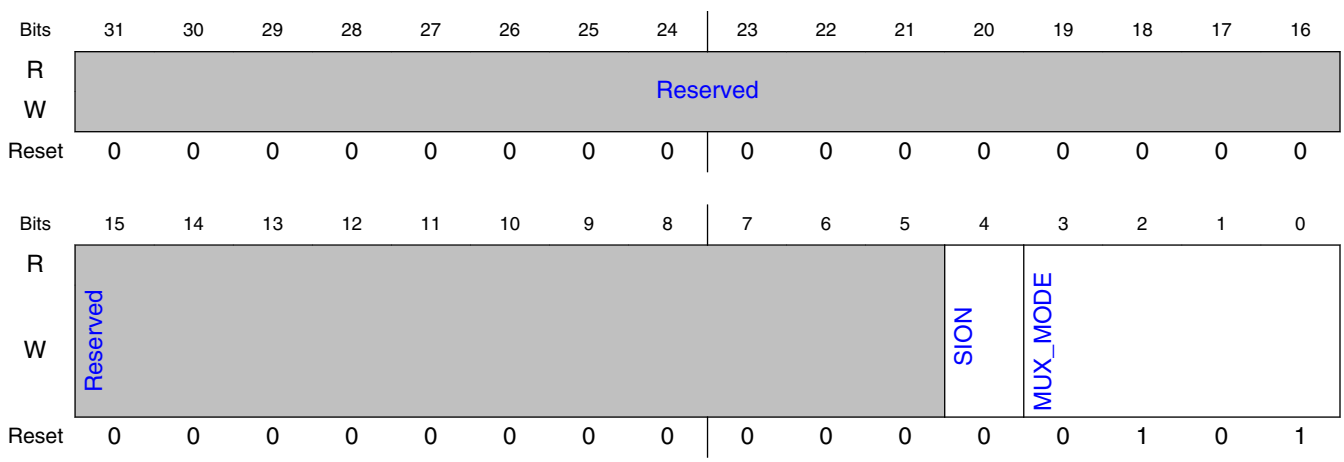
12.4.6.89.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_24	16Ch

12.4.6.89.2 Function

SW_MUX_CTL Register

12.4.6.89.3 Diagram



12.4.6.89.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_24
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_24. 0000b - Select mux mode: ALT0 mux port: LPUART1_TXD of instance: LPUART1 0001b - Select mux mode: ALT1 mux port: LPSPI2_SCK of instance: LPSPI2 0010b - Select mux mode: ALT2 mux port: VIDEO_MUX_CSI_DATA00 of instance: VIDEO_MUX 0011b - Select mux mode: ALT3 mux port: ENET_RX_EN of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM0_A of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO23 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW05 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D24 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: LPI2C4_SCL of instance: LPI2C4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO23 of instance: GPIO9

12.4.6.90 SW_MUX_CTL_PAD_GPIO_AD_25 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_25)

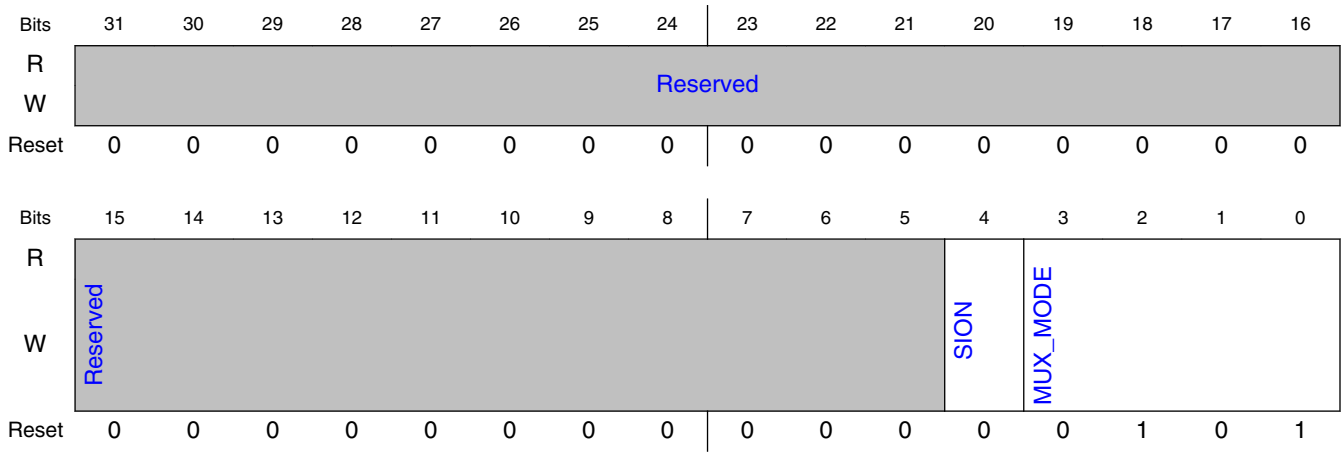
12.4.6.90.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_25	170h

12.4.6.90.2 Function

SW_MUX_CTL Register

12.4.6.90.3 Diagram



12.4.6.90.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_25
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_25. 0000b - Select mux mode: ALT0 mux port: LPUART1_RXD of instance: LPUART1 0001b - Select mux mode: ALT1 mux port: LPSPI2_PCS0 of instance: LPSPI2 0010b - Select mux mode: ALT2 mux port: VIDEO_MUX_CSI_DATA01 of instance: VIDEO_MUX 0011b - Select mux mode: ALT3 mux port: ENET_RX_ER of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM0_B of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO24 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL05 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D25 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: LPI2C4_SDA of instance: LPI2C4 1010b - Select mux mode: ALT10 mux port: GPIO9_IO24 of instance: GPIO9

12.4.6.91 SW_MUX_CTL_PAD_GPIO_AD_26 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_26)

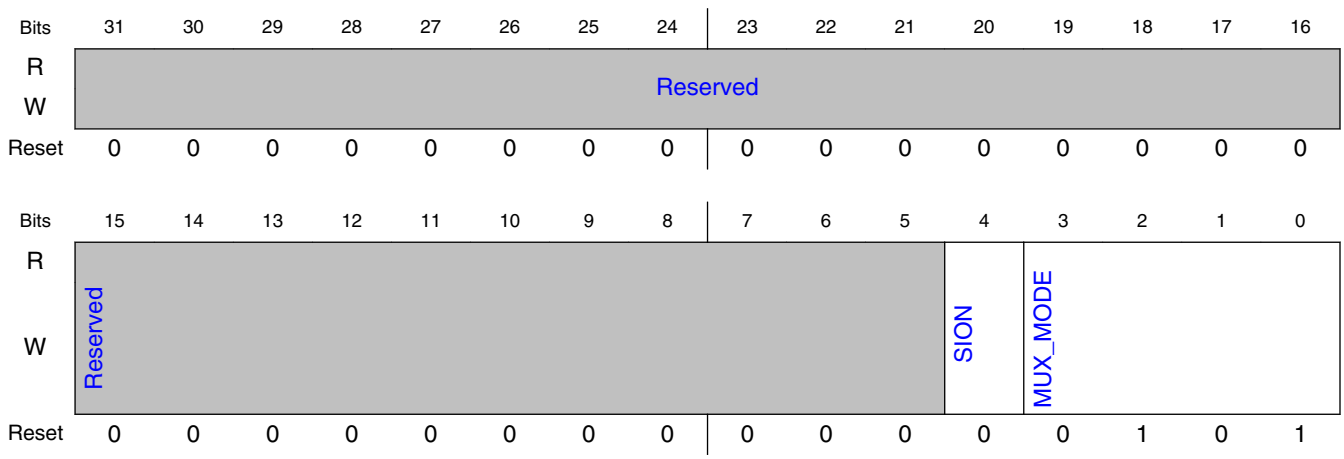
12.4.6.91.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_26	174h

12.4.6.91.2 Function

SW_MUX_CTL Register

12.4.6.91.3 Diagram



12.4.6.91.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_26
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_26. 0000b - Select mux mode: ALT0 mux port: LPUART1_CTS_B of instance: LPUART1 0001b - Select mux mode: ALT1 mux port: LPSPI2_SOUT of instance: LPSPI2 0010b - Select mux mode: ALT2 mux port: SEMC_CSX01 of instance: SEMC 0011b - Select mux mode: ALT3 mux port: ENET_RX_DATA00 of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM1_A of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO25 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW04 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D26 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_QOS_MDC of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO25 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: USDHC2_CD_B of instance: USDHC2

12.4.6.92 SW_MUX_CTL_PAD_GPIO_AD_27 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_27)

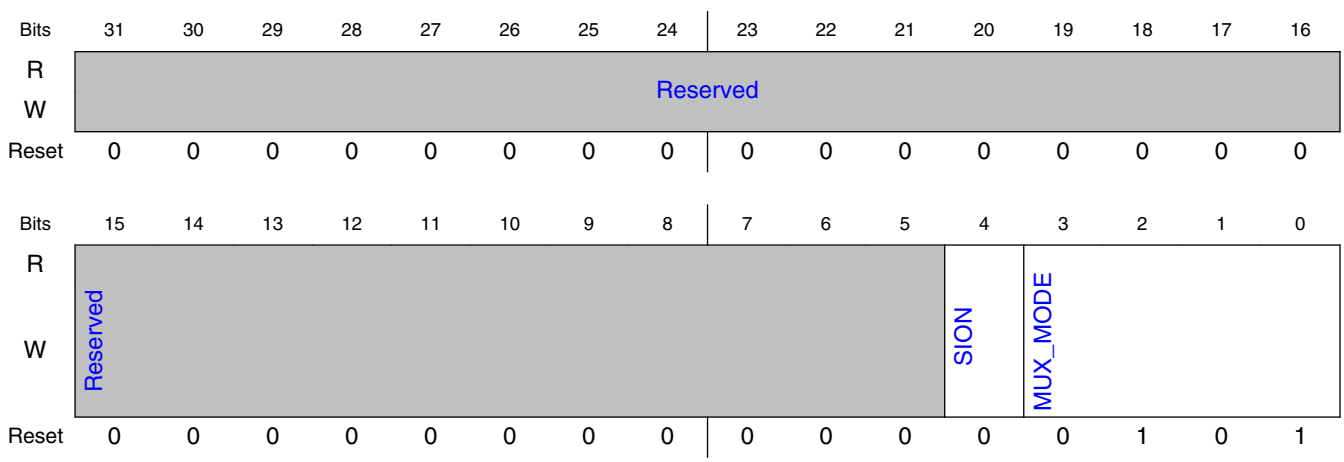
12.4.6.92.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_27	178h

12.4.6.92.2 Function

SW_MUX_CTL Register

12.4.6.92.3 Diagram



12.4.6.92.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_27
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_27. 0000b - Select mux mode: ALT0 mux port: LPUART1_RTS_B of instance: LPUART1 0001b - Select mux mode: ALT1 mux port: LPSPI2_SIN of instance: LPSPI2 0010b - Select mux mode: ALT2 mux port: SEMC_CSX02 of instance: SEMC 0011b - Select mux mode: ALT3 mux port: ENET_RX_DATA01 of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM1_B of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO26 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL04 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D27 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: ENET_QOS_MDIO of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO9_IO26 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: USDHC2_WP of instance: USDHC2

12.4.6.93 SW_MUX_CTL_PAD_GPIO_AD_28 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_28)

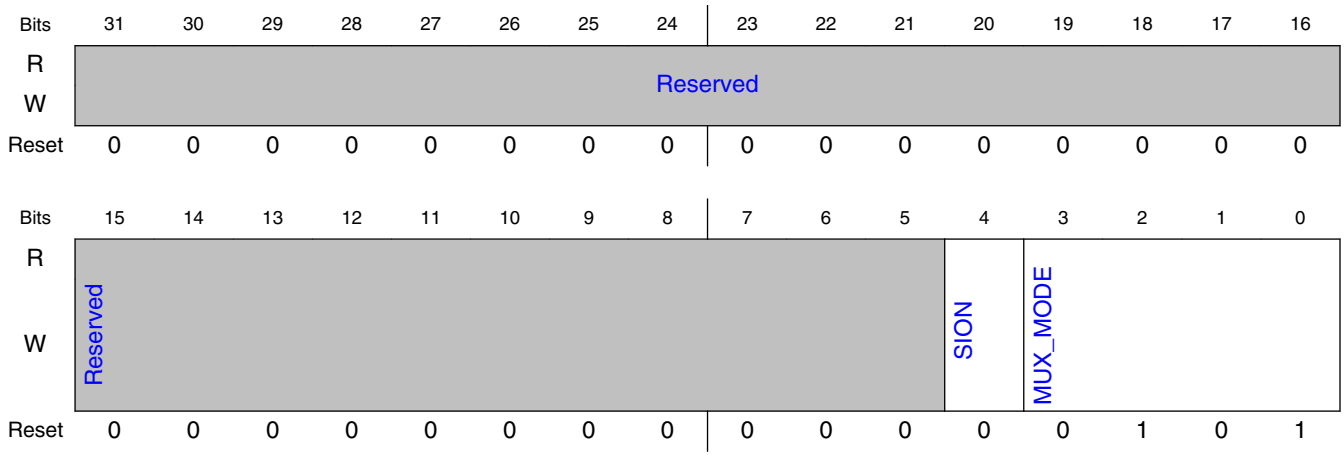
12.4.6.93.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_28	17Ch

12.4.6.93.2 Function

SW_MUX_CTL Register

12.4.6.93.3 Diagram



12.4.6.93.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_28
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_28. 0000b - Select mux mode: ALT0 mux port: LPSPI1_SCK of instance: LPSPI1 0001b - Select mux mode: ALT1 mux port: LPUART5_TXD of instance: LPUART5 0010b - Select mux mode: ALT2 mux port: SEMC_CSX03 of instance: SEMC 0011b - Select mux mode: ALT3 mux port: ENET_TX_EN of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM2_A of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO27 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW03 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D28 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: VIDEO_MUX_EXT_DCIC1 of instance: VIDEO_MUX 1010b - Select mux mode: ALT10 mux port: GPIO9_IO27 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: USDHC2_VSELECT of instance: USDHC2

12.4.6.94 SW_MUX_CTL_PAD_GPIO_AD_29 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_29)

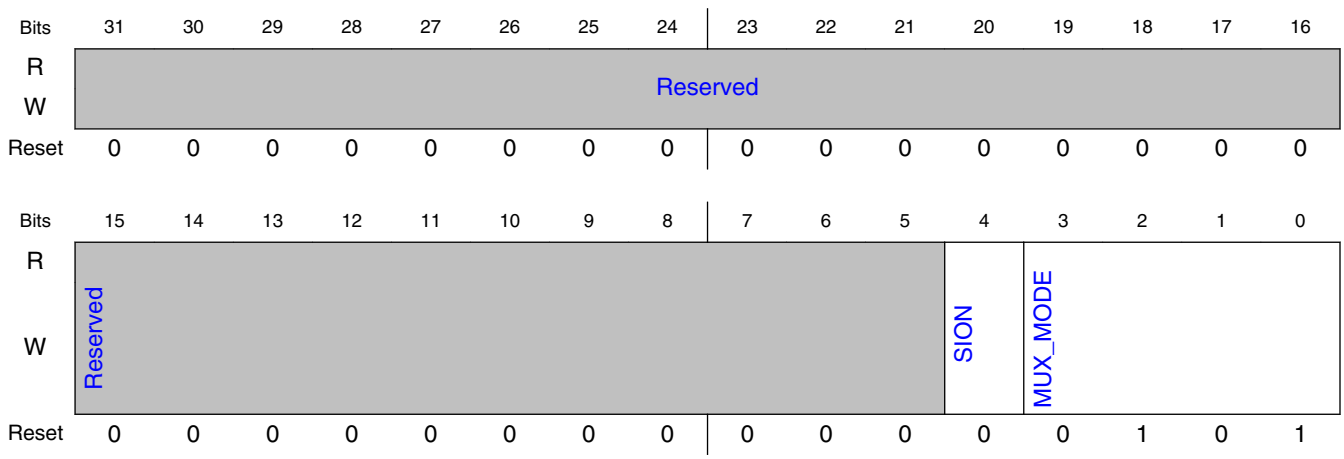
12.4.6.94.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_29	180h

12.4.6.94.2 Function

SW_MUX_CTL Register

12.4.6.94.3 Diagram



12.4.6.94.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_29
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_AD_29. 0000b - Select mux mode: ALT0 mux port: LPSPI1_PCS0 of instance: LPSPI1 0001b - Select mux mode: ALT1 mux port: LPUART5_RXD of instance: LPUART5 0010b - Select mux mode: ALT2 mux port: ENET_REF_CLK of instance: ENET 0011b - Select mux mode: ALT3 mux port: ENET_TX_CLK of instance: ENET 0100b - Select mux mode: ALT4 mux port: FLEXPWM2_PWM2_B of instance: FLEXPWM2 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO28 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL03 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D29 of instance: FLEXIO2

Field	Function
	1001b - Select mux mode: ALT9 mux port: VIDEO_MUX_EXT_DCIC2 of instance: VIDEO_MUX 1010b - Select mux mode: ALT10 mux port: GPIO9_IO28 of instance: GPIO9 1011b - Select mux mode: ALT11 mux port: USDHC2_RESET_B of instance: USDHC2

12.4.6.95 SW_MUX_CTL_PAD_GPIO_AD_30 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_30)

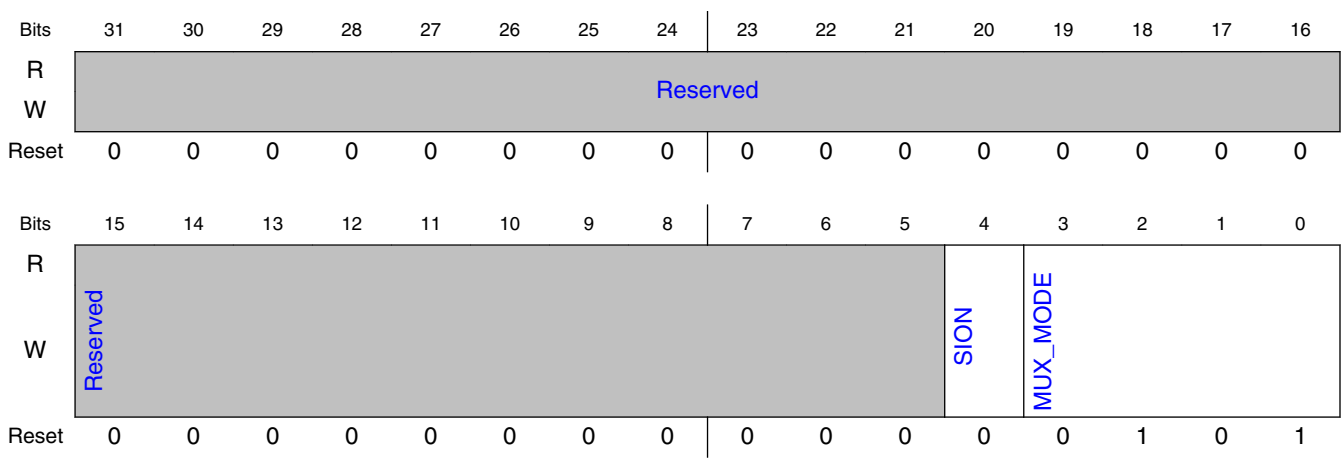
12.4.6.95.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_30	184h

12.4.6.95.2 Function

SW_MUX_CTL Register

12.4.6.95.3 Diagram



12.4.6.95.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_30
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_30. 0000b - Select mux mode: ALT0 mux port: LPSPI1_SOUT of instance: LPSPI1 0001b - Select mux mode: ALT1 mux port: USB_OTG2_OC of instance: USB 0010b - Select mux mode: ALT2 mux port: FLEXCAN2_TX of instance: FLEXCAN2 0011b - Select mux mode: ALT3 mux port: ENET_TX_DATA00 of instance: ENET 0100b - Select mux mode: ALT4 mux port: LPUART3_TXD of instance: LPUART3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO29 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW02 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D30 of instance: FLEXIO2 1001b - Select mux mode: WDOG2_RESET_B_DEB of instance: WDOG2 1010b - Select mux mode: ALT10 mux port: GPIO9_IO29 of instance: GPIO9

12.4.6.96 SW_MUX_CTL_PAD_GPIO_AD_31 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_31)

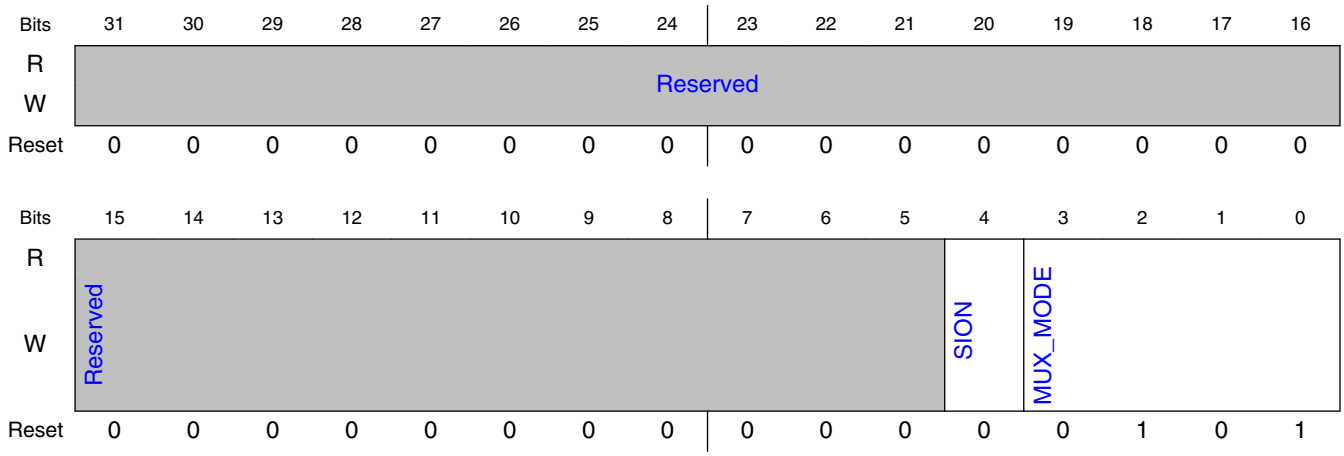
12.4.6.96.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_31	188h

12.4.6.96.2 Function

SW_MUX_CTL Register

12.4.6.96.3 Diagram



12.4.6.96.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_31
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_31. 0000b - Select mux mode: ALT0 mux port: LPSPI1_SIN of instance: LPSPI1 0001b - Select mux mode: ALT1 mux port: USB_OTG2_PWR of instance: USB 0010b - Select mux mode: ALT2 mux port: FLEXCAN2_RX of instance: FLEXCAN2 0011b - Select mux mode: ALT3 mux port: ENET_TX_DATA01 of instance: ENET 0100b - Select mux mode: ALT4 mux port: LPUART3_RXD of instance: LPUART3 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO30 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_COL02 of instance: KPP 1000b - Select mux mode: ALT8 mux port: FLEXIO2_D31 of instance: FLEXIO2 1001b - Select mux mode: ALT9 mux port: WDOG1_RESET_B_DEB of instance: WDOG1 1010b - Select mux mode: ALT10 mux port: GPIO9_IO30 of instance: GPIO9

12.4.6.97 SW_MUX_CTL_PAD_GPIO_AD_32 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_32)

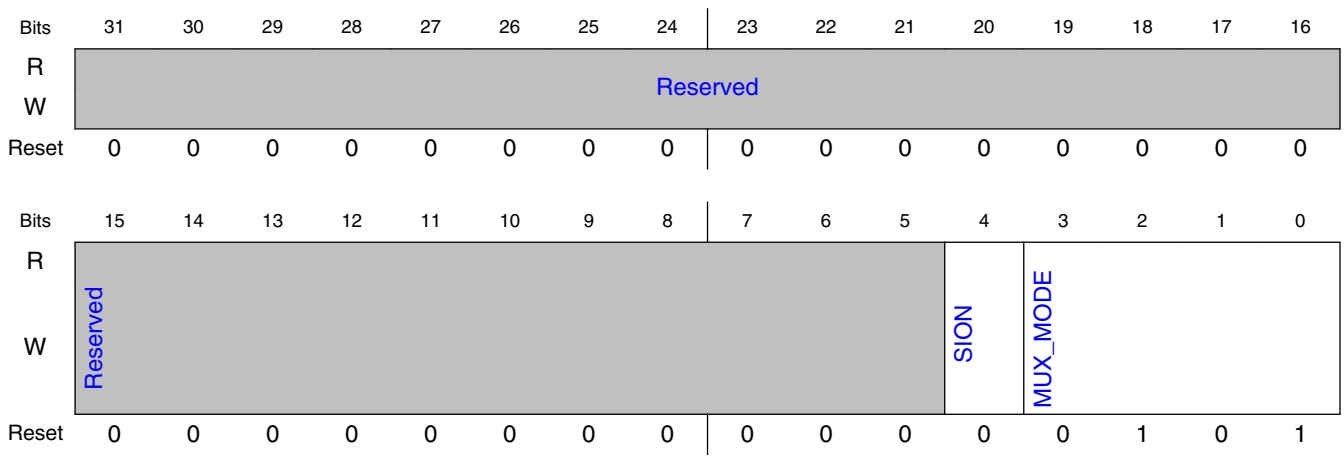
12.4.6.97.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_32	18Ch

12.4.6.97.2 Function

SW_MUX_CTL Register

12.4.6.97.3 Diagram



12.4.6.97.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_32
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_32. 0000b - Select mux mode: ALT0 mux port: LPI2C1_SCL of instance: LPI2C1 0001b - Select mux mode: ALT1 mux port: USBPHY2_OTG_ID of instance: USBPHY2 0010b - Select mux mode: ALT2 mux port: PGMC_PMIC_RDY of instance: pgmc 0011b - Select mux mode: ALT3 mux port: ENET_MDC of instance: ENET 0100b - Select mux mode: ALT4 mux port: USDHC1_CD_B of instance: USDHC1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX3_IO31 of instance: GPIO_MUX3 0110b - Select mux mode: ALT6 mux port: KPP_ROW01 of instance: KPP 1000b - Select mux mode: ALT8 mux port: LPUART10_TXD of instance: LPUART10

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_1G_MDC of instance: ENET_1G 1010b - Select mux mode: ALT10 mux port: GPIO9_IO31 of instance: GPIO9

12.4.6.98 SW_MUX_CTL_PAD_GPIO_AD_33 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_33)

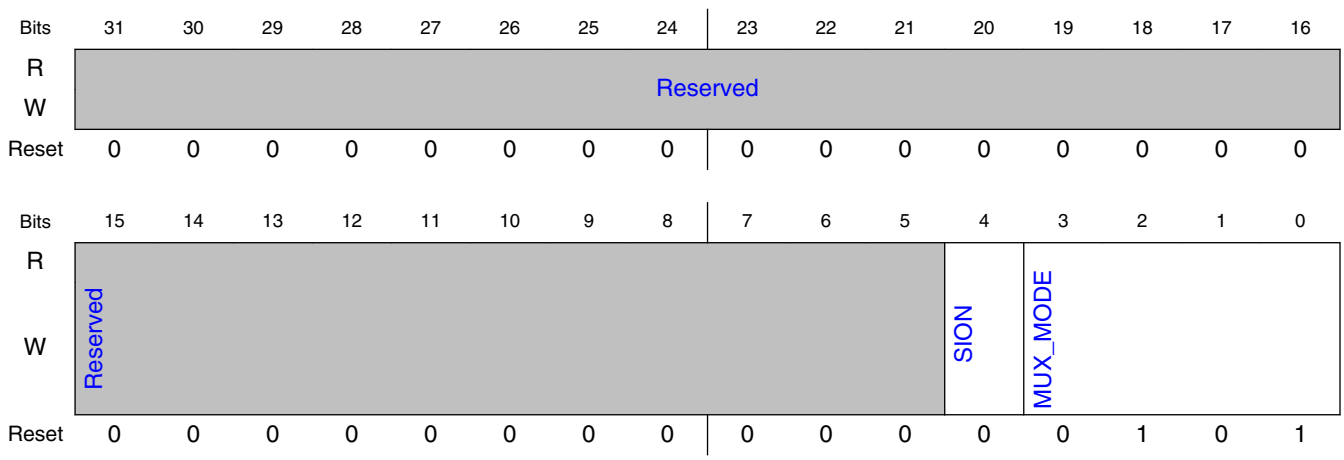
12.4.6.98.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_33	190h

12.4.6.98.2 Function

SW_MUX_CTL Register

12.4.6.98.3 Diagram



12.4.6.98.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_33
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_33. 0000b - Select mux mode: ALT0 mux port: LPI2C1_SDA of instance: LPI2C1 0001b - Select mux mode: ALT1 mux port: USBPHY1_OTG_ID of instance: USBPHY1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT17 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: ENET_MDIO of instance: ENET 0100b - Select mux mode: ALT4 mux port: USDHC1_WP of instance: USDHC1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO00 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: KPP_COL01 of instance: KPP 1000b - Select mux mode: ALT8 mux port: LPUART10_RXD of instance: LPUART10 1001b - Select mux mode: ALT9 mux port: ENET_1G_MDIO of instance: ENET_1G 1010b - Select mux mode: ALT10 mux port: GPIO10_IO00 of instance: GPIO10

12.4.6.99 SW_MUX_CTL_PAD_GPIO_AD_34 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_34)

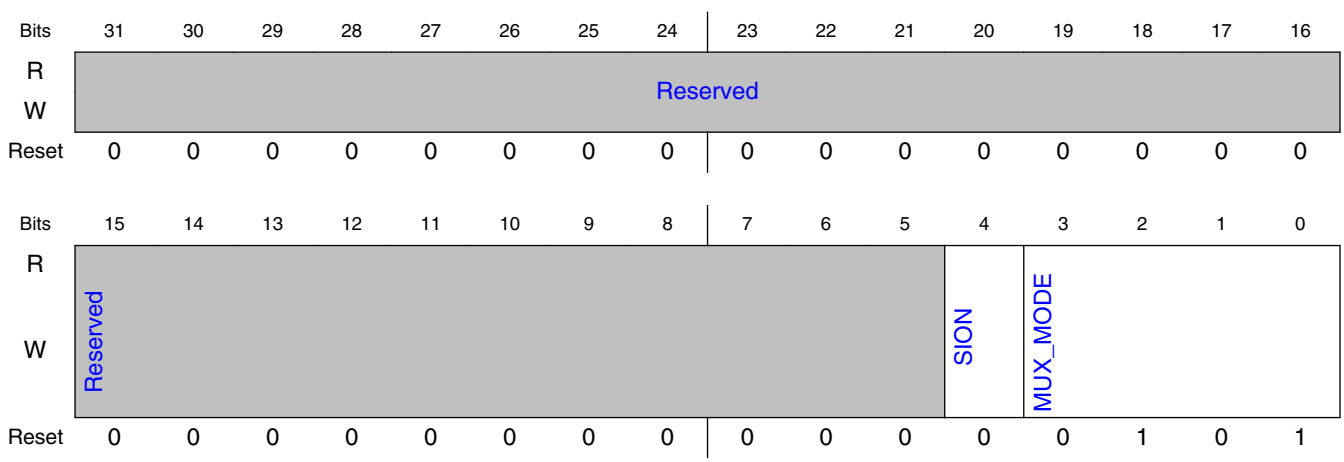
12.4.6.99.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_34	194h

12.4.6.99.2 Function

SW_MUX_CTL Register

12.4.6.99.3 Diagram



12.4.6.99.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_34
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_34. 0000b - Select mux mode: ALT0 mux port: ENET_1G_1588_EVENT0_IN of instance: ENET_1G 0001b - Select mux mode: ALT1 mux port: USB_OTG1_PWR of instance: USB 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT18 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: ENET_1588_EVENT0_IN of instance: ENET 0100b - Select mux mode: ALT4 mux port: USDHC1_VSELECT of instance: USDHC1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO01 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: KPP_ROW00 of instance: KPP 1000b - Select mux mode: ALT8 mux port: LPUART10_CTS_B of instance: LPUART10 1001b - Select mux mode: ALT9 mux port: WDOG1_ANY of instance: WDOG1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO01 of instance: GPIO10

12.4.6.100 SW_MUX_CTL_PAD_GPIO_AD_35 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_AD_35)

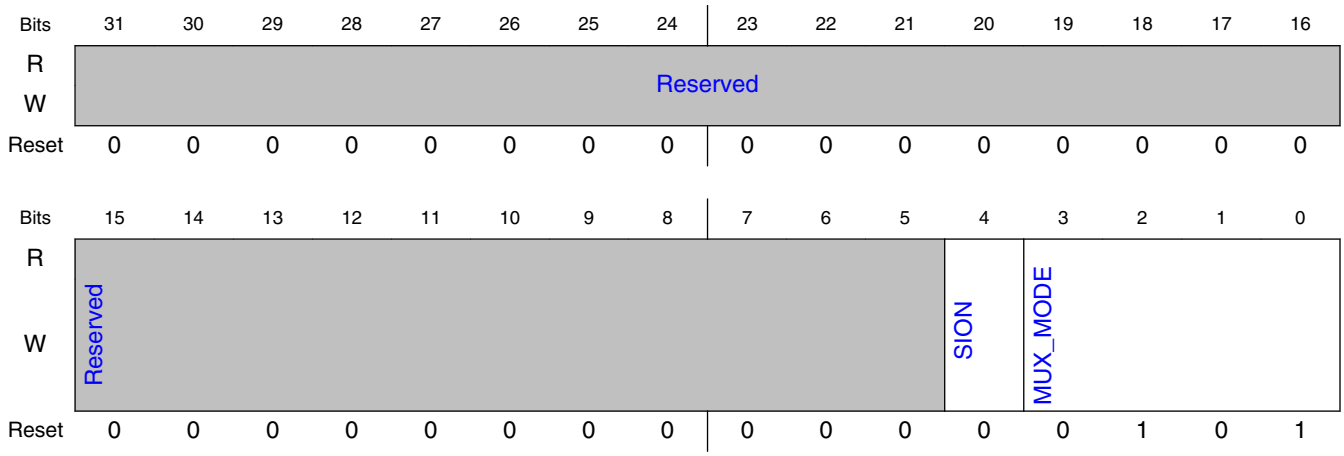
12.4.6.100.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_AD_35	198h

12.4.6.100.2 Function

SW_MUX_CTL Register

12.4.6.100.3 Diagram



12.4.6.100.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_AD_35
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_AD_35. 0000b - Select mux mode: ALT0 mux port: ENET_1G_1588_EVENT0_OUT of instance: ENET_1G 0001b - Select mux mode: ALT1 mux port: USB_OTG1_OC of instance: USB 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT19 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: ENET_1588_EVENT0_OUT of instance: ENET 0100b - Select mux mode: ALT4 mux port: USDHC1_RESET_B of instance: USDHC1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO02 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: KPP_COL00 of instance: KPP 1000b - Select mux mode: ALT8 mux port: LPUART10_RTS_B of instance: LPUART10 1001b - Select mux mode: ALT9 mux port: FLEXSPI1_B_SS1_B of instance: FLEXSPI1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO02 of instance: GPIO10

12.4.6.101 SW_MUX_CTL_PAD_GPIO_SD_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_00)

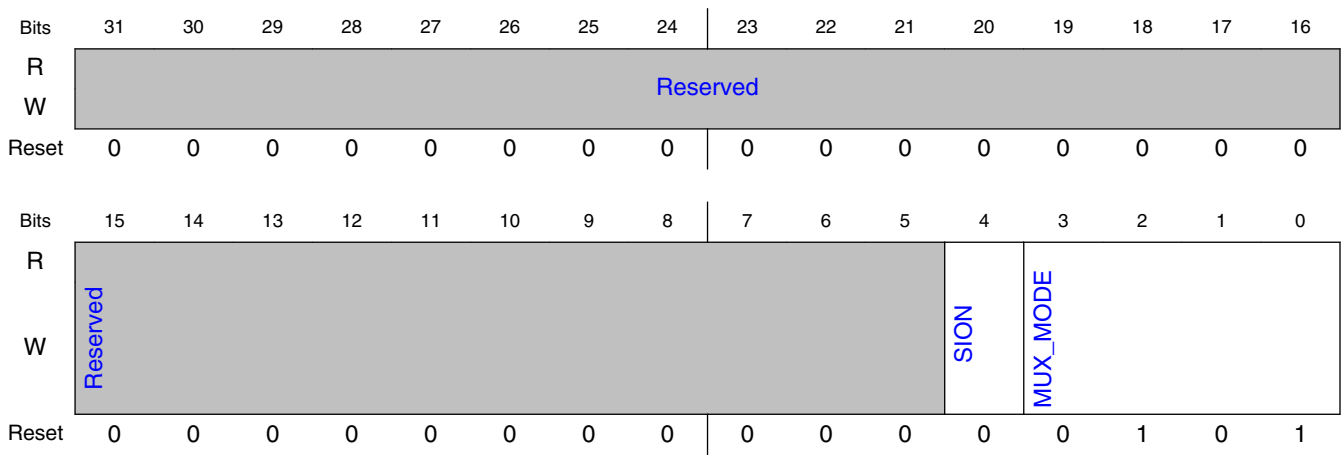
12.4.6.101.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_00	19Ch

12.4.6.101.2 Function

SW_MUX_CTL Register

12.4.6.101.3 Diagram



12.4.6.101.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_SD_B1_00. 0000b - Select mux mode: ALT0 mux port: USDHC1_CMD of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT20 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_CAPTURE1 of instance: GPT4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO03 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_SS0_B of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: KPP_ROW07 of instance: KPP 1010b - Select mux mode: ALT10 mux port: GPIO10_IO03 of instance: GPIO10

12.4.6.102 SW_MUX_CTL_PAD_GPIO_SD_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_01)

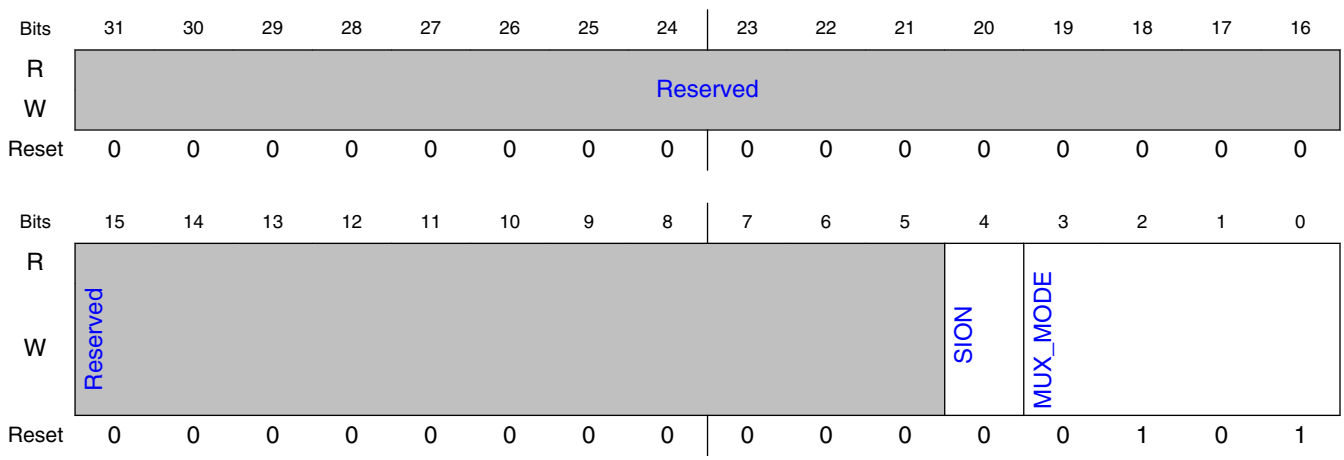
12.4.6.102.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_01	1A0h

12.4.6.102.2 Function

SW_MUX_CTL Register

12.4.6.102.3 Diagram



12.4.6.102.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_SD_B1_01.

Field	Function
	0000b - Select mux mode: ALT0 mux port: USDHC1_CLK of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT21 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_CAPTURE2 of instance: GPT4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO04 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_SCLK of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: KPP_COL07 of instance: KPP 1010b - Select mux mode: ALT10 mux port: GPIO10_IO04 of instance: GPIO10

12.4.6.103 SW_MUX_CTL_PAD_GPIO_SD_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_02)

12.4.6.103.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_02	1A4h

12.4.6.103.2 Function

SW_MUX_CTL Register

12.4.6.103.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	MUX_MODE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

12.4.6.103.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B1_02. 0000b - Select mux mode: ALT0 mux port: USDHC1_DATA0 of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT22 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_COMPARE1 of instance: GPT4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO05 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_DATA00 of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: KPP_ROW06 of instance: KPP 1001b - Select mux mode: ALT9 mux port: FLEXSPI1_A_SS1_B of instance: FLEXSPI1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO05 of instance: GPIO10

12.4.6.104 SW_MUX_CTL_PAD_GPIO_SD_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_03)

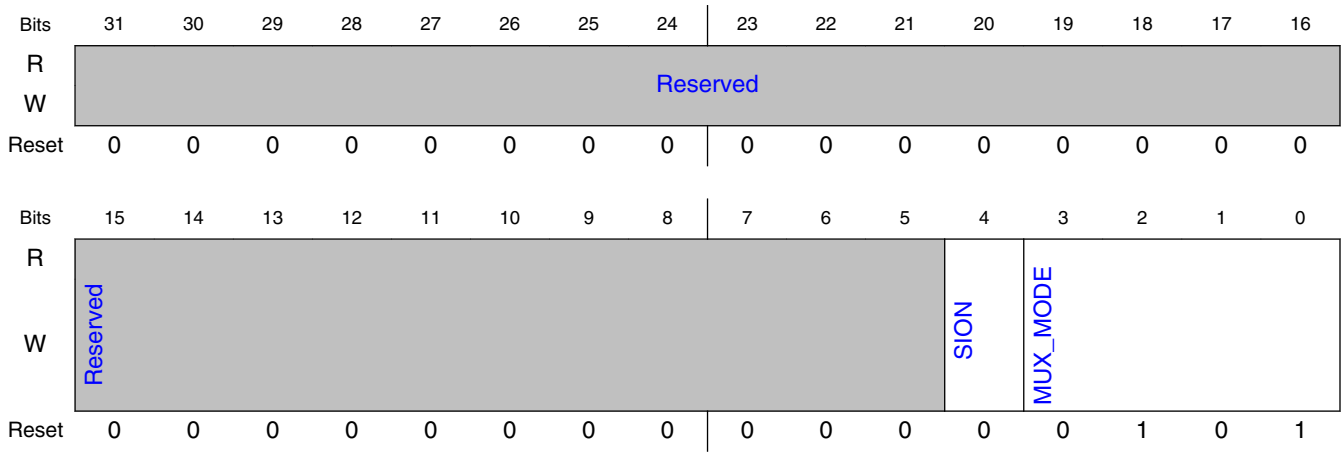
12.4.6.104.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_03	1A8h

12.4.6.104.2 Function

SW_MUX_CTL Register

12.4.6.104.3 Diagram



12.4.6.104.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B1_03. 0000b - Select mux mode: ALT0 mux port: USDHC1_DATA1 of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT23 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_COMPARE2 of instance: GPT4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO06 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_DATA01 of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: KPP_COL06 of instance: KPP 1001b - Select mux mode: ALT9 mux port: FLEXSPI1_B_SS1_B of instance: FLEXSPI1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO06 of instance: GPIO10

12.4.6.105 SW_MUX_CTL_PAD_GPIO_SD_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_04)

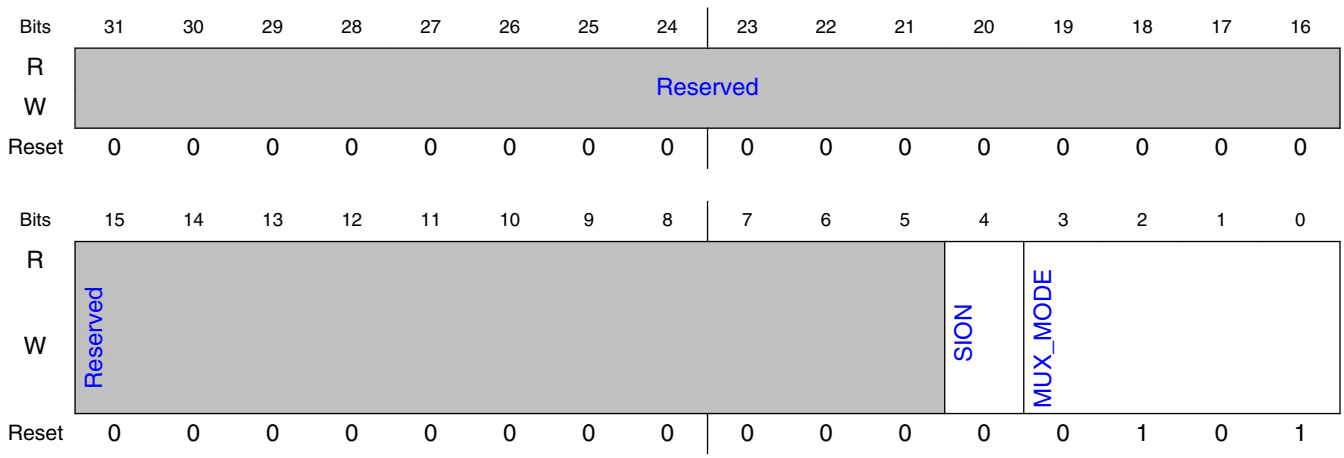
12.4.6.105.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_04	1ACh

12.4.6.105.2 Function

SW_MUX_CTL Register

12.4.6.105.3 Diagram



12.4.6.105.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B1_04. 0000b - Select mux mode: ALT0 mux port: USDHC1_DATA2 of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT24 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_COMPARE3 of instance: GPT4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO07 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_DATA02 of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: FLEXSPI1_B_SS0_B of instance: FLEXSPI1 1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT2_AUX_IN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO10_IO07 of instance: GPIO10

12.4.6.106 SW_MUX_CTL_PAD_GPIO_SD_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B1_05)

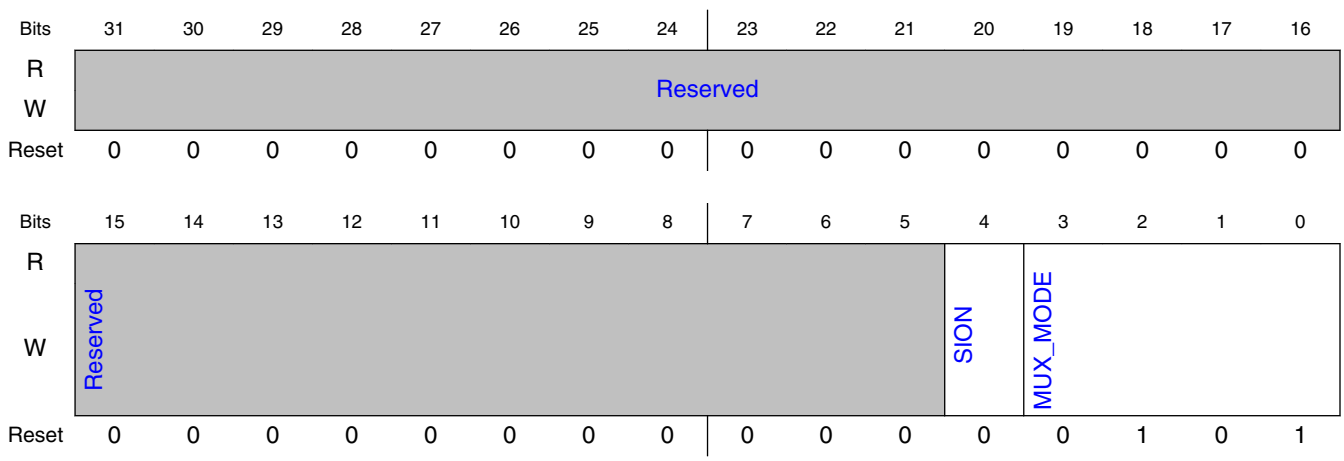
12.4.6.106.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B1_05	1B0h

12.4.6.106.2 Function

SW_MUX_CTL Register

12.4.6.106.3 Diagram



12.4.6.106.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B1_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B1_05. 0000b - Select mux mode: ALT0 mux port: USDHC1_DATA3 of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: XBAR1_INOUT25 of instance: XBAR1 0011b - Select mux mode: ALT3 mux port: GPT4_CLK of instance: GPT4

Field	Function
	0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO08 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: FLEXSPI2_A_DATA03 of instance: FLEXSPI2 1000b - Select mux mode: ALT8 mux port: FLEXSPI1_B_DQS of instance: FLEXSPI1 1001b - Select mux mode: ALT9 mux port: ENET_QOS_1588_EVENT3_AUX_IN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO10_IO08 of instance: GPIO10

12.4.6.107 SW_MUX_CTL_PAD_GPIO_SD_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_00)

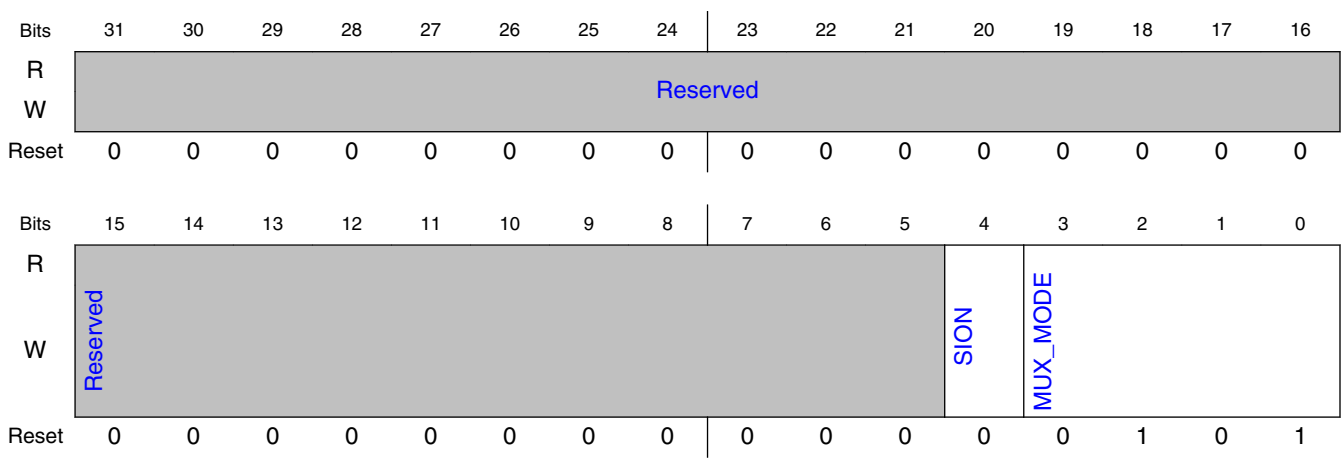
12.4.6.107.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_00	1B4h

12.4.6.107.2 Function

SW_MUX_CTL Register

12.4.6.107.3 Diagram



12.4.6.107.4 Fields

Field	Function
31-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_00. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA3 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_B_DATA03 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_EN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART9_TXD of instance: LPUART9 0100b - Select mux mode: ALT4 mux port: LPSPI4_SCK of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO09 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO09 of instance: GPIO10

12.4.6.108 SW_MUX_CTL_PAD_GPIO_SD_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_01)

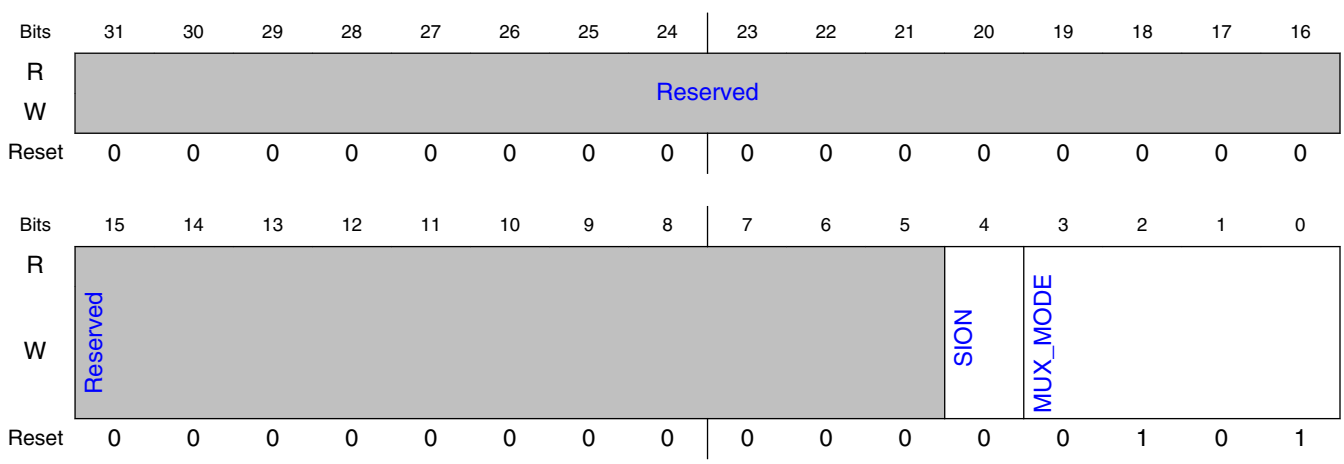
12.4.6.108.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_01	1B8h

12.4.6.108.2 Function

SW_MUX_CTL Register

12.4.6.108.3 Diagram



12.4.6.108.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_01. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA2 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_B_DATA02 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_CLK of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART9_RXD of instance: LPUART9 0100b - Select mux mode: ALT4 mux port: LPSPI4_PCS0 of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO10 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO10 of instance: GPIO10

12.4.6.109 SW_MUX_CTL_PAD_GPIO_SD_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_02)

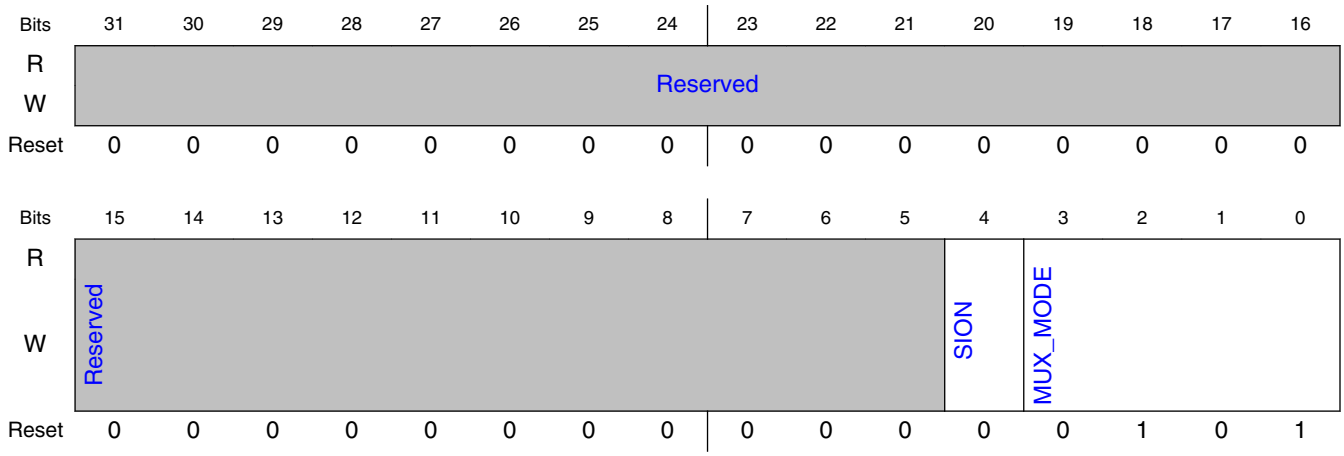
12.4.6.109.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_02	1BCh

12.4.6.109.2 Function

SW_MUX_CTL Register

12.4.6.109.3 Diagram



12.4.6.109.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_02. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA1 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_B_DATA01 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA00 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART9_CTS_B of instance: LPUART9 0100b - Select mux mode: ALT4 mux port: LPSPI4_SOUT of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO11 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO11 of instance: GPIO10

12.4.6.110 SW_MUX_CTL_PAD_GPIO_SD_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_03)

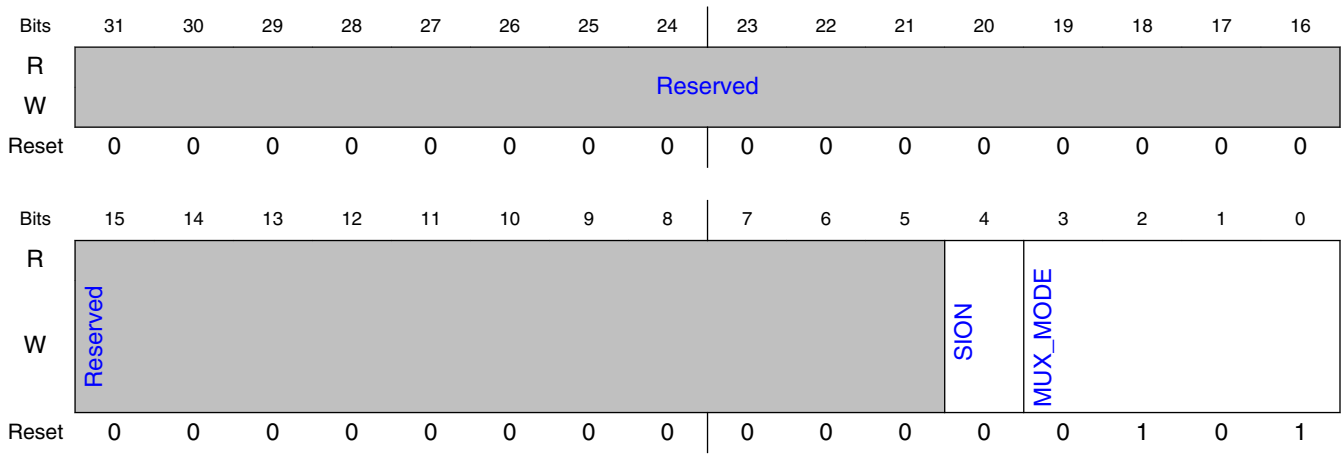
12.4.6.110.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_03	1C0h

12.4.6.110.2 Function

SW_MUX_CTL Register

12.4.6.110.3 Diagram



12.4.6.110.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_03. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA0 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_B_DATA00 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA01 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART9_RTS_B of instance: LPUART9 0100b - Select mux mode: ALT4 mux port: LPSPI4_SIN of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO12 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO12 of instance: GPIO10

12.4.6.111 SW_MUX_CTL_PAD_GPIO_SD_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_04)

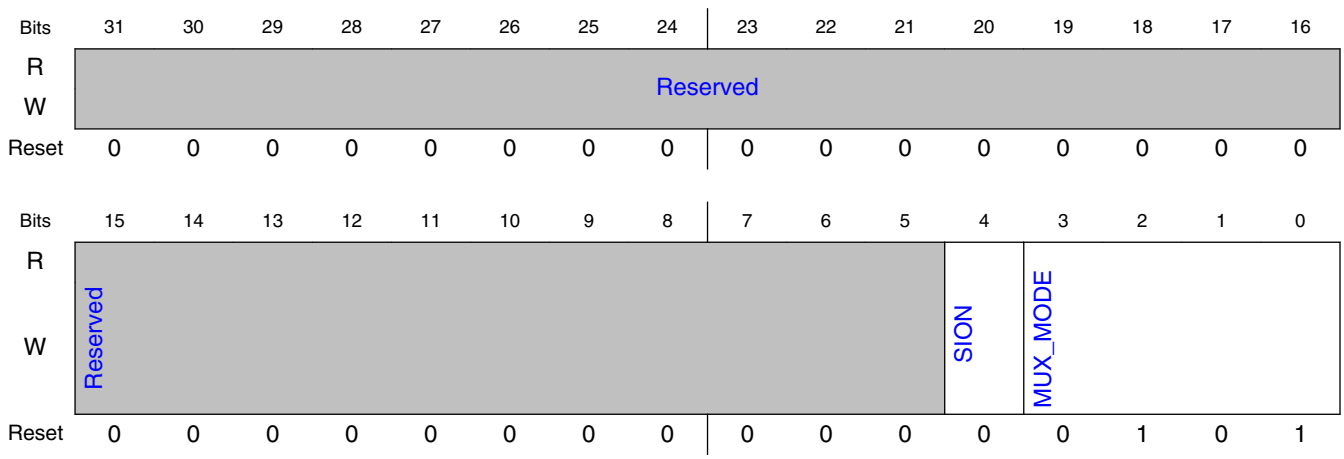
12.4.6.111.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_04	1C4h

12.4.6.111.2 Function

SW_MUX_CTL Register

12.4.6.111.3 Diagram



12.4.6.111.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_04. 0000b - Select mux mode: ALT0 mux port: USDHC2_CLK of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_B_SCLK of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA02 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_A_SS1_B of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: LPSPI4_PCS1 of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO13 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO13 of instance: GPIO10

12.4.6.112 SW_MUX_CTL_PAD_GPIO_SD_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_05)

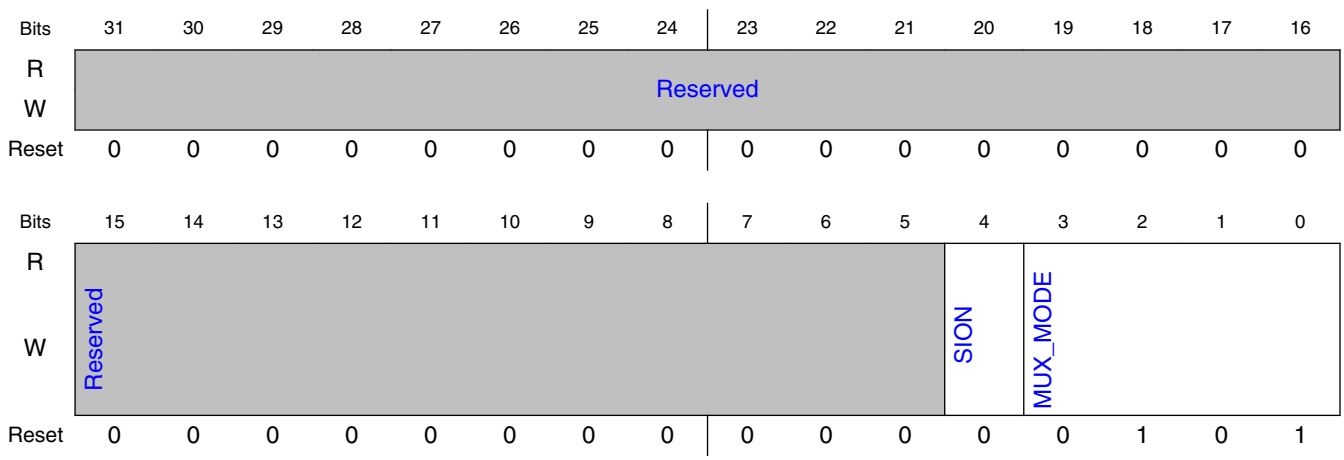
12.4.6.112.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_05	1C8h

12.4.6.112.2 Function

SW_MUX_CTL Register

12.4.6.112.3 Diagram



12.4.6.112.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_05.

Field	Function
	0000b - Select mux mode: ALT0 mux port: USDHC2_CMD of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_DQS of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_DATA03 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: FLEXSPI1_B_SS0_B of instance: FLEXSPI1 0100b - Select mux mode: ALT4 mux port: LPSPI4_PCS2 of instance: LPSPI4 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO14 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO14 of instance: GPIO10

12.4.6.113 SW_MUX_CTL_PAD_GPIO_SD_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_06)

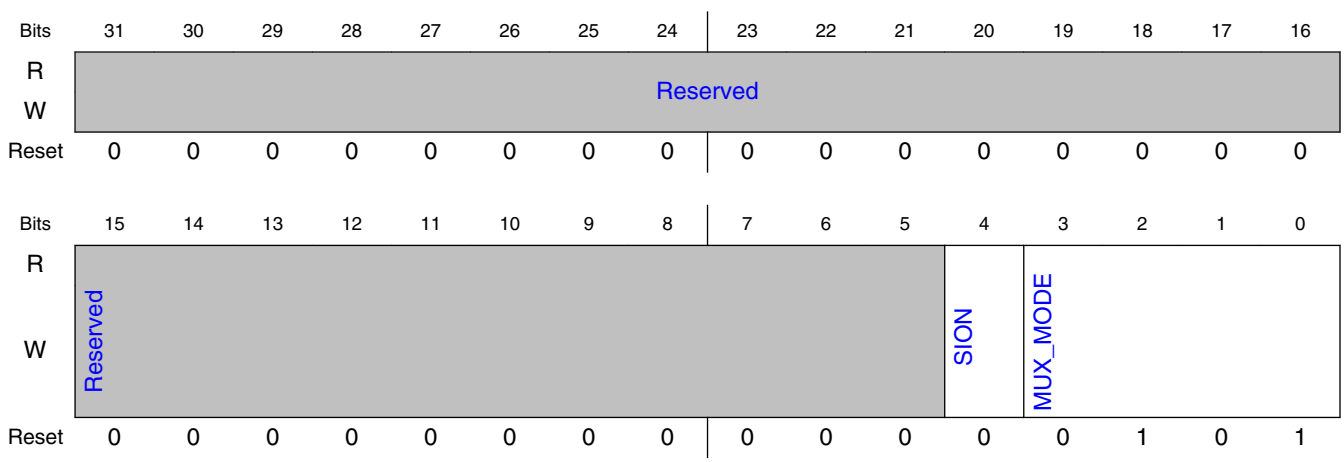
12.4.6.113.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_06	1CCh

12.4.6.113.2 Function

SW_MUX_CTL Register

12.4.6.113.3 Diagram



12.4.6.113.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_SD_B2_06. 0000b - Select mux mode: ALT0 mux port: USDHC2_RESET_B of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_SS0_B of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA03 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPSPI4_PCS3 of instance: LPSPI4 0100b - Select mux mode: ALT4 mux port: GPT6_CAPTURE1 of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO15 of instance: GPIO_MUX4 1010b - Select mux mode: ALT10 mux port: GPIO10_IO15 of instance: GPIO10

12.4.6.114 SW_MUX_CTL_PAD_GPIO_SD_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_07)

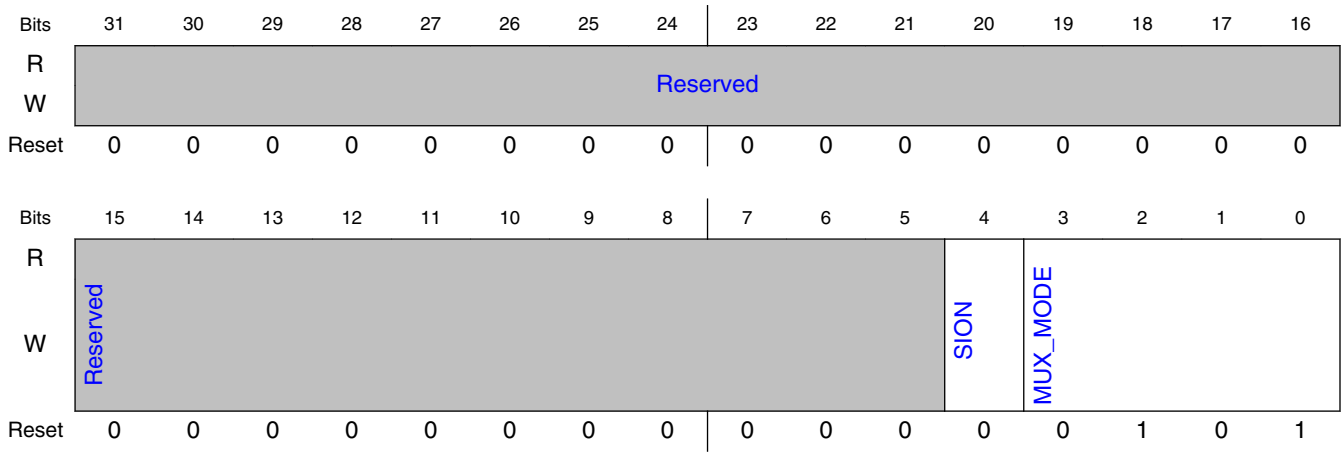
12.4.6.114.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_07	1D0h

12.4.6.114.2 Function

SW_MUX_CTL Register

12.4.6.114.3 Diagram



12.4.6.114.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_SD_B2_07. 0000b - Select mux mode: ALT0 mux port: USDHC2_STROBE of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_SCLK of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA02 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART3_CTS_B of instance: LPUART3 0100b - Select mux mode: ALT4 mux port: GPT6_CAPTURE2 of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO16 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: LPSPI2_SCK of instance: LPSPI2 1000b - Select mux mode: ALT8 mux port: ENET_TX_ER of instance: ENET 1001b - Select mux mode: ALT9 mux port: ENET_QOS_REF_CLK of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO10_IO16 of instance: GPIO10

12.4.6.115 SW_MUX_CTL_PAD_GPIO_SD_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_08)

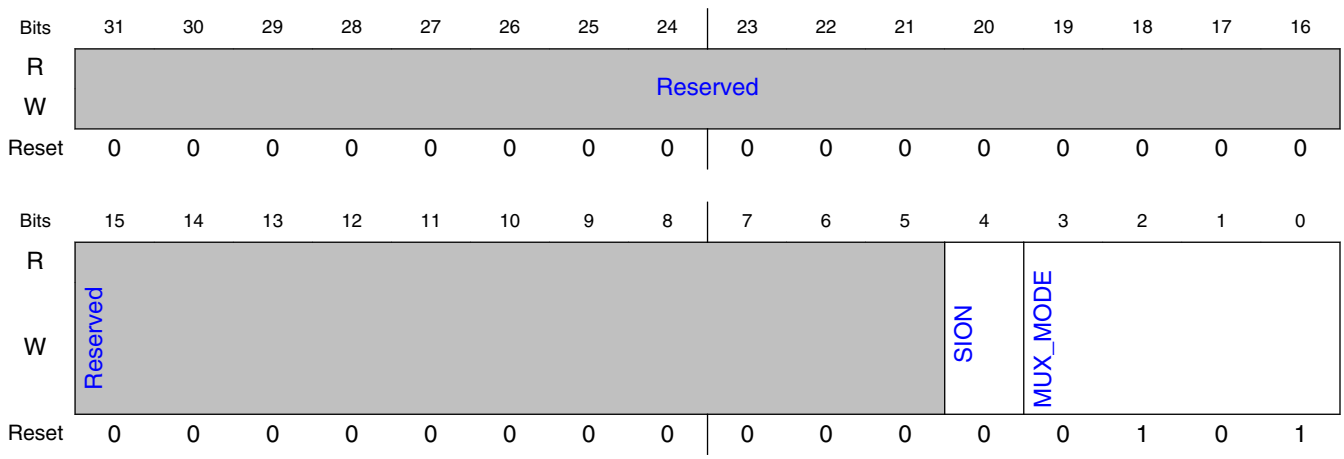
12.4.6.115.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_08	1D4h

12.4.6.115.2 Function

SW_MUX_CTL Register

12.4.6.115.3 Diagram



12.4.6.115.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B2_08. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA4 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA00 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA01 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART3_RTS_B of instance: LPUART3 0100b - Select mux mode: ALT4 mux port: GPT6_COMPARE1 of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO17 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: LPSPI2_PCS0 of instance: LPSPI2 1010b - Select mux mode: ALT10 mux port: GPIO10_IO17 of instance: GPIO10

12.4.6.116 SW_MUX_CTL_PAD_GPIO_SD_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_09)

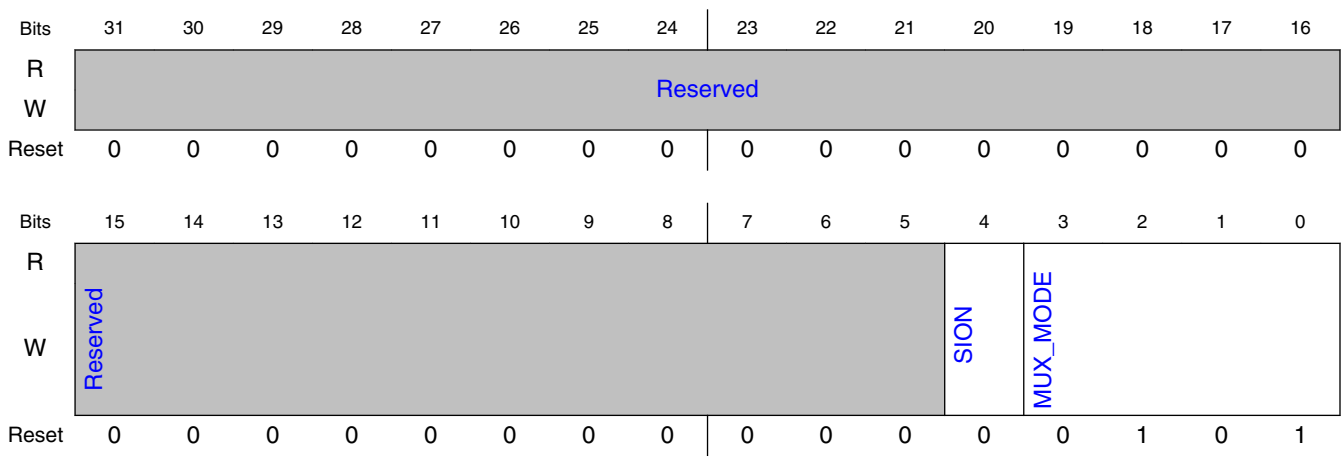
12.4.6.116.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_09	1D8h

12.4.6.116.2 Function

SW_MUX_CTL Register

12.4.6.116.3 Diagram



12.4.6.116.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B2_09.

Field	Function
	0000b - Select mux mode: ALT0 mux port: USDHC2_DATA5 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA01 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_DATA00 of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART5_CTS_B of instance: LPUART5 0100b - Select mux mode: ALT4 mux port: GPT6_COMPARE2 of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO18 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: LPSPI2_SOUT of instance: LPSPI2 1010b - Select mux mode: ALT10 mux port: GPIO10_IO18 of instance: GPIO10

12.4.6.117 SW_MUX_CTL_PAD_GPIO_SD_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_10)

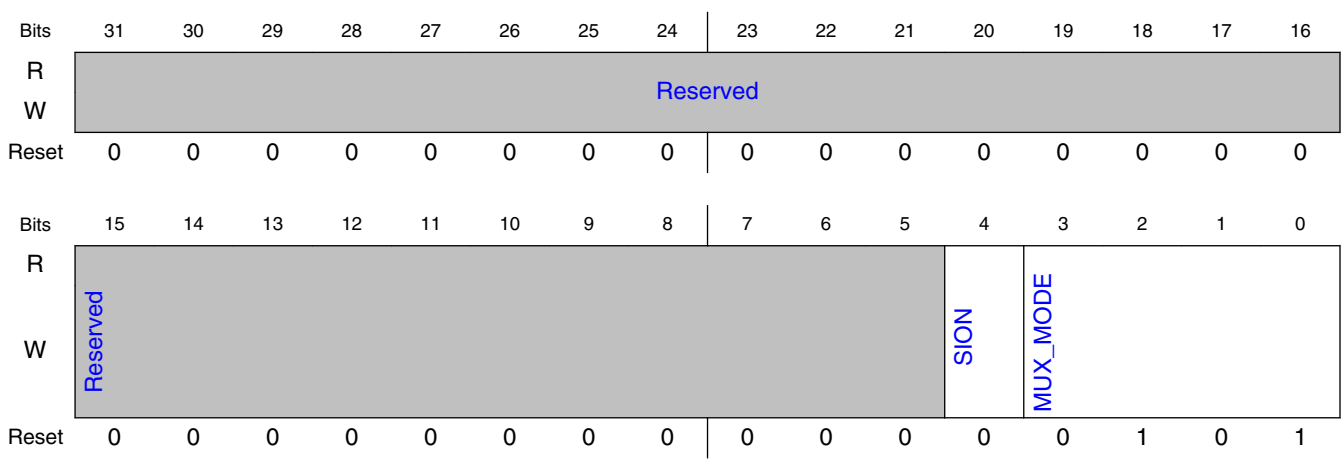
12.4.6.117.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_10	1DCh

12.4.6.117.2 Function

SW_MUX_CTL Register

12.4.6.117.3 Diagram



12.4.6.117.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B2_10. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA6 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA02 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_EN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: LPUART5_RTS_B of instance: LPUART5 0100b - Select mux mode: ALT4 mux port: GPT6_COMPARE3 of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO19 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: LPSPI2_SIN of instance: LPSPI2 1010b - Select mux mode: ALT10 mux port: GPIO10_IO19 of instance: GPIO10

12.4.6.118 SW_MUX_CTL_PAD_GPIO_SD_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_SD_B2_11)

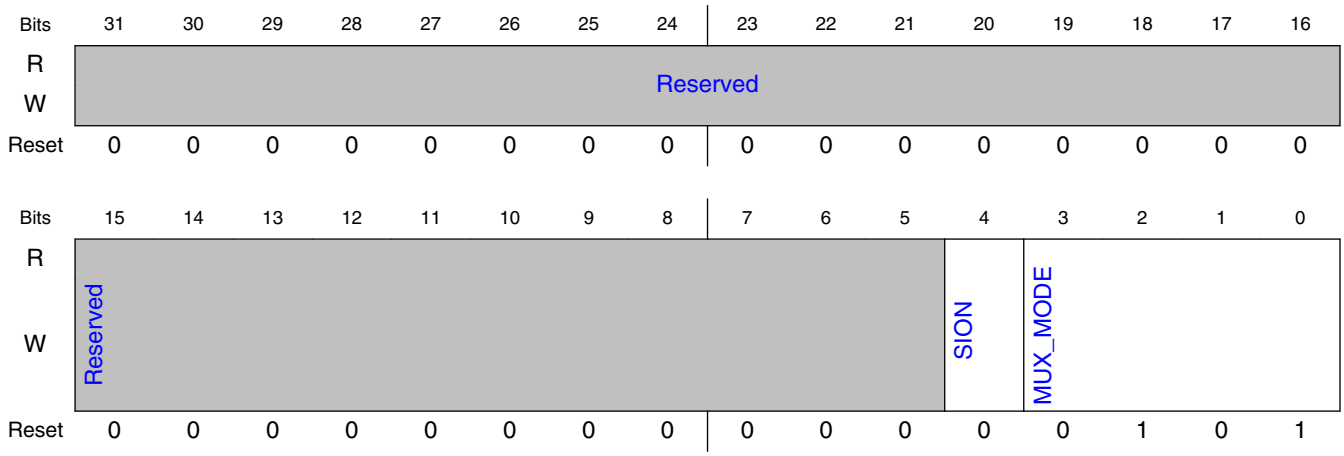
12.4.6.118.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_SD_B2_11	1E0h

12.4.6.118.2 Function

SW_MUX_CTL Register

12.4.6.118.3 Diagram



12.4.6.118.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_SD_B2_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_SD_B2_11. 0000b - Select mux mode: ALT0 mux port: USDHC2_DATA7 of instance: USDHC2 0001b - Select mux mode: ALT1 mux port: FLEXSPI1_A_DATA03 of instance: FLEXSPI1 0010b - Select mux mode: ALT2 mux port: ENET_1G_TX_CLK_IO of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: ENET_1G_REF_CLK of instance: ENET_1G 0100b - Select mux mode: ALT4 mux port: GPT6_CLK of instance: GPT6 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO20 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: LPSPI2_PCS1 of instance: LPSPI2 1010b - Select mux mode: ALT10 mux port: GPIO10_IO20 of instance: GPIO10

12.4.6.119 SW_MUX_CTL_PAD_GPIO_DISP_B1_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_00)

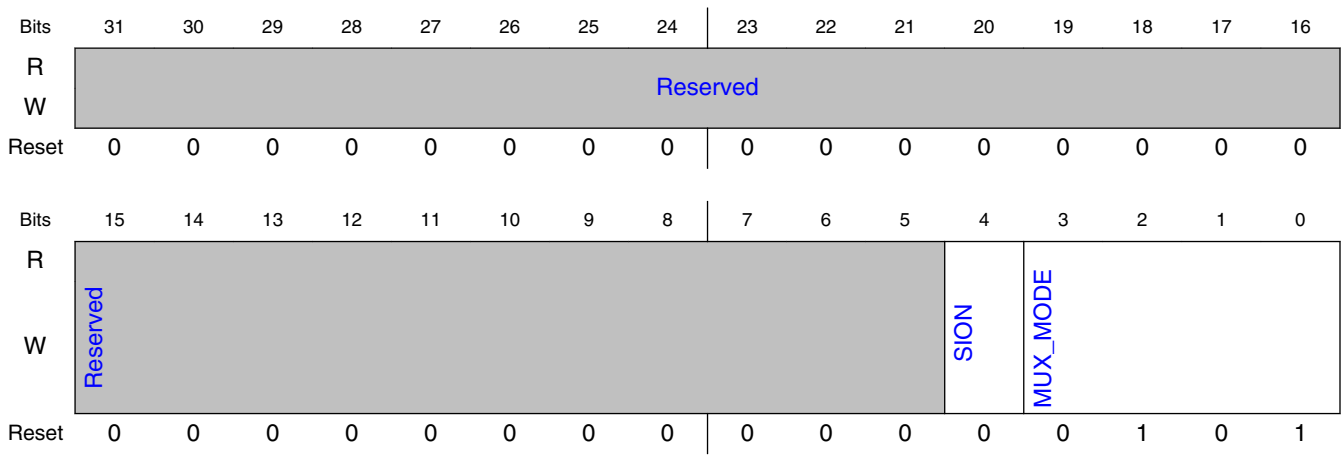
12.4.6.119.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_00	1E4h

12.4.6.119.2 Function

SW_MUX_CTL Register

12.4.6.119.3 Diagram



12.4.6.119.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_DISP_B1_00. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_CLK of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_EN of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: TMR1_TIMER0 of instance: TMR1 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT26 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO21 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_EN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO10_IO21 of instance: GPIO10

12.4.6.120 SW_MUX_CTL_PAD_GPIO_DISP_B1_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_01)

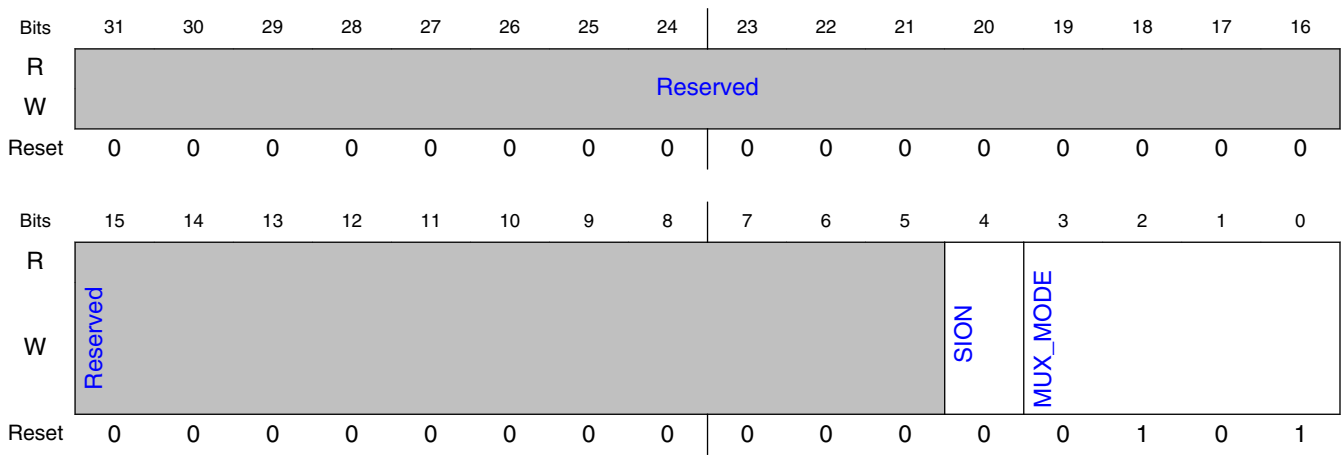
12.4.6.120.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_01	1E8h

12.4.6.120.2 Function

SW_MUX_CTL Register

12.4.6.120.3 Diagram



12.4.6.120.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_01. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_ENABLE of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_CLK of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: ENET_1G_RX_ER of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: TMR1_TIMER1 of instance: TMR1 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT27 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO22 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_CLK of instance: ENET_QOS

Field	Function
	1001b - Select mux mode: ALT9 mux port: ENET_QOS_RX_ER of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO10_IO22 of instance: GPIO10

12.4.6.121 SW_MUX_CTL_PAD_GPIO_DISP_B1_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_02)

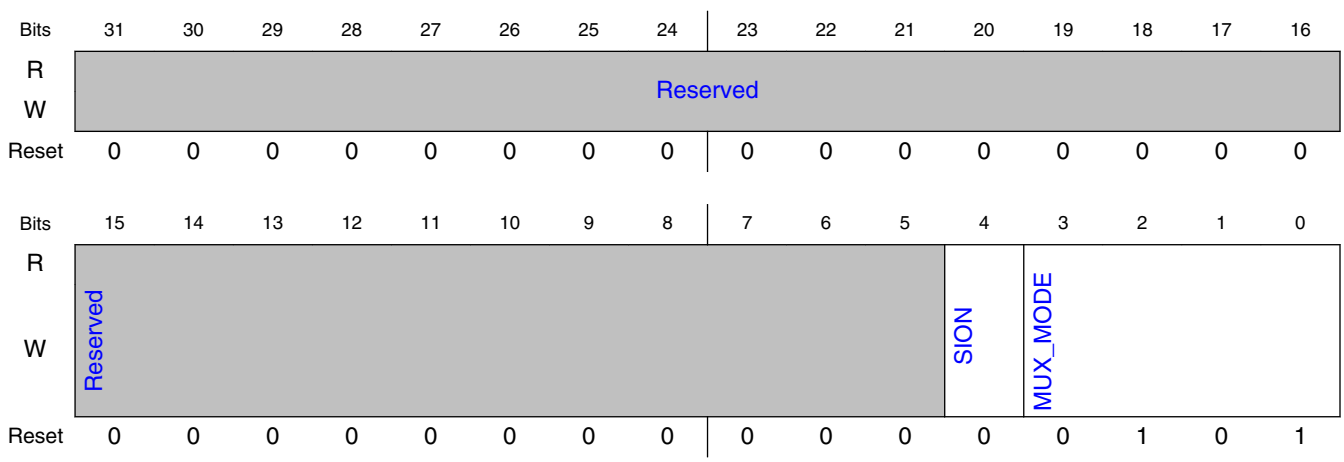
12.4.6.121.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_02	1ECh

12.4.6.121.2 Function

SW_MUX_CTL Register

12.4.6.121.3 Diagram



12.4.6.121.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_02. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_HSYNC of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_DATA00 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPI2C3_SCL of instance: LPI2C3 0011b - Select mux mode: ALT3 mux port: TMR1_TIMER2 of instance: TMR1 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT28 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO23 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA00 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPUART1_TXD of instance: LPUART1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO23 of instance: GPIO10

12.4.6.122 SW_MUX_CTL_PAD_GPIO_DISP_B1_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_03)

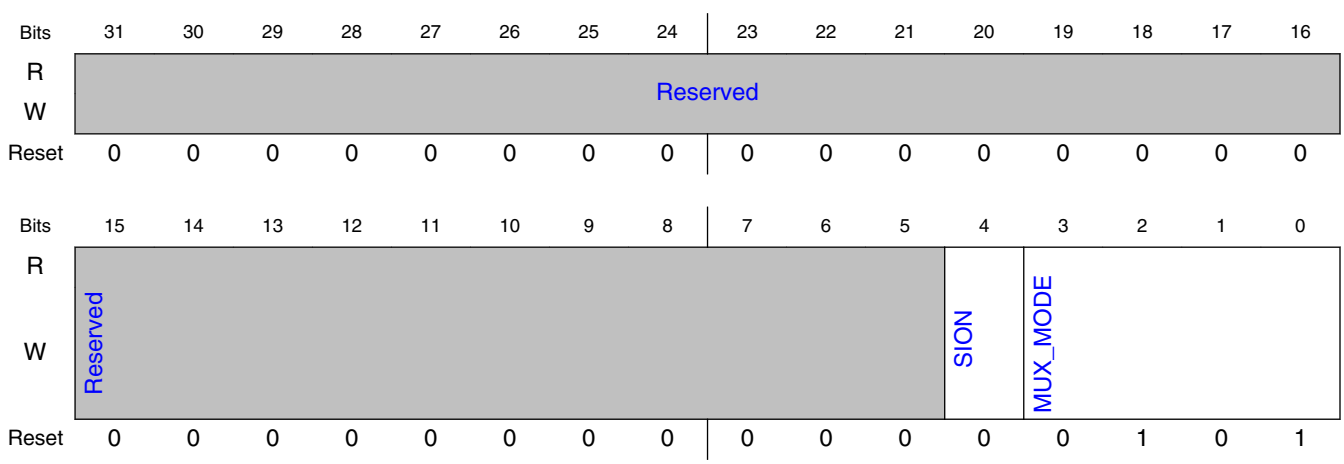
12.4.6.122.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_03	1F0h

12.4.6.122.2 Function

SW_MUX_CTL Register

12.4.6.122.3 Diagram



12.4.6.122.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_03. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_VSYNC of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_DATA01 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPI2C3_SDA of instance: LPI2C3 0011b - Select mux mode: ALT3 mux port: TMR2_TIMER0 of instance: TMR2 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT29 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO24 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA01 of instance: ENET_QOS 1001b - Select mux mode: LPUART1_RXD of instance: LPUART1 1010b - Select mux mode: ALT10 mux port: GPIO10_IO24 of instance: GPIO10

12.4.6.123 SW_MUX_CTL_PAD_GPIO_DISP_B1_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_04)

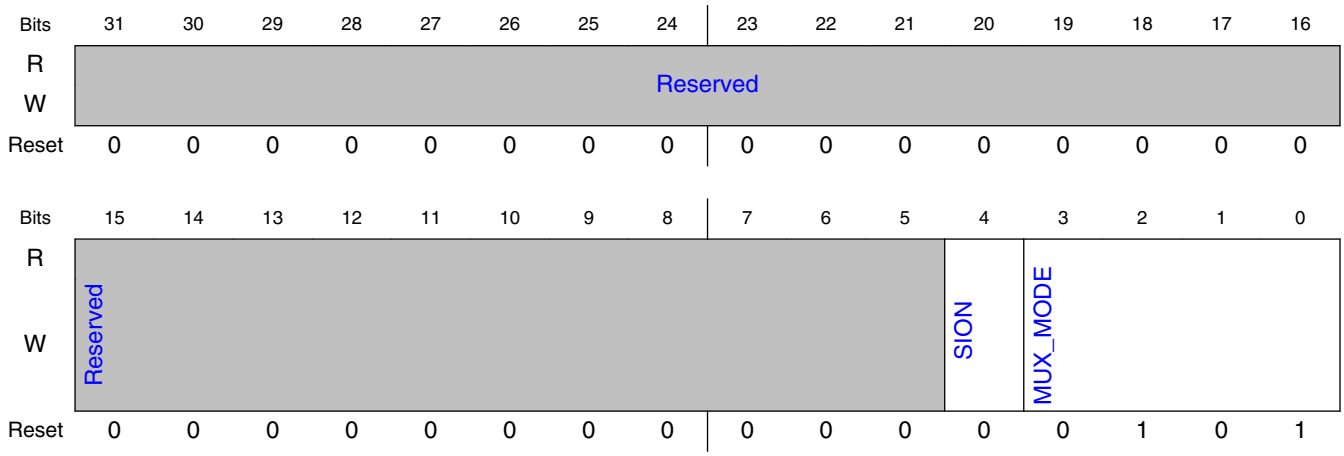
12.4.6.123.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_04	1F4h

12.4.6.123.2 Function

SW_MUX_CTL Register

12.4.6.123.3 Diagram



12.4.6.123.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_04. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA00 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_DATA02 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPUART4_RXD of instance: LPUART4 0011b - Select mux mode: ALT3 mux port: TMR2_TIMER1 of instance: TMR2 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT30 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO25 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA02 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_SCK of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO25 of instance: GPIO10

12.4.6.124 SW_MUX_CTL_PAD_GPIO_DISP_B1_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_05)

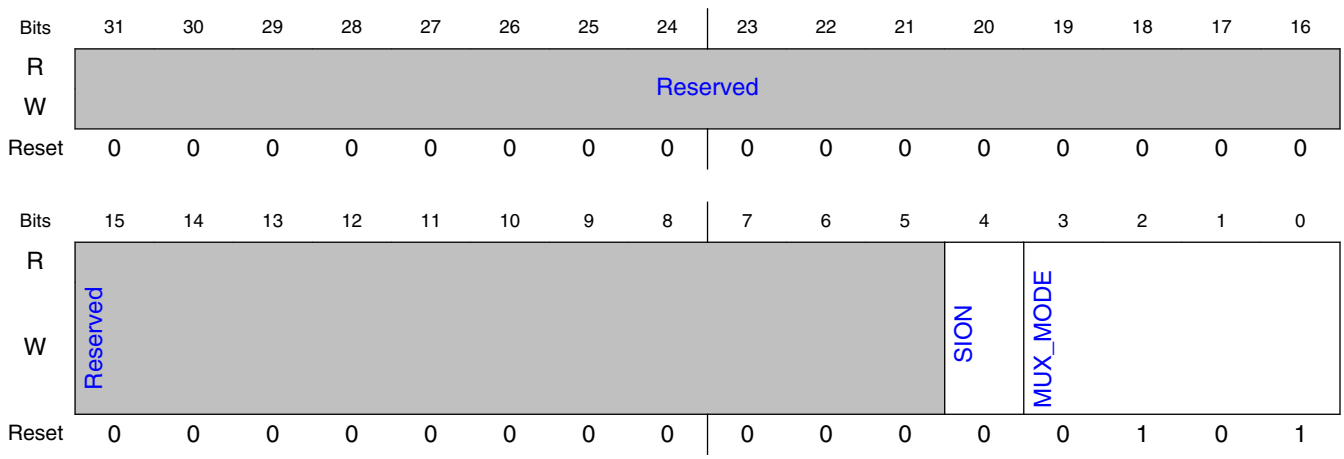
12.4.6.124.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_05	1F8h

12.4.6.124.2 Function

SW_MUX_CTL Register

12.4.6.124.3 Diagram



12.4.6.124.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_05. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA01 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_RX_DATA03 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPUART4_CTS_B of instance: LPUART4 0011b - Select mux mode: ALT3 mux port: TMR2_TIMER2 of instance: TMR2 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT31 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO26 of instance: GPIO_MUX4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA03 of instance: ENET_QOS

Field	Function
	1001b - Select mux mode: ALT9 mux port: LPSPI3_SIN of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO26 of instance: GPIO10

12.4.6.125 SW_MUX_CTL_PAD_GPIO_DISP_B1_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_06)

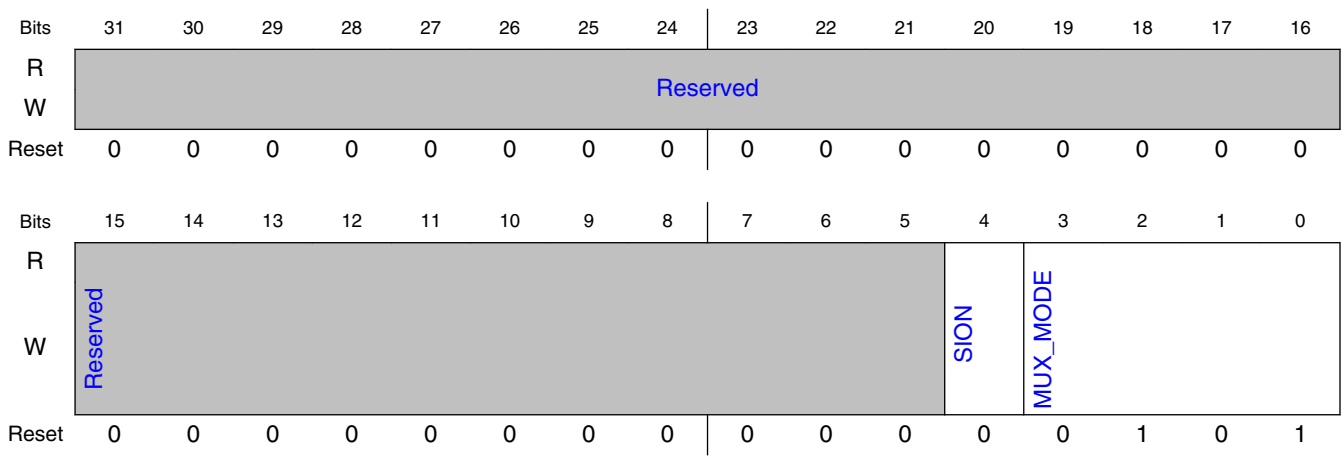
12.4.6.125.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_06	1FCh

12.4.6.125.2 Function

SW_MUX_CTL Register

12.4.6.125.3 Diagram



12.4.6.125.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_06. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA02 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_DATA03 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPUART4_TXD of instance: LPUART4 0011b - Select mux mode: ALT3 mux port: TMR3_TIMER0 of instance: TMR3 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT32 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO27 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG00 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA03 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_SOUT of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO27 of instance: GPIO10

12.4.6.126 SW_MUX_CTL_PAD_GPIO_DISP_B1_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_07)

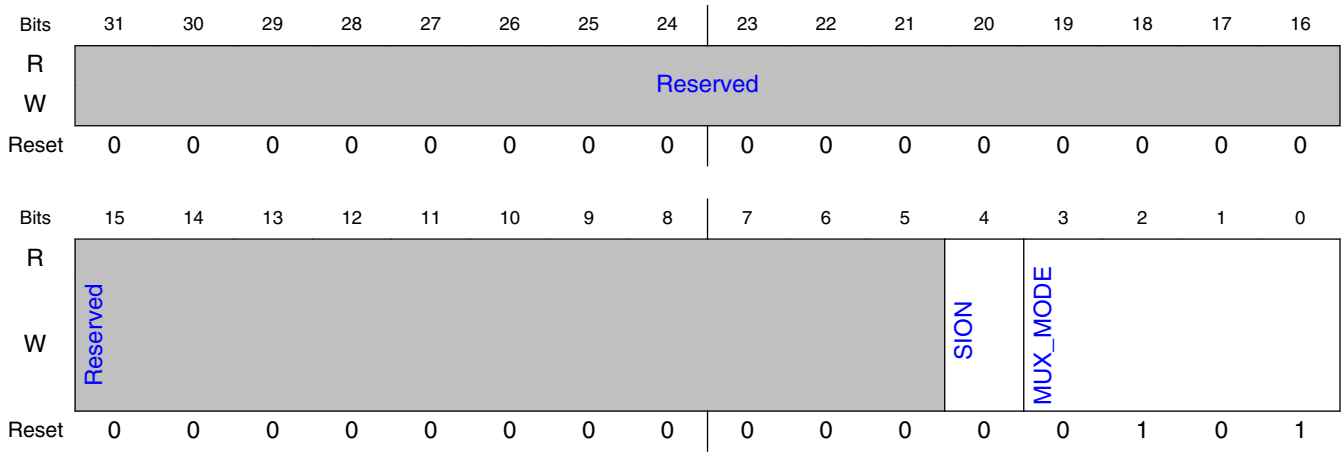
12.4.6.126.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_07	200h

12.4.6.126.2 Function

SW_MUX_CTL Register

12.4.6.126.3 Diagram



12.4.6.126.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_07. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA03 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_DATA02 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: LPUART4_RTS_B of instance: LPUART4 0011b - Select mux mode: ALT3 mux port: TMR3_TIMER1 of instance: TMR3 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT33 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO28 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG01 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA02 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_PCS0 of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO28 of instance: GPIO10

12.4.6.127 SW_MUX_CTL_PAD_GPIO_DISP_B1_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_08)

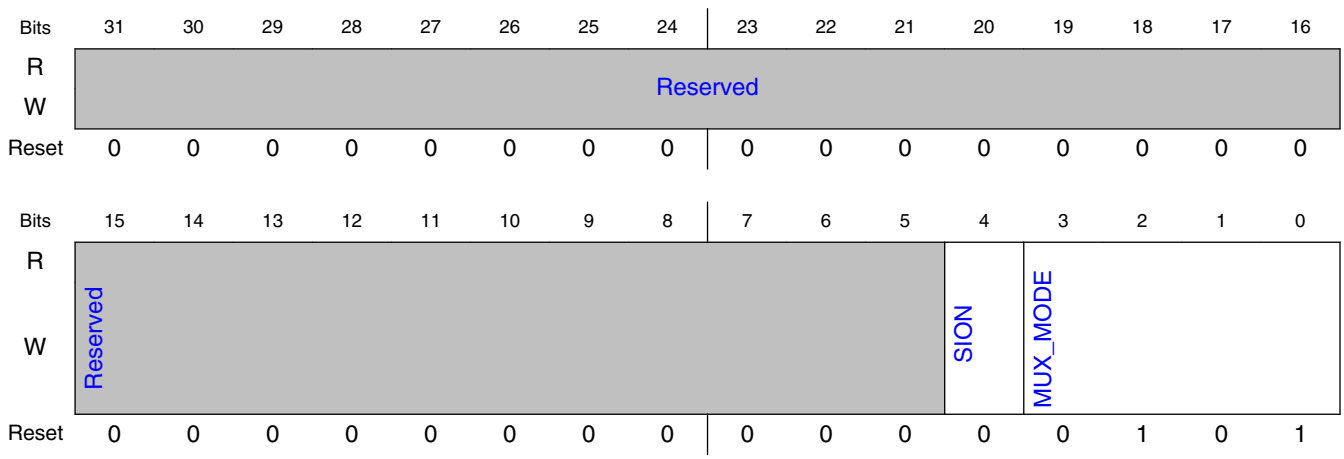
12.4.6.127.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_08	204h

12.4.6.127.2 Function

SW_MUX_CTL Register

12.4.6.127.3 Diagram



12.4.6.127.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_08. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA04 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_DATA01 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: USDHC1_CD_B of instance: USDHC1 0011b - Select mux mode: ALT3 mux port: TMR3_TIMER2 of instance: TMR3 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT34 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO29 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG02 of instance: SRC

Field	Function
	1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA01 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_PCS1 of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO29 of instance: GPIO10

12.4.6.128 SW_MUX_CTL_PAD_GPIO_DISP_B1_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_09)

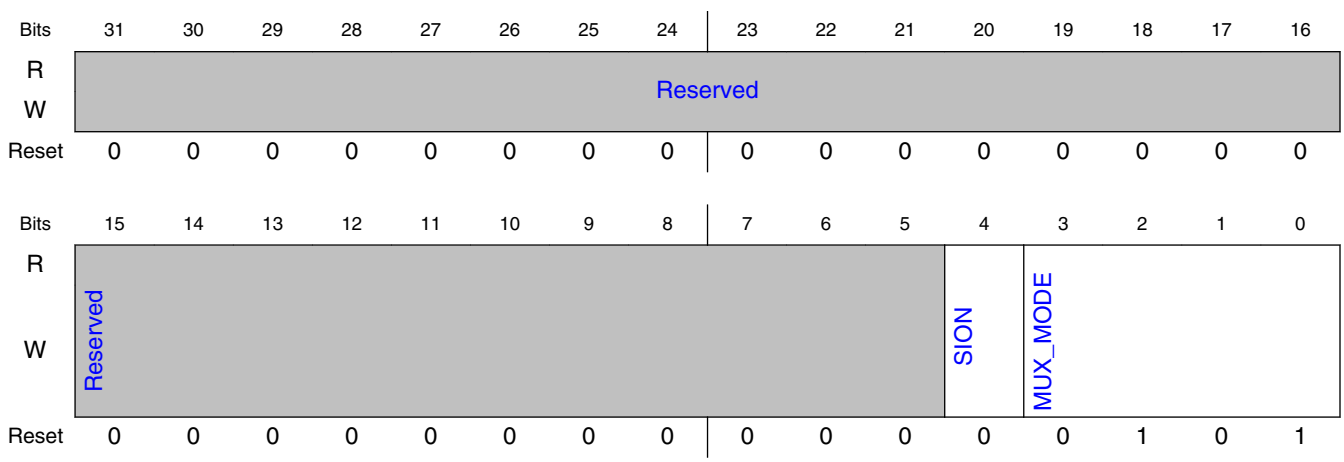
12.4.6.128.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_09	208h

12.4.6.128.2 Function

SW_MUX_CTL Register

12.4.6.128.3 Diagram



12.4.6.128.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_09. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA05 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_DATA00 of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: USDHC1_WP of instance: USDHC1 0011b - Select mux mode: ALT3 mux port: TMR4_TIMER0 of instance: TMR4 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT35 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO30 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG03 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA00 of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_PCS2 of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO30 of instance: GPIO10

12.4.6.129 SW_MUX_CTL_PAD_GPIO_DISP_B1_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_10)

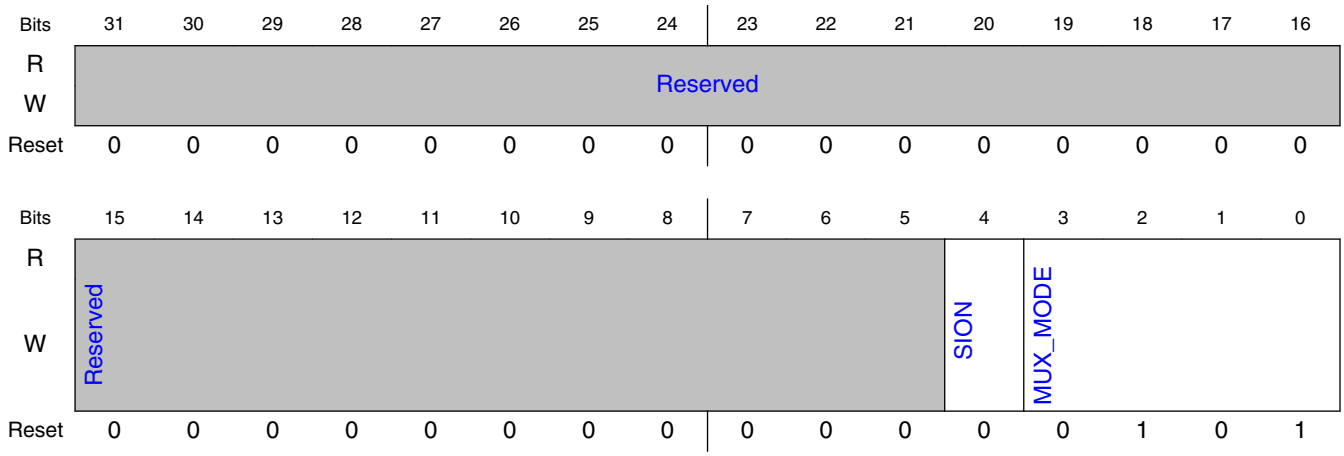
12.4.6.129.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_10	20Ch

12.4.6.129.2 Function

SW_MUX_CTL Register

12.4.6.129.3 Diagram



12.4.6.129.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_10. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA06 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_EN of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: USDHC1_RESET_B of instance: USDHC1 0011b - Select mux mode: ALT3 mux port: TMR4_TIMER1 of instance: TMR4 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT36 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX4_IO31 of instance: GPIO_MUX4 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG04 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_EN of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI3_PCS3 of instance: LPSPI3 1010b - Select mux mode: ALT10 mux port: GPIO10_IO31 of instance: GPIO10

12.4.6.130 SW_MUX_CTL_PAD_GPIO_DISP_B1_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B1_11)

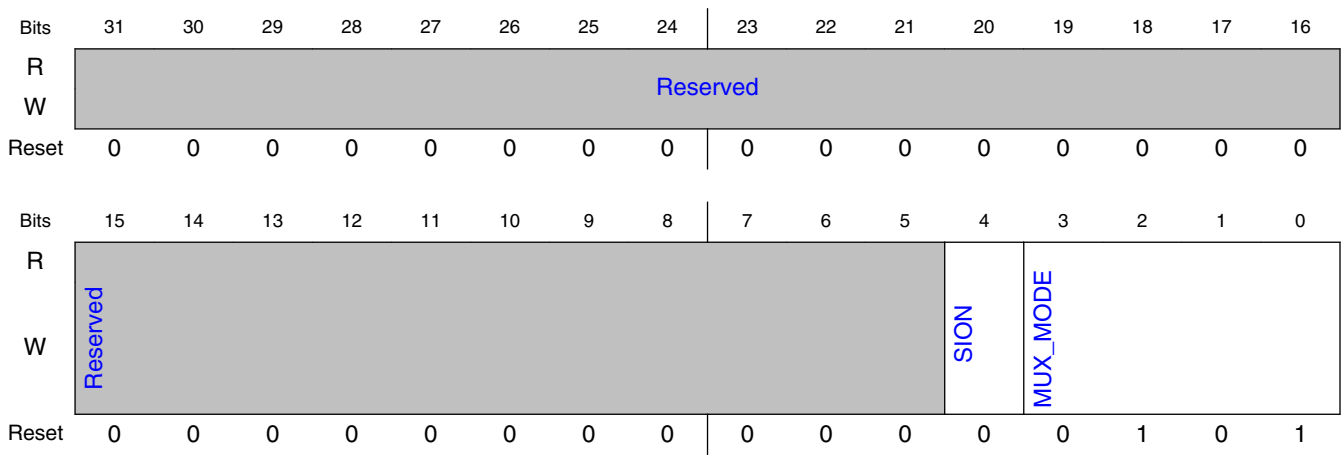
12.4.6.130.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B1_11	210h

12.4.6.130.2 Function

SW_MUX_CTL Register

12.4.6.130.3 Diagram



12.4.6.130.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B1_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B1_11. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA07 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_1G_TX_CLK_IO of instance: ENET_1G 0010b - Select mux mode: ALT2 mux port: ENET_1G_REF_CLK of instance: ENET_1G 0011b - Select mux mode: ALT3 mux port: TMR4_TIMER2 of instance: TMR4 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT37 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO00 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG05 of instance: SRC

Field	Function
	1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_CLK of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: ENET_QOS_REF_CLK of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO00 of instance: GPIO11

12.4.6.131 SW_MUX_CTL_PAD_GPIO_DISP_B2_00 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_00)

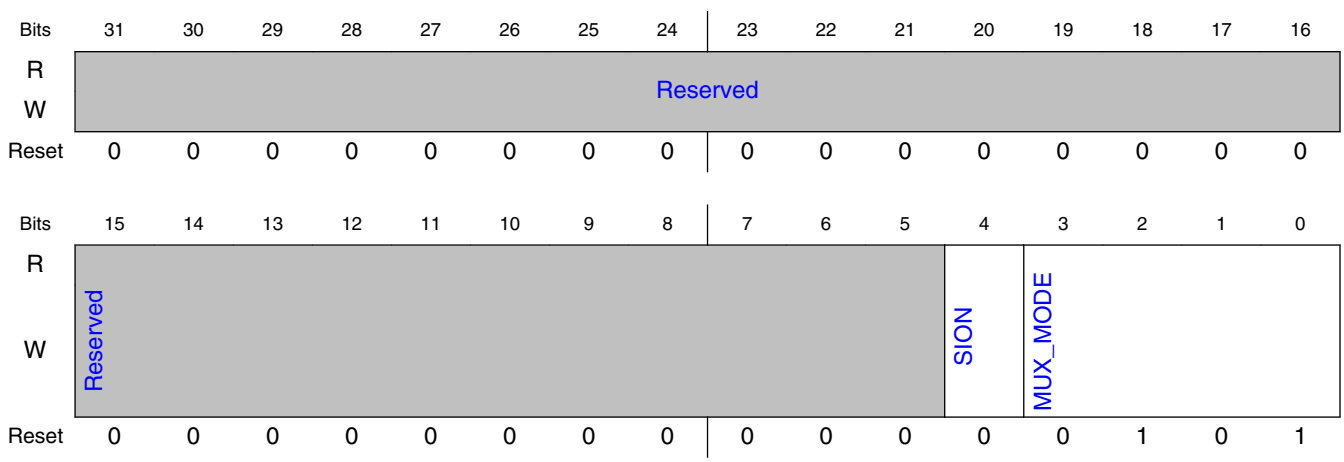
12.4.6.131.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_00	214h

12.4.6.131.2 Function

SW_MUX_CTL Register

12.4.6.131.3 Diagram



12.4.6.131.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_00
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_DISP_B2_00. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA08 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: WDOG1_B of instance: WDOG1 0010b - Select mux mode: ALT2 mux port: MQS_RIGHT of instance: MQS 0011b - Select mux mode: ALT3 mux port: ENET_1G_TX_ER of instance: ENET_1G 0100b - Select mux mode: ALT4 mux port: SAI1_TX_DATA03 of instance SAI1 as input, and SAI1_RX_DATA01 of instance SAI1 as output 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO01 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG06 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_ER of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO01 of instance: GPIO11

12.4.6.132 SW_MUX_CTL_PAD_GPIO_DISP_B2_01 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_01)

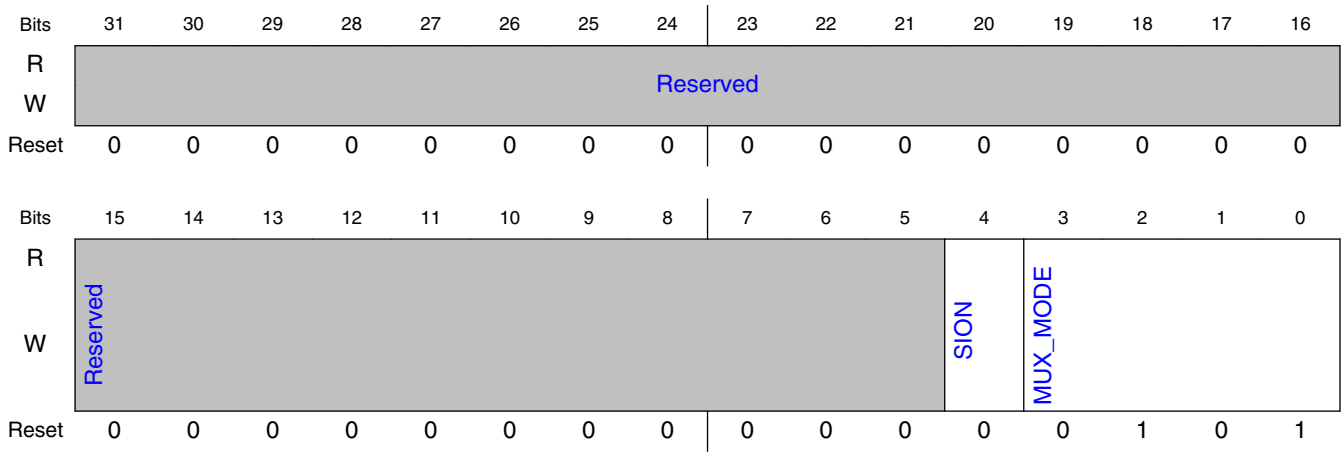
12.4.6.132.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_01	218h

12.4.6.132.2 Function

SW_MUX_CTL Register

12.4.6.132.3 Diagram



12.4.6.132.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_01
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_DISP_B2_01. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA09 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: USDHC1_VSELECT of instance: USDHC1 0010b - Select mux mode: ALT2 mux port: MQS_LEFT of instance: MQS 0011b - Select mux mode: ALT3 mux port: WDOG2_B of instance: WDOG2 0100b - Select mux mode: ALT4 mux port: SAI1_TX_DATA02 of instance SAI1 as input, and SAI1_RX_DATA2 of instance SAI1 as output 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO02 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG07 of instance: SRC 1000b - Select mux mode: ALT8 mux port: EWM_OUT_B of instance: EWM 1001b - Select mux mode: ALT9 mux port: CCM_ENET_REF_CLK_25M of instance: CCM 1010b - Select mux mode: ALT10 mux port: GPIO11_IO02 of instance: GPIO11

12.4.6.133 SW_MUX_CTL_PAD_GPIO_DISP_B2_02 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_02)

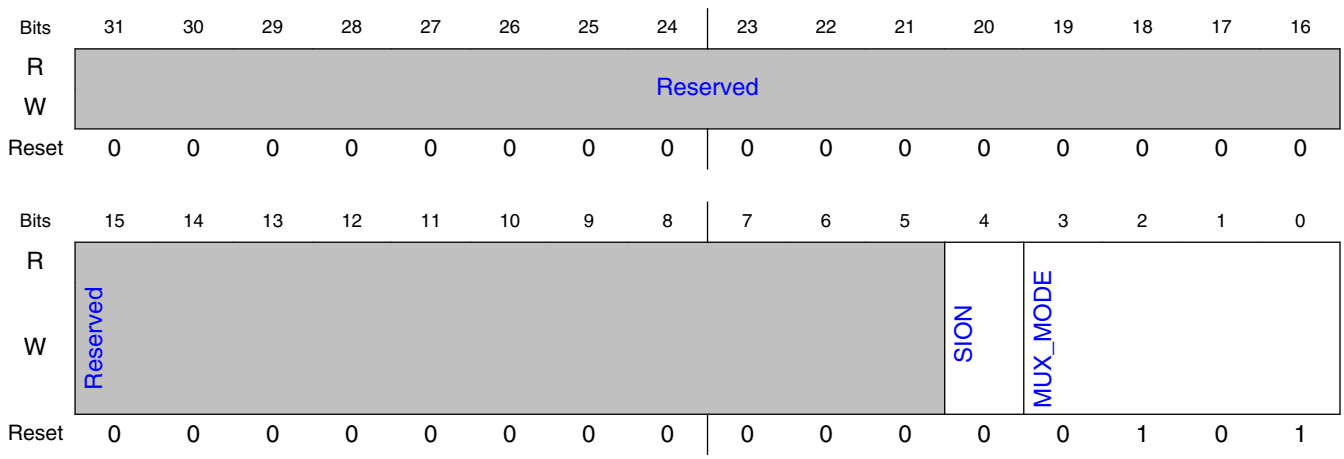
12.4.6.133.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_02	21Ch

12.4.6.133.2 Function

SW_MUX_CTL Register

12.4.6.133.3 Diagram



12.4.6.133.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_02
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_DISP_B2_02. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA10 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_TX_DATA00 of instance: ENET 0010b - Select mux mode: ALT2 mux port: PIT1_TRIGGER3 of instance: PIT1 0011b - Select mux mode: ALT3 mux port: ARM_TRACE00 of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_TX_DATA01 of instance SAI1 as input or SAI1_RX_DATA3 of instance SAI as output 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO03 of instance: GPIO_MUX5

Field	Function
	0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG08 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA00 of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO03 of instance: GPIO11

12.4.6.134 SW_MUX_CTL_PAD_GPIO_DISP_B2_03 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_03)

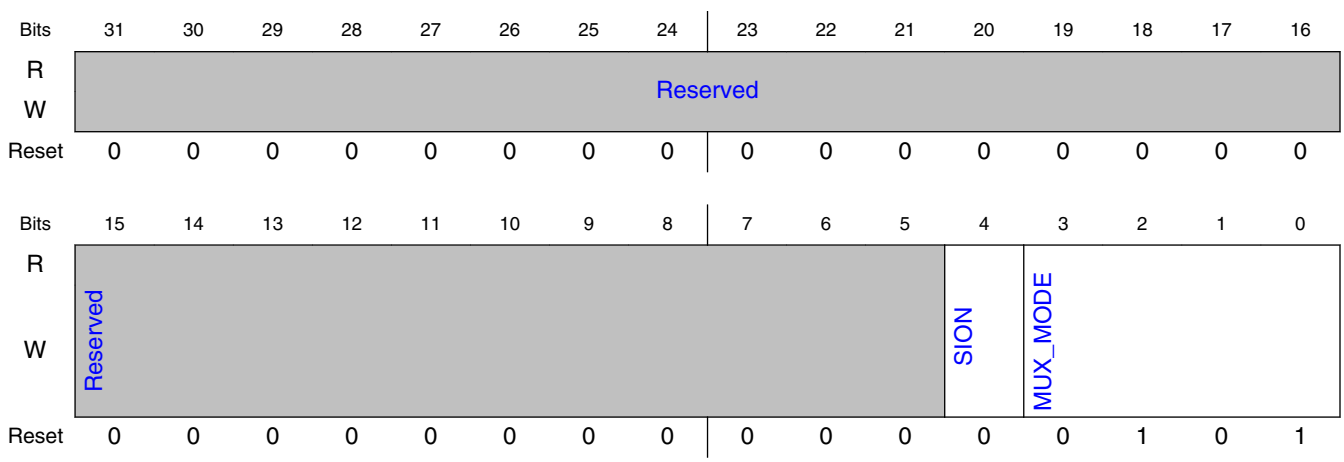
12.4.6.134.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_03	220h

12.4.6.134.2 Function

SW_MUX_CTL Register

12.4.6.134.3 Diagram



12.4.6.134.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_03
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_DISP_B2_03. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA11 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_TX_DATA01 of instance: ENET 0010b - Select mux mode: ALT2 mux port: PIT1_TRIGGER2 of instance: PIT1 0011b - Select mux mode: ALT3 mux port: ARM_TRACE01 of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_MCLK of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO04 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG09 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_DATA01 of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO04 of instance: GPIO11

12.4.6.135 SW_MUX_CTL_PAD_GPIO_DISP_B2_04 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_04)

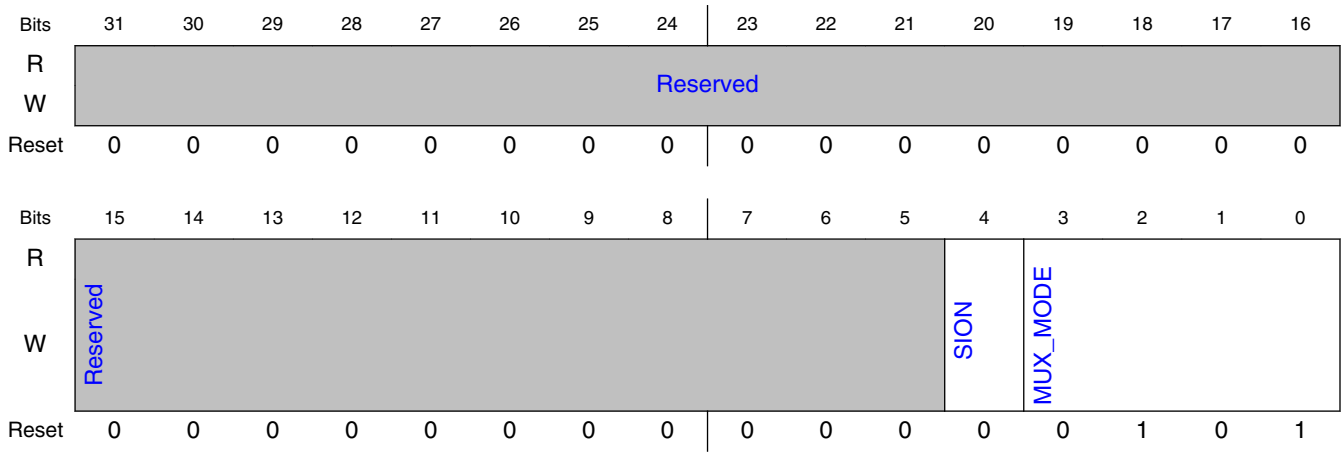
12.4.6.135.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_04	224h

12.4.6.135.2 Function

SW_MUX_CTL Register

12.4.6.135.3 Diagram



12.4.6.135.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_04
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_DISP_B2_04. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA12 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_TX_EN of instance: ENET 0010b - Select mux mode: ALT2 mux port: PIT1_TRIGGER1 of instance: PIT1 0011b - Select mux mode: ALT3 mux port: ARM_TRACE02 of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_RX_SYNC of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO05 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG10 of instance: SRC 1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_EN of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO05 of instance: GPIO11

12.4.6.136 SW_MUX_CTL_PAD_GPIO_DISP_B2_05 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_05)

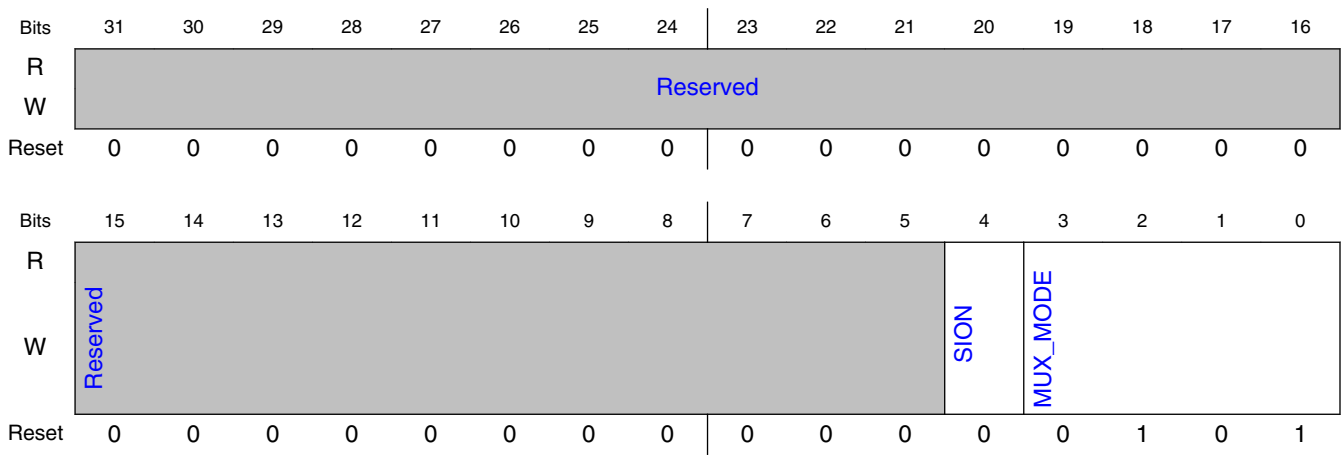
12.4.6.136.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_05	228h

12.4.6.136.2 Function

SW_MUX_CTL Register

12.4.6.136.3 Diagram



12.4.6.136.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_05
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 10 iomux modes to be used for pad: GPIO_DISP_B2_05. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA13 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_TX_CLK of instance: ENET 0010b - Select mux mode: ALT2 mux port: ENET_REF_CLK of instance: ENET 0011b - Select mux mode: ALT3 mux port: ARM_TRACE03 of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_RX_BCLK of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO06 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: SRC_BT_CFG11 of instance: SRC

Field	Function
	1000b - Select mux mode: ALT8 mux port: ENET_QOS_TX_CLK of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO06 of instance: GPIO11

12.4.6.137 SW_MUX_CTL_PAD_GPIO_DISP_B2_06 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_06)

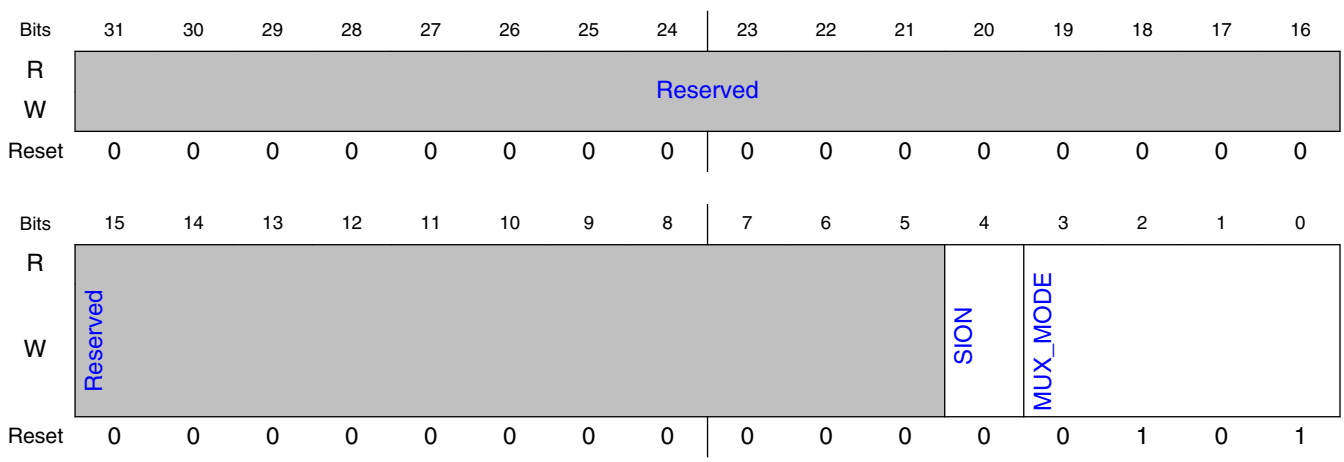
12.4.6.137.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_06	22Ch

12.4.6.137.2 Function

SW_MUX_CTL Register

12.4.6.137.3 Diagram



12.4.6.137.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_06
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_DISP_B2_06. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA14 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_RX_DATA00 of instance: ENET 0010b - Select mux mode: ALT2 mux port: LPUART7_TXD of instance: LPUART7 0011b - Select mux mode: ALT3 mux port: ARM_TRACE_CLK of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_RX_DATA00 of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO07 of instance: GPIO_MUX5 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA00 of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO07 of instance: GPIO11

12.4.6.138 SW_MUX_CTL_PAD_GPIO_DISP_B2_07 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_07)

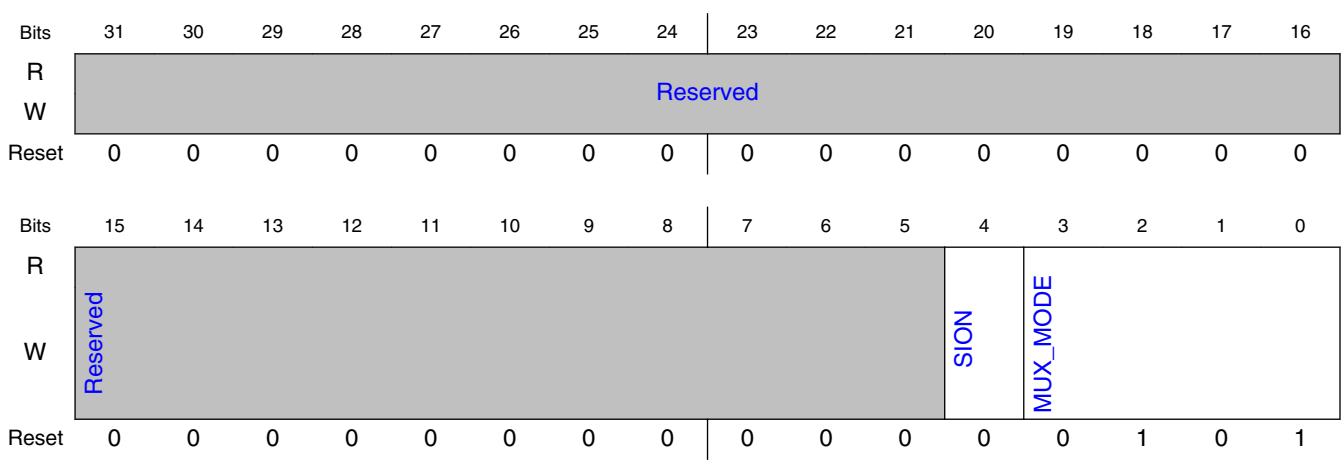
12.4.6.138.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_07	230h

12.4.6.138.2 Function

SW_MUX_CTL Register

12.4.6.138.3 Diagram



12.4.6.138.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_07
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO_DISP_B2_07. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA15 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_RX_DATA01 of instance: ENET 0010b - Select mux mode: ALT2 mux port: LPUART7_RXD of instance: LPUART7 0011b - Select mux mode: ALT3 mux port: ARM_TRACE_SWO of instance: ARM 0100b - Select mux mode: ALT4 mux port: SAI1_TX_DATA00 of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO08 of instance: GPIO_MUX5 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_DATA01 of instance: ENET_QOS 1010b - Select mux mode: ALT10 mux port: GPIO11_IO08 of instance: GPIO11

12.4.6.139 SW_MUX_CTL_PAD_GPIO_DISP_B2_08 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_08)

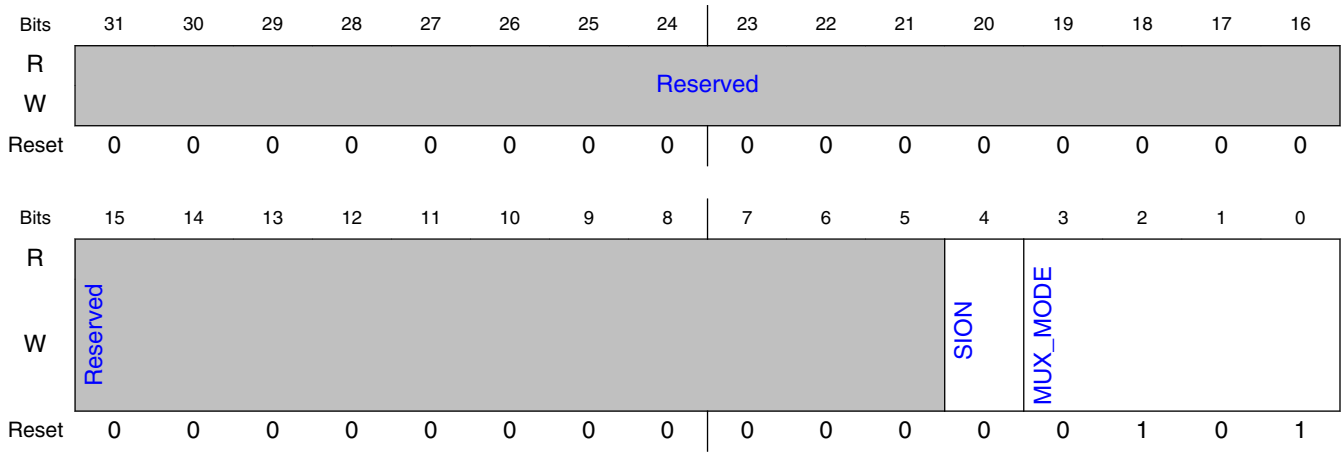
12.4.6.139.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_08	234h

12.4.6.139.2 Function

SW_MUX_CTL Register

12.4.6.139.3 Diagram



12.4.6.139.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_08
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_DISP_B2_08. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA16 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_RX_EN of instance: ENET 0010b - Select mux mode: ALT2 mux port: LPUART8_TXD of instance: LPUART8 0011b - Select mux mode: ALT3 mux port: ARM_CM7_EVENTO of instance: CM7 0100b - Select mux mode: ALT4 mux port: SAI1_TX_BCLK of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO09 of instance: GPIO_MUX5 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_EN of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPUART1_TXD of instance: LPUART1 1010b - Select mux mode: ALT10 mux port: GPIO11_IO09 of instance: GPIO11

12.4.6.140 SW_MUX_CTL_PAD_GPIO_DISP_B2_09 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_09)

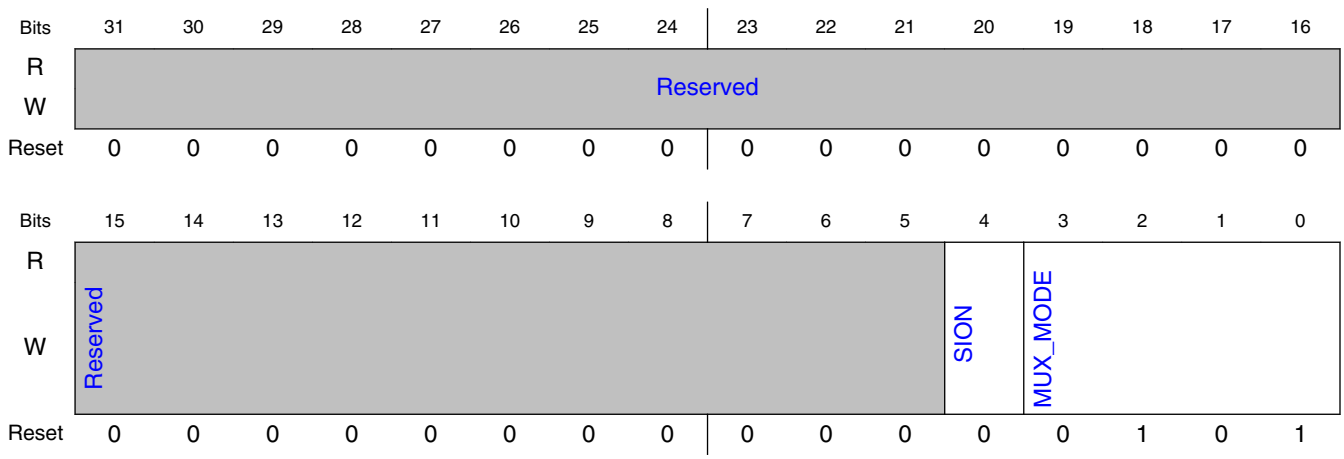
12.4.6.140.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_09	238h

12.4.6.140.2 Function

SW_MUX_CTL Register

12.4.6.140.3 Diagram



12.4.6.140.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_09
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_DISP_B2_09. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA17 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: ENET_RX_ER of instance: ENET 0010b - Select mux mode: ALT2 mux port: LPUART8_RXD of instance: LPUART8 0011b - Select mux mode: ALT3 mux port: ARM_CM7_EVENTI of instance: CM7 0100b - Select mux mode: ALT4 mux port: SAI1_TX_SYNC of instance: SAI1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO10 of instance: GPIO_MUX5 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_ER of instance: ENET_QOS

Field	Function
	1001b - Select mux mode: ALT9 mux port: LPUART1_RXD of instance: LPUART1 1010b - Select mux mode: ALT10 mux port: GPIO11_IO10 of instance: GPIO11

12.4.6.141 SW_MUX_CTL_PAD_GPIO_DISP_B2_10 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_10)

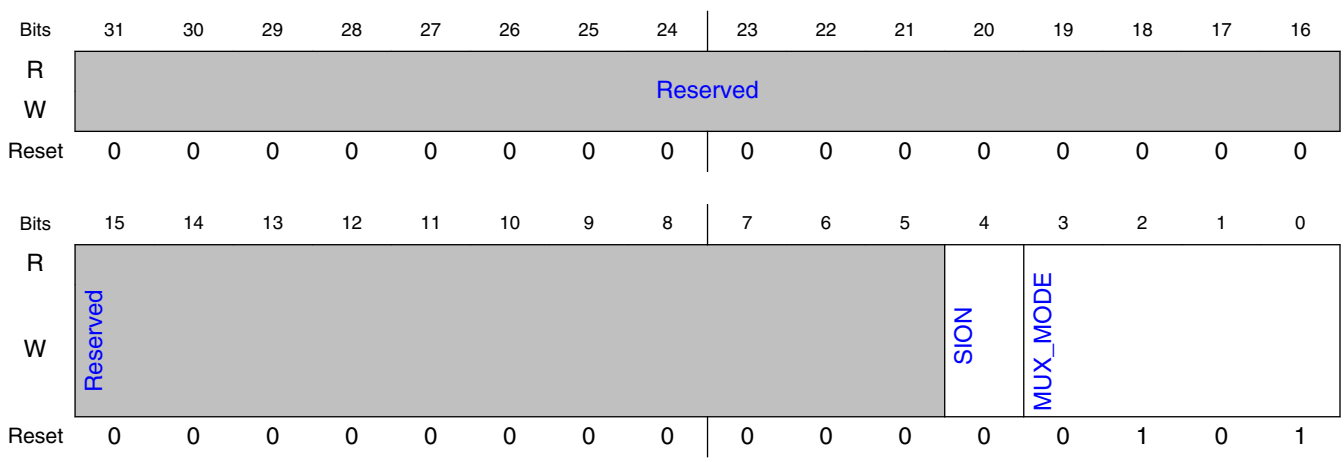
12.4.6.141.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_10	23Ch

12.4.6.141.2 Function

SW_MUX_CTL Register

12.4.6.141.3 Diagram



12.4.6.141.4 Fields

Field	Function
31-5 —	- Reserved
4	Software Input On Field.

Table continues on the next page...

Field	Function
SION	Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_10
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B2_10. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA18 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_IO of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: LPUART2_TXD of instance: LPUART2 0011b - Select mux mode: ALT3 mux port: WDOG2_RESET_B_DEB of instance: WDOG2 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT38 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO11 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: LPI2C3_SCL of instance: LPI2C3 1000b - Select mux mode: ALT8 mux port: ENET_QOS_RX_ER of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: SPDIF_IN of instance: SPDIF 1010b - Select mux mode: ALT10 mux port: GPIO11_IO11 of instance: GPIO11

12.4.6.142 SW_MUX_CTL_PAD_GPIO_DISP_B2_11 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_11)

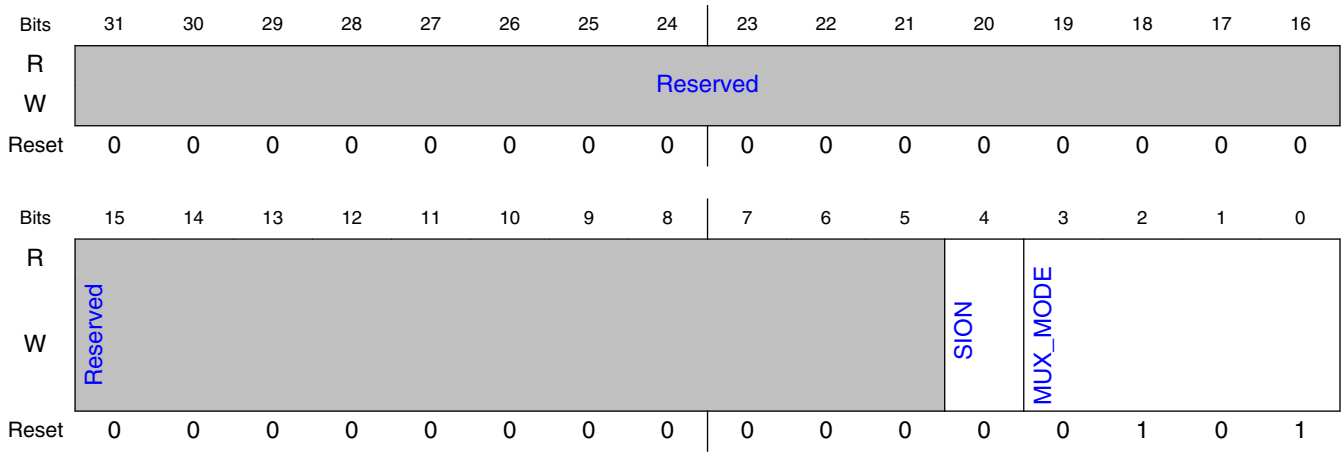
12.4.6.142.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_11	240h

12.4.6.142.2 Function

SW_MUX_CTL Register

12.4.6.142.3 Diagram



12.4.6.142.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_11
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B2_11. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA19 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_CLK of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: LPUART2_RXD of instance: LPUART2 0011b - Select mux mode: ALT3 mux port: WDOG1_RESET_B_DEB of instance: WDOG1 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT39 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO12 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: LPI2C3_SDA of instance: LPI2C3 1000b - Select mux mode: ALT8 mux port: ENET_QOS_CRS of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: SPDIF_OUT of instance: SPDIF 1010b - Select mux mode: ALT10 mux port: GPIO11_IO12 of instance: GPIO11

12.4.6.143 SW_MUX_CTL_PAD_GPIO_DISP_B2_12 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_12)

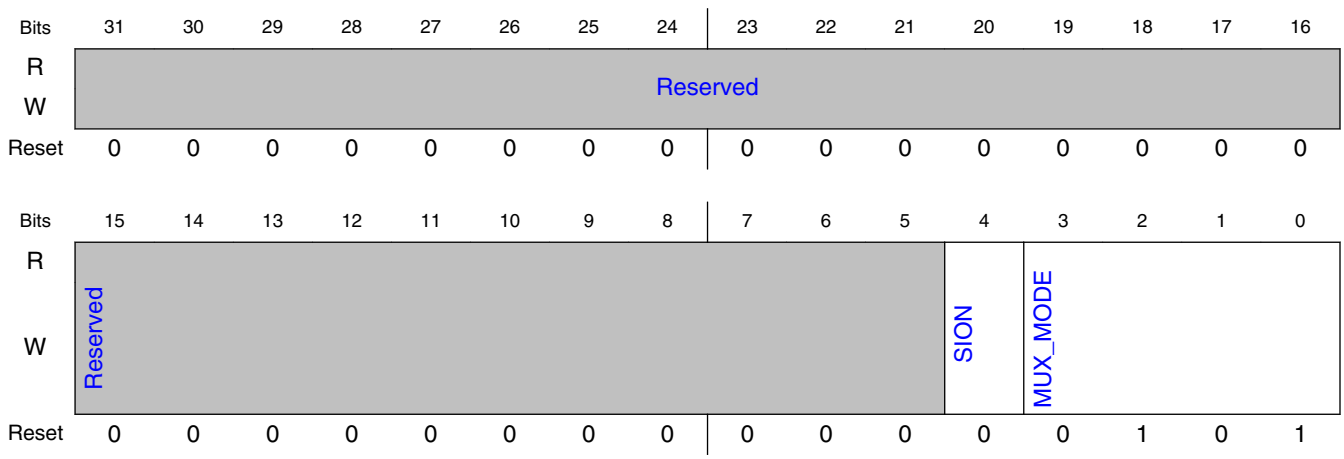
12.4.6.143.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_12	244h

12.4.6.143.2 Function

SW_MUX_CTL Register

12.4.6.143.3 Diagram



12.4.6.143.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_12
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B2_12. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA20 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_RST of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: FLEXCAN1_TX of instance: FLEXCAN1 0011b - Select mux mode: ALT3 mux port: LPUART2_CTS_B of instance: LPUART2 0100b - Select mux mode: ALT4 mux port: XBAR1_INOUT40 of instance: XBAR1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO13 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: LPI2C4_SCL of instance: LPI2C4

Field	Function
	1000b - Select mux mode: ALT8 mux port: ENET_QOS_COL of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI4_SCK of instance: LPSPI4 1010b - Select mux mode: ALT10 mux port: GPIO11_IO13 of instance: GPIO11

12.4.6.144 SW_MUX_CTL_PAD_GPIO_DISP_B2_13 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_13)

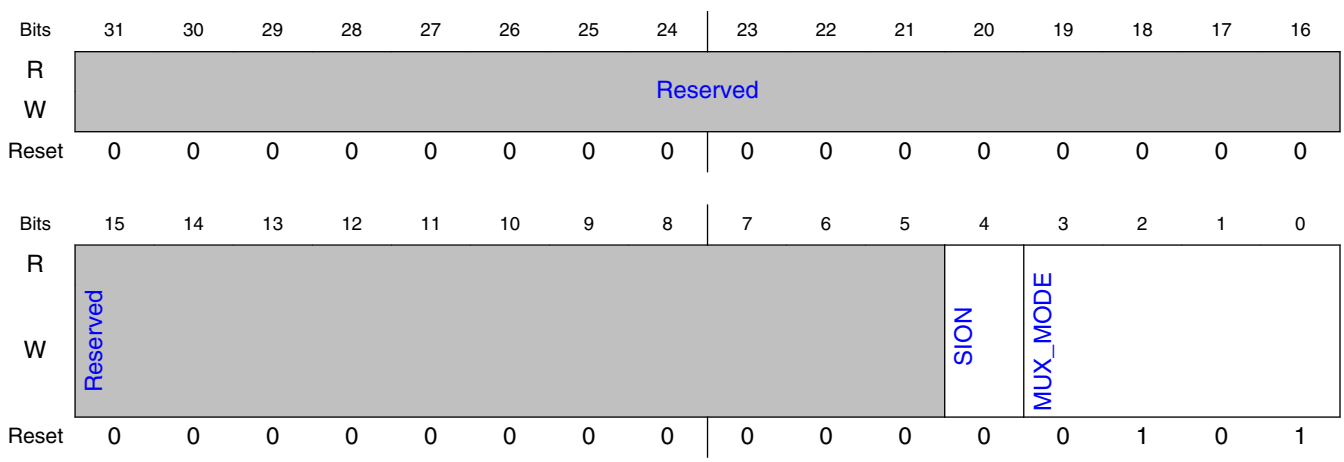
12.4.6.144.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_13	248h

12.4.6.144.2 Function

SW_MUX_CTL Register

12.4.6.144.3 Diagram



12.4.6.144.4 Fields

Field	Function
31-5 —	- Reserved

Table continues on the next page...

Field	Function
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_13
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 12 iomux modes to be used for pad: GPIO_DISP_B2_13. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA21 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_SVEN of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: FLEXCAN1_RX of instance: FLEXCAN1 0011b - Select mux mode: ALT3 mux port: LPUART2_RTS_B of instance: LPUART2 0100b - Select mux mode: ALT4 mux port: ENET_REF_CLK of instance: ENET 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO14 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: LPI2C4_SDA of instance: LPI2C4 1000b - Select mux mode: ALT8 mux port: ENET_QOS_1588_EVENT0_OUT of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI4_SIN of instance: LPSPI4 1010b - Select mux mode: ALT10 mux port: GPIO11_IO14 of instance: GPIO11

12.4.6.145 SW_MUX_CTL_PAD_GPIO_DISP_B2_14 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_14)

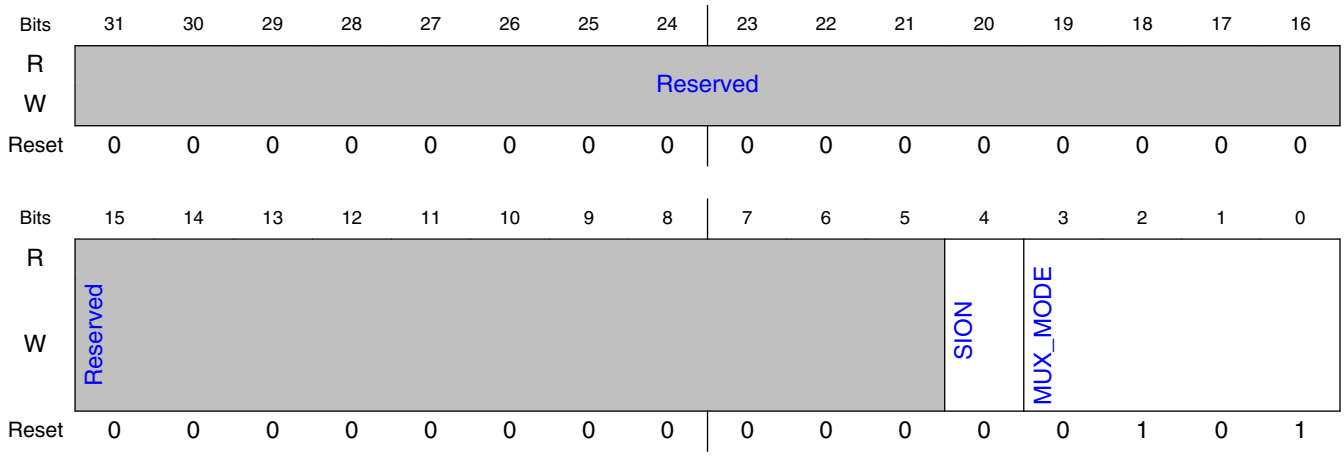
12.4.6.145.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_14	24Ch

12.4.6.145.2 Function

SW_MUX_CTL Register

12.4.6.145.3 Diagram



12.4.6.145.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_14
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_DISP_B2_14. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA22 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_PD of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: WDOG2_B of instance: WDOG2 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_EXT_DCIC1 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: ENET_1G_REF_CLK of instance: ENET_1G 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO15 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: FLEXCAN1_TX of instance: FLEXCAN1 1000b - Select mux mode: ALT8 mux port: ENET_QOS_1588_EVENT0_IN of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI4_SOUT of instance: LPSPI4 1010b - Select mux mode: ALT10 mux port: GPIO11_IO15 of instance: GPIO11

12.4.6.146 SW_MUX_CTL_PAD_GPIO_DISP_B2_15 SW MUX Control Register (SW_MUX_CTL_PAD_GPIO_DISP_B2_15)

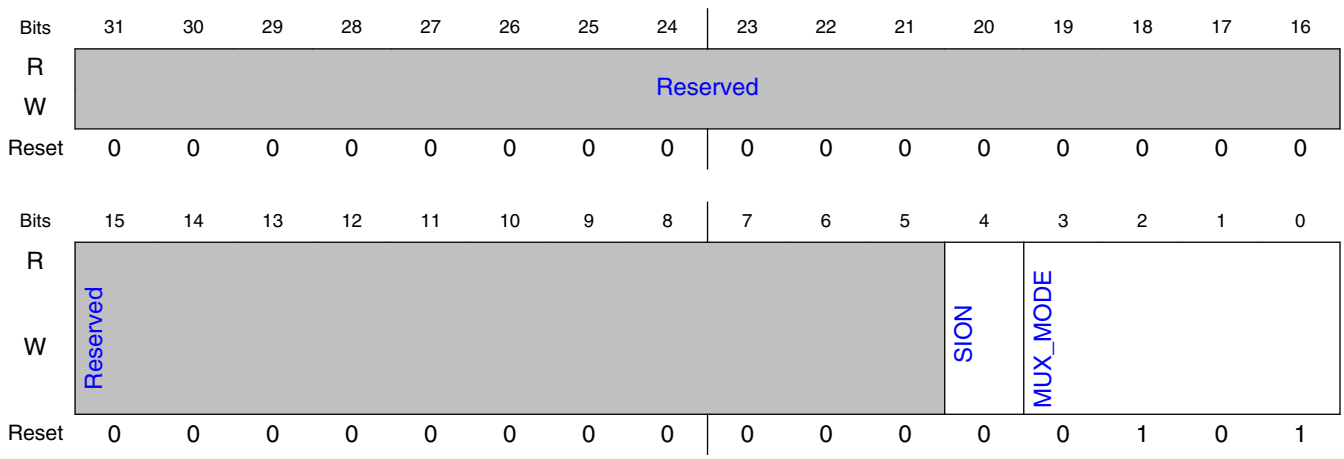
12.4.6.146.1 Offset

Register	Offset
SW_MUX_CTL_PAD_GPIO_DISP_B2_15	250h

12.4.6.146.2 Function

SW_MUX_CTL Register

12.4.6.146.3 Diagram



12.4.6.146.4 Fields

Field	Function
31-5 —	- Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 0b - Input Path is determined by functionality 1b - Force input path of pad GPIO_DISP_B2_15
3-0 MUX_MODE	MUX Mode Select Field. Select 1 of 11 iomux modes to be used for pad: GPIO_DISP_B2_15. 0000b - Select mux mode: ALT0 mux port: VIDEO_MUX_LCDIF_DATA23 of instance: VIDEO_MUX 0001b - Select mux mode: ALT1 mux port: EMVSIM2_POWER_FAIL of instance: EMVSIM2 0010b - Select mux mode: ALT2 mux port: WDOG1_B of instance: WDOG1 0011b - Select mux mode: ALT3 mux port: VIDEO_MUX_EXT_DCIC2 of instance: VIDEO_MUX 0100b - Select mux mode: ALT4 mux port: PIT1_TRIGGER0 of instance: PIT1 0101b - Select mux mode: ALT5 mux port: GPIO_MUX5_IO16 of instance: GPIO_MUX5 0110b - Select mux mode: ALT6 mux port: FLEXCAN1_RX of instance: FLEXCAN1

Field	Function
	1000b - Select mux mode: ALT8 mux port: ENET_QOS_1588_EVENT0_AUX_IN of instance: ENET_QOS 1001b - Select mux mode: ALT9 mux port: LPSPI4_PCS0 of instance: LPSPI4 1010b - Select mux mode: ALT10 mux port: GPIO11_IO16 of instance: GPIO11

12.4.6.147 SW_PAD_CTL_PAD_GPIO_EMC_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_00)

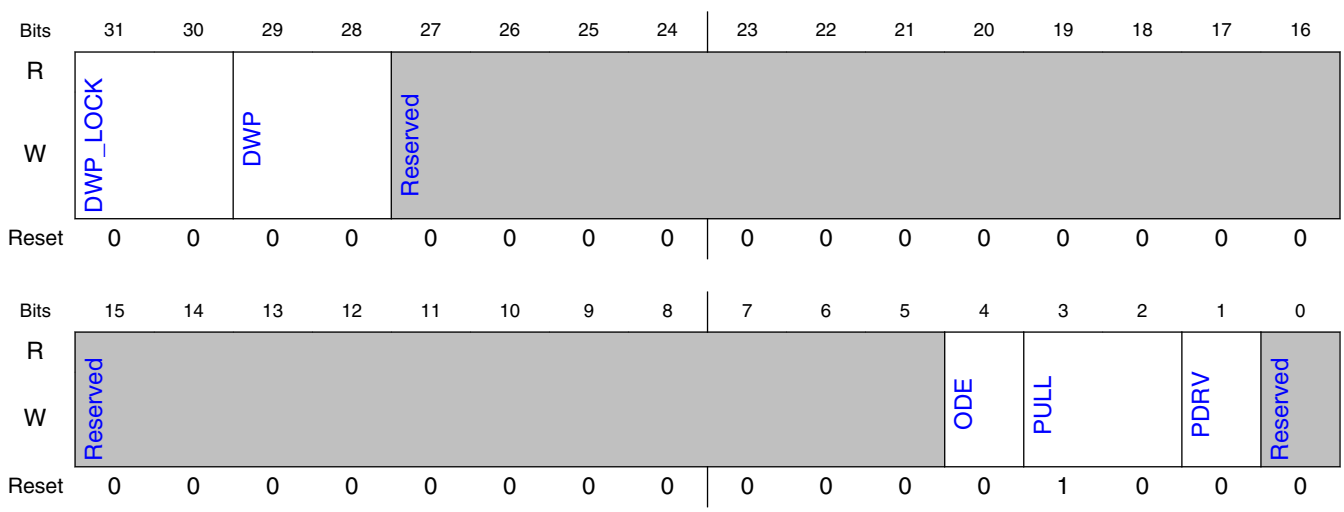
12.4.6.147.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_00	254h

12.4.6.147.2 Function

SW_PAD_CTL Register

12.4.6.147.3 Diagram



12.4.6.147.4 Fields

Field	Function
31-30	Domain write protection lock

Table continues on the next page...

Field	Function
DWP_LOCK	Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_00 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_00 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_00 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.148 SW_PAD_CTL_PAD_GPIO_EMC_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_01)

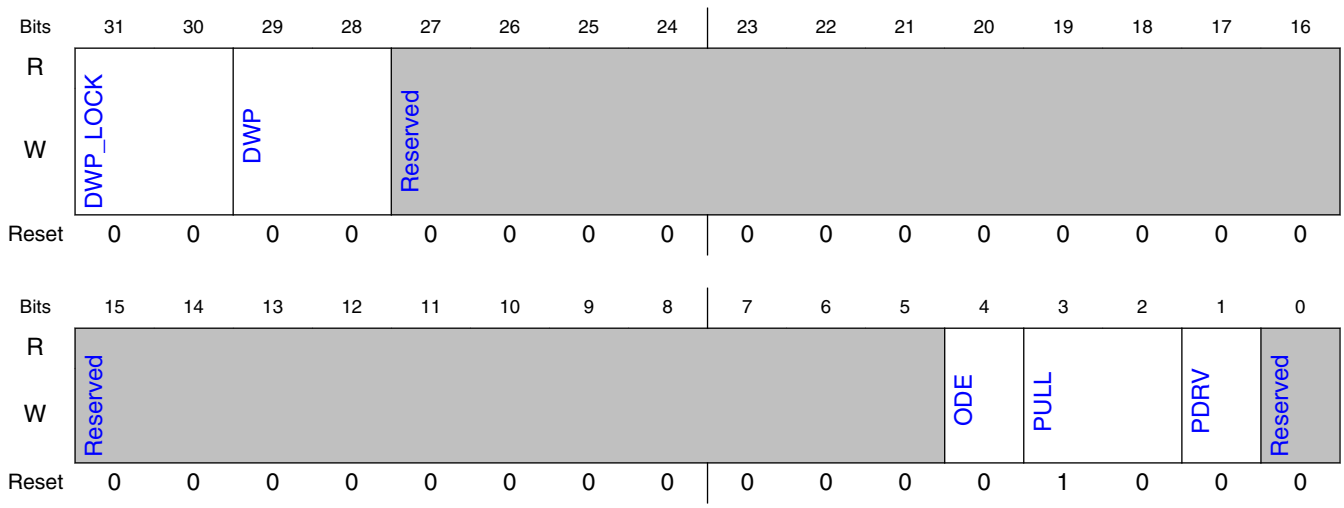
12.4.6.148.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_01	258h

12.4.6.148.2 Function

SW_PAD_CTL Register

12.4.6.148.3 Diagram



12.4.6.148.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_01

Table continues on the next page...

Field	Function
	0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_01 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_01 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.149 SW_PAD_CTL_PAD_GPIO_EMC_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_02)

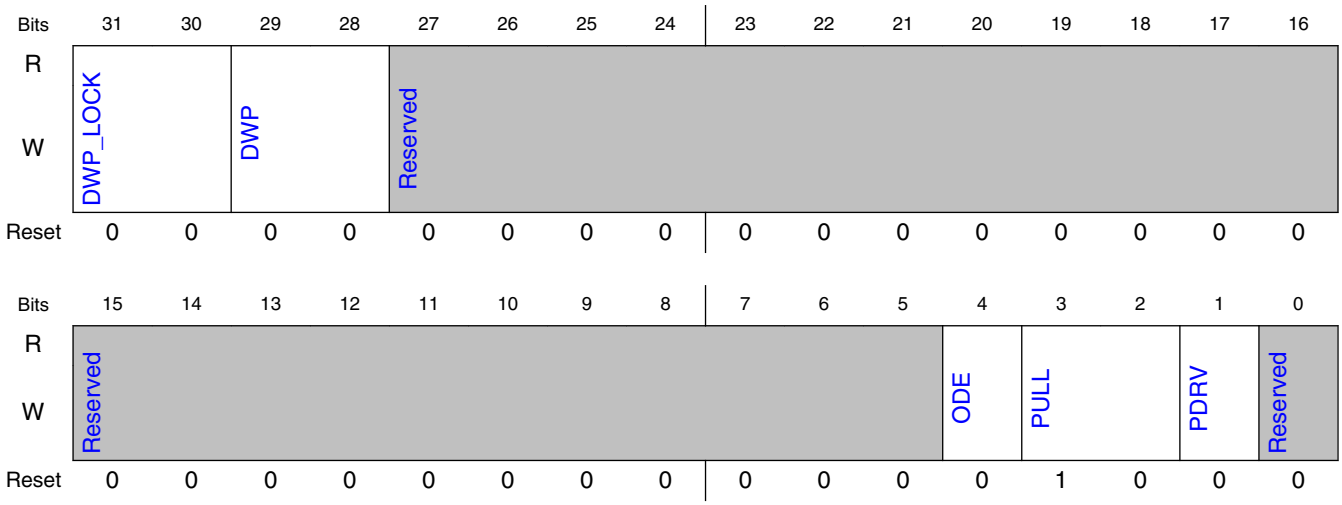
12.4.6.149.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_02	25Ch

12.4.6.149.2 Function

SW_PAD_CTL Register

12.4.6.149.3 Diagram



12.4.6.149.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_02 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_02 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_02 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.150 SW_PAD_CTL_PAD_GPIO_EMC_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_03)

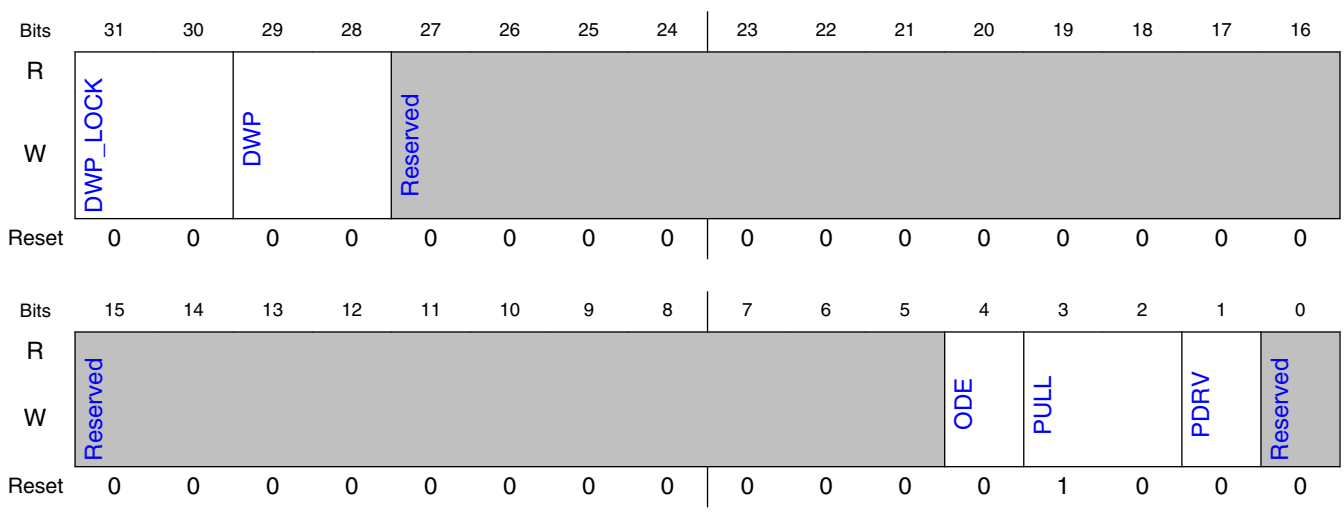
12.4.6.150.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_03	260h

12.4.6.150.2 Function

SW_PAD_CTL Register

12.4.6.150.3 Diagram



12.4.6.150.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_03 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_03 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_03 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.151 SW_PAD_CTL_PAD_GPIO_EMC_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_04)

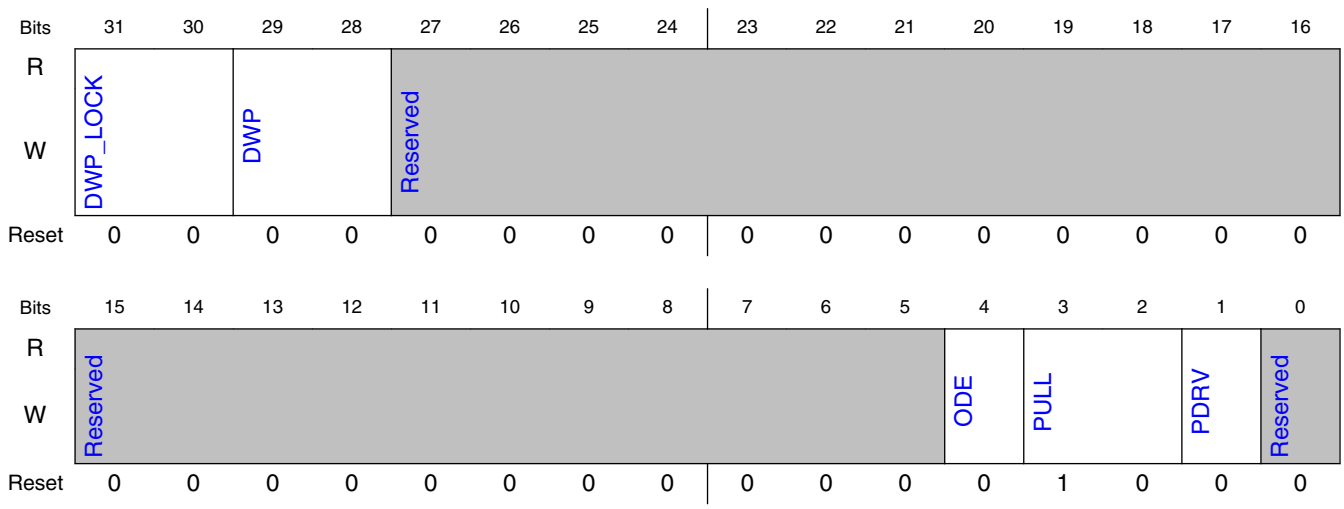
12.4.6.151.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_04	264h

12.4.6.151.2 Function

SW_PAD_CTL Register

12.4.6.151.3 Diagram



12.4.6.151.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_04 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_04 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_04 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.152 SW_PAD_CTL_PAD_GPIO_EMC_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_05)

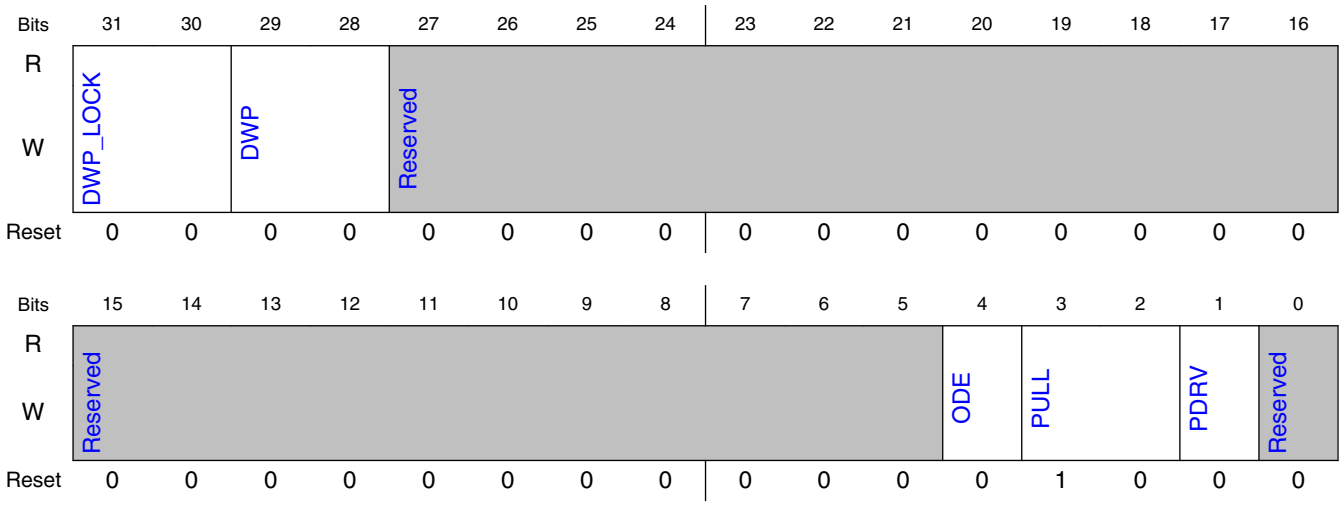
12.4.6.152.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_05	268h

12.4.6.152.2 Function

SW_PAD_CTL Register

12.4.6.152.3 Diagram



12.4.6.152.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_05 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_05 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_05 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.153 SW_PAD_CTL_PAD_GPIO_EMC_B1_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_06)

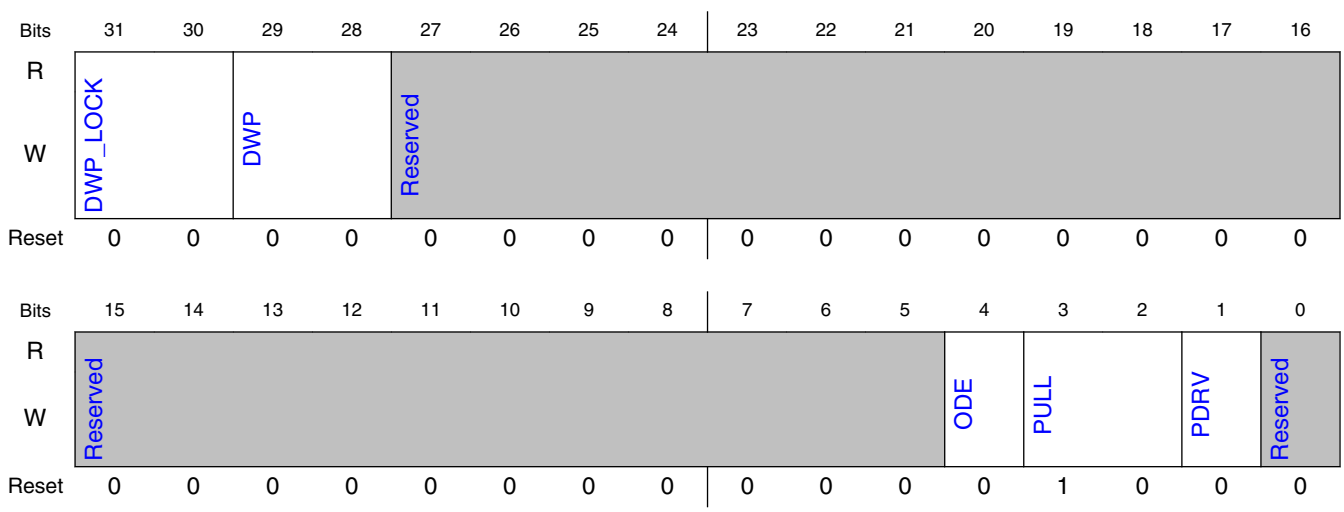
12.4.6.153.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_06	26Ch

12.4.6.153.2 Function

SW_PAD_CTL Register

12.4.6.153.3 Diagram



12.4.6.153.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_06 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_06 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_06 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.154 SW_PAD_CTL_PAD_GPIO_EMC_B1_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_07)

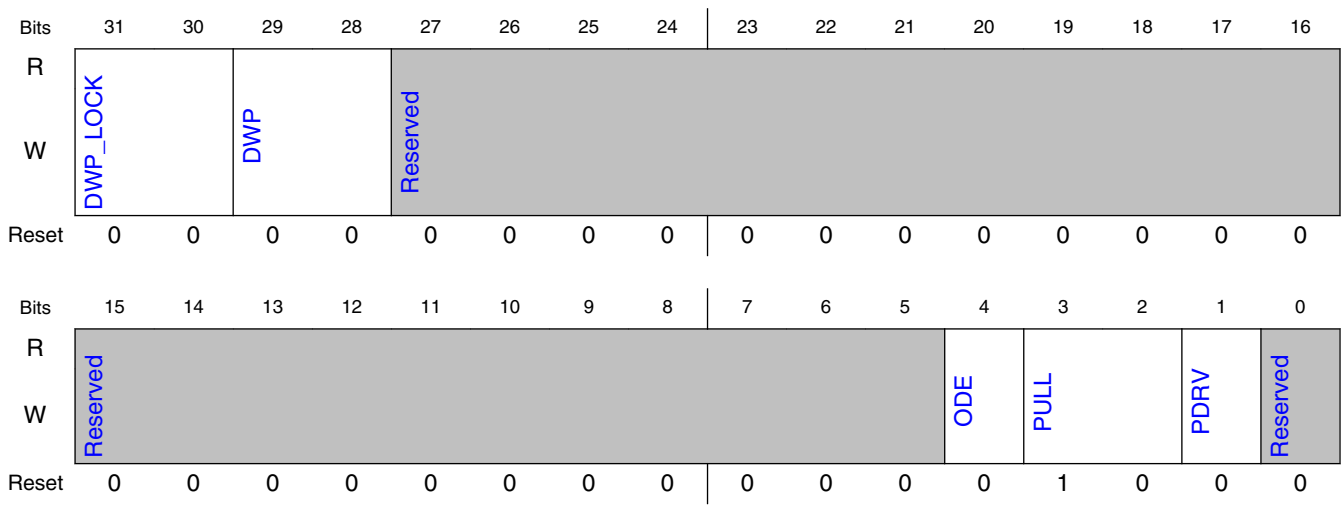
12.4.6.154.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_07	270h

12.4.6.154.2 Function

SW_PAD_CTL Register

12.4.6.154.3 Diagram



12.4.6.154.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_07 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_07 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_07 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.155 SW_PAD_CTL_PAD_GPIO_EMC_B1_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_08)

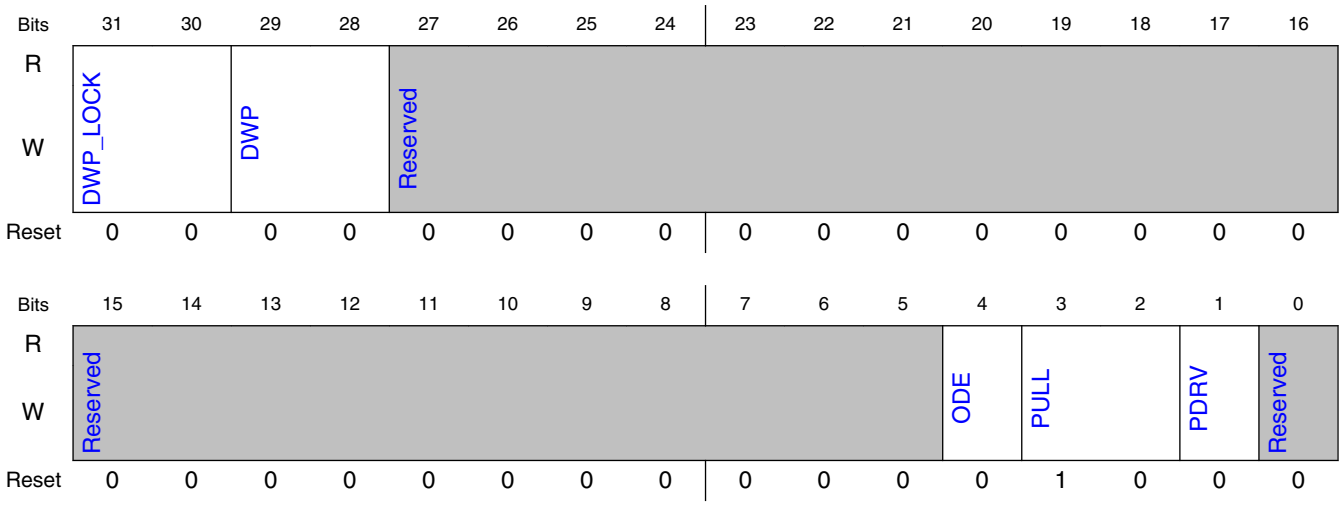
12.4.6.155.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_08	274h

12.4.6.155.2 Function

SW_PAD_CTL Register

12.4.6.155.3 Diagram



12.4.6.155.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_08 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_08 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_08 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.156 SW_PAD_CTL_PAD_GPIO_EMC_B1_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_09)

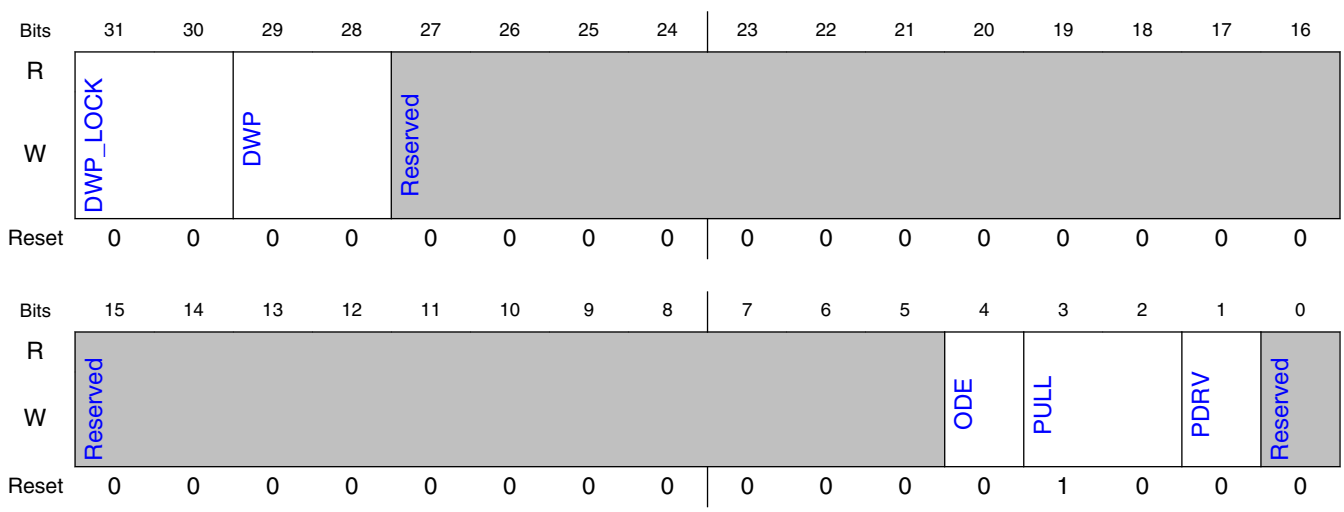
12.4.6.156.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_09	278h

12.4.6.156.2 Function

SW_PAD_CTL Register

12.4.6.156.3 Diagram



12.4.6.156.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_09 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_09 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_09 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.157 SW_PAD_CTL_PAD_GPIO_EMC_B1_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_10)

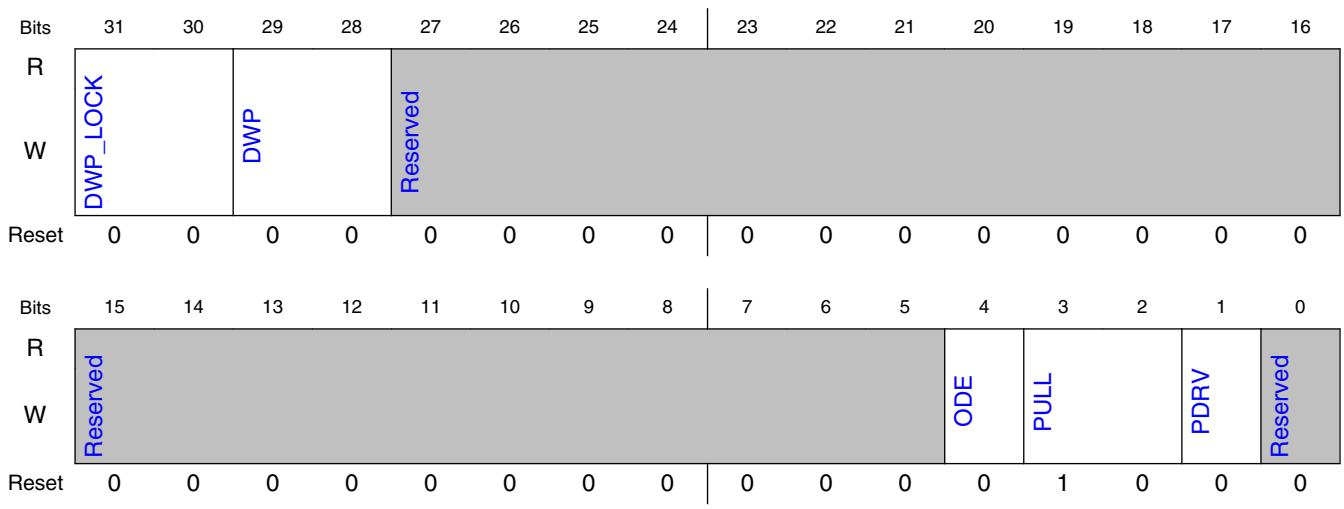
12.4.6.157.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_10	27Ch

12.4.6.157.2 Function

SW_PAD_CTL Register

12.4.6.157.3 Diagram



12.4.6.157.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_10 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_10 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_10 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.158 SW_PAD_CTL_PAD_GPIO_EMC_B1_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_11)

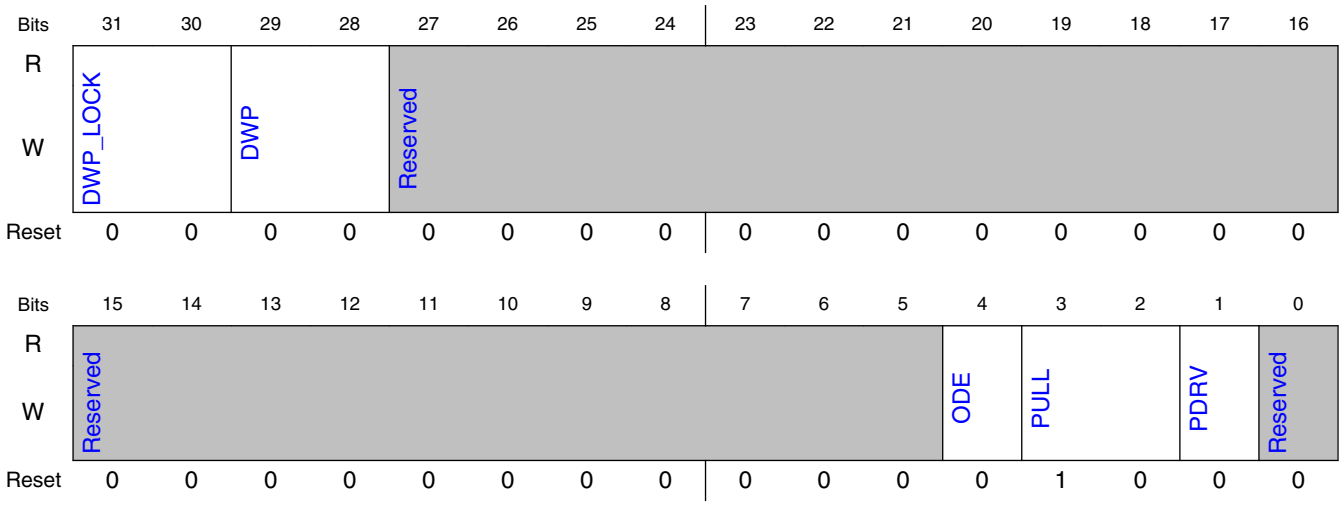
12.4.6.158.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_11	280h

12.4.6.158.2 Function

SW_PAD_CTL Register

12.4.6.158.3 Diagram



12.4.6.158.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_11 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_11 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_11 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.159 SW_PAD_CTL_PAD_GPIO_EMC_B1_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_12)

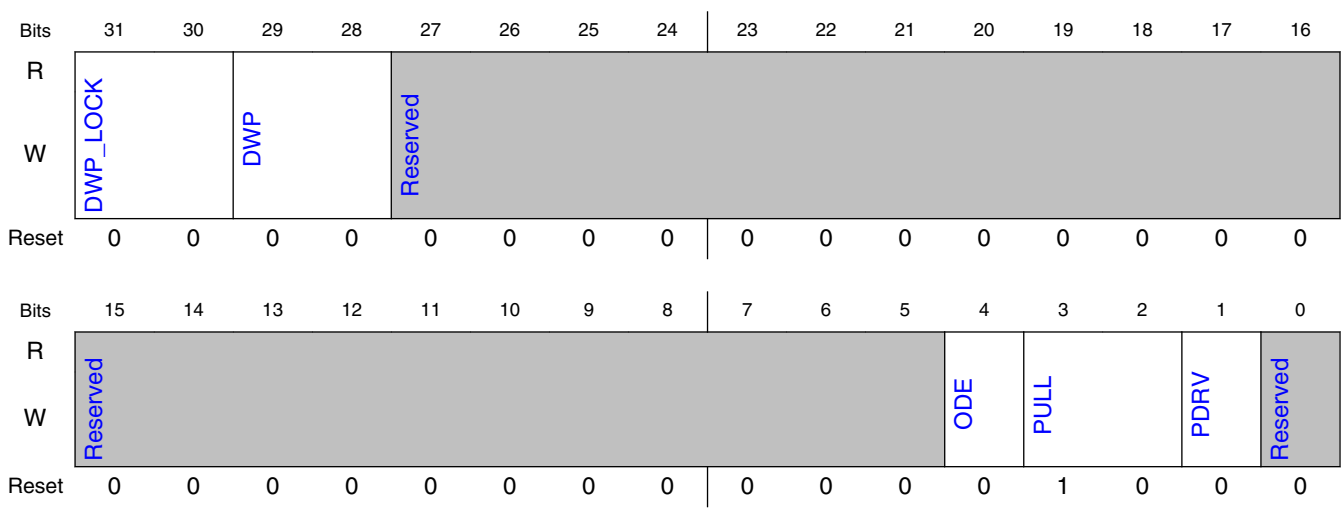
12.4.6.159.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_12	284h

12.4.6.159.2 Function

SW_PAD_CTL Register

12.4.6.159.3 Diagram



12.4.6.159.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_12 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_12 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_12 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.160 SW_PAD_CTL_PAD_GPIO_EMC_B1_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_13)

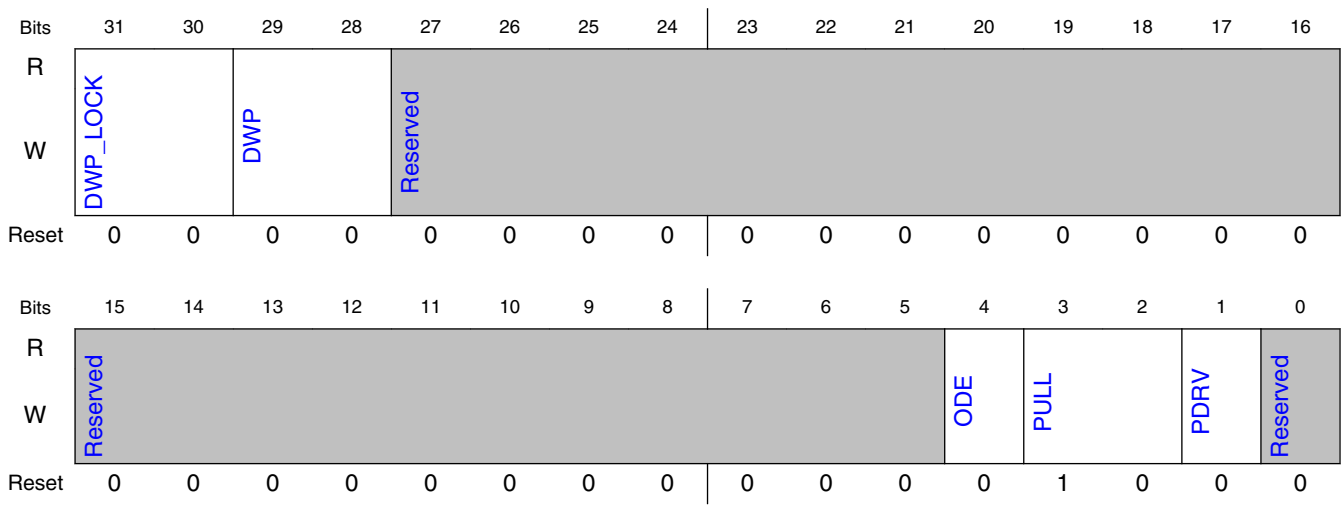
12.4.6.160.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_13	288h

12.4.6.160.2 Function

SW_PAD_CTL Register

12.4.6.160.3 Diagram



12.4.6.160.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_13 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_13 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_13 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.161 SW_PAD_CTL_PAD_GPIO_EMC_B1_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_14)

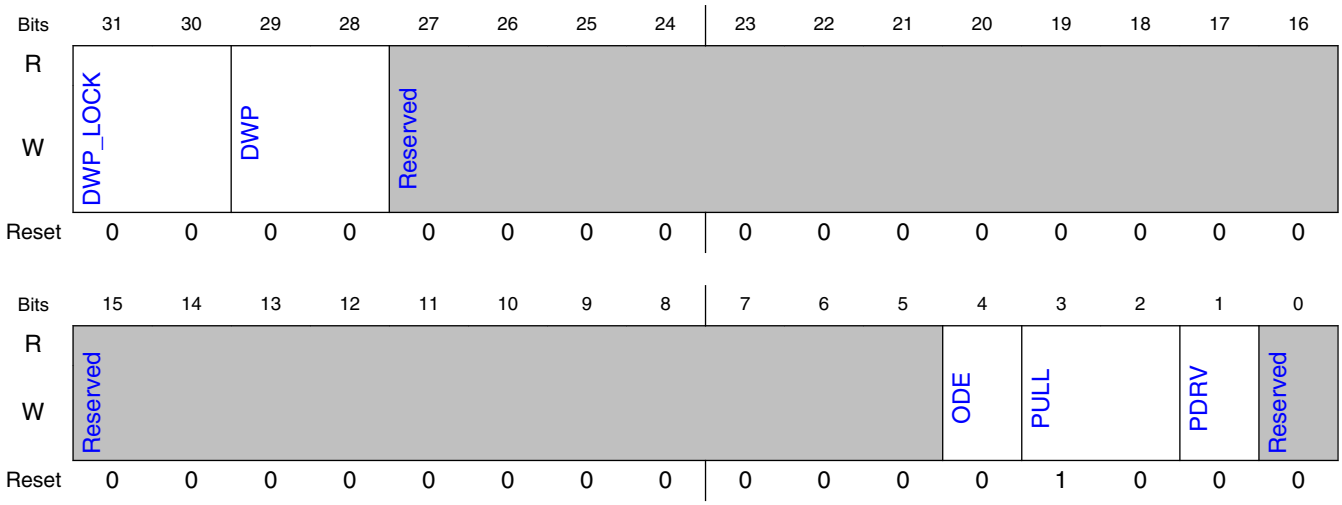
12.4.6.161.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_14	28Ch

12.4.6.161.2 Function

SW_PAD_CTL Register

12.4.6.161.3 Diagram



12.4.6.161.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_14 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_14 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_14 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.162 SW_PAD_CTL_PAD_GPIO_EMC_B1_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_15)

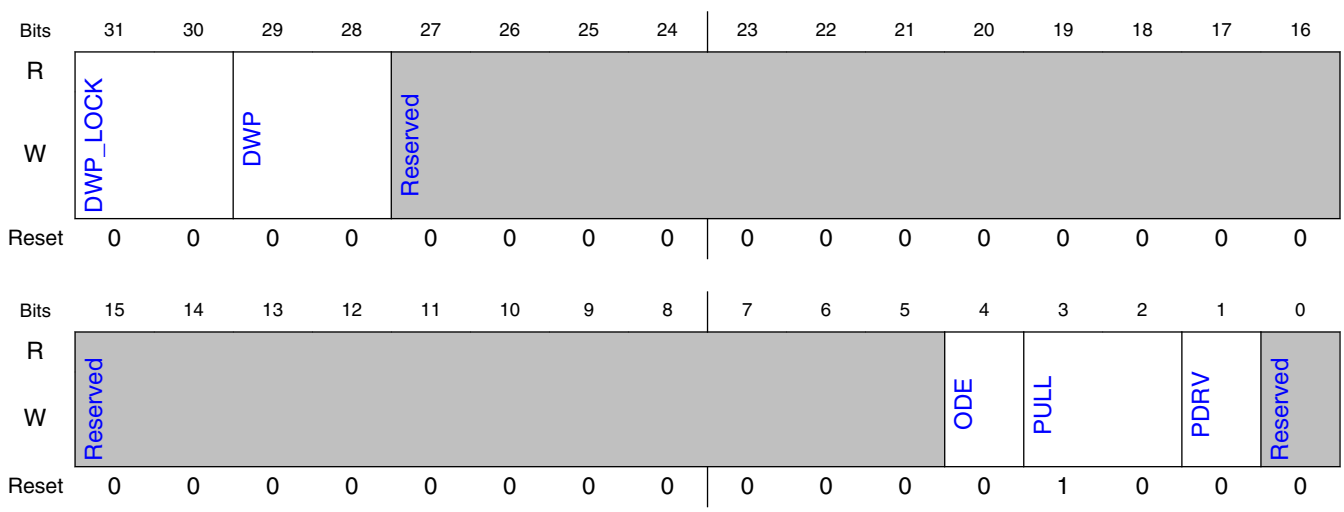
12.4.6.162.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_15	290h

12.4.6.162.2 Function

SW_PAD_CTL Register

12.4.6.162.3 Diagram



12.4.6.162.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_15 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_15 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_15 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.163 SW_PAD_CTL_PAD_GPIO_EMC_B1_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_16)

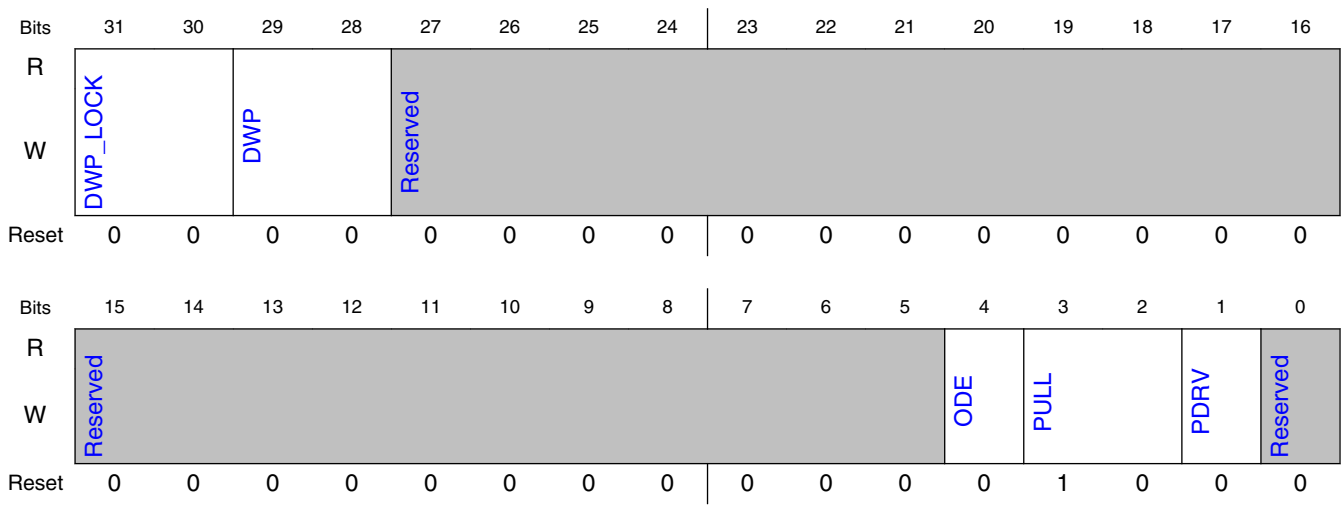
12.4.6.163.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_16	294h

12.4.6.163.2 Function

SW_PAD_CTL Register

12.4.6.163.3 Diagram



12.4.6.163.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_16 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_16 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_16 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.164 SW_PAD_CTL_PAD_GPIO_EMC_B1_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_17)

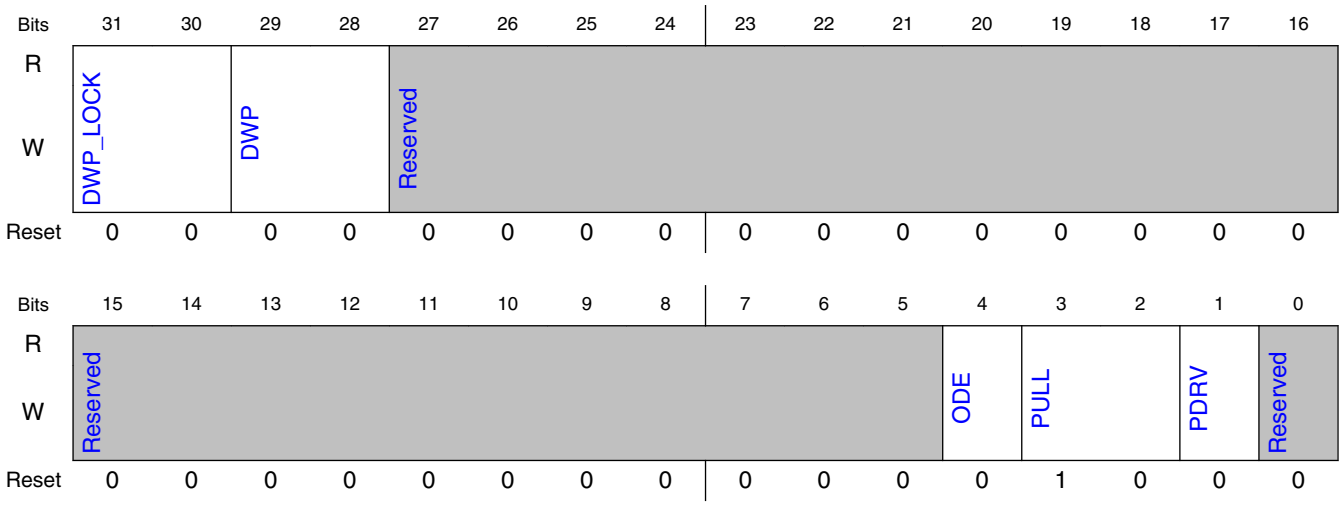
12.4.6.164.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_17	298h

12.4.6.164.2 Function

SW_PAD_CTL Register

12.4.6.164.3 Diagram



12.4.6.164.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_17 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_17 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_17 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.165 SW_PAD_CTL_PAD_GPIO_EMC_B1_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_18)

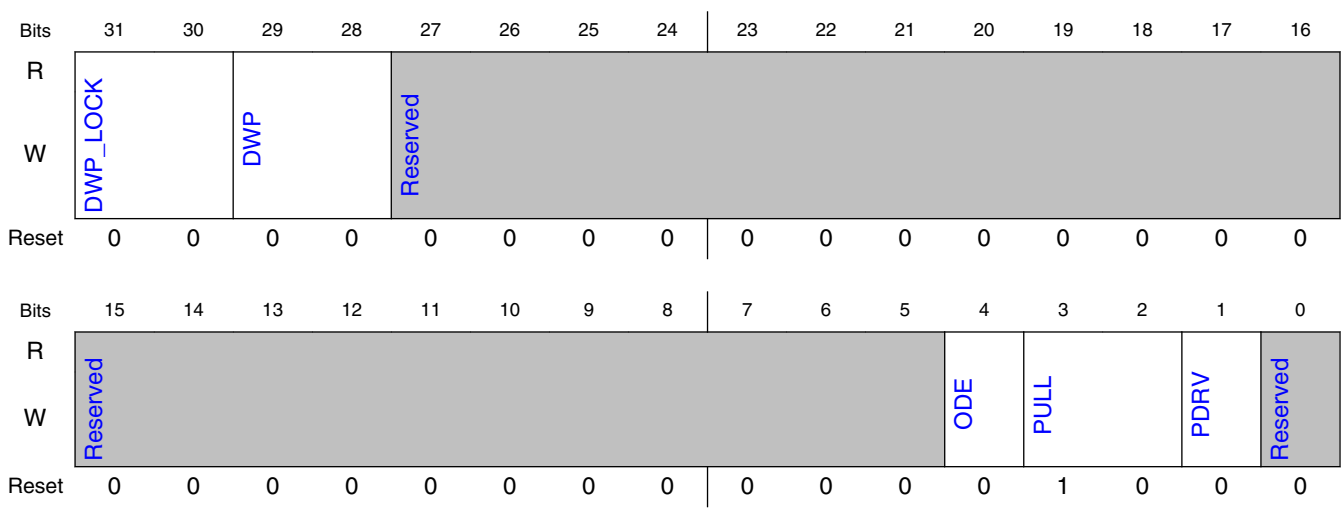
12.4.6.165.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_18	29Ch

12.4.6.165.2 Function

SW_PAD_CTL Register

12.4.6.165.3 Diagram



12.4.6.165.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_18 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_18 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_18 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.166 SW_PAD_CTL_PAD_GPIO_EMC_B1_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_19)

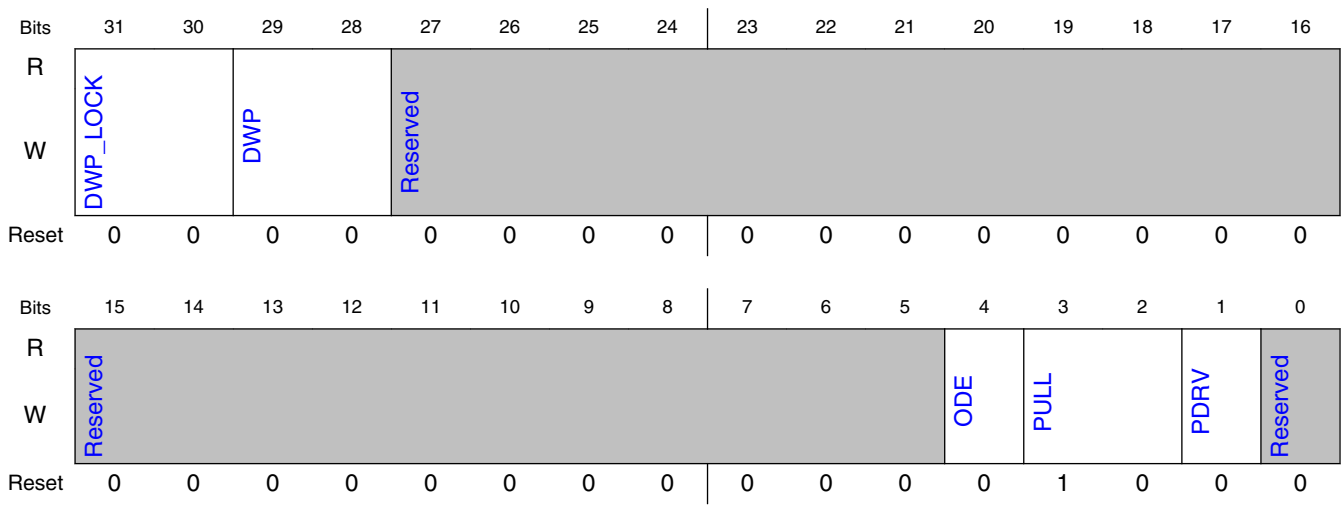
12.4.6.166.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_19	2A0h

12.4.6.166.2 Function

SW_PAD_CTL Register

12.4.6.166.3 Diagram



12.4.6.166.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_19 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_19 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_19 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.167 SW_PAD_CTL_PAD_GPIO_EMC_B1_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_20)

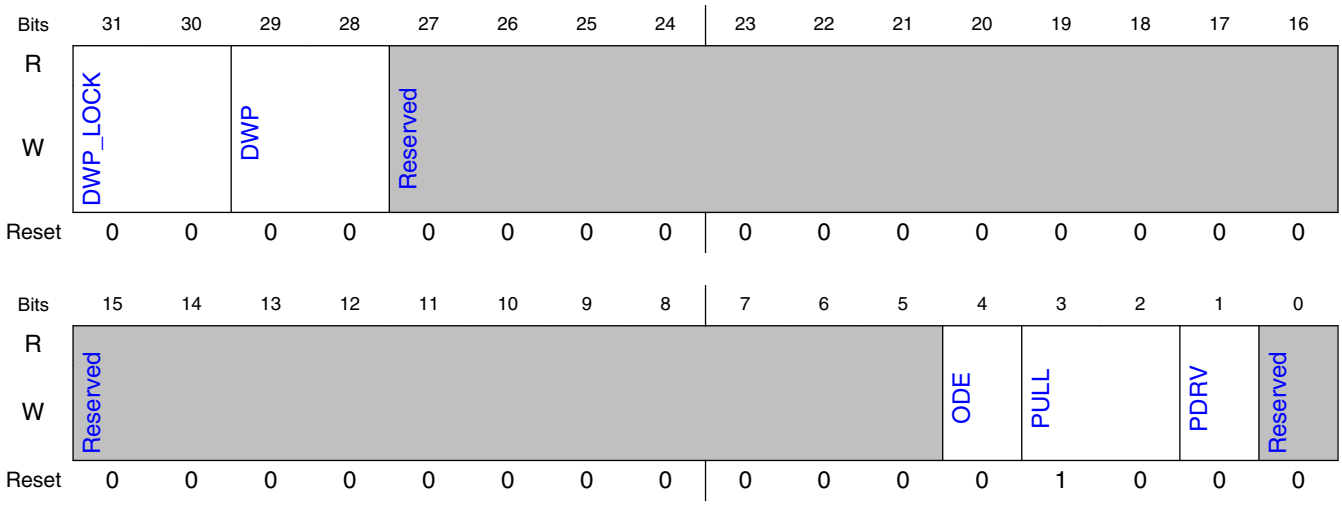
12.4.6.167.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_20	2A4h

12.4.6.167.2 Function

SW_PAD_CTL Register

12.4.6.167.3 Diagram



12.4.6.167.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_20 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_20 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_20 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.168 SW_PAD_CTL_PAD_GPIO_EMC_B1_21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_21)

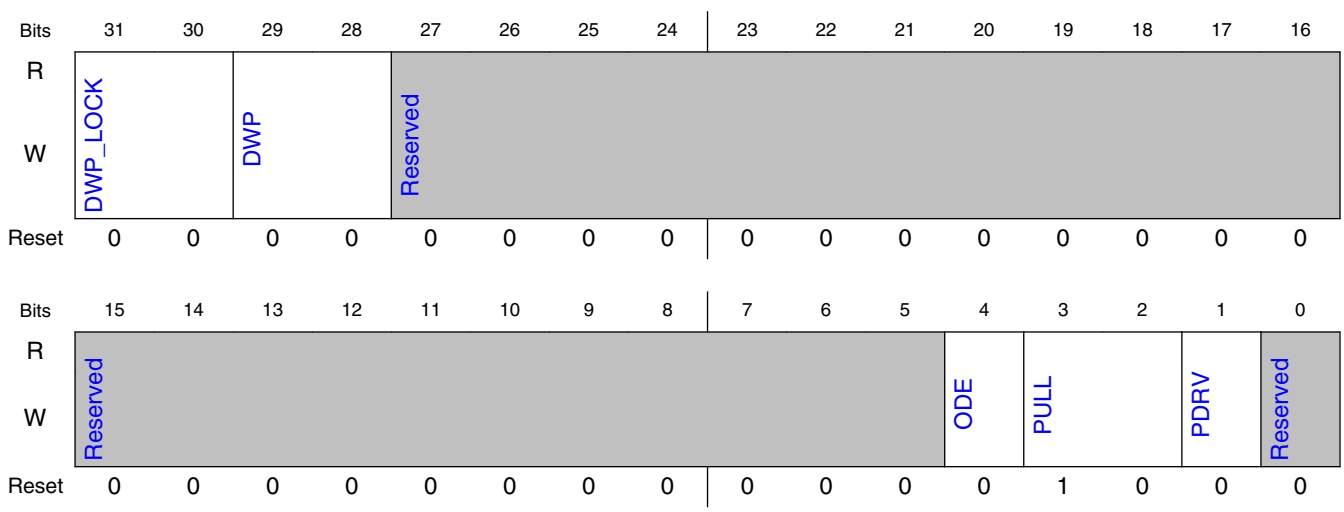
12.4.6.168.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_21	2A8h

12.4.6.168.2 Function

SW_PAD_CTL Register

12.4.6.168.3 Diagram



12.4.6.168.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_21 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_21 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_21 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.169 SW_PAD_CTL_PAD_GPIO_EMC_B1_22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_22)

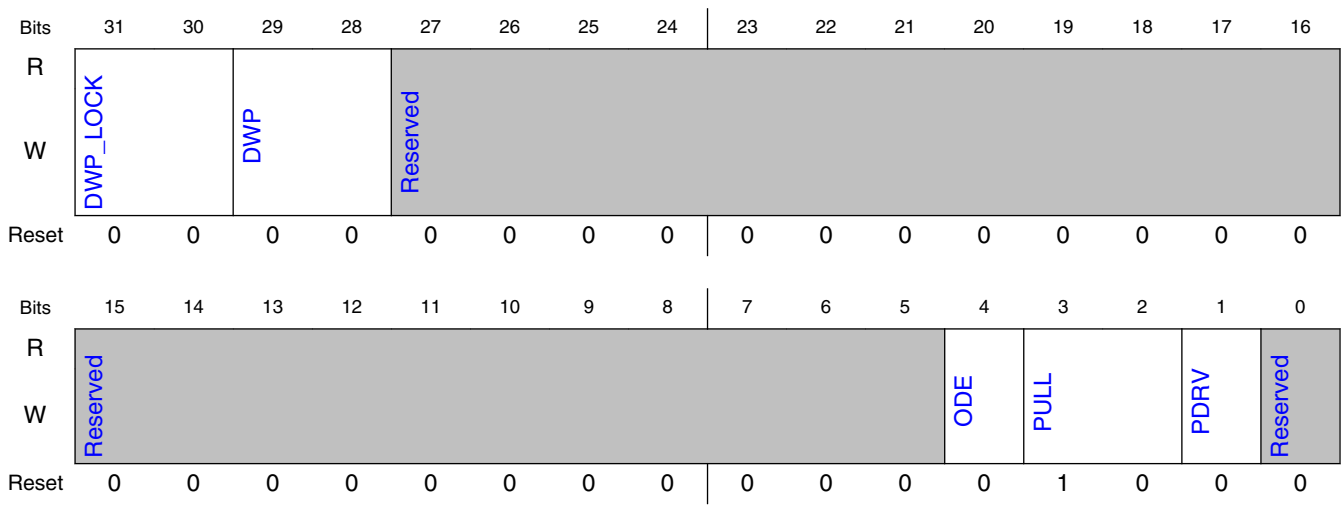
12.4.6.169.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_22	2ACh

12.4.6.169.2 Function

SW_PAD_CTL Register

12.4.6.169.3 Diagram



12.4.6.169.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_22 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_22 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_22 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.170 SW_PAD_CTL_PAD_GPIO_EMC_B1_23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_23)

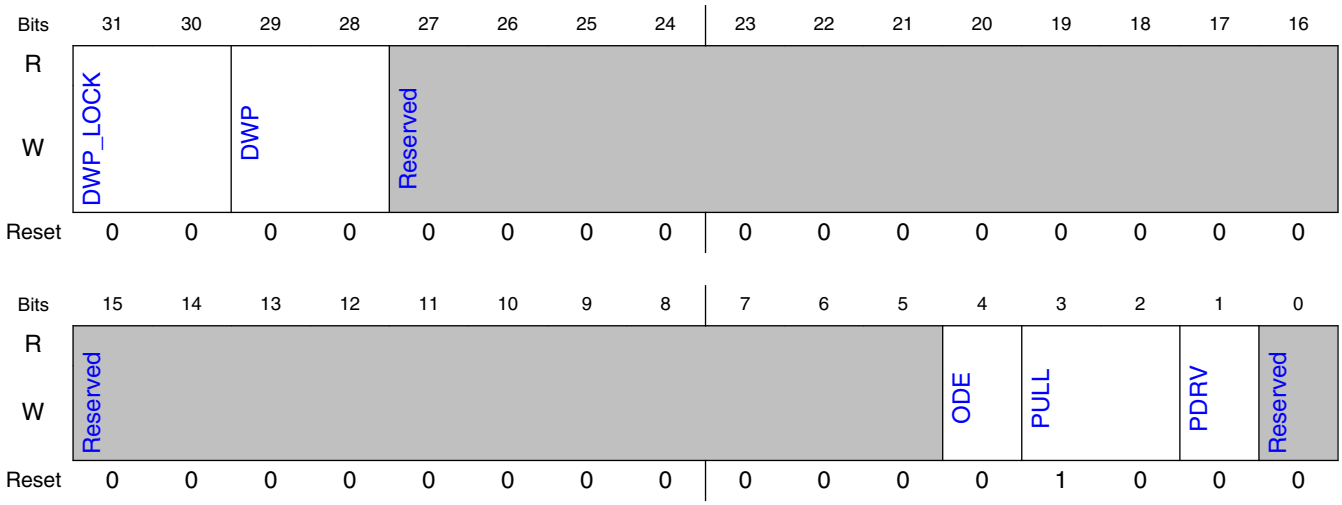
12.4.6.170.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_23	2B0h

12.4.6.170.2 Function

SW_PAD_CTL Register

12.4.6.170.3 Diagram



12.4.6.170.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_23 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_23 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_23 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.171 SW_PAD_CTL_PAD_GPIO_EMC_B1_24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_24)

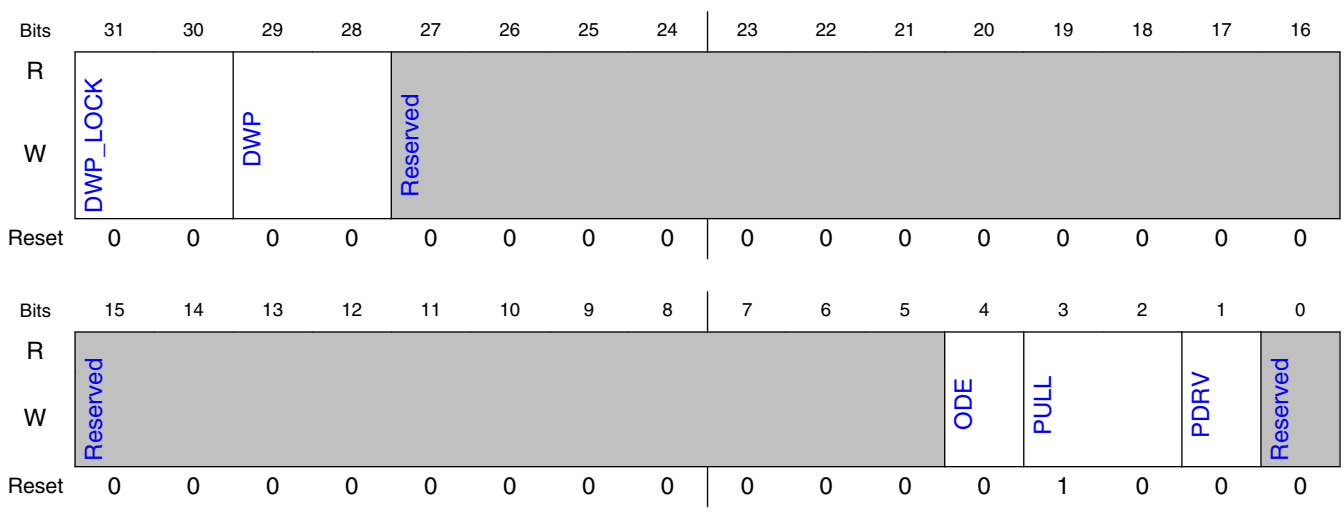
12.4.6.171.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_24	2B4h

12.4.6.171.2 Function

SW_PAD_CTL Register

12.4.6.171.3 Diagram



12.4.6.171.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_24 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_24 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_24 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.172 SW_PAD_CTL_PAD_GPIO_EMC_B1_25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_25)

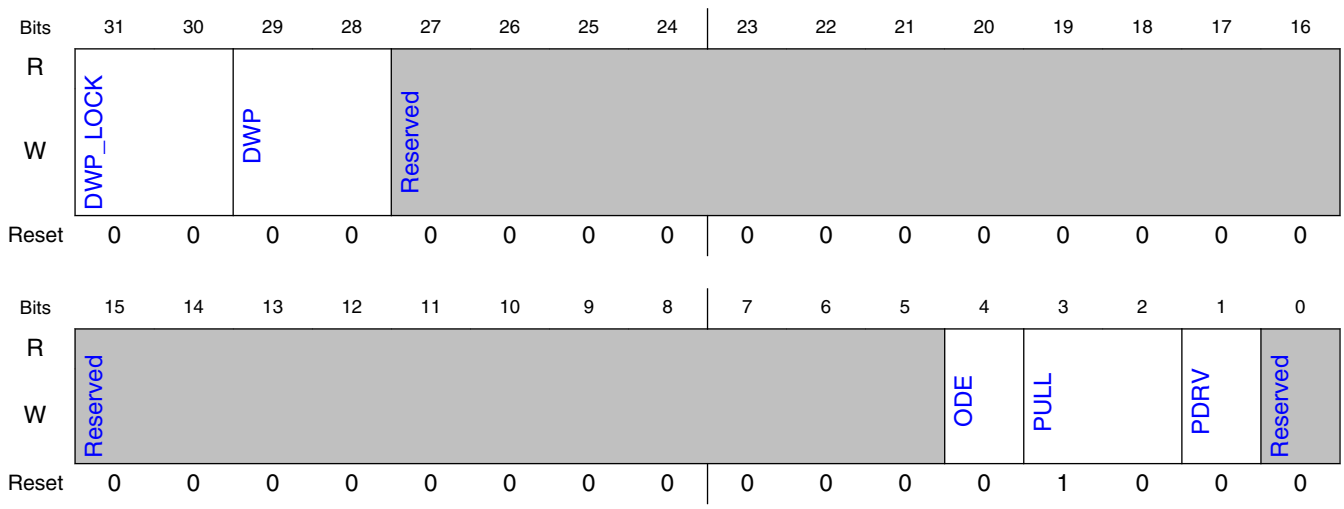
12.4.6.172.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_25	2B8h

12.4.6.172.2 Function

SW_PAD_CTL Register

12.4.6.172.3 Diagram



12.4.6.172.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_25 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_25 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_25 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.173 SW_PAD_CTL_PAD_GPIO_EMC_B1_26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_26)

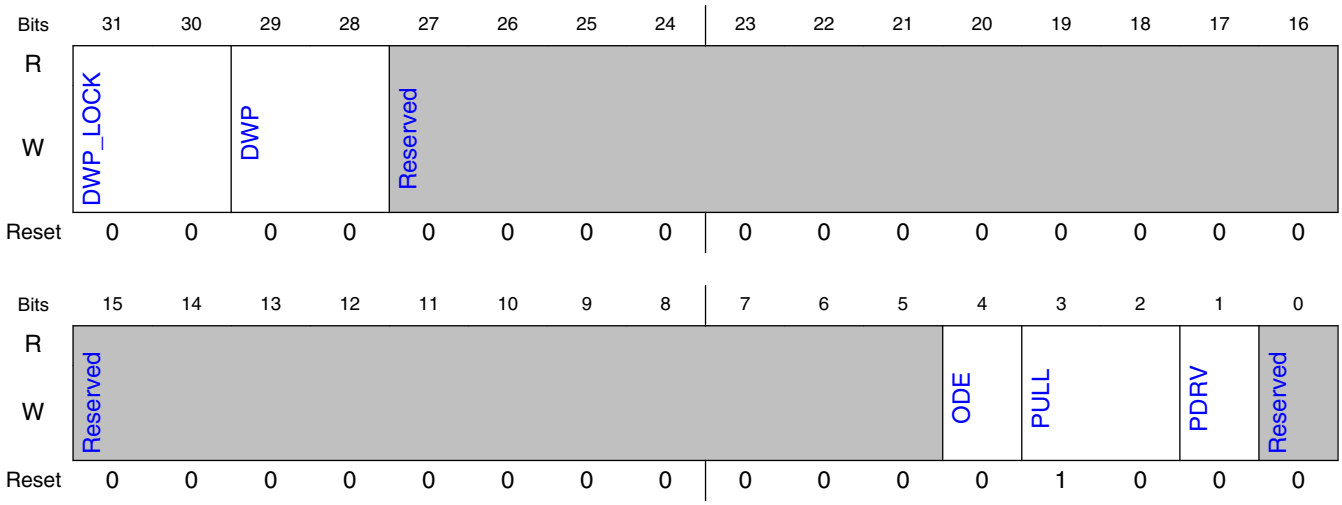
12.4.6.173.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_26	2BCh

12.4.6.173.2 Function

SW_PAD_CTL Register

12.4.6.173.3 Diagram



12.4.6.173.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_26 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_26 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_26 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.174 SW_PAD_CTL_PAD_GPIO_EMC_B1_27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_27)

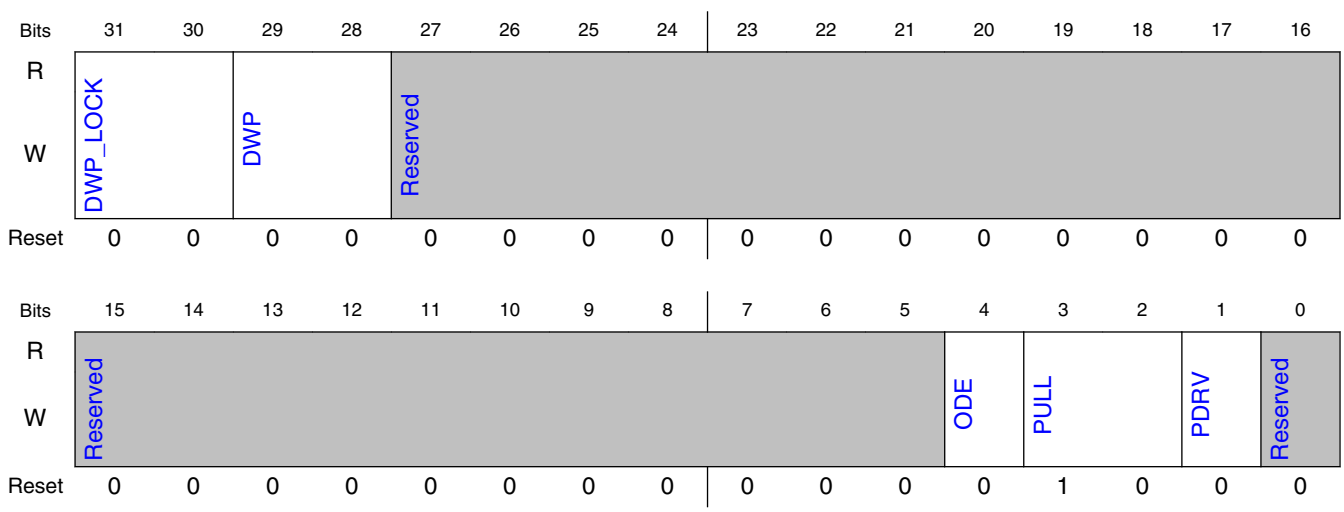
12.4.6.174.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_27	2C0h

12.4.6.174.2 Function

SW_PAD_CTL Register

12.4.6.174.3 Diagram



12.4.6.174.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_27 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_27 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_27 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.175 SW_PAD_CTL_PAD_GPIO_EMC_B1_28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_28)

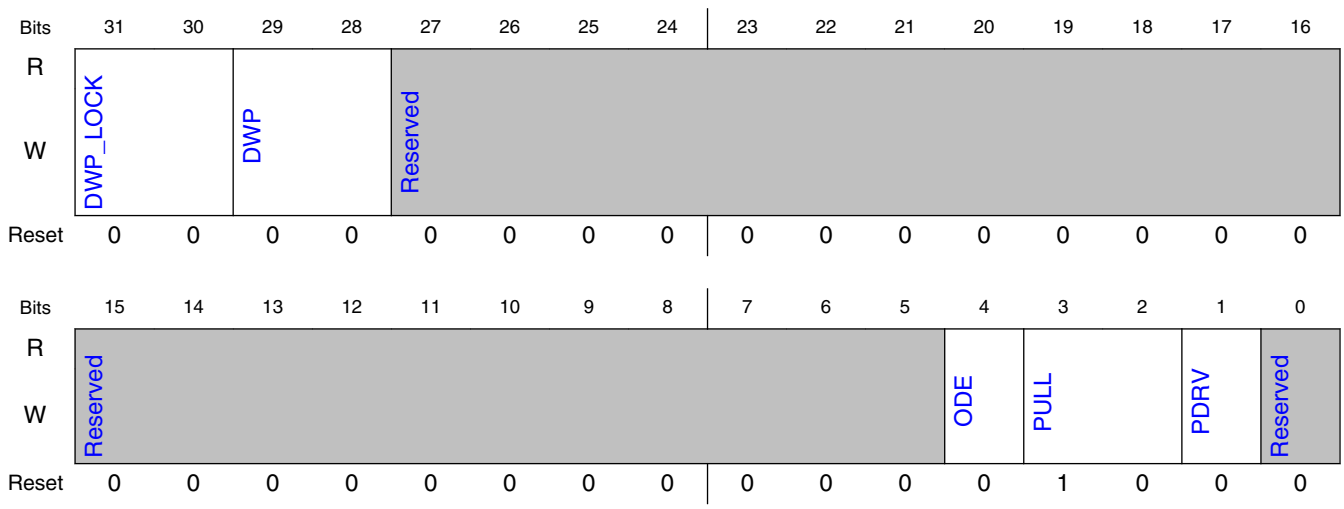
12.4.6.175.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_28	2C4h

12.4.6.175.2 Function

SW_PAD_CTL Register

12.4.6.175.3 Diagram



12.4.6.175.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_28 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_28 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_28 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.176 SW_PAD_CTL_PAD_GPIO_EMC_B1_29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_29)

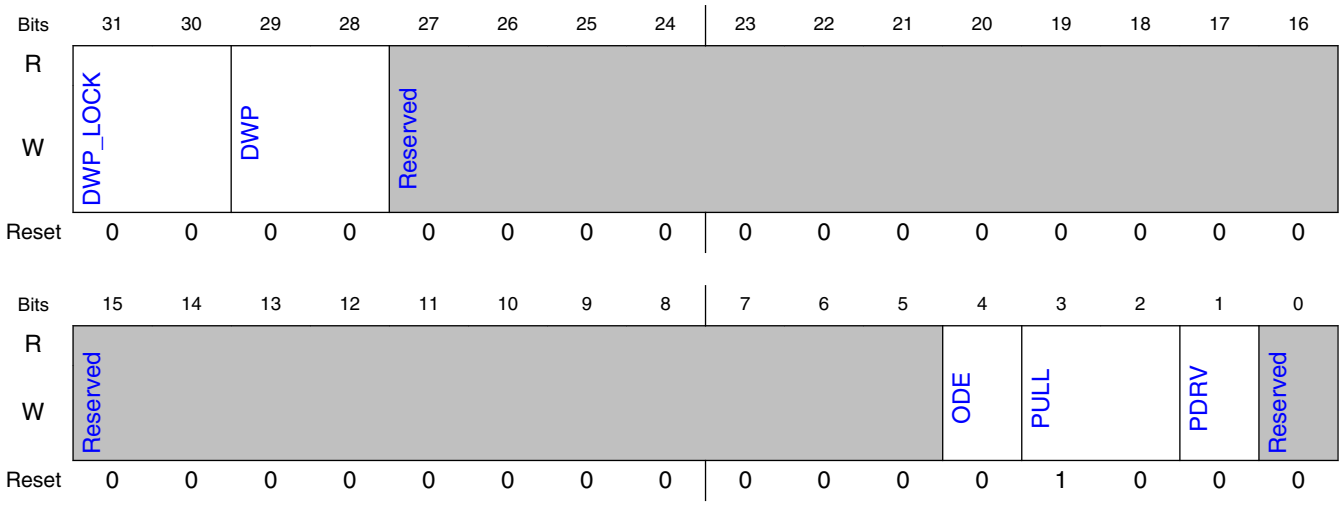
12.4.6.176.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_29	2C8h

12.4.6.176.2 Function

SW_PAD_CTL Register

12.4.6.176.3 Diagram



12.4.6.176.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_29 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_29 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_29 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.177 SW_PAD_CTL_PAD_GPIO_EMC_B1_30 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_30)

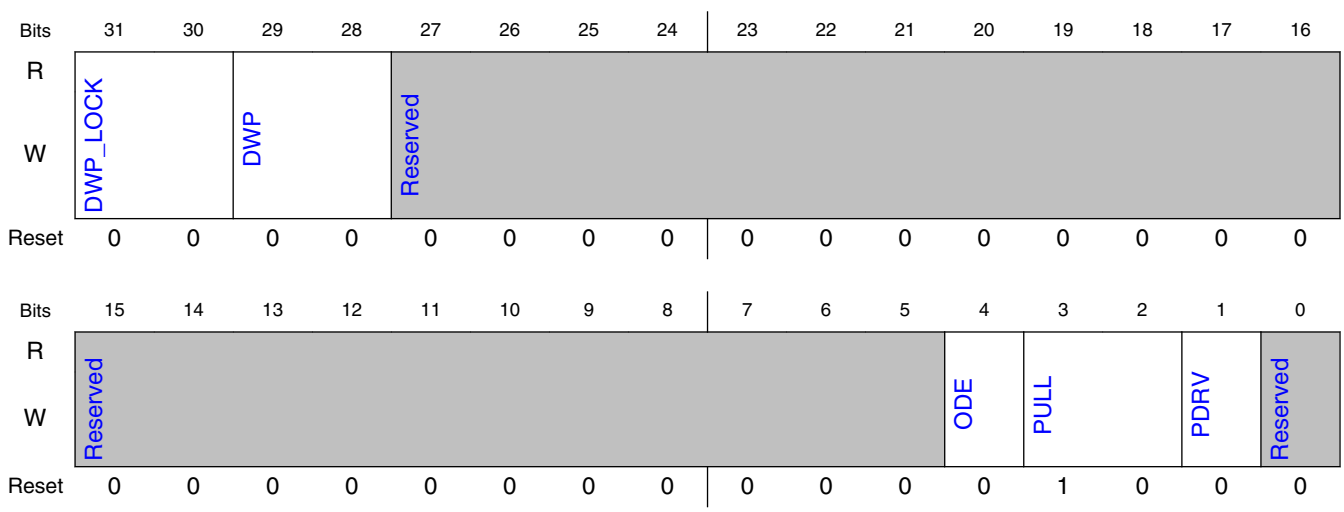
12.4.6.177.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_30	2CCh

12.4.6.177.2 Function

SW_PAD_CTL Register

12.4.6.177.3 Diagram



12.4.6.177.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_30 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_30 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_30 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.178 SW_PAD_CTL_PAD_GPIO_EMC_B1_31 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_31)

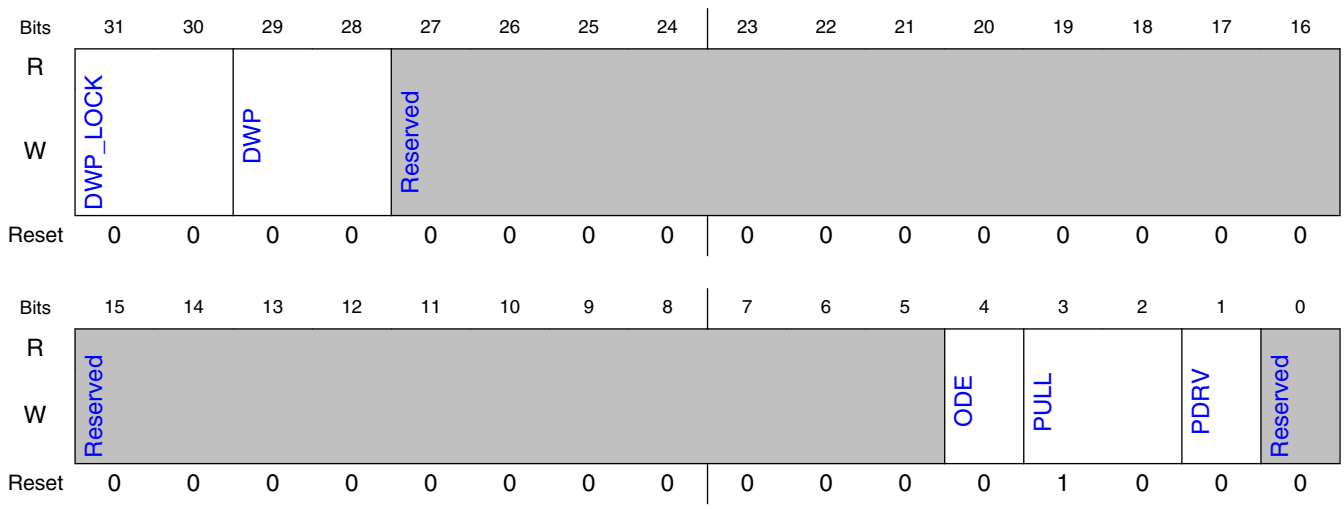
12.4.6.178.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_31	2D0h

12.4.6.178.2 Function

SW_PAD_CTL Register

12.4.6.178.3 Diagram



12.4.6.178.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_31 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_31 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_31 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.179 SW_PAD_CTL_PAD_GPIO_EMC_B1_32 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_32)

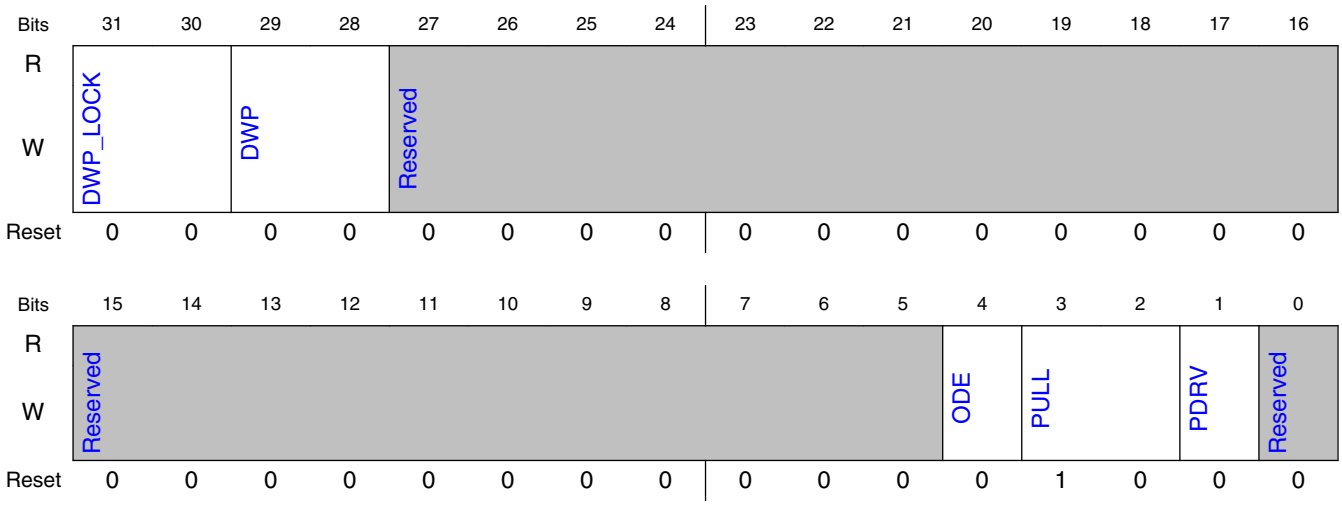
12.4.6.179.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_32	2D4h

12.4.6.179.2 Function

SW_PAD_CTL Register

12.4.6.179.3 Diagram



12.4.6.179.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_32 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_32 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_32 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.180 SW_PAD_CTL_PAD_GPIO_EMC_B1_33 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_33)

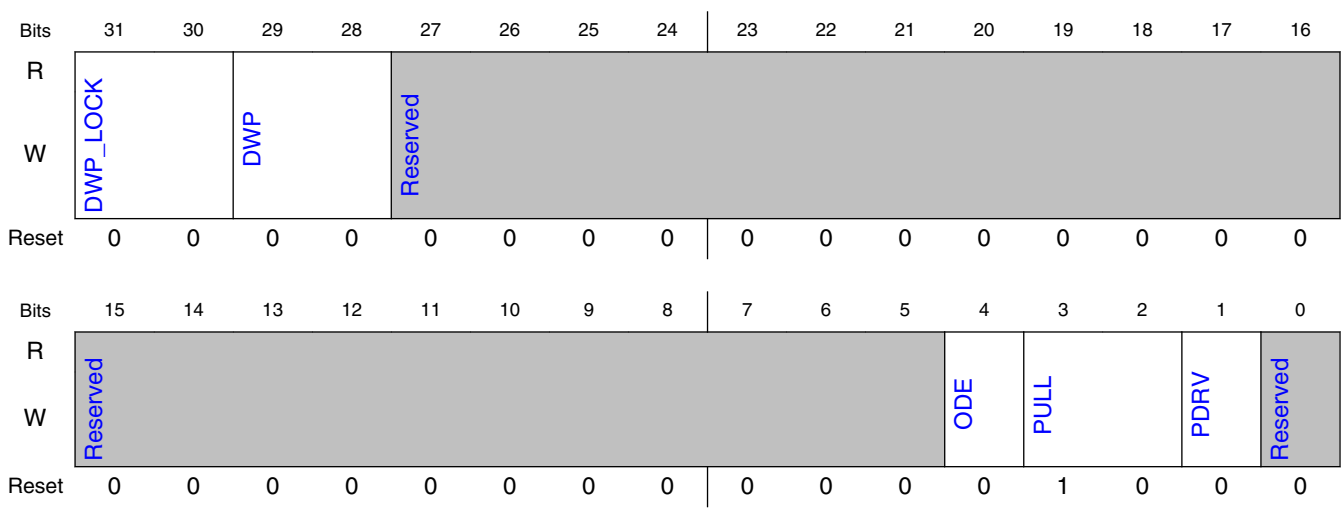
12.4.6.180.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_33	2D8h

12.4.6.180.2 Function

SW_PAD_CTL Register

12.4.6.180.3 Diagram



12.4.6.180.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_33 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_33 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_33 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.181 SW_PAD_CTL_PAD_GPIO_EMC_B1_34 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_34)

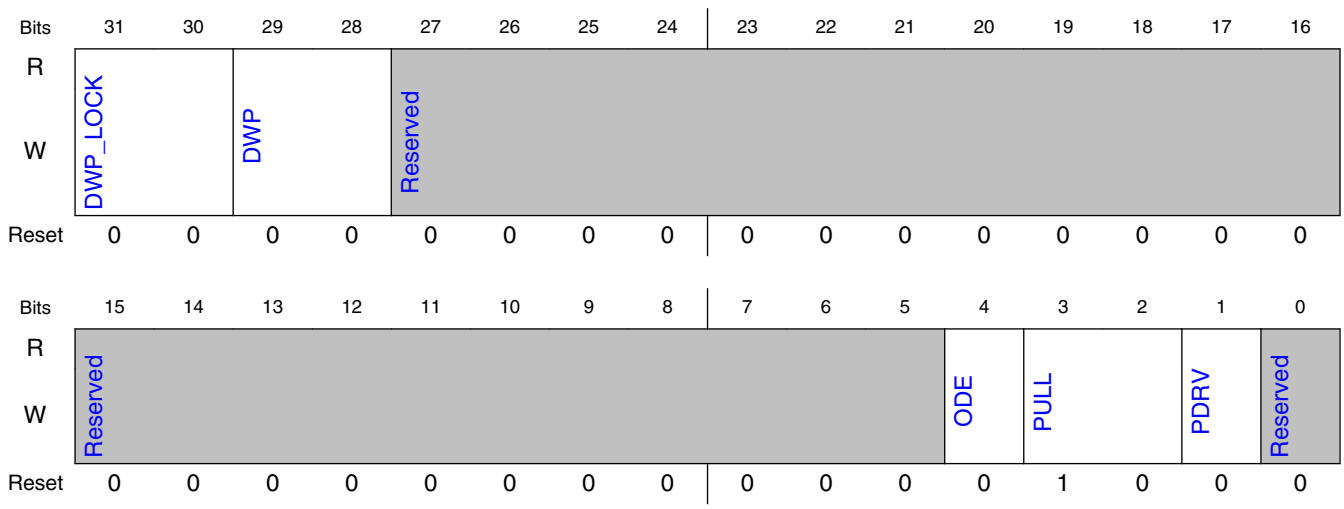
12.4.6.181.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_34	2DCh

12.4.6.181.2 Function

SW_PAD_CTL Register

12.4.6.181.3 Diagram



12.4.6.181.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_34 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_34 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_34 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.182 SW_PAD_CTL_PAD_GPIO_EMC_B1_35 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_35)

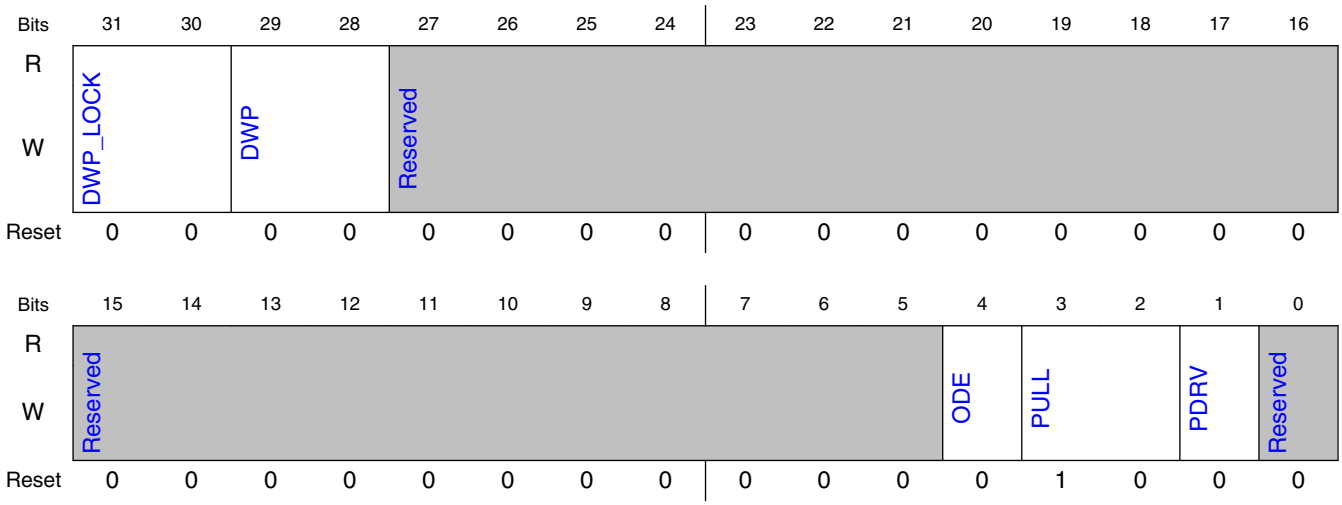
12.4.6.182.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_35	2E0h

12.4.6.182.2 Function

SW_PAD_CTL Register

12.4.6.182.3 Diagram



12.4.6.182.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_35 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_35 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_35 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.183 SW_PAD_CTL_PAD_GPIO_EMC_B1_36 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_36)

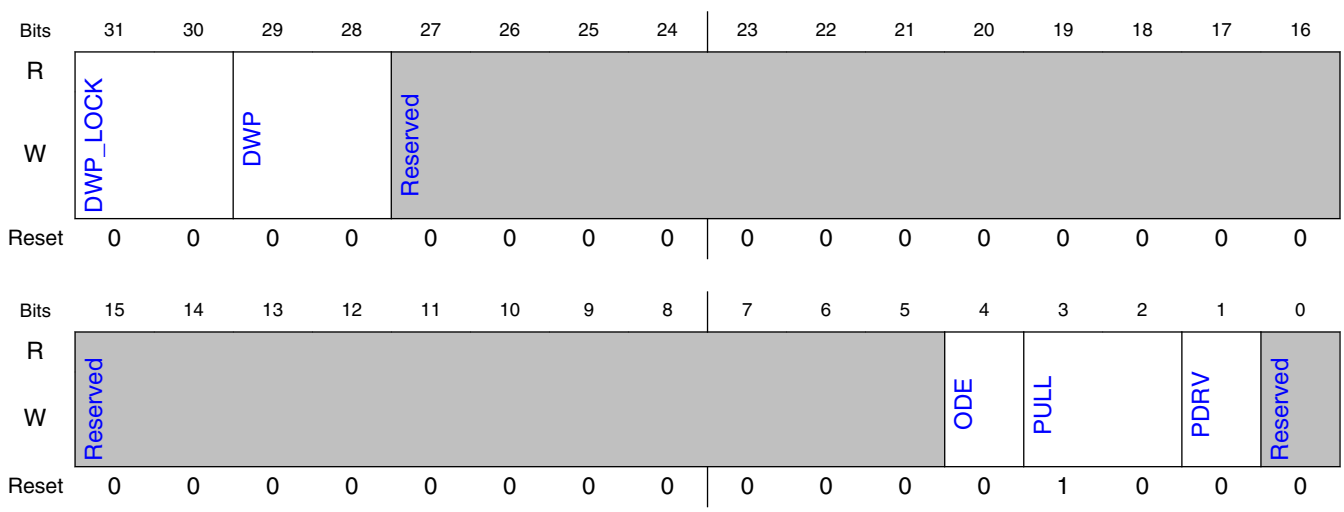
12.4.6.183.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_36	2E4h

12.4.6.183.2 Function

SW_PAD_CTL Register

12.4.6.183.3 Diagram



12.4.6.183.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_36 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_36 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_36 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.184 SW_PAD_CTL_PAD_GPIO_EMC_B1_37 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_37)

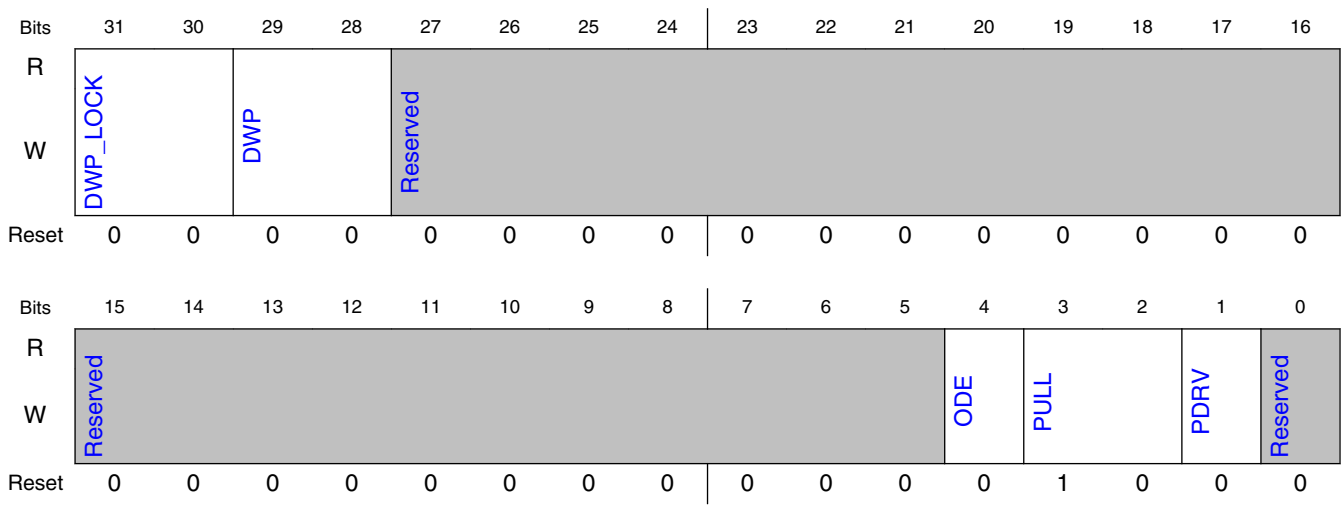
12.4.6.184.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_37	2E8h

12.4.6.184.2 Function

SW_PAD_CTL Register

12.4.6.184.3 Diagram



12.4.6.184.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_37 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_37 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_37 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.185 SW_PAD_CTL_PAD_GPIO_EMC_B1_38 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_38)

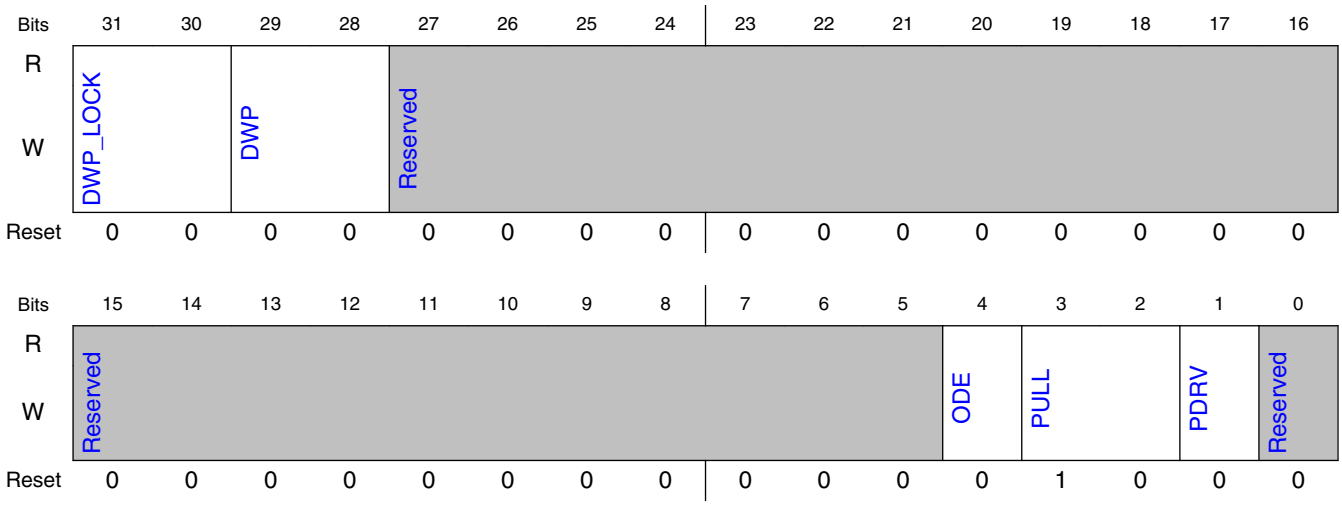
12.4.6.185.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_38	2ECh

12.4.6.185.2 Function

SW_PAD_CTL Register

12.4.6.185.3 Diagram



12.4.6.185.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_38 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_38 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_38 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.186 SW_PAD_CTL_PAD_GPIO_EMC_B1_39 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_39)

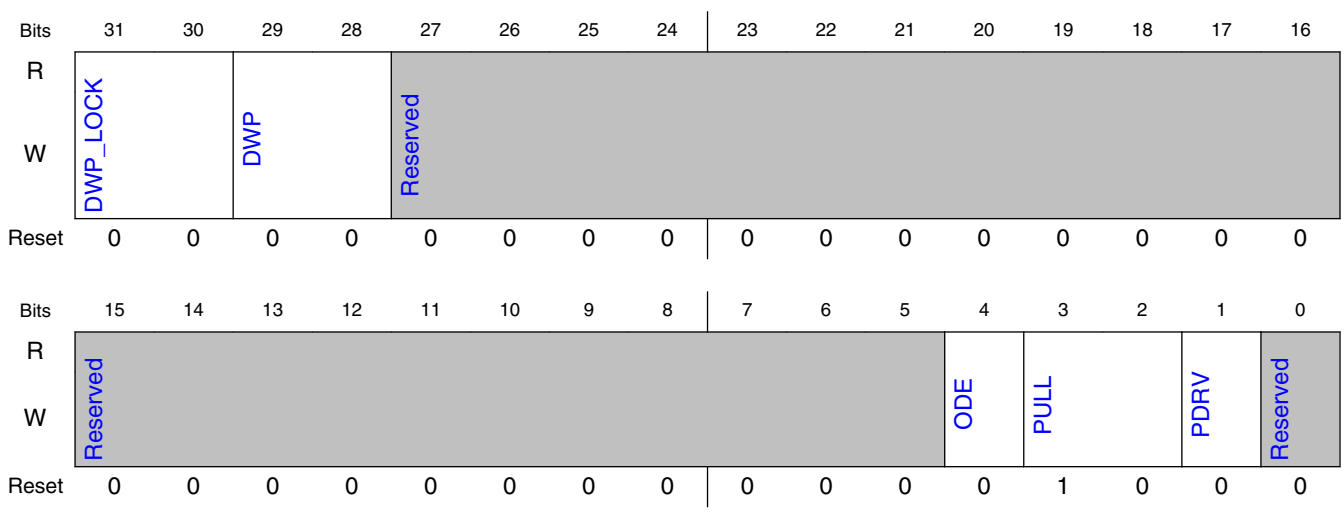
12.4.6.186.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_39	2F0h

12.4.6.186.2 Function

SW_PAD_CTL Register

12.4.6.186.3 Diagram



12.4.6.186.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_39 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_39 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_39 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.187 SW_PAD_CTL_PAD_GPIO_EMC_B1_40 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_40)

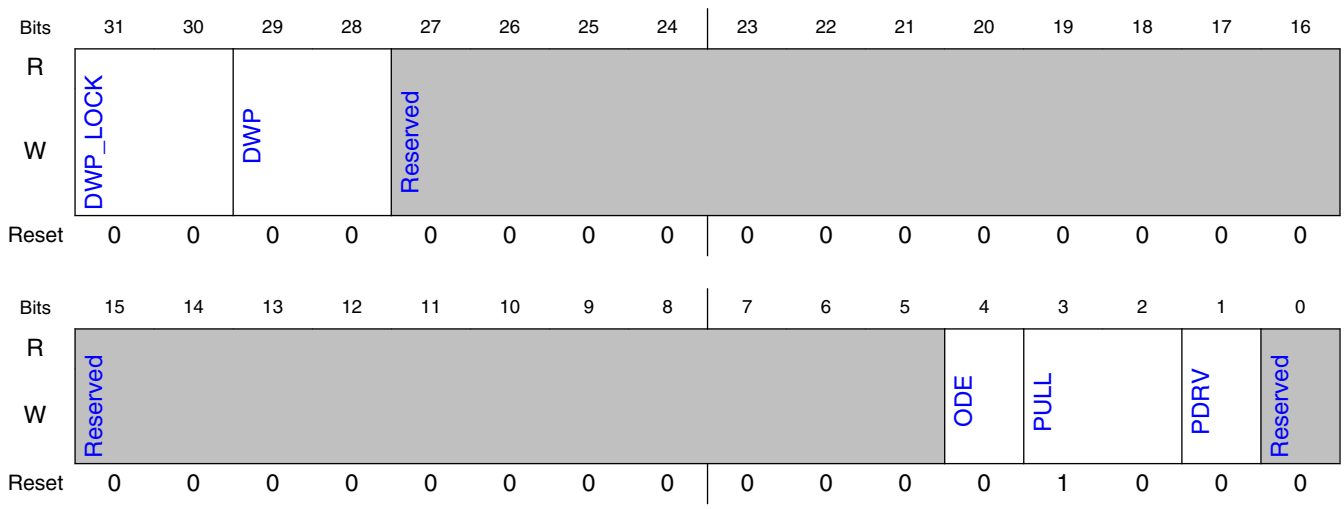
12.4.6.187.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_40	2F4h

12.4.6.187.2 Function

SW_PAD_CTL Register

12.4.6.187.3 Diagram



12.4.6.187.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_40 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_40 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_40 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.188 SW_PAD_CTL_PAD_GPIO_EMC_B1_41 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B1_41)

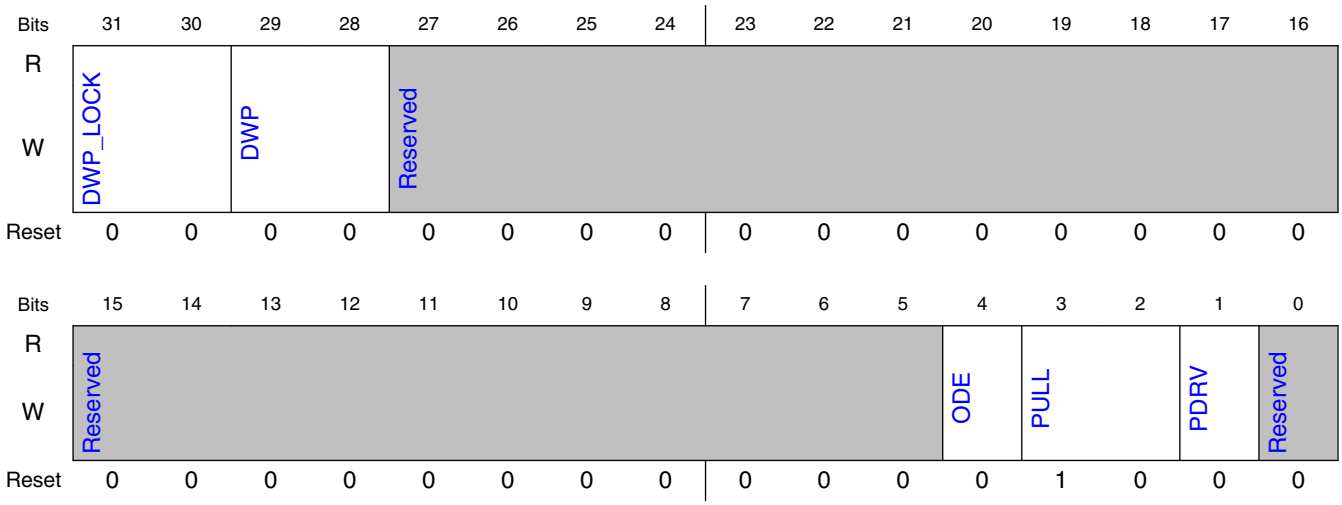
12.4.6.188.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B1_41	2F8h

12.4.6.188.2 Function

SW_PAD_CTL Register

12.4.6.188.3 Diagram



12.4.6.188.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B1_41 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B1_41 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B1_41 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.189 SW_PAD_CTL_PAD_GPIO_EMC_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_00)

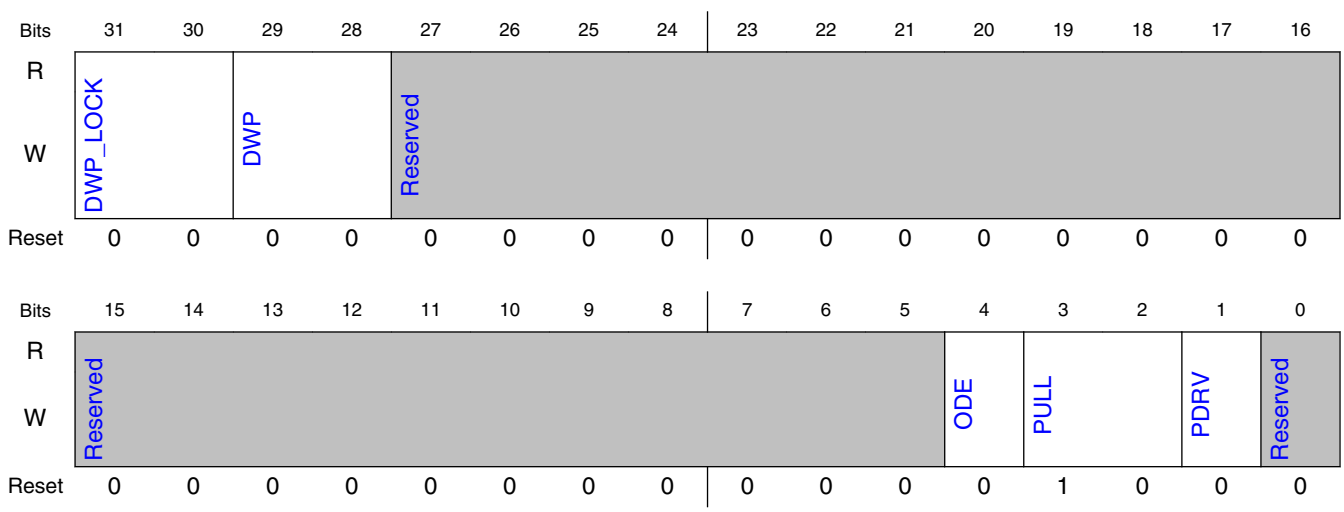
12.4.6.189.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_00	2FCh

12.4.6.189.2 Function

SW_PAD_CTL Register

12.4.6.189.3 Diagram



12.4.6.189.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_00 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_00 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_00 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.190 SW_PAD_CTL_PAD_GPIO_EMC_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_01)

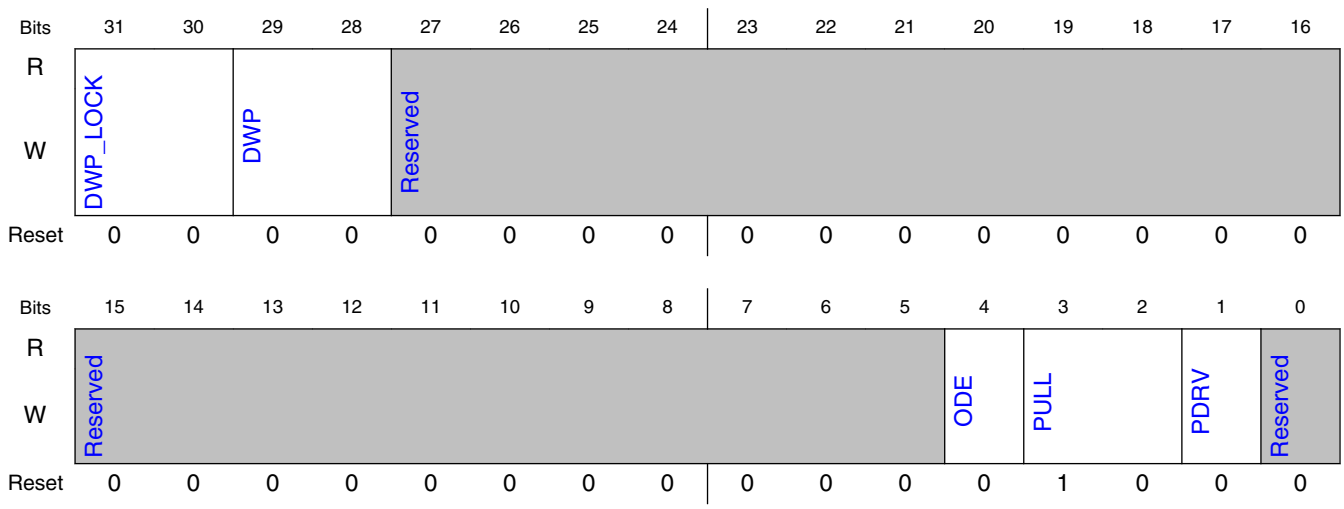
12.4.6.190.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_01	300h

12.4.6.190.2 Function

SW_PAD_CTL Register

12.4.6.190.3 Diagram



12.4.6.190.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_01 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_01 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_01 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.191 SW_PAD_CTL_PAD_GPIO_EMC_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_02)

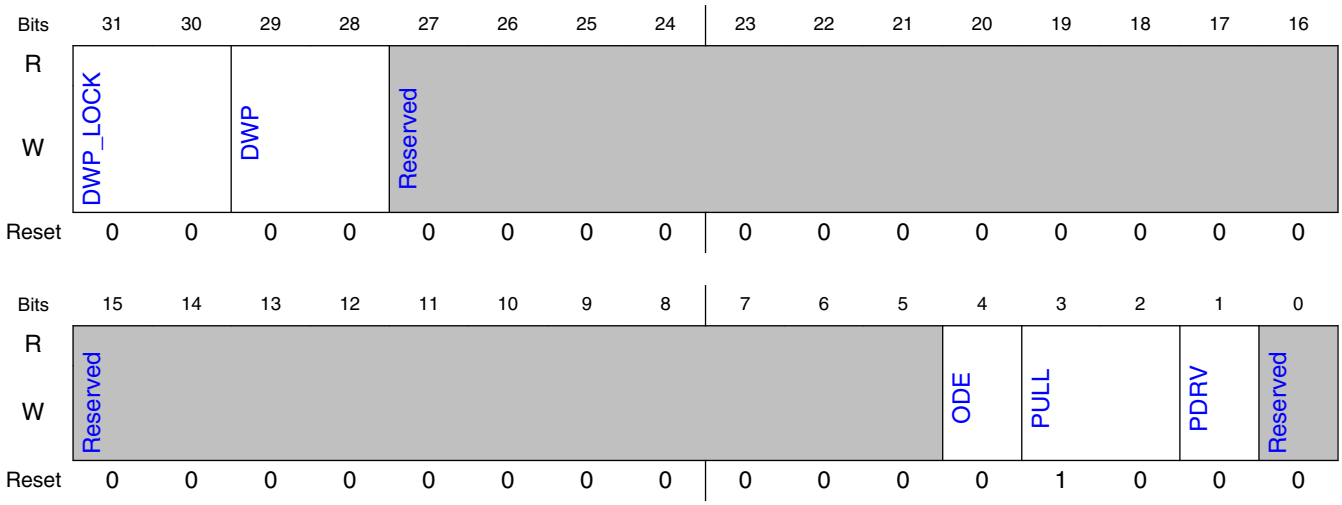
12.4.6.191.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_02	304h

12.4.6.191.2 Function

SW_PAD_CTL Register

12.4.6.191.3 Diagram



12.4.6.191.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_02 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_02 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_02 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.192 SW_PAD_CTL_PAD_GPIO_EMC_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_03)

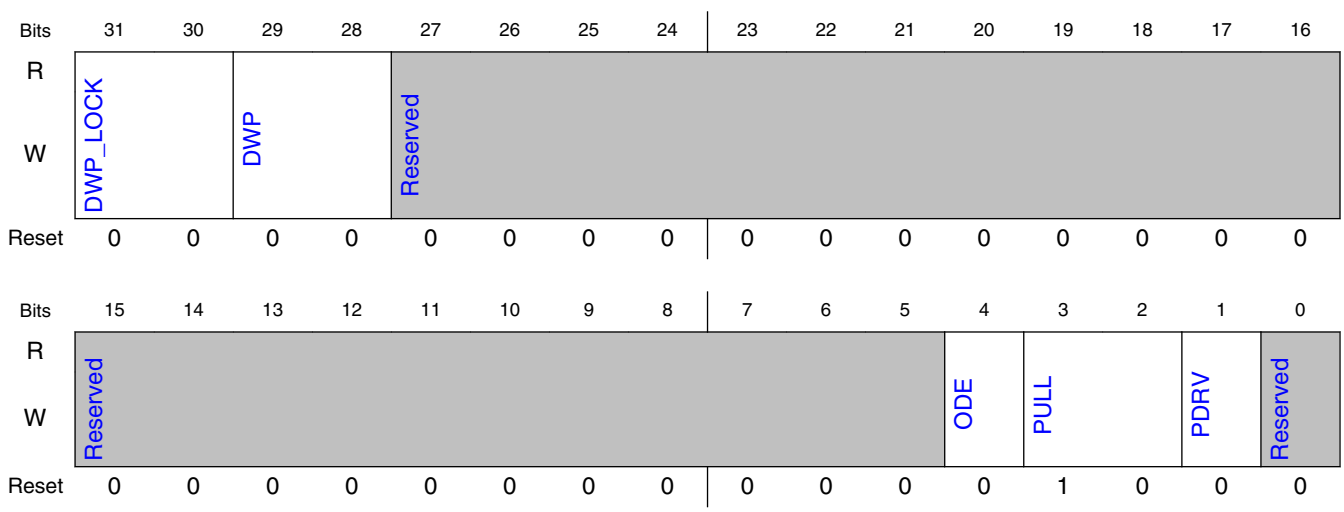
12.4.6.192.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_03	308h

12.4.6.192.2 Function

SW_PAD_CTL Register

12.4.6.192.3 Diagram



12.4.6.192.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_03 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_03 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_03 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.193 SW_PAD_CTL_PAD_GPIO_EMC_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_04)

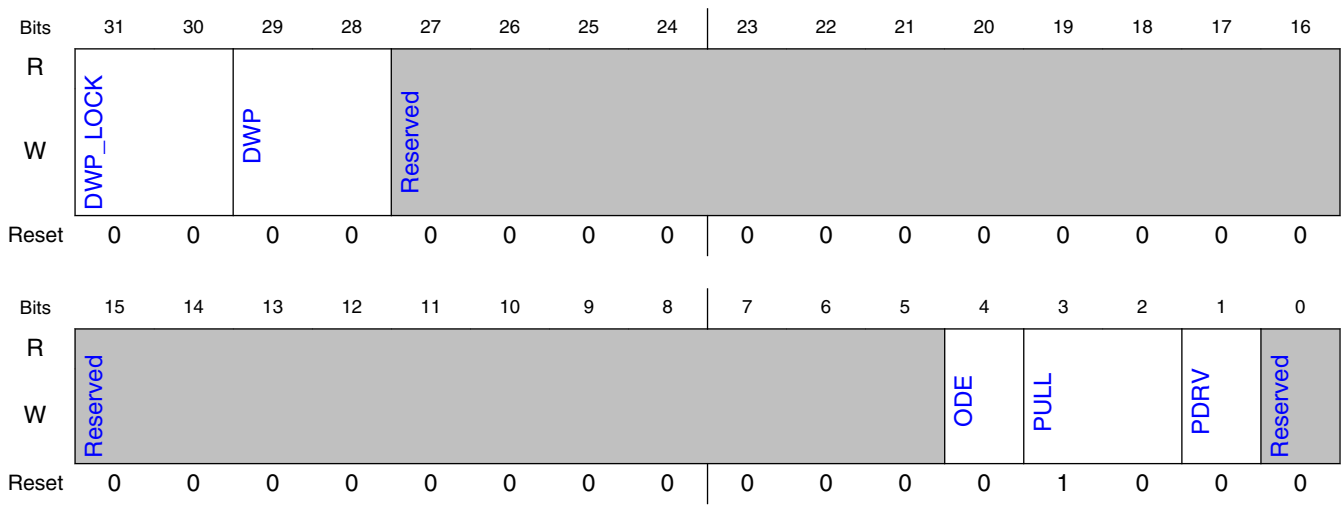
12.4.6.193.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_04	30Ch

12.4.6.193.2 Function

SW_PAD_CTL Register

12.4.6.193.3 Diagram



12.4.6.193.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_04 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_04 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_04 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.194 SW_PAD_CTL_PAD_GPIO_EMC_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_05)

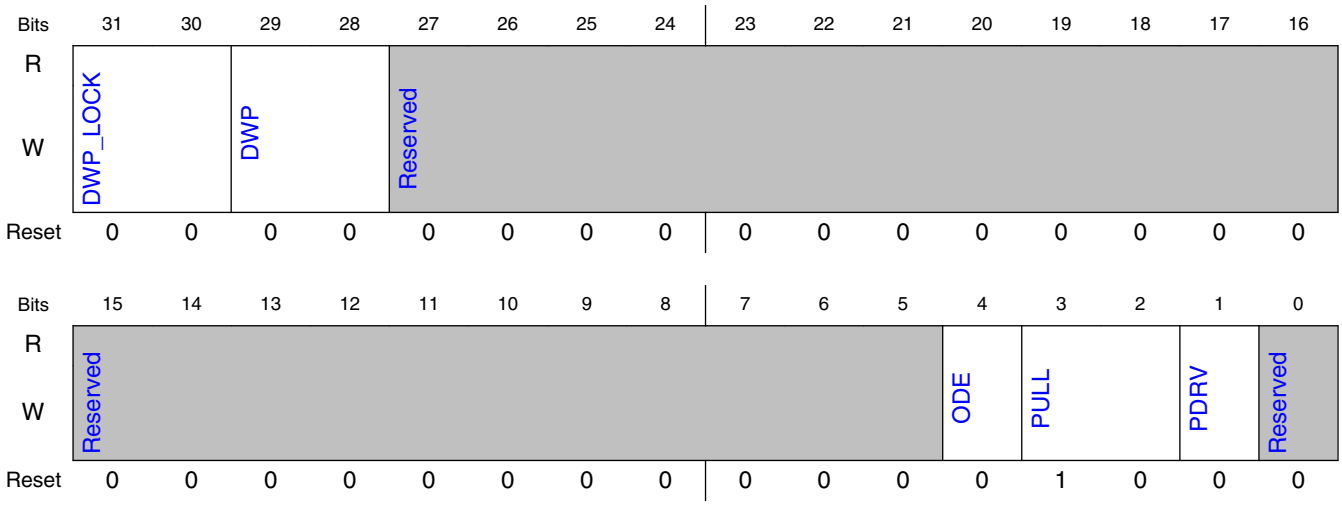
12.4.6.194.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_05	310h

12.4.6.194.2 Function

SW_PAD_CTL Register

12.4.6.194.3 Diagram



12.4.6.194.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_05 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_05 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_05 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.195 SW_PAD_CTL_PAD_GPIO_EMC_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_06)

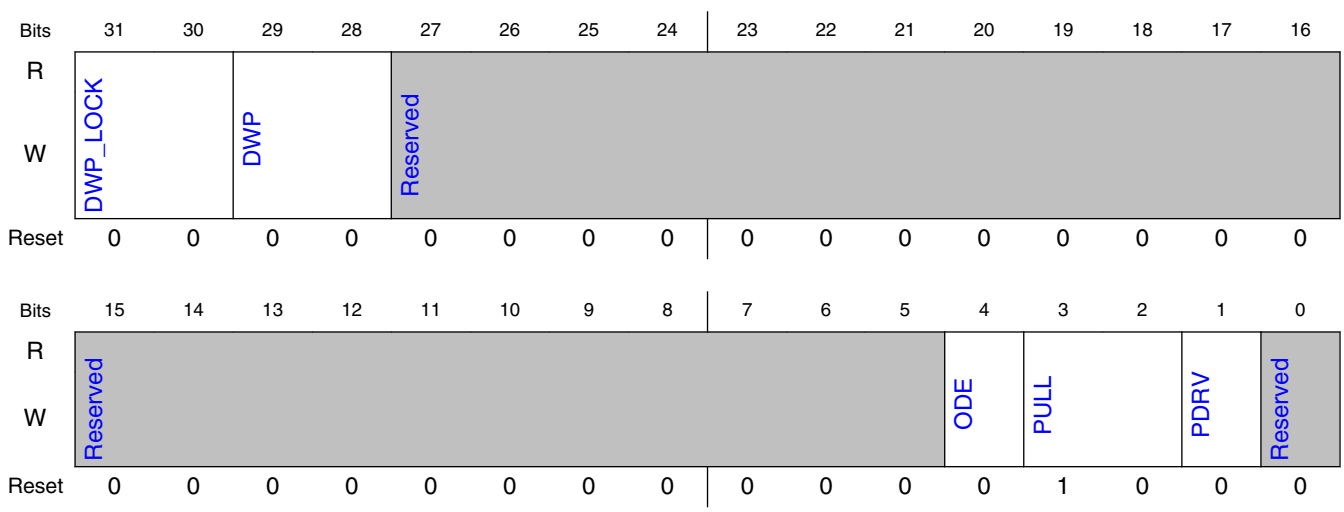
12.4.6.195.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_06	314h

12.4.6.195.2 Function

SW_PAD_CTL Register

12.4.6.195.3 Diagram



12.4.6.195.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_06 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_06 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_06 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.196 SW_PAD_CTL_PAD_GPIO_EMC_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_07)

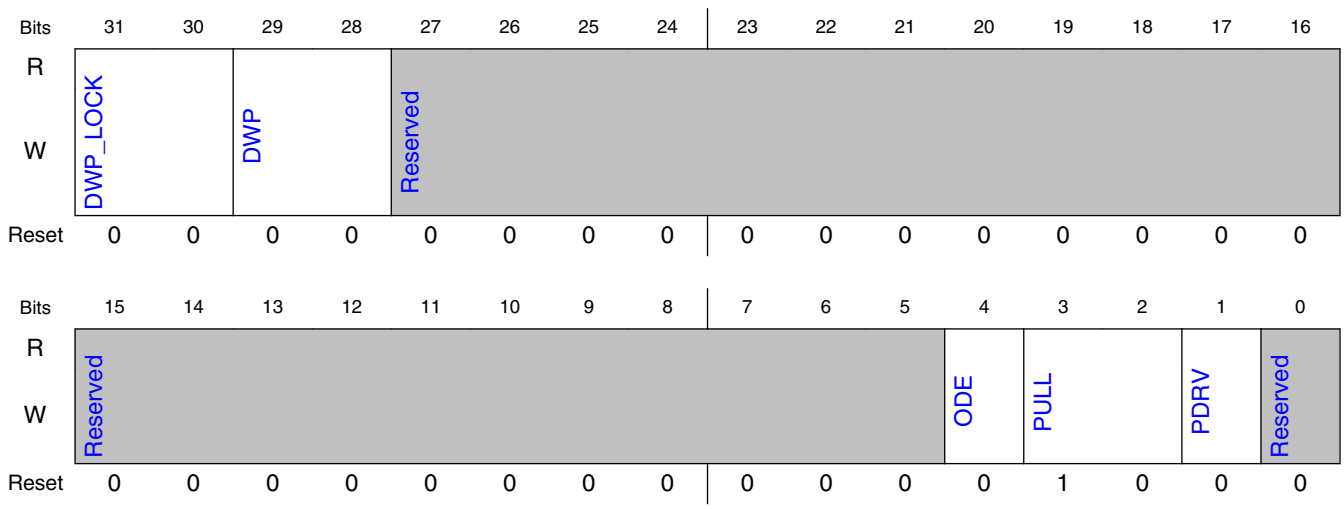
12.4.6.196.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_07	318h

12.4.6.196.2 Function

SW_PAD_CTL Register

12.4.6.196.3 Diagram



12.4.6.196.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_07 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_07 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_07 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.197 SW_PAD_CTL_PAD_GPIO_EMC_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_08)

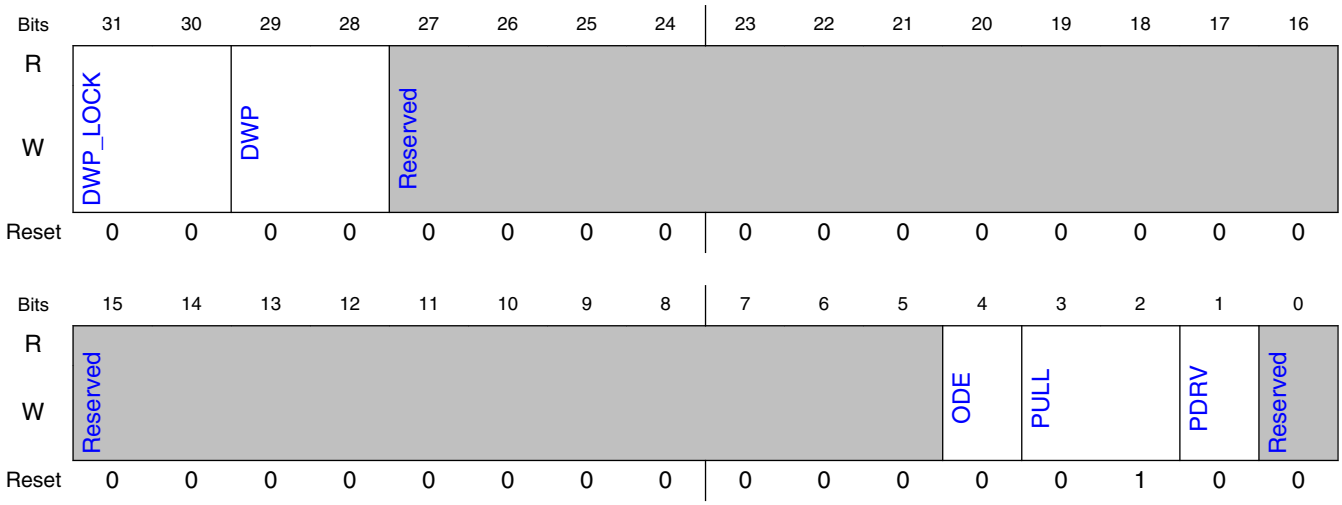
12.4.6.197.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_08	31Ch

12.4.6.197.2 Function

SW_PAD_CTL Register

12.4.6.197.3 Diagram



12.4.6.197.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_08 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_08 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_08 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.198 SW_PAD_CTL_PAD_GPIO_EMC_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_09)

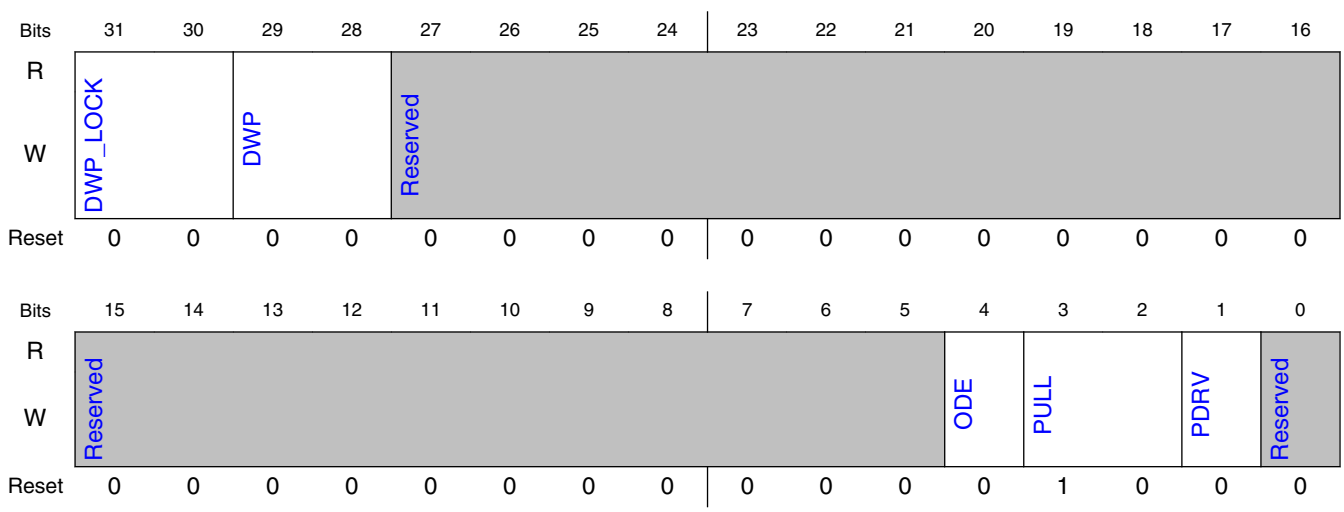
12.4.6.198.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_09	320h

12.4.6.198.2 Function

SW_PAD_CTL Register

12.4.6.198.3 Diagram



12.4.6.198.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_09 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_09 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_09 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.199 SW_PAD_CTL_PAD_GPIO_EMC_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_10)

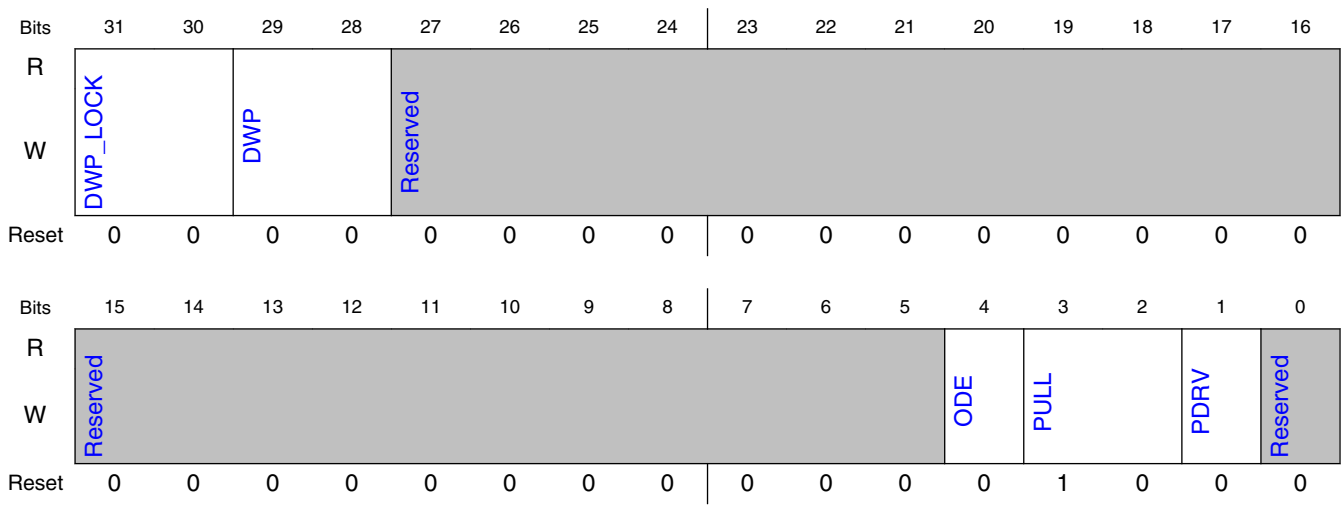
12.4.6.199.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_10	324h

12.4.6.199.2 Function

SW_PAD_CTL Register

12.4.6.199.3 Diagram



12.4.6.199.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_10 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_10 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_10 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.200 SW_PAD_CTL_PAD_GPIO_EMC_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_11)

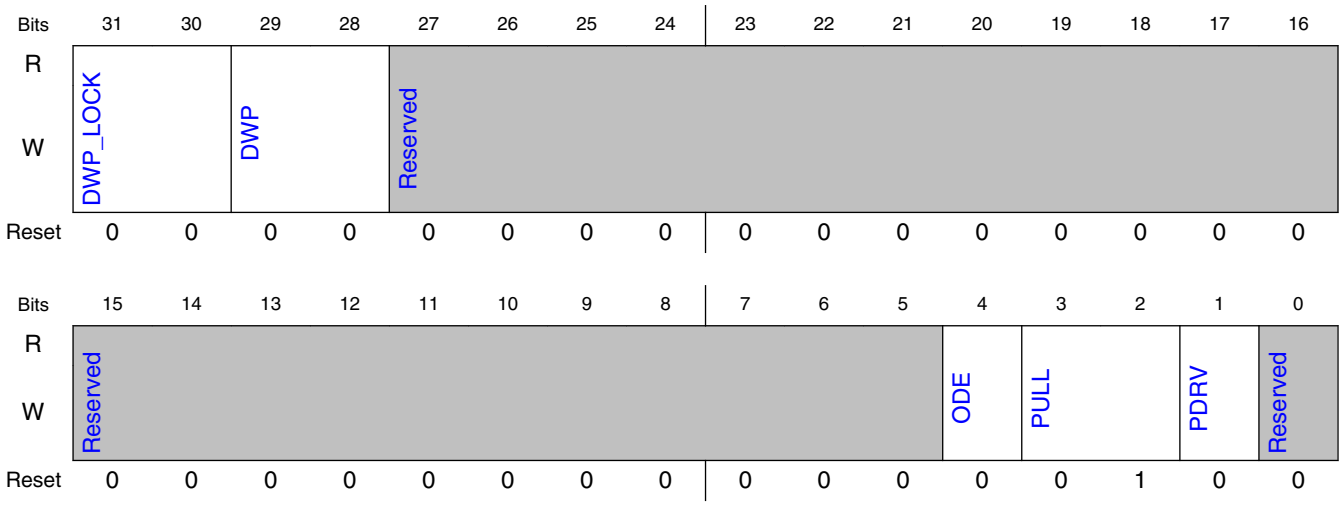
12.4.6.200.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_11	328h

12.4.6.200.2 Function

SW_PAD_CTL Register

12.4.6.200.3 Diagram



12.4.6.200.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_11 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_11 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_11 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.201 SW_PAD_CTL_PAD_GPIO_EMC_B2_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_12)

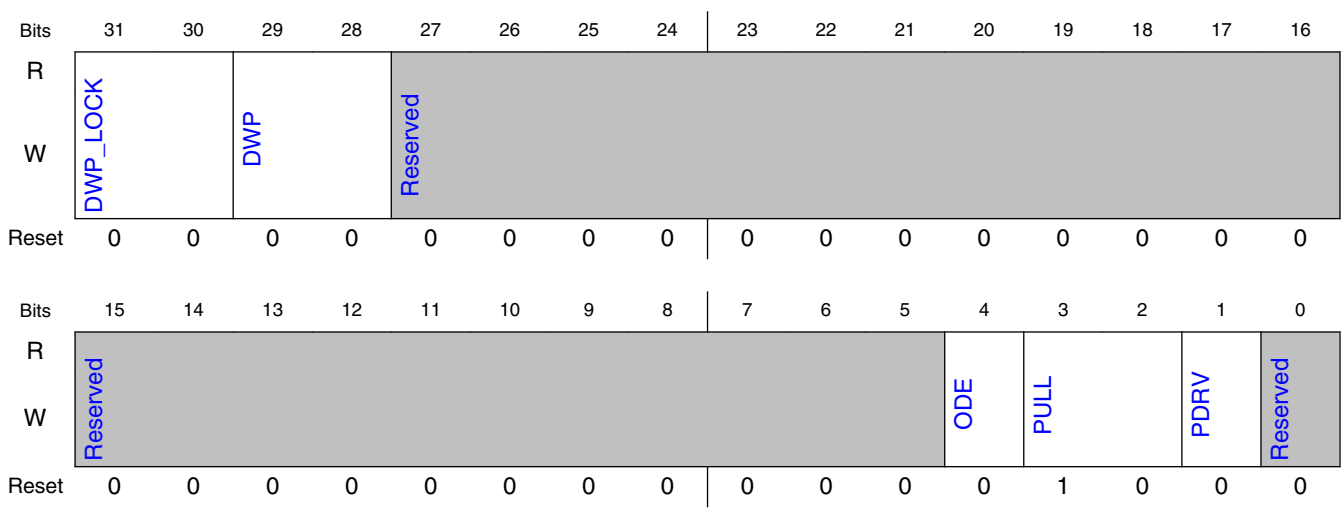
12.4.6.201.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_12	32Ch

12.4.6.201.2 Function

SW_PAD_CTL Register

12.4.6.201.3 Diagram



12.4.6.201.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_12 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_12 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_12 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.202 SW_PAD_CTL_PAD_GPIO_EMC_B2_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_13)

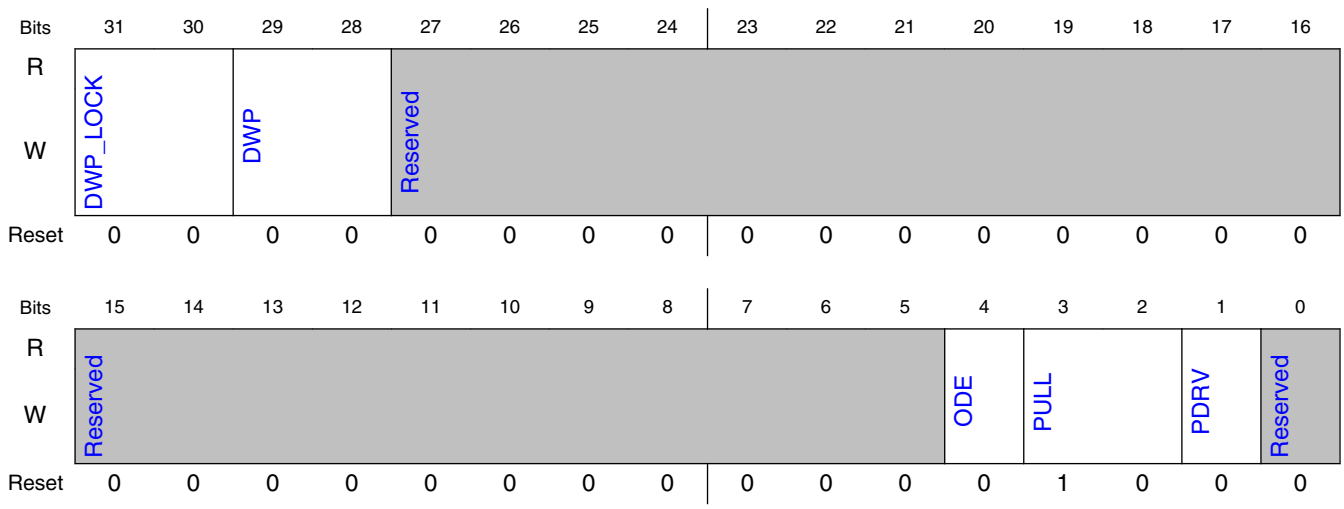
12.4.6.202.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_13	330h

12.4.6.202.2 Function

SW_PAD_CTL Register

12.4.6.202.3 Diagram



12.4.6.202.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_13 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_13 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_13 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.203 SW_PAD_CTL_PAD_GPIO_EMC_B2_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_14)

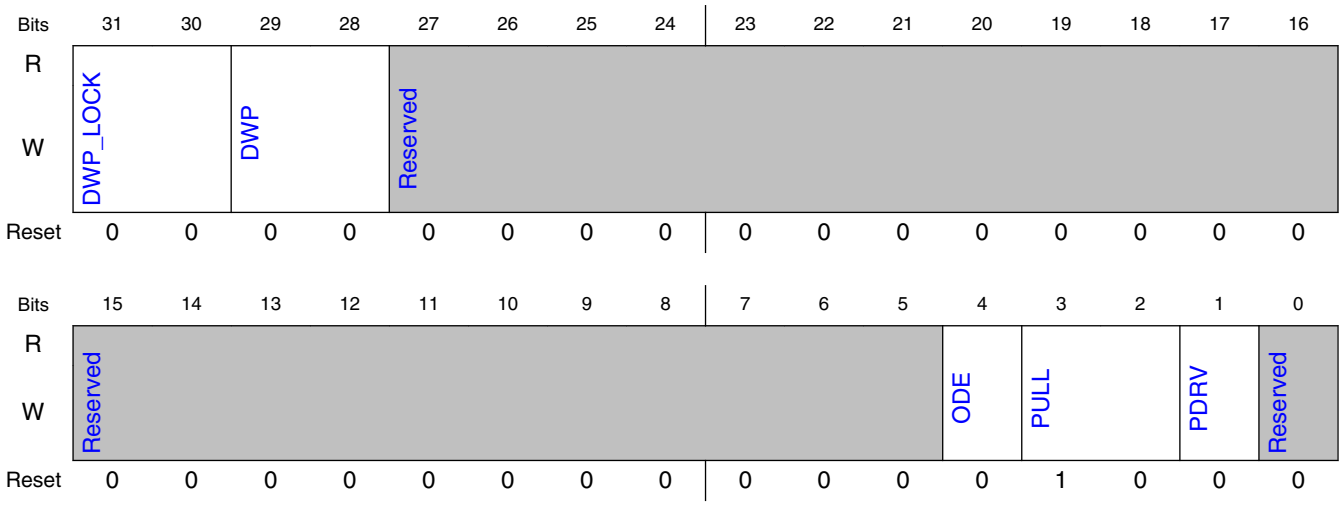
12.4.6.203.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_14	334h

12.4.6.203.2 Function

SW_PAD_CTL Register

12.4.6.203.3 Diagram



12.4.6.203.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_14 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_14 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_14 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.204 SW_PAD_CTL_PAD_GPIO_EMC_B2_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_15)

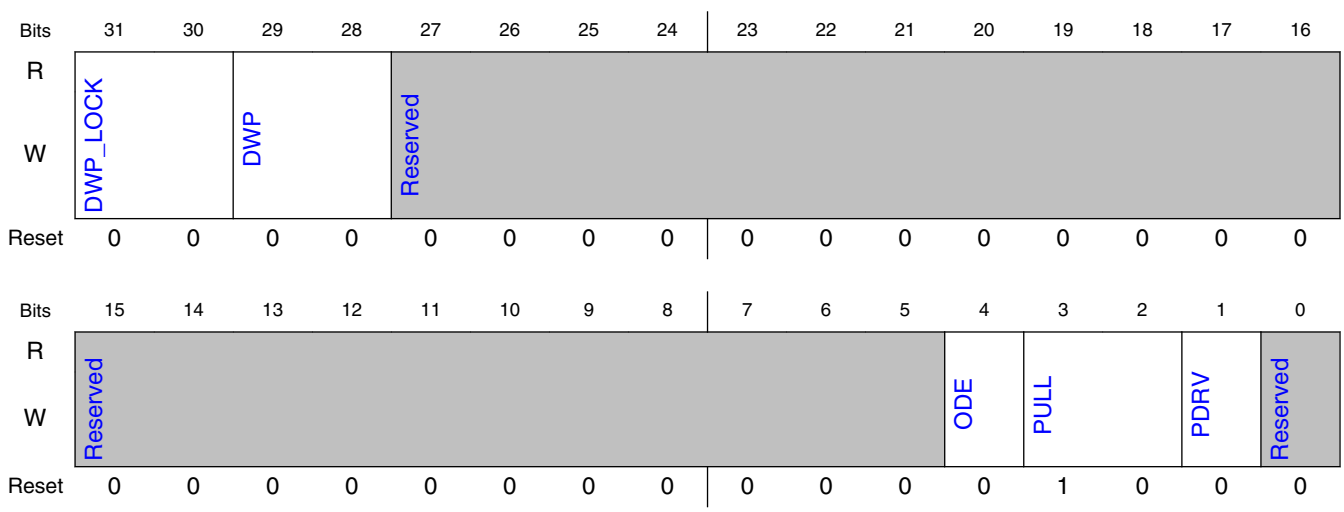
12.4.6.204.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_15	338h

12.4.6.204.2 Function

SW_PAD_CTL Register

12.4.6.204.3 Diagram



12.4.6.204.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_15 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_15 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_15 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.205 SW_PAD_CTL_PAD_GPIO_EMC_B2_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_16)

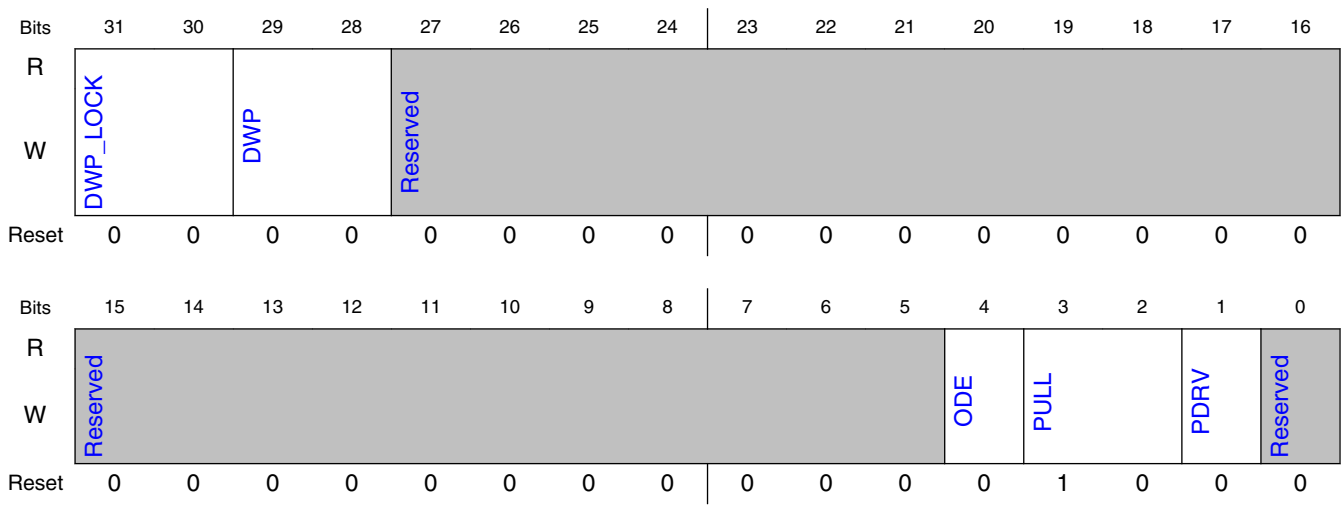
12.4.6.205.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_16	33Ch

12.4.6.205.2 Function

SW_PAD_CTL Register

12.4.6.205.3 Diagram



12.4.6.205.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_16 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_16 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_16 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.206 SW_PAD_CTL_PAD_GPIO_EMC_B2_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_17)

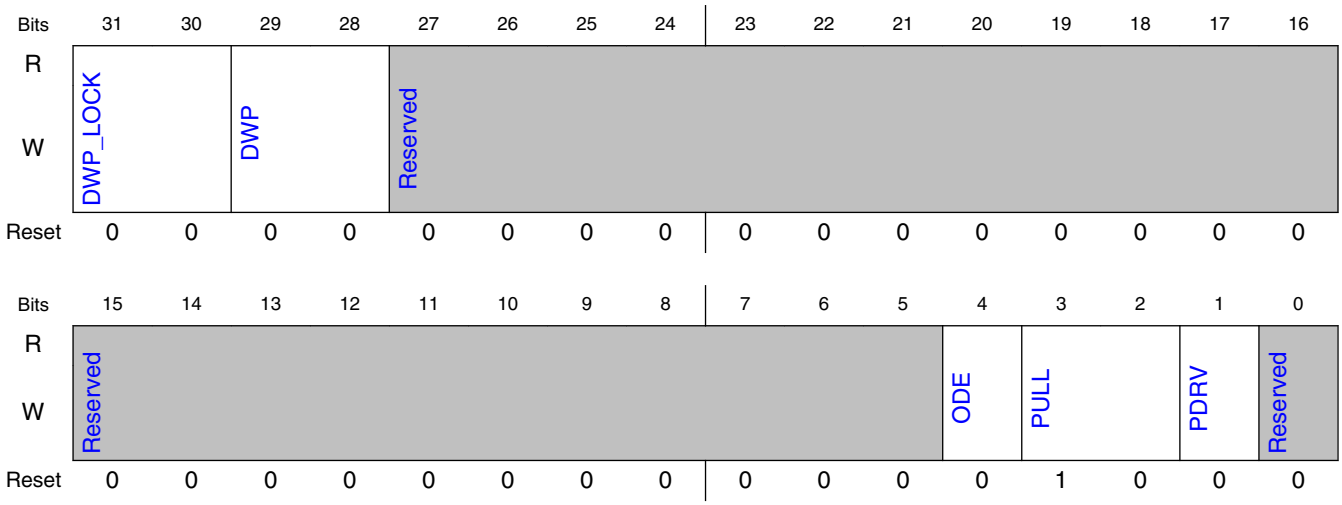
12.4.6.206.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_17	340h

12.4.6.206.2 Function

SW_PAD_CTL Register

12.4.6.206.3 Diagram



12.4.6.206.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_17 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_17 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_17 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.207 SW_PAD_CTL_PAD_GPIO_EMC_B2_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_18)

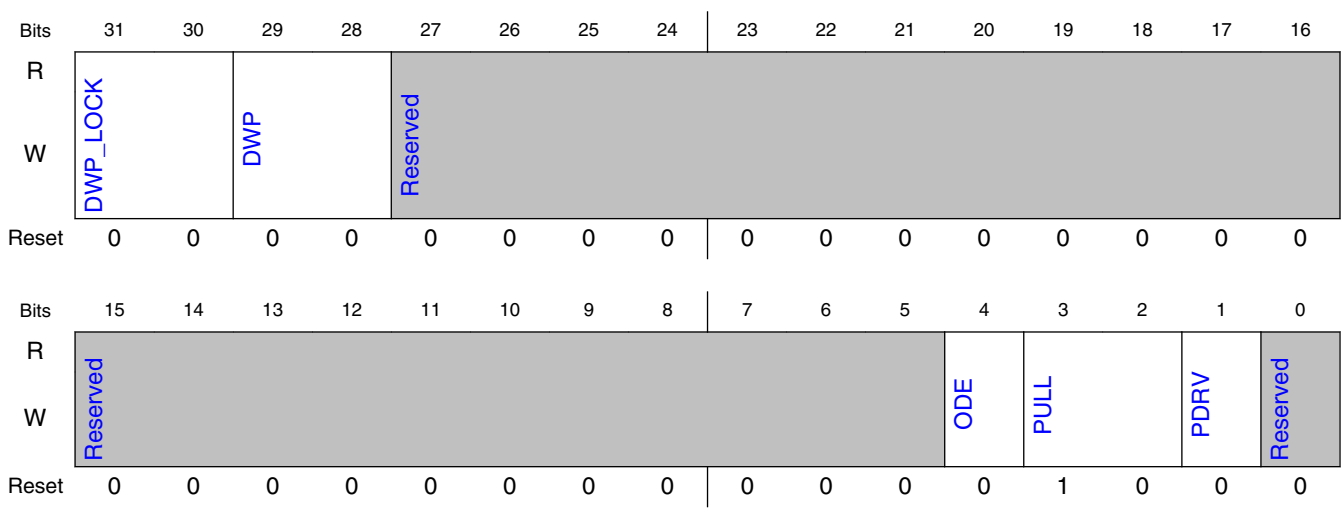
12.4.6.207.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_18	344h

12.4.6.207.2 Function

SW_PAD_CTL Register

12.4.6.207.3 Diagram



12.4.6.207.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_18 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_18 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_18 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.208 SW_PAD_CTL_PAD_GPIO_EMC_B2_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_19)

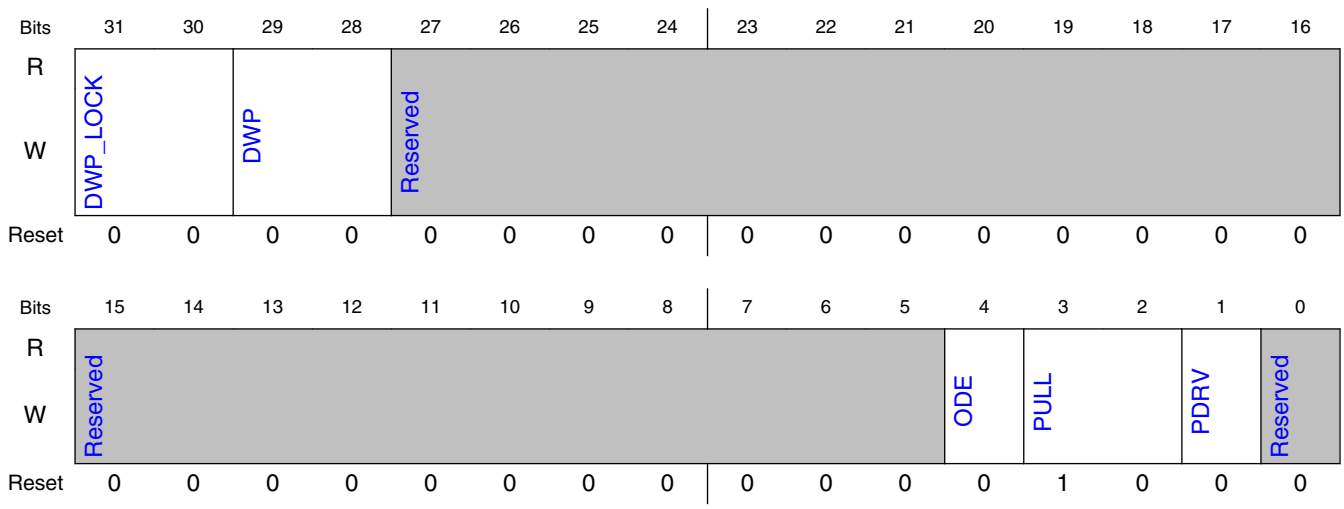
12.4.6.208.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_19	348h

12.4.6.208.2 Function

SW_PAD_CTL Register

12.4.6.208.3 Diagram



12.4.6.208.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_19 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_19 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_19 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.209 SW_PAD_CTL_PAD_GPIO_EMC_B2_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_EMC_B2_20)

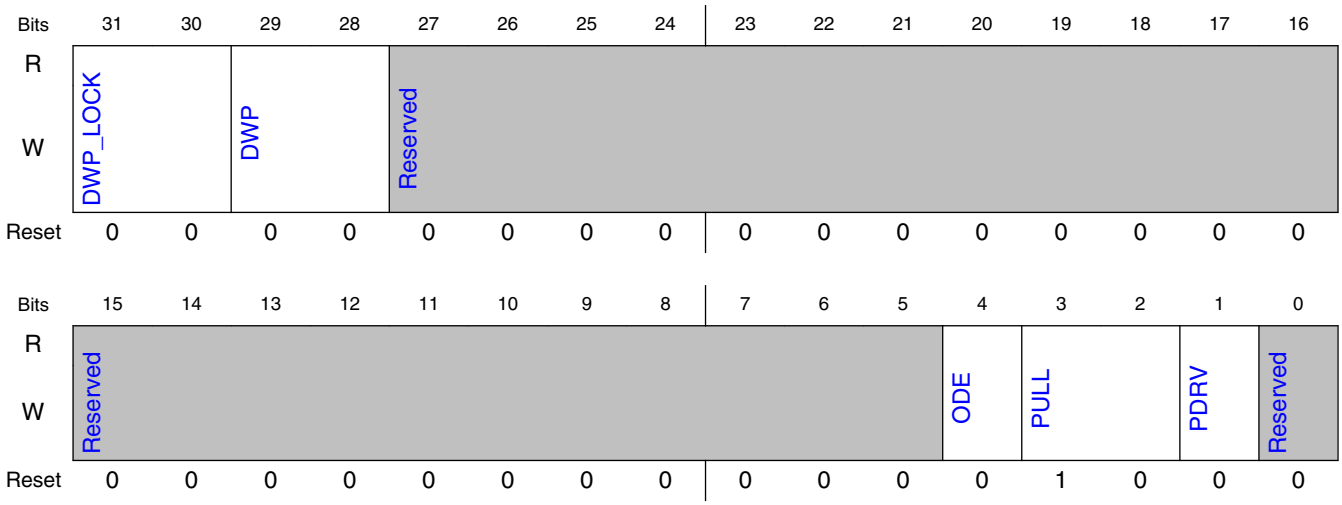
12.4.6.209.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_EMC_B2_20	34Ch

12.4.6.209.2 Function

SW_PAD_CTL Register

12.4.6.209.3 Diagram



12.4.6.209.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_EMC_B2_20 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_EMC_B2_20 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_EMC_B2_20 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.210 SW_PAD_CTL_PAD_GPIO_AD_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_00)

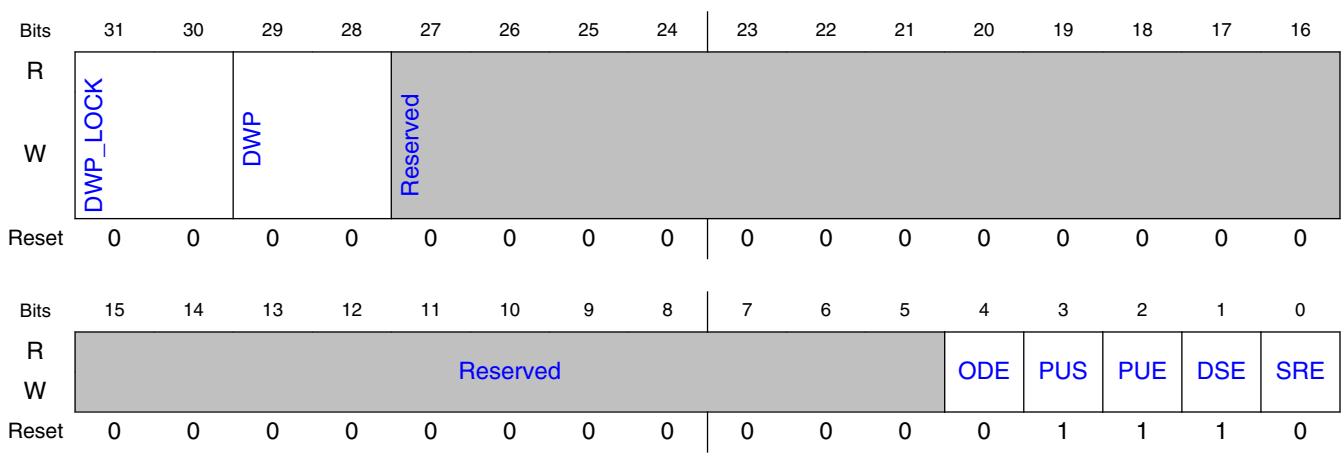
12.4.6.210.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_00	350h

12.4.6.210.2 Function

SW_PAD_CTL Register

12.4.6.210.3 Diagram



12.4.6.210.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_00 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_00 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_00 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_00 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_00 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.211 SW_PAD_CTL_PAD_GPIO_AD_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_01)

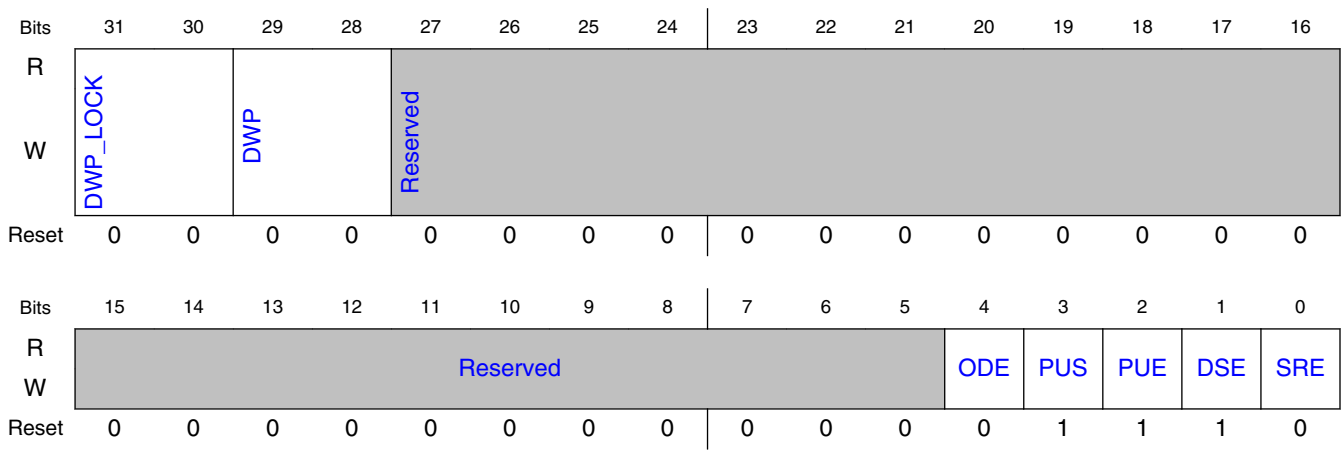
12.4.6.211.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_01	354h

12.4.6.211.2 Function

SW_PAD_CTL Register

12.4.6.211.3 Diagram



12.4.6.211.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_01 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_01 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_01 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_01 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_01 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.212 SW_PAD_CTL_PAD_GPIO_AD_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_02)

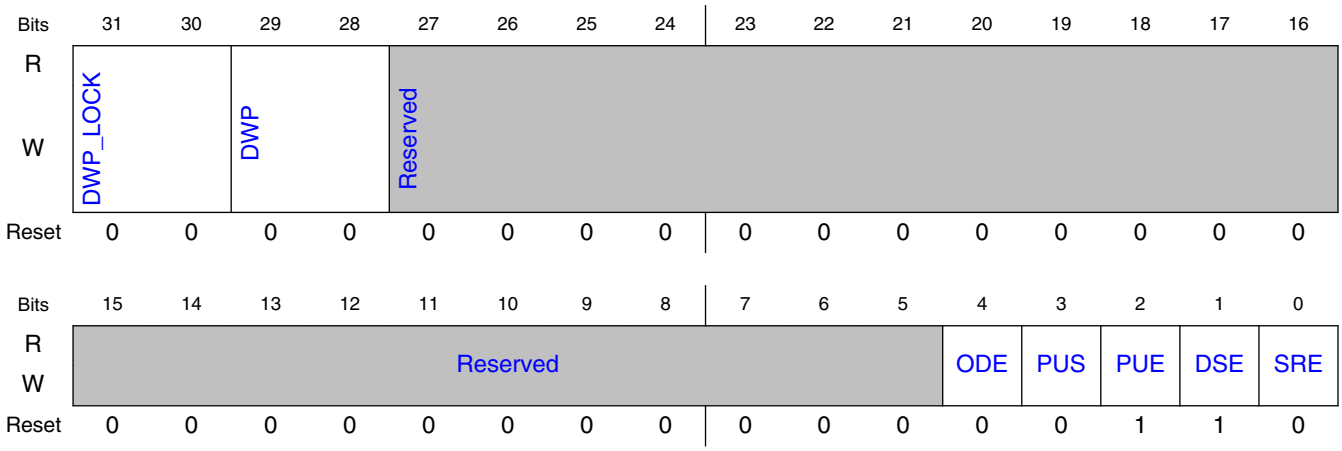
12.4.6.212.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_02	358h

12.4.6.212.2 Function

SW_PAD_CTL Register

12.4.6.212.3 Diagram



12.4.6.212.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_02 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_02 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_02

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_02 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_02 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.213 SW_PAD_CTL_PAD_GPIO_AD_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_03)

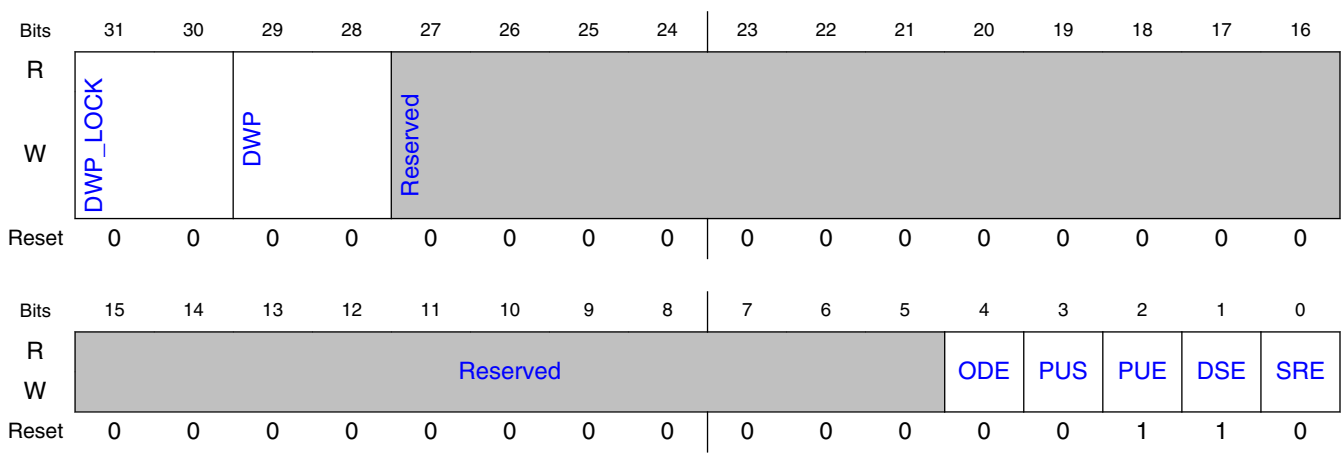
12.4.6.213.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_03	35Ch

12.4.6.213.2 Function

SW_PAD_CTL Register

12.4.6.213.3 Diagram



12.4.6.213.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_03 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_03 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_03 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_03 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_03 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.214 SW_PAD_CTL_PAD_GPIO_AD_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_04)

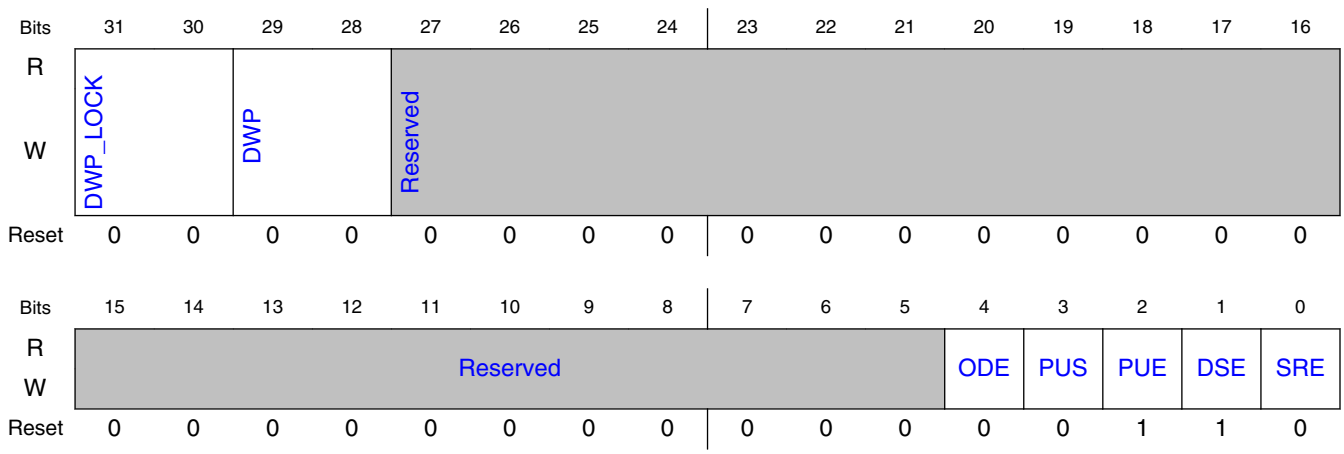
12.4.6.214.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_04	360h

12.4.6.214.2 Function

SW_PAD_CTL Register

12.4.6.214.3 Diagram



12.4.6.214.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_04 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_04 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_04 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_04 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_04 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.215 SW_PAD_CTL_PAD_GPIO_AD_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_05)

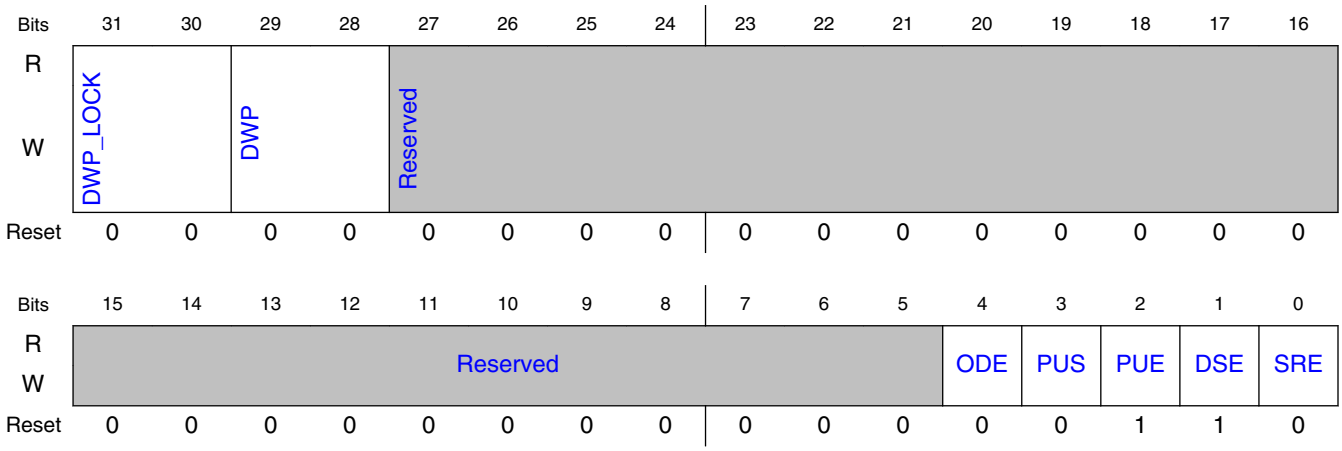
12.4.6.215.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_05	364h

12.4.6.215.2 Function

SW_PAD_CTL Register

12.4.6.215.3 Diagram



12.4.6.215.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_05 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_05 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_05

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_05 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_05 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.216 SW_PAD_CTL_PAD_GPIO_AD_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_06)

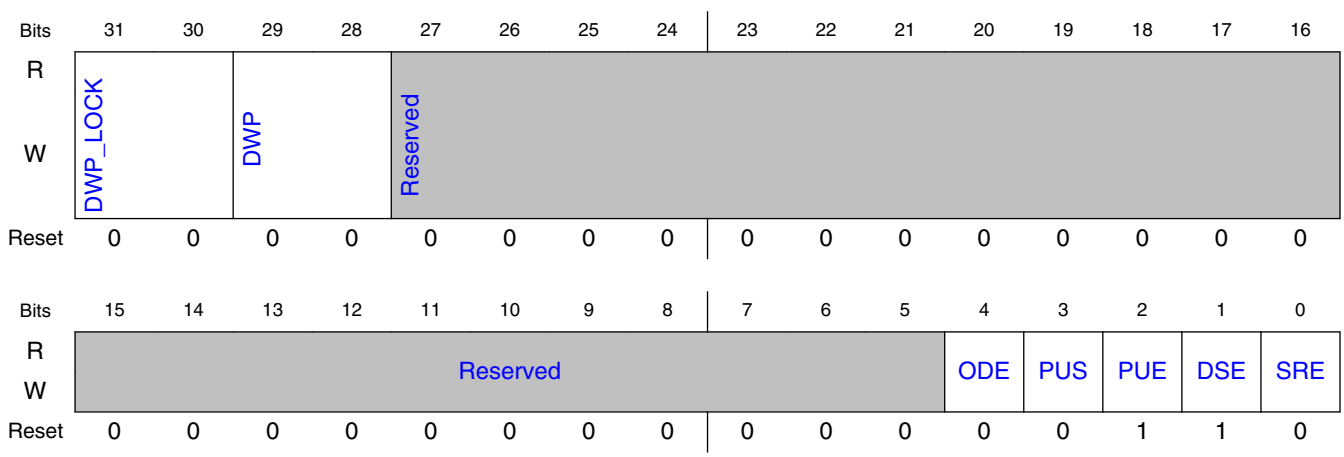
12.4.6.216.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_06	368h

12.4.6.216.2 Function

SW_PAD_CTL Register

12.4.6.216.3 Diagram



12.4.6.216.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_06 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_06 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_06 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_06 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_06 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.217 SW_PAD_CTL_PAD_GPIO_AD_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_07)

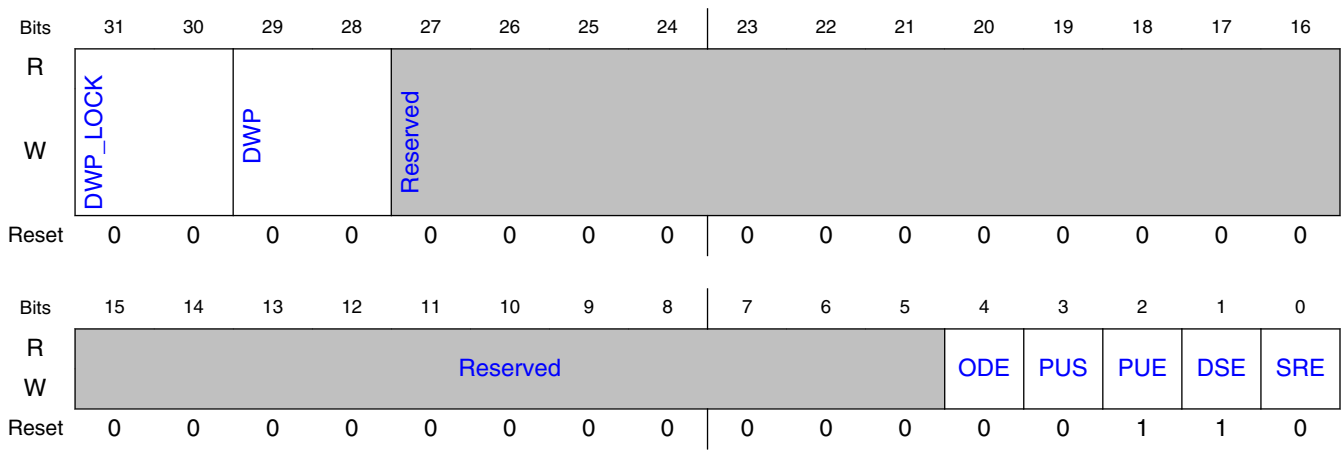
12.4.6.217.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_07	36Ch

12.4.6.217.2 Function

SW_PAD_CTL Register

12.4.6.217.3 Diagram



12.4.6.217.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_07 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_07 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_07 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_07 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_07 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.218 SW_PAD_CTL_PAD_GPIO_AD_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_08)

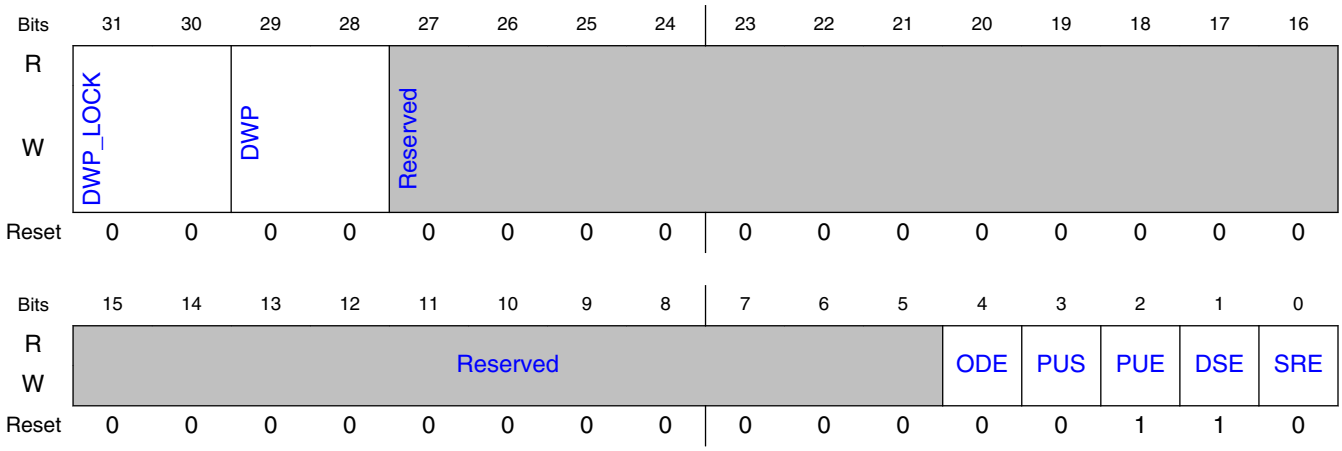
12.4.6.218.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_08	370h

12.4.6.218.2 Function

SW_PAD_CTL Register

12.4.6.218.3 Diagram



12.4.6.218.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_08 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_08 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_08

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_08 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_08 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.219 SW_PAD_CTL_PAD_GPIO_AD_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_09)

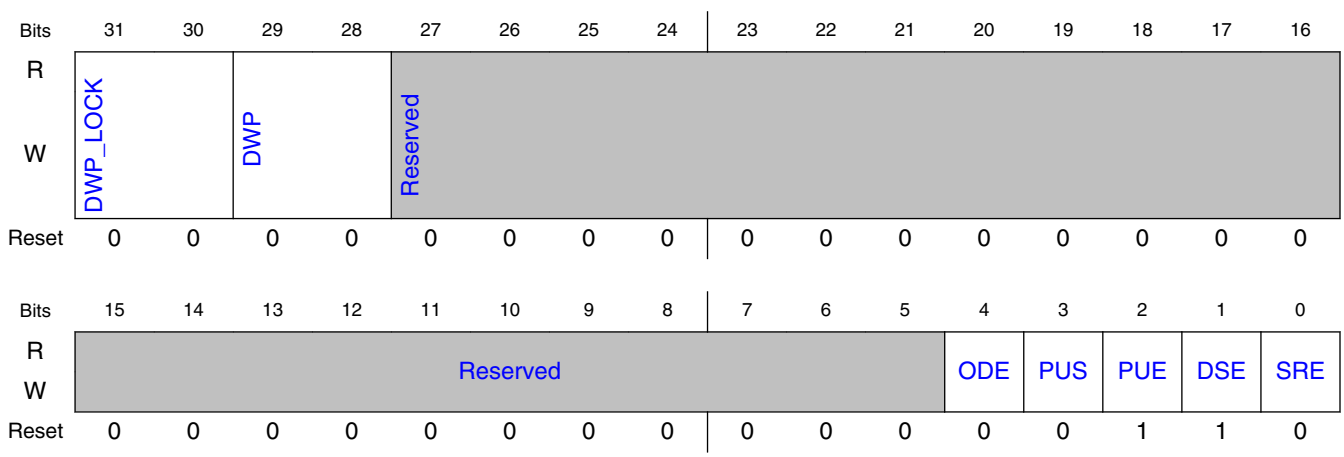
12.4.6.219.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_09	374h

12.4.6.219.2 Function

SW_PAD_CTL Register

12.4.6.219.3 Diagram



12.4.6.219.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_09 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_09 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_09 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_09 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_09 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.220 SW_PAD_CTL_PAD_GPIO_AD_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_10)

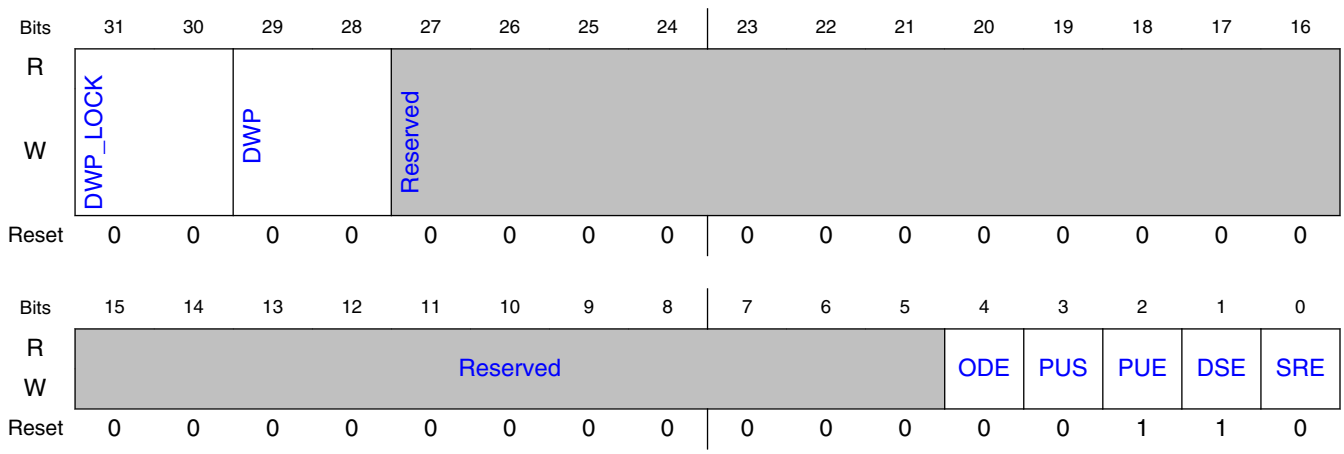
12.4.6.220.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_10	378h

12.4.6.220.2 Function

SW_PAD_CTL Register

12.4.6.220.3 Diagram



12.4.6.220.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_10 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_10 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_10 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_10 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_10 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.221 SW_PAD_CTL_PAD_GPIO_AD_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_11)

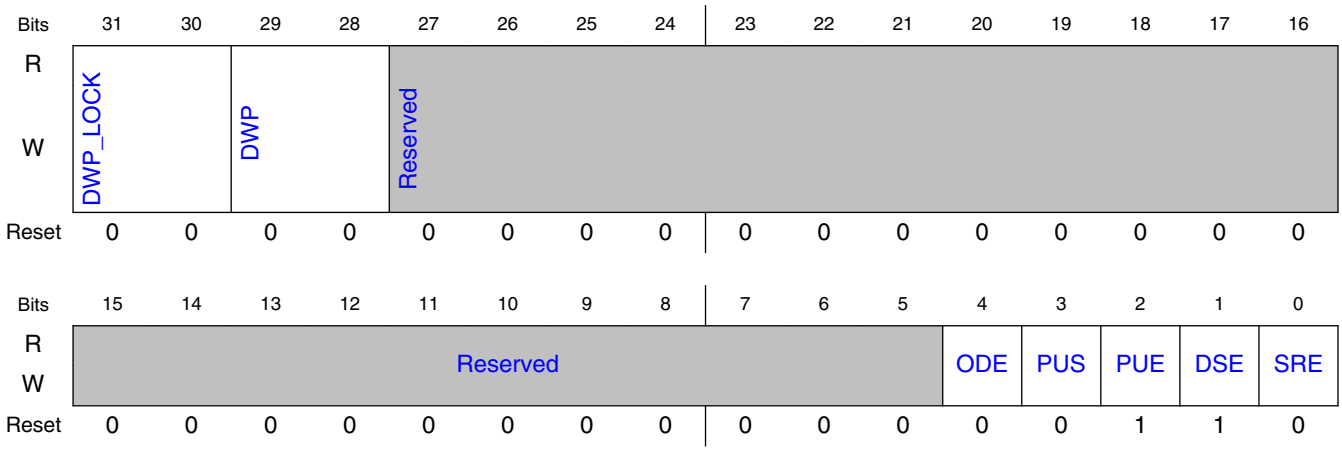
12.4.6.221.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_11	37Ch

12.4.6.221.2 Function

SW_PAD_CTL Register

12.4.6.221.3 Diagram



12.4.6.221.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_11 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_11 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_11

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_11 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_11 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.222 SW_PAD_CTL_PAD_GPIO_AD_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_12)

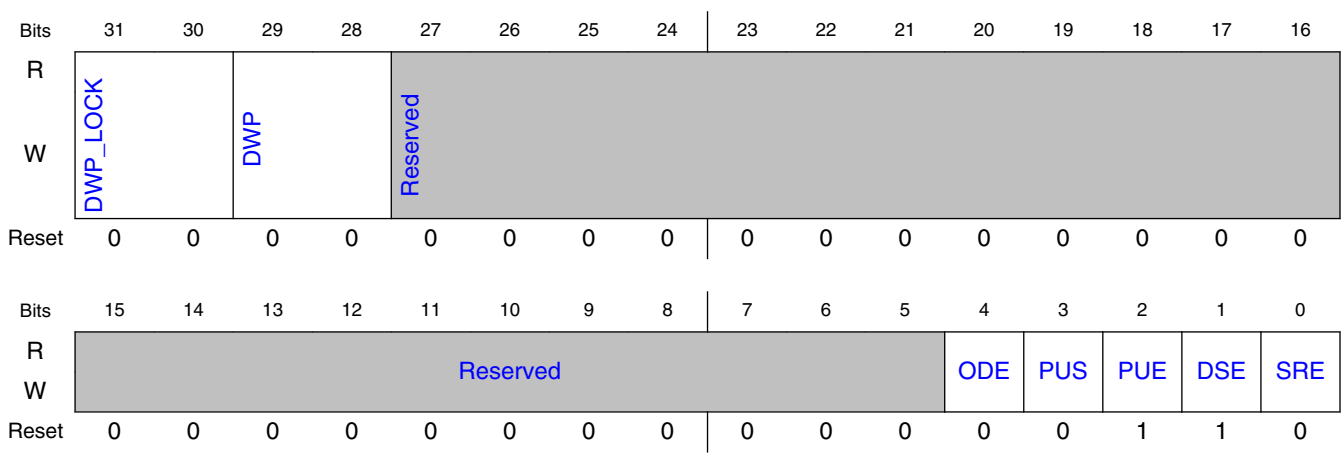
12.4.6.222.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_12	380h

12.4.6.222.2 Function

SW_PAD_CTL Register

12.4.6.222.3 Diagram



12.4.6.222.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_12 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_12 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_12 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_12 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_12 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.223 SW_PAD_CTL_PAD_GPIO_AD_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_13)

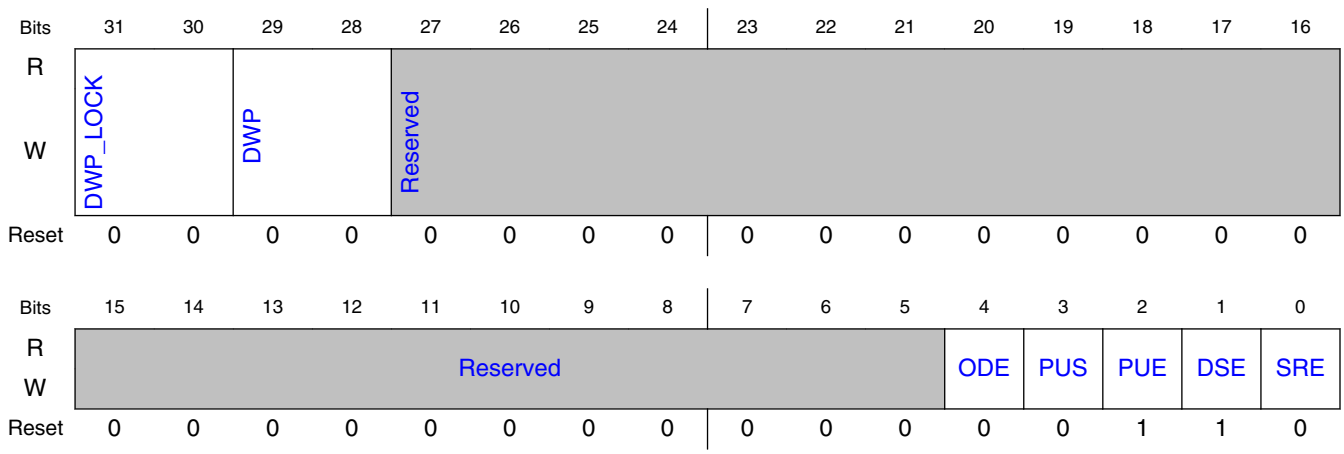
12.4.6.223.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_13	384h

12.4.6.223.2 Function

SW_PAD_CTL Register

12.4.6.223.3 Diagram



12.4.6.223.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_13 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_13 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_13 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_13 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_13 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.224 SW_PAD_CTL_PAD_GPIO_AD_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_14)

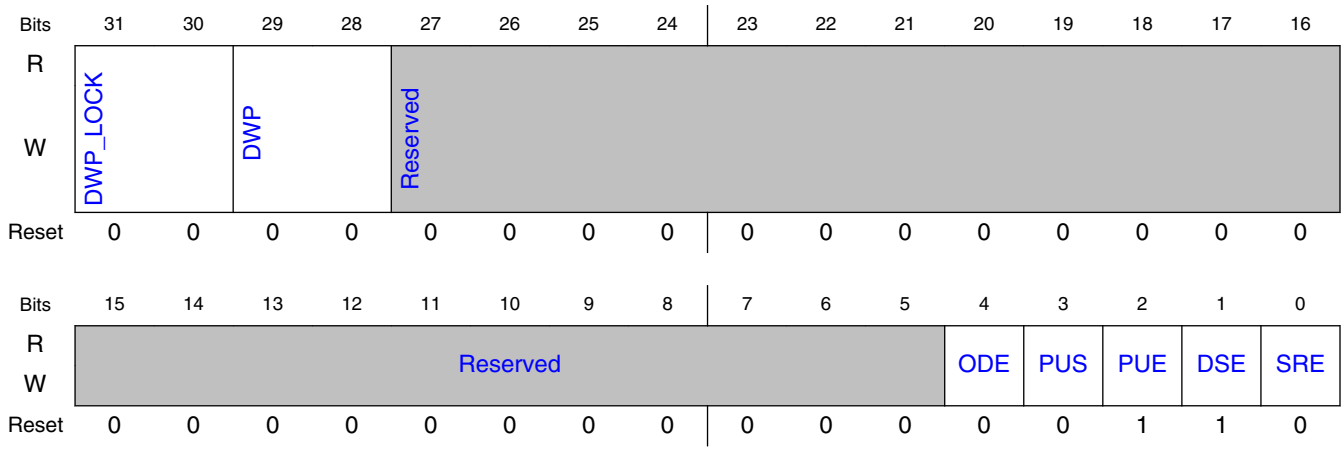
12.4.6.224.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_14	388h

12.4.6.224.2 Function

SW_PAD_CTL Register

12.4.6.224.3 Diagram



12.4.6.224.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_14 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_14 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_14

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_14 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_14 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.225 SW_PAD_CTL_PAD_GPIO_AD_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_15)

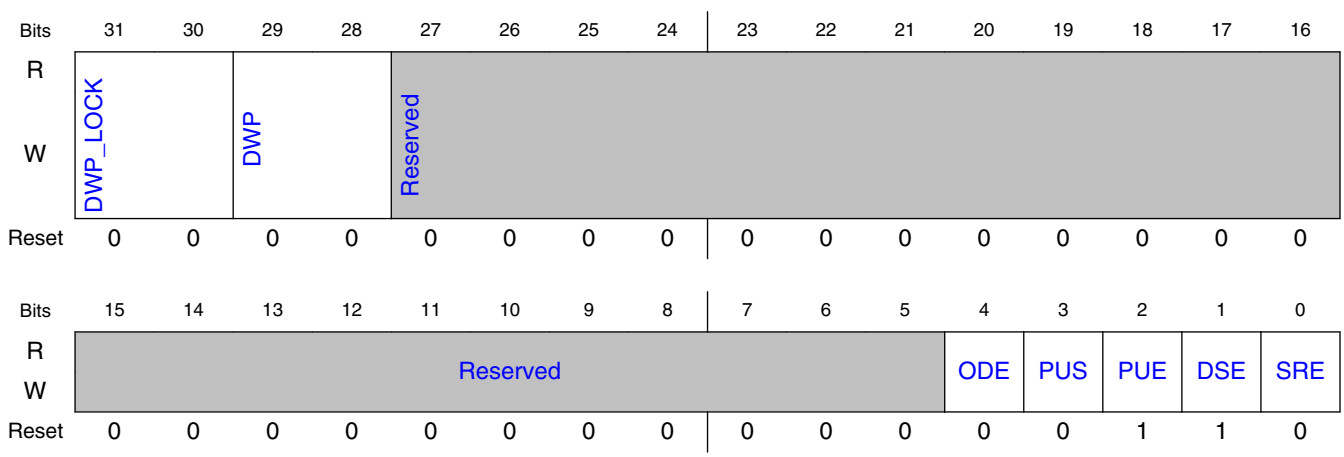
12.4.6.225.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_15	38Ch

12.4.6.225.2 Function

SW_PAD_CTL Register

12.4.6.225.3 Diagram



12.4.6.225.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_15 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_15 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_15 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_15 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_15 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.226 SW_PAD_CTL_PAD_GPIO_AD_16 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_16)

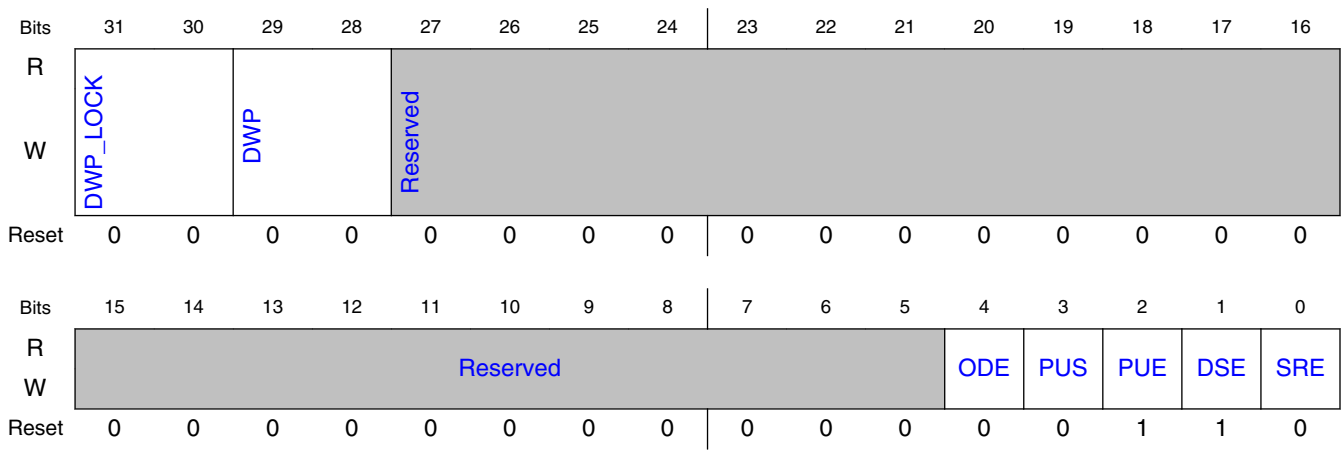
12.4.6.226.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_16	390h

12.4.6.226.2 Function

SW_PAD_CTL Register

12.4.6.226.3 Diagram



12.4.6.226.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_16 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_16 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_16 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_16 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_16 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.227 SW_PAD_CTL_PAD_GPIO_AD_17 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_17)

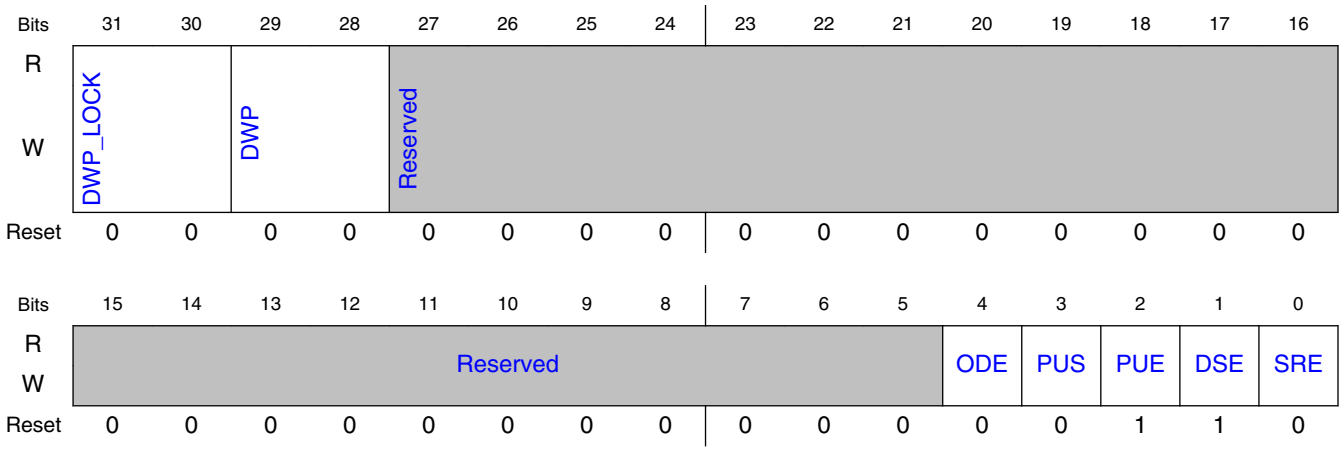
12.4.6.227.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_17	394h

12.4.6.227.2 Function

SW_PAD_CTL Register

12.4.6.227.3 Diagram



12.4.6.227.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_17 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_17 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_17

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_17 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_17 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.228 SW_PAD_CTL_PAD_GPIO_AD_18 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_18)

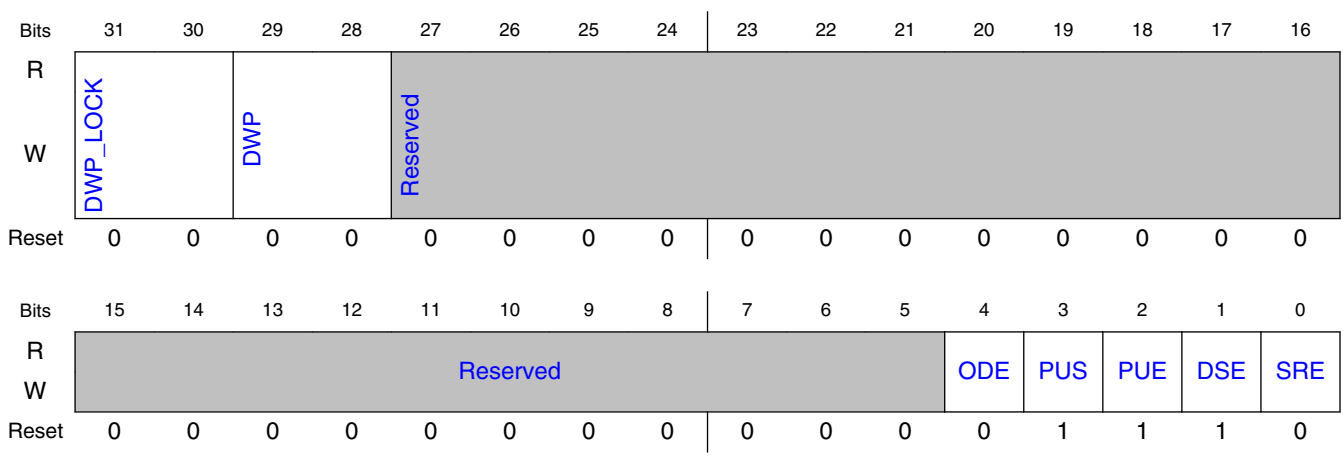
12.4.6.228.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_18	398h

12.4.6.228.2 Function

SW_PAD_CTL Register

12.4.6.228.3 Diagram



12.4.6.228.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_18 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_18 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_18 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_18 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_18 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.229 SW_PAD_CTL_PAD_GPIO_AD_19 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_19)

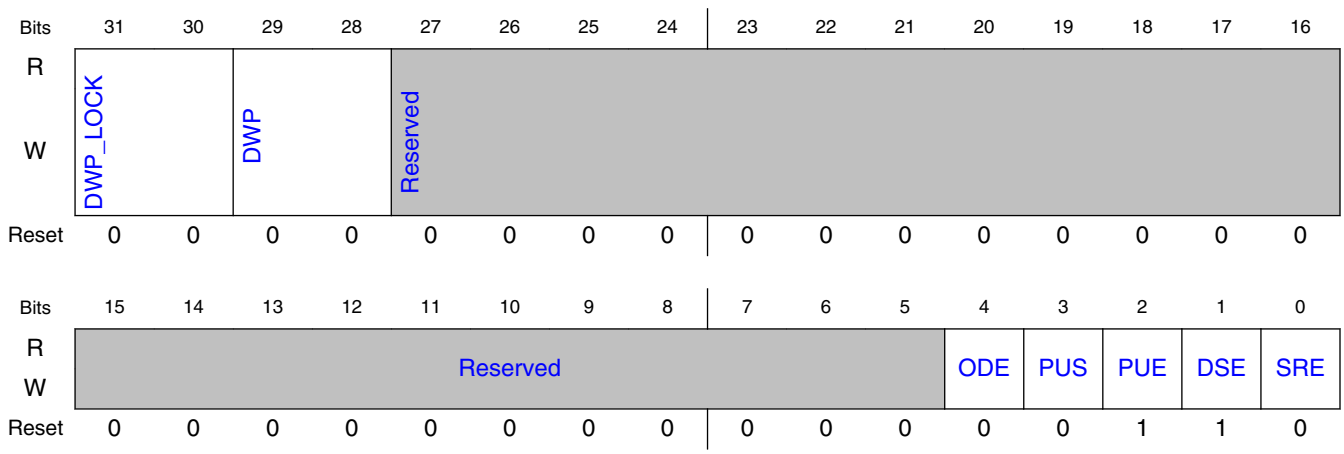
12.4.6.229.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_19	39Ch

12.4.6.229.2 Function

SW_PAD_CTL Register

12.4.6.229.3 Diagram



12.4.6.229.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_19 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_19 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_19 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_19 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_19 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.230 SW_PAD_CTL_PAD_GPIO_AD_20 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_20)

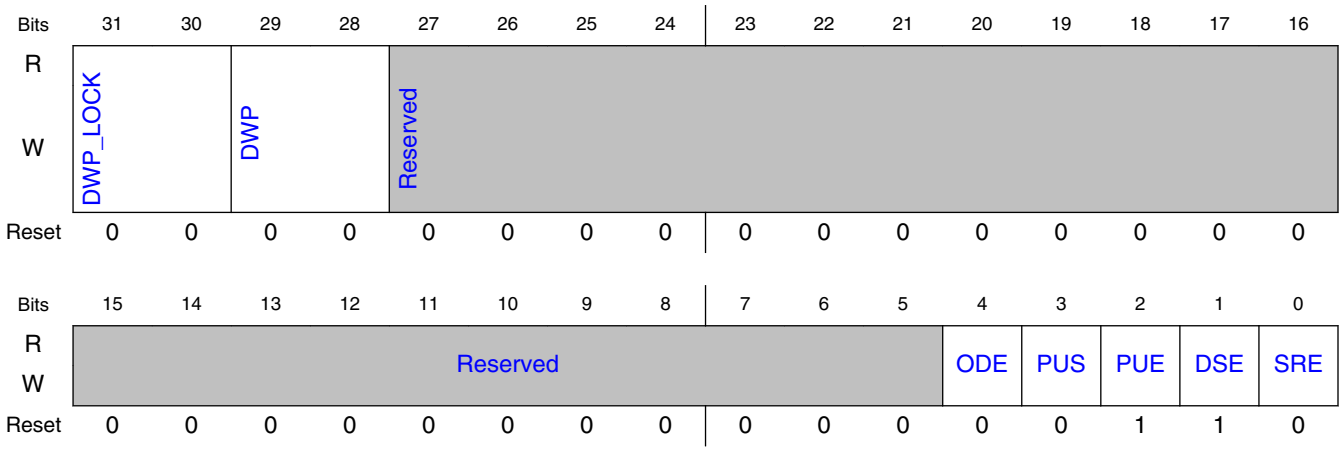
12.4.6.230.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_20	3A0h

12.4.6.230.2 Function

SW_PAD_CTL Register

12.4.6.230.3 Diagram



12.4.6.230.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_20 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_20 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_20

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_20 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_20 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.231 SW_PAD_CTL_PAD_GPIO_AD_21 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_21)

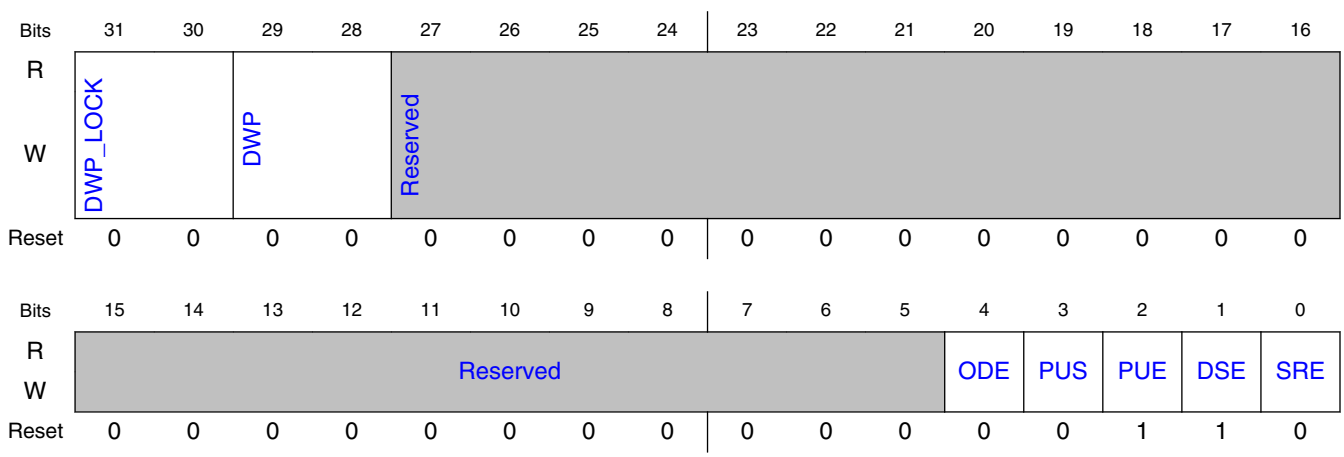
12.4.6.231.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_21	3A4h

12.4.6.231.2 Function

SW_PAD_CTL Register

12.4.6.231.3 Diagram



12.4.6.231.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_21 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_21 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_21 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_21 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_21 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.232 SW_PAD_CTL_PAD_GPIO_AD_22 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_22)

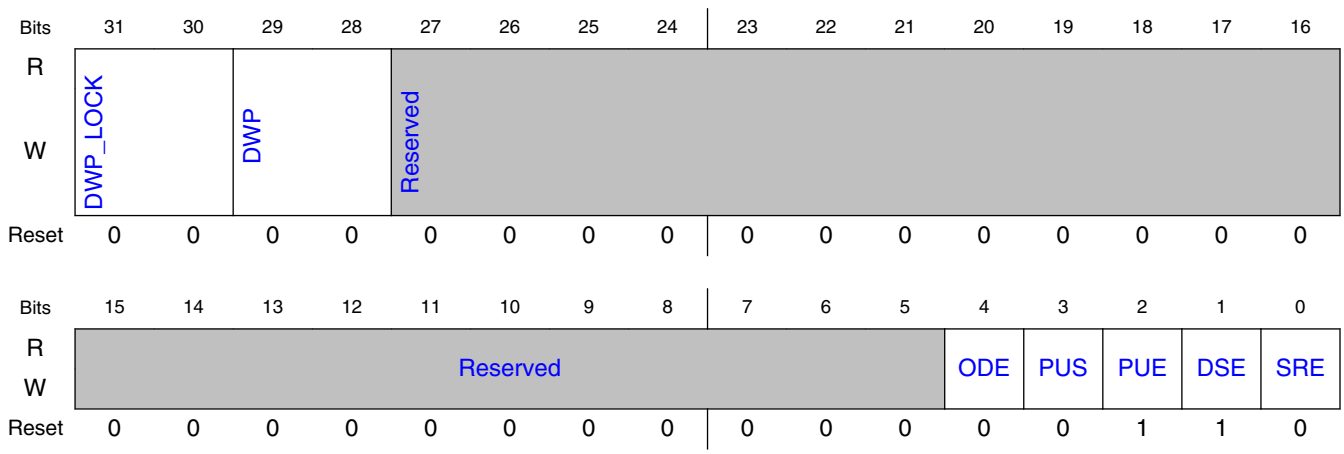
12.4.6.232.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_22	3A8h

12.4.6.232.2 Function

SW_PAD_CTL Register

12.4.6.232.3 Diagram



12.4.6.232.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_22 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_22 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_22 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_22 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_22 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.233 SW_PAD_CTL_PAD_GPIO_AD_23 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_23)

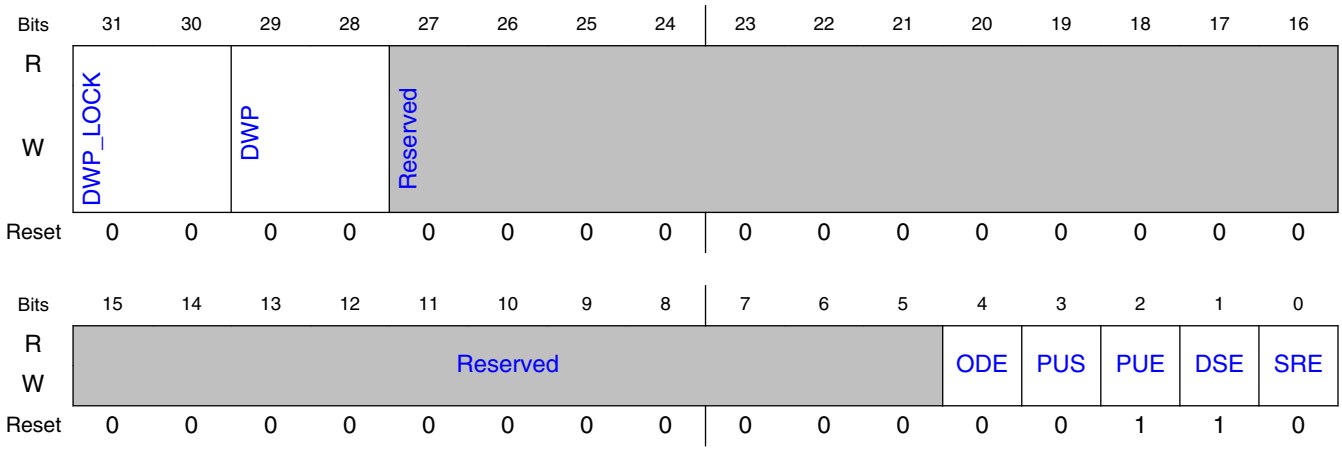
12.4.6.233.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_23	3ACh

12.4.6.233.2 Function

SW_PAD_CTL Register

12.4.6.233.3 Diagram



12.4.6.233.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_23 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_23 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_23

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_23 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_23 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.234 SW_PAD_CTL_PAD_GPIO_AD_24 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_24)

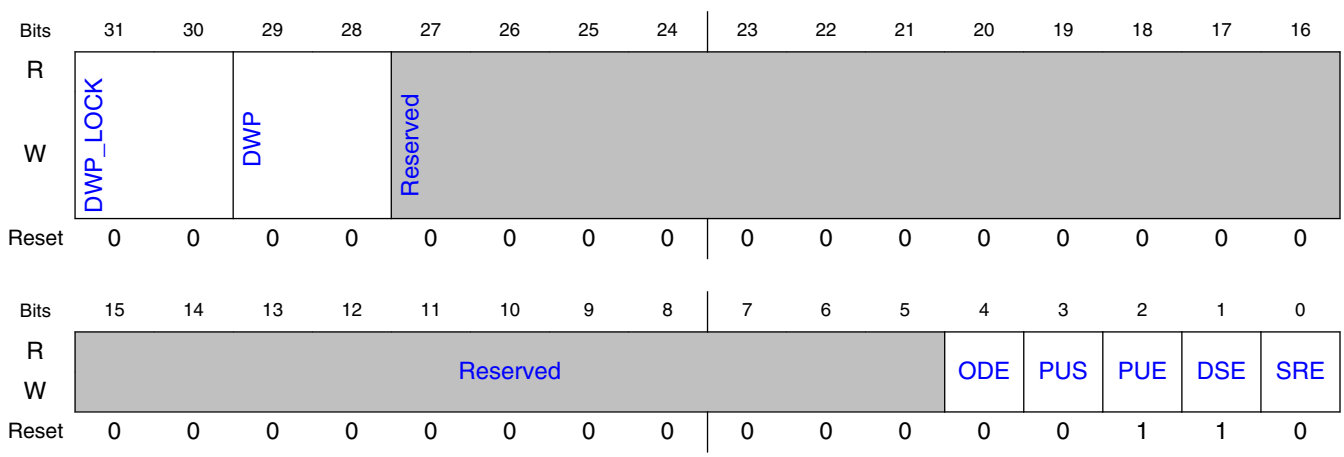
12.4.6.234.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_24	3B0h

12.4.6.234.2 Function

SW_PAD_CTL Register

12.4.6.234.3 Diagram



12.4.6.234.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_24 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_24 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_24 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_24 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_24 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.235 SW_PAD_CTL_PAD_GPIO_AD_25 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_25)

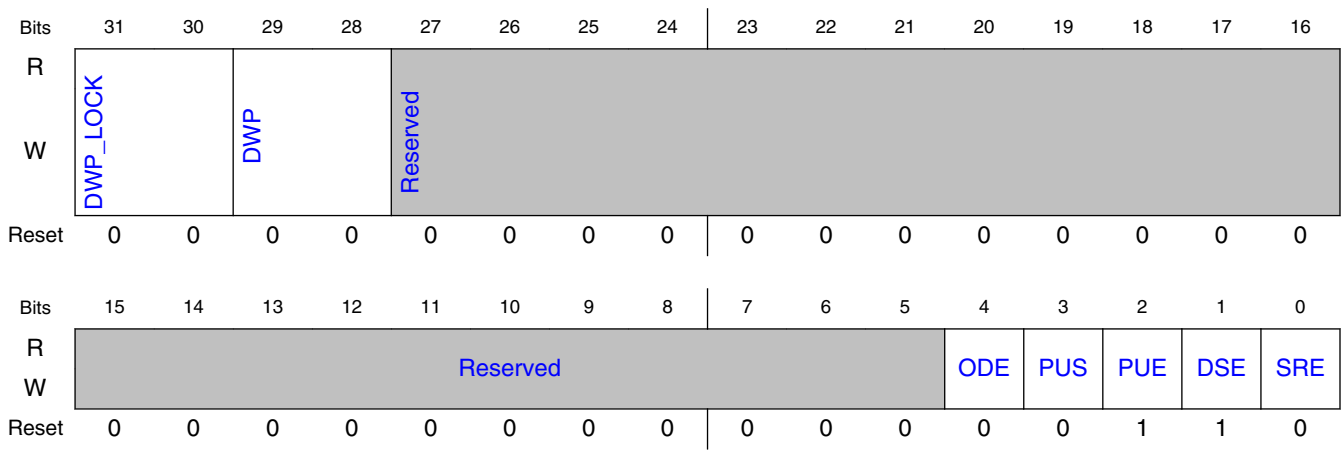
12.4.6.235.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_25	3B4h

12.4.6.235.2 Function

SW_PAD_CTL Register

12.4.6.235.3 Diagram



12.4.6.235.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_25 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_25 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_25 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_25 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_25 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.236 SW_PAD_CTL_PAD_GPIO_AD_26 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_26)

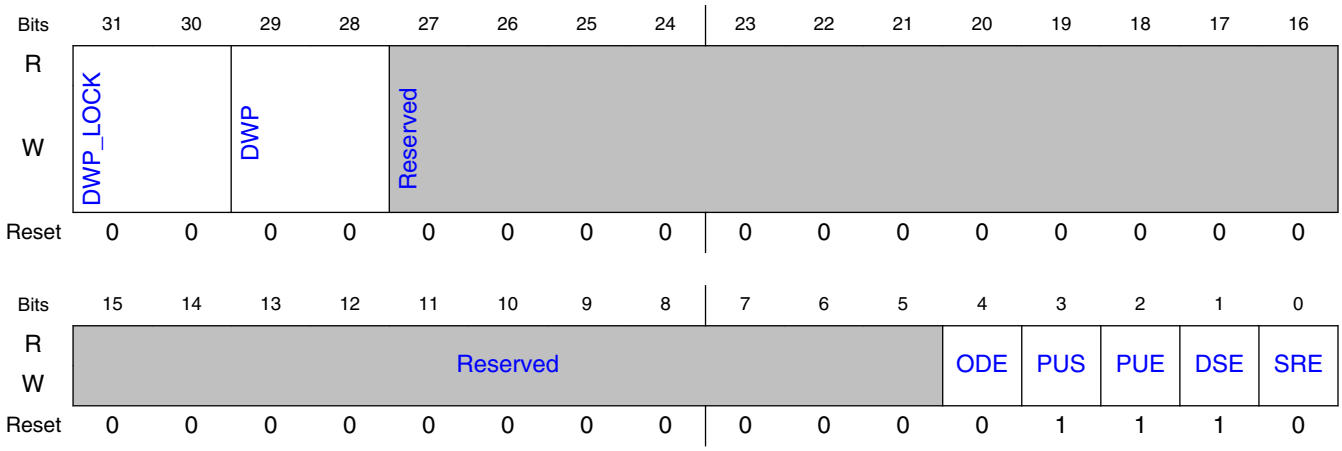
12.4.6.236.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_26	3B8h

12.4.6.236.2 Function

SW_PAD_CTL Register

12.4.6.236.3 Diagram



12.4.6.236.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_26 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_26 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_26

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_26 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_26 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.237 SW_PAD_CTL_PAD_GPIO_AD_27 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_27)

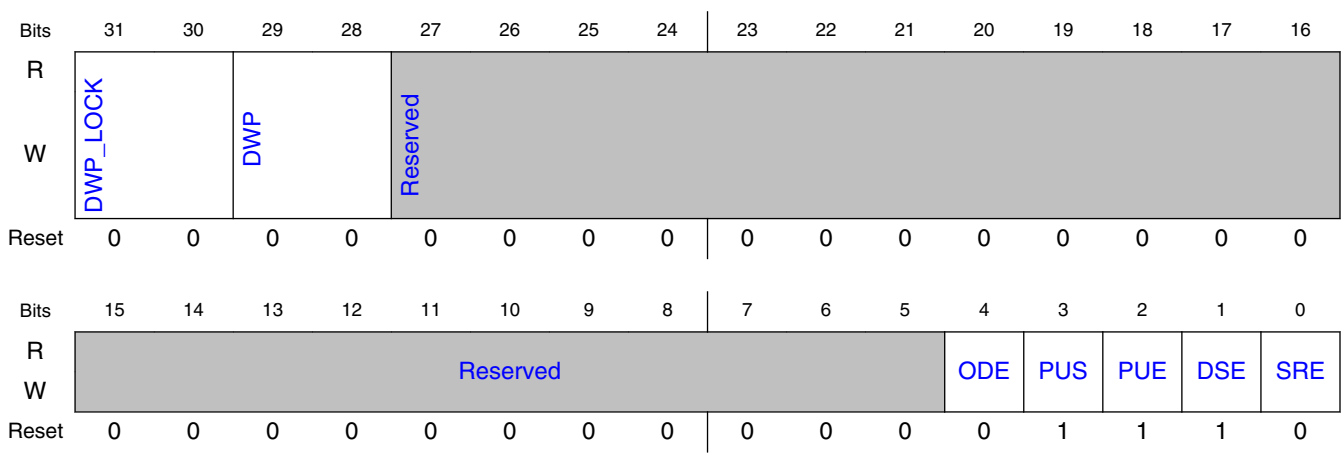
12.4.6.237.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_27	3BCh

12.4.6.237.2 Function

SW_PAD_CTL Register

12.4.6.237.3 Diagram



12.4.6.237.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_27 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_27 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_27 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_27 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_27 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.238 SW_PAD_CTL_PAD_GPIO_AD_28 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_28)

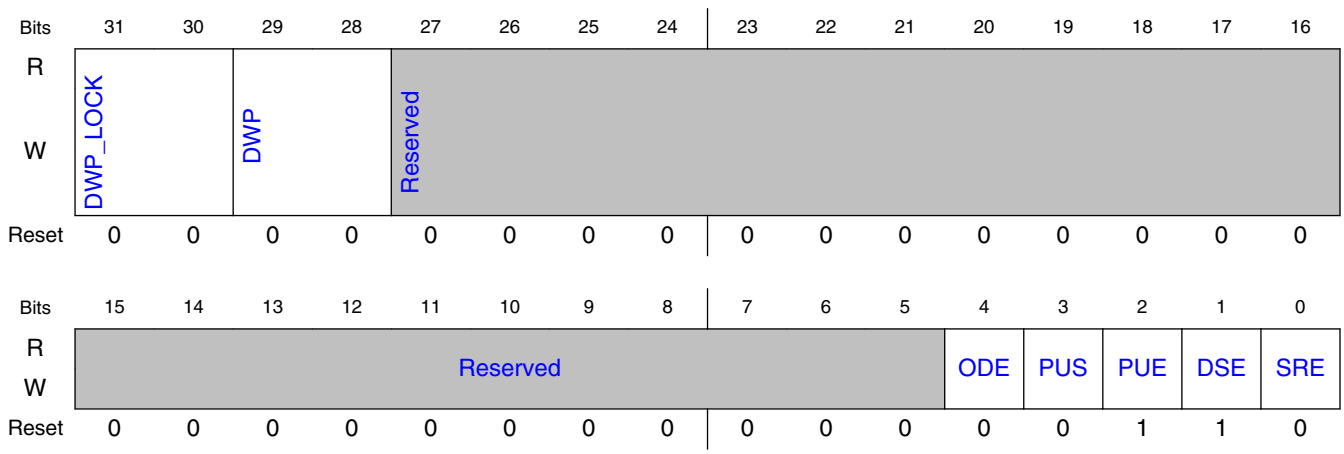
12.4.6.238.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_28	3C0h

12.4.6.238.2 Function

SW_PAD_CTL Register

12.4.6.238.3 Diagram



12.4.6.238.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_28 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_28 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_28 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_28 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_28 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.239 SW_PAD_CTL_PAD_GPIO_AD_29 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_29)

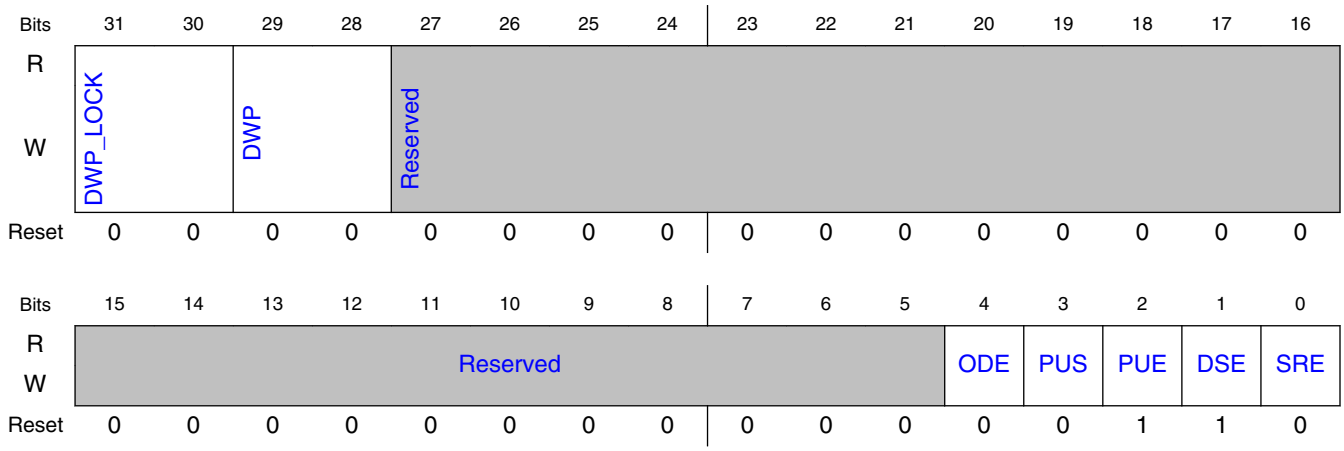
12.4.6.239.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_29	3C4h

12.4.6.239.2 Function

SW_PAD_CTL Register

12.4.6.239.3 Diagram



12.4.6.239.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_29 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_29 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_29

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_29 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_29 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.240 SW_PAD_CTL_PAD_GPIO_AD_30 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_30)

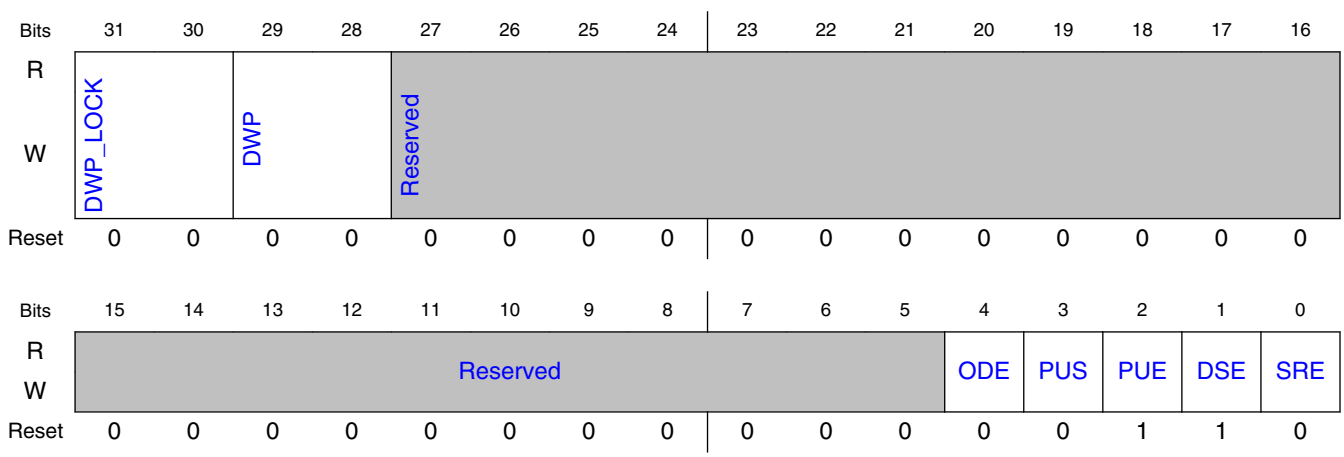
12.4.6.240.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_30	3C8h

12.4.6.240.2 Function

SW_PAD_CTL Register

12.4.6.240.3 Diagram



12.4.6.240.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_30 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_30 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_30 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_30 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_30 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.241 SW_PAD_CTL_PAD_GPIO_AD_31 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_31)

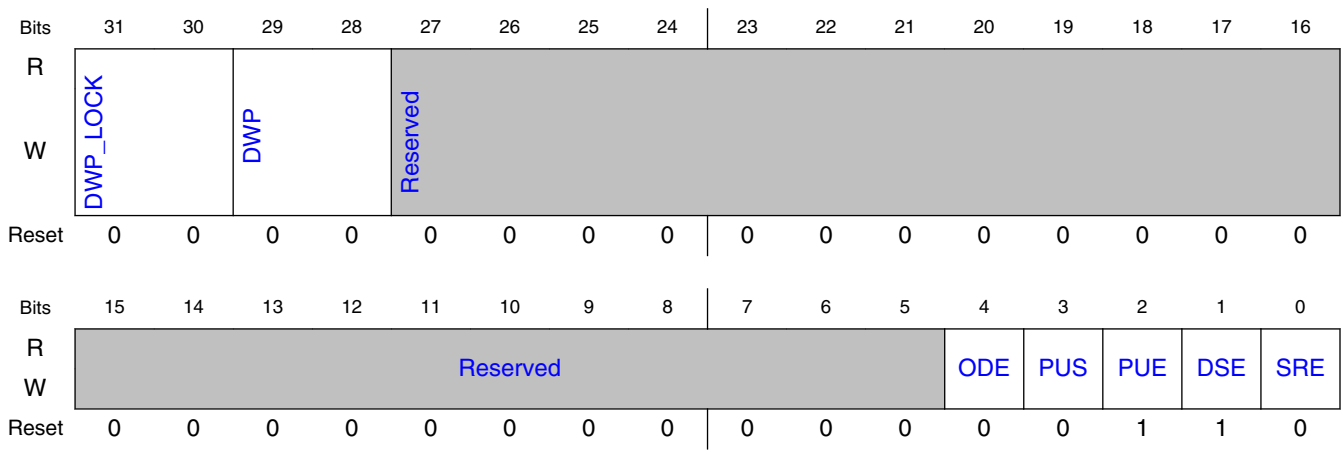
12.4.6.241.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_31	3CCh

12.4.6.241.2 Function

SW_PAD_CTL Register

12.4.6.241.3 Diagram



12.4.6.241.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_31 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_31 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_31 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_31 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_31 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.242 SW_PAD_CTL_PAD_GPIO_AD_32 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_32)

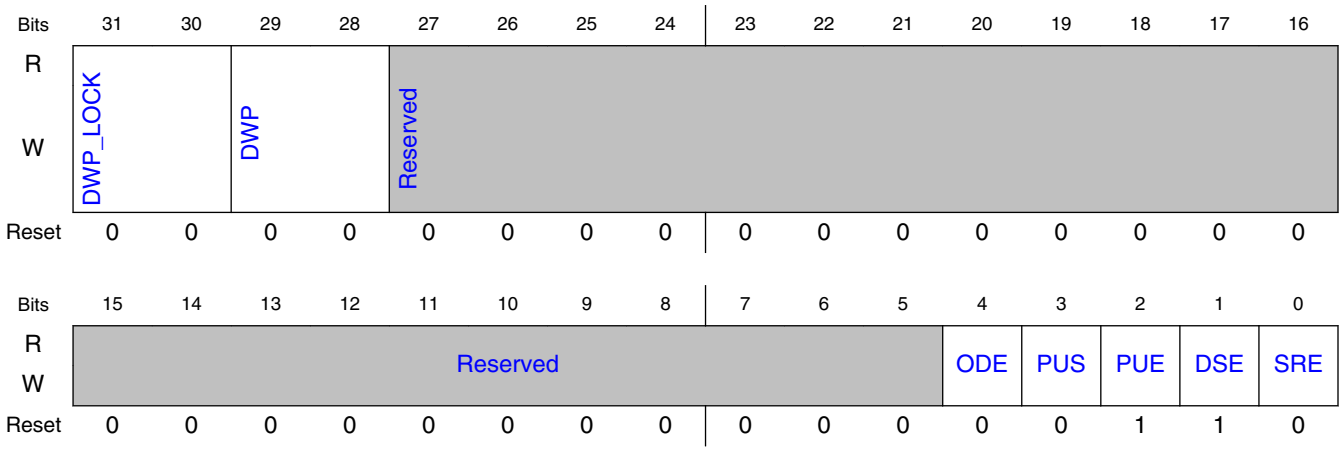
12.4.6.242.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_32	3D0h

12.4.6.242.2 Function

SW_PAD_CTL Register

12.4.6.242.3 Diagram



12.4.6.242.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_32 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_32 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_32

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_32 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_32 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.243 SW_PAD_CTL_PAD_GPIO_AD_33 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_33)

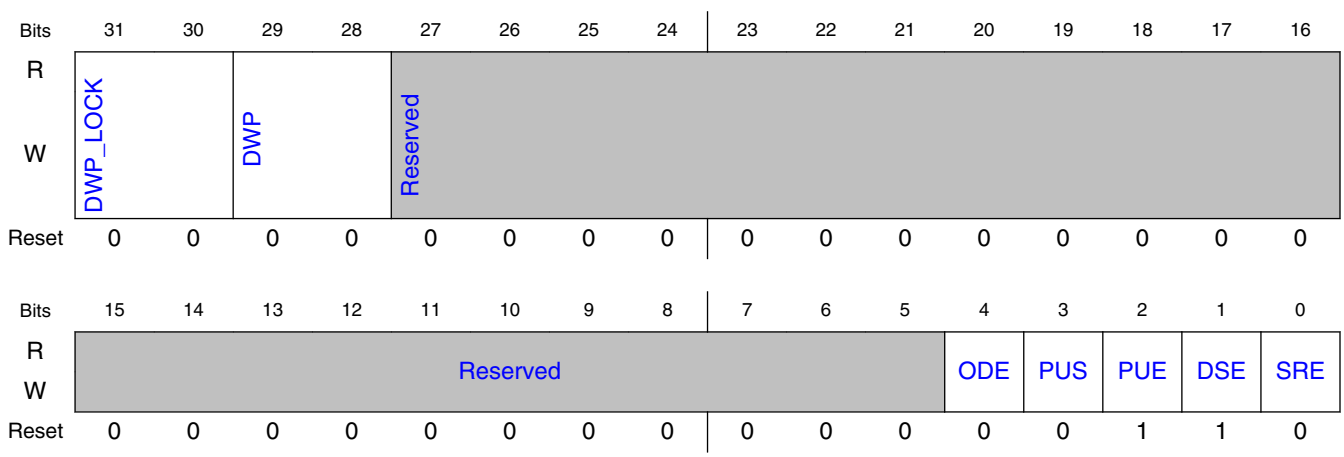
12.4.6.243.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_33	3D4h

12.4.6.243.2 Function

SW_PAD_CTL Register

12.4.6.243.3 Diagram



12.4.6.243.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_33 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_33 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_33 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_33 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_33 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.244 SW_PAD_CTL_PAD_GPIO_AD_34 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_34)

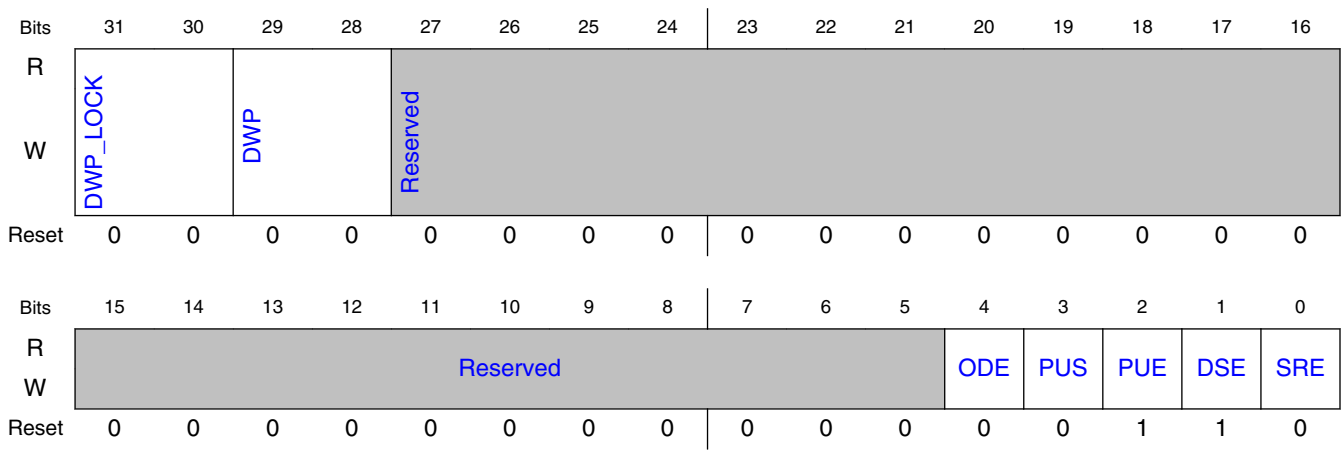
12.4.6.244.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_34	3D8h

12.4.6.244.2 Function

SW_PAD_CTL Register

12.4.6.244.3 Diagram



12.4.6.244.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_34 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_34 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_34 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_34 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_34 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.245 SW_PAD_CTL_PAD_GPIO_AD_35 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_AD_35)

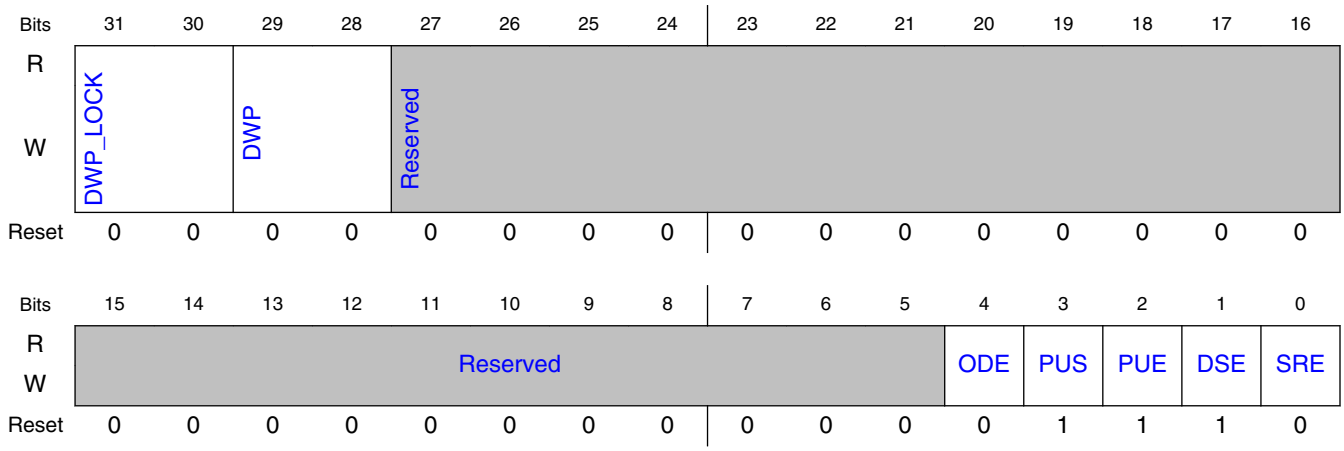
12.4.6.245.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_AD_35	3DCh

12.4.6.245.2 Function

SW_PAD_CTL Register

12.4.6.245.3 Diagram



12.4.6.245.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_AD_35 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_AD_35 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_AD_35

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_AD_35 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_AD_35 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.246 SW_PAD_CTL_PAD_GPIO_SD_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_00)

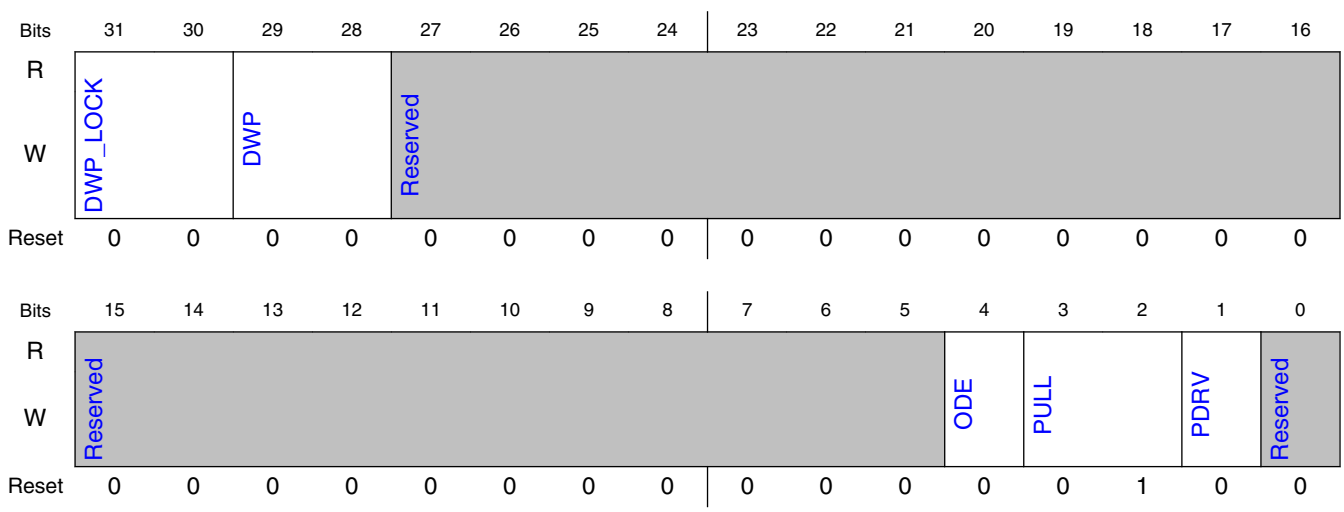
12.4.6.246.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_00	3E0h

12.4.6.246.2 Function

SW_PAD_CTL Register

12.4.6.246.3 Diagram



12.4.6.246.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_00 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_00 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_00 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.247 SW_PAD_CTL_PAD_GPIO_SD_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_01)

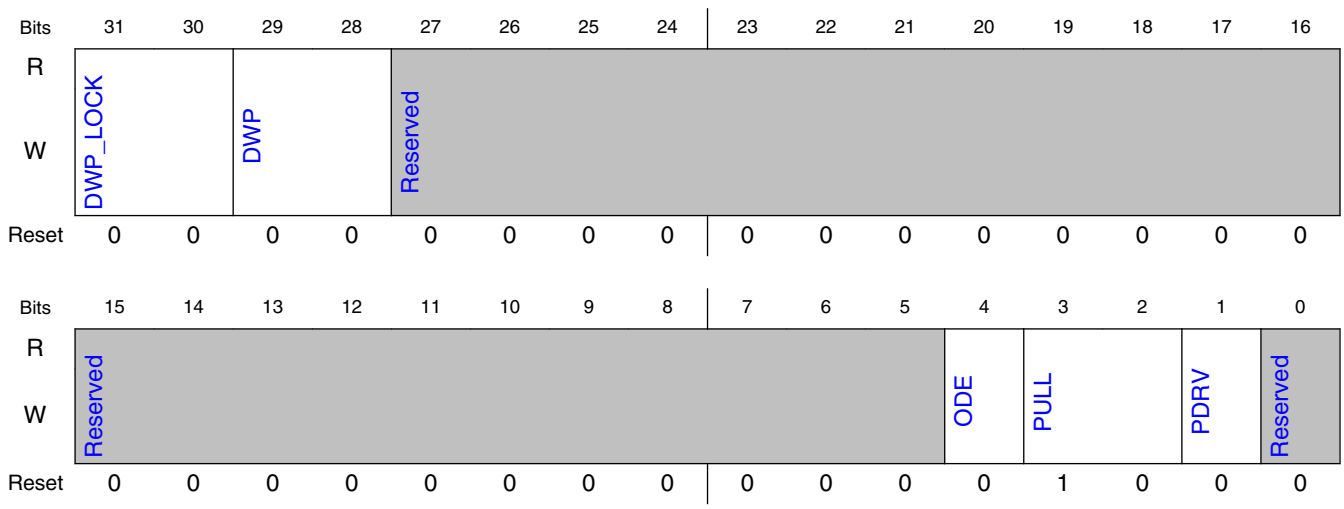
12.4.6.247.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_01	3E4h

12.4.6.247.2 Function

SW_PAD_CTL Register

12.4.6.247.3 Diagram



12.4.6.247.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_01 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_01 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_01 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.248 SW_PAD_CTL_PAD_GPIO_SD_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_02)

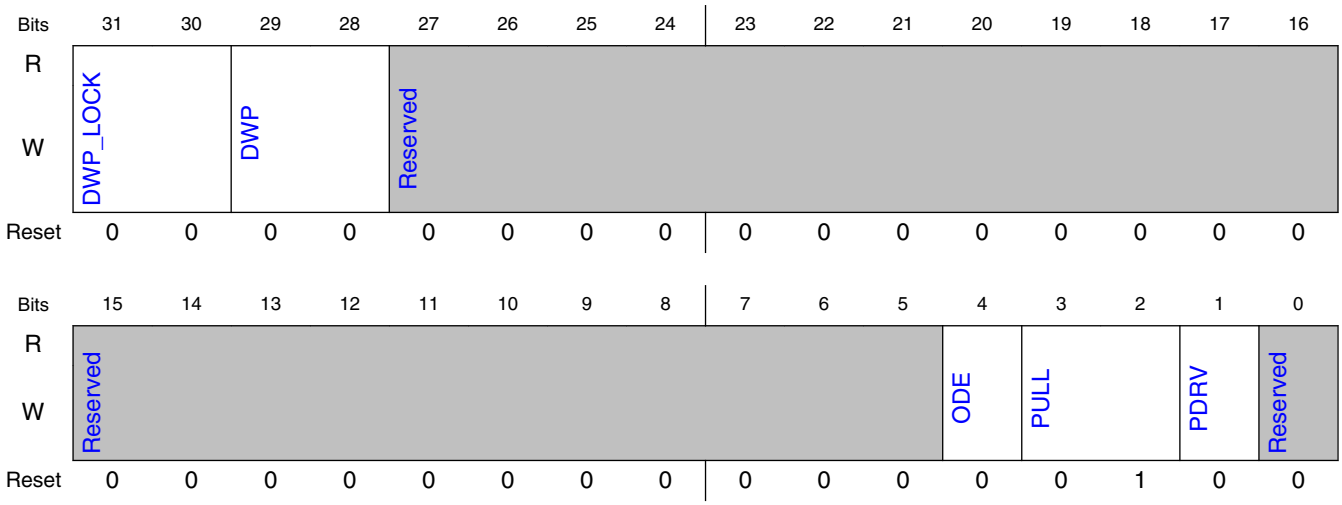
12.4.6.248.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_02	3E8h

12.4.6.248.2 Function

SW_PAD_CTL Register

12.4.6.248.3 Diagram



12.4.6.248.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_02 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_02 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_02 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.249 SW_PAD_CTL_PAD_GPIO_SD_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_03)

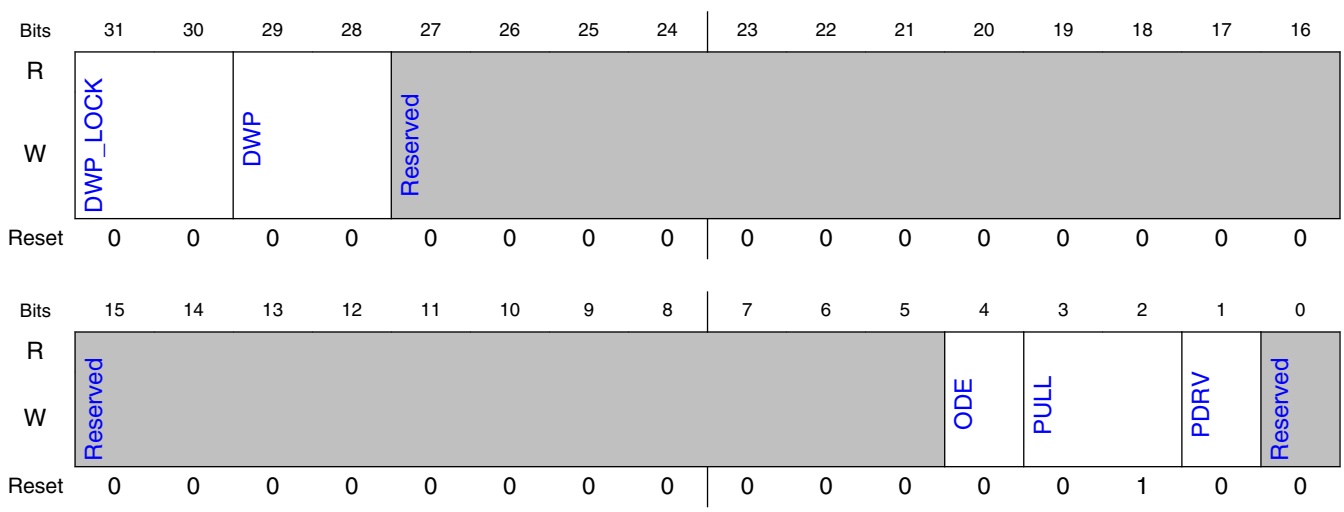
12.4.6.249.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_03	3ECh

12.4.6.249.2 Function

SW_PAD_CTL Register

12.4.6.249.3 Diagram



12.4.6.249.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_03 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_03 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_03 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.250 SW_PAD_CTL_PAD_GPIO_SD_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_04)

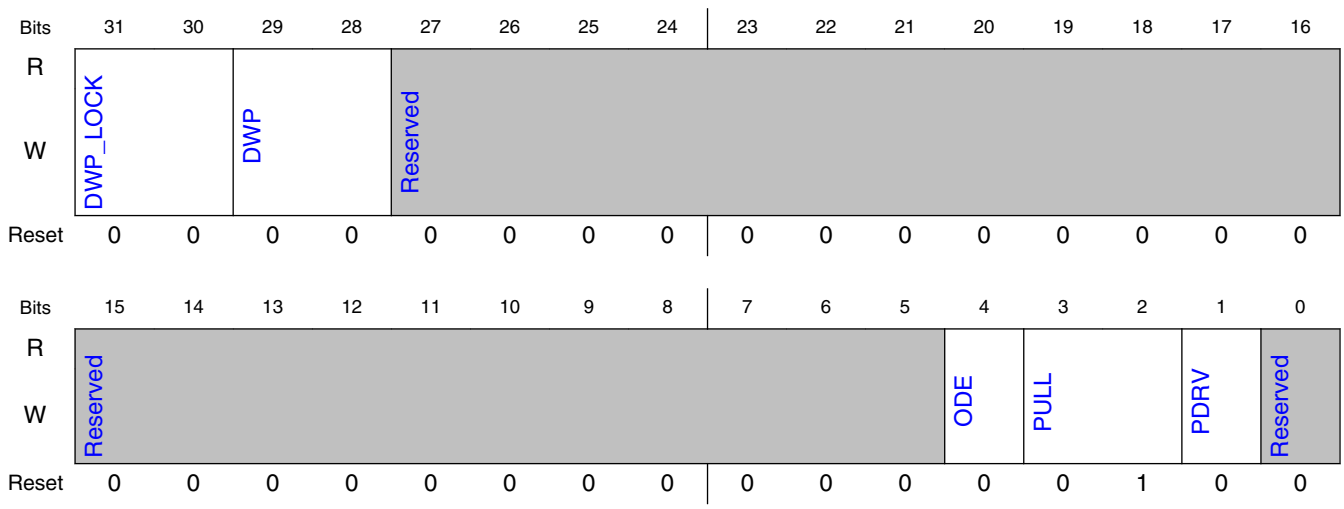
12.4.6.250.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_04	3F0h

12.4.6.250.2 Function

SW_PAD_CTL Register

12.4.6.250.3 Diagram



12.4.6.250.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_04 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_04 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_04 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.251 SW_PAD_CTL_PAD_GPIO_SD_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B1_05)

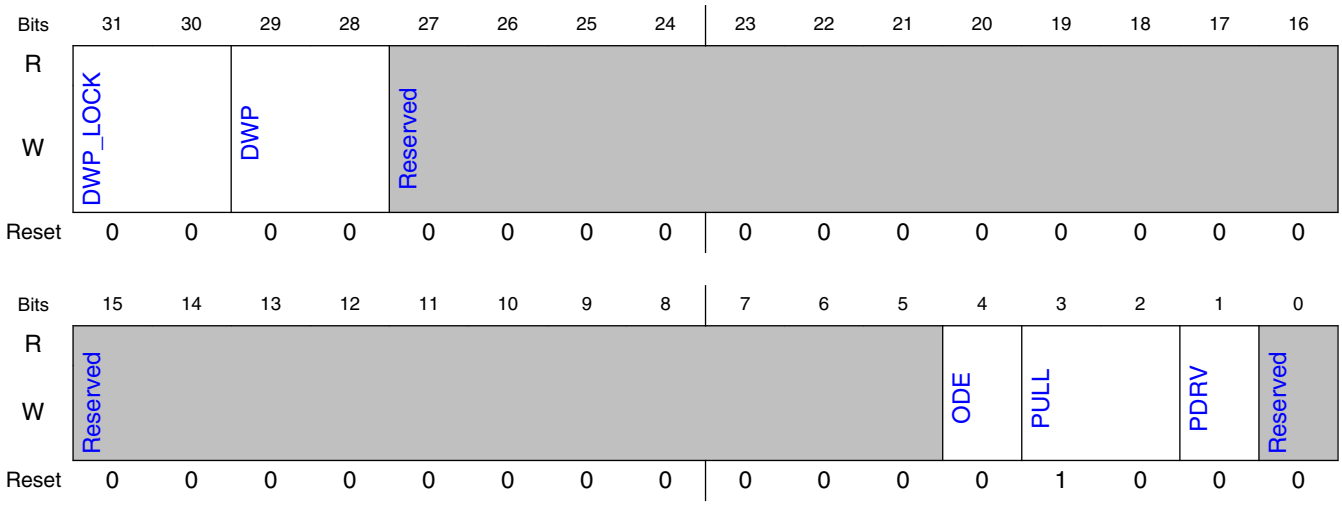
12.4.6.251.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B1_05	3F4h

12.4.6.251.2 Function

SW_PAD_CTL Register

12.4.6.251.3 Diagram



12.4.6.251.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B1_05 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B1_05 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B1_05 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.252 SW_PAD_CTL_PAD_GPIO_SD_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_00)

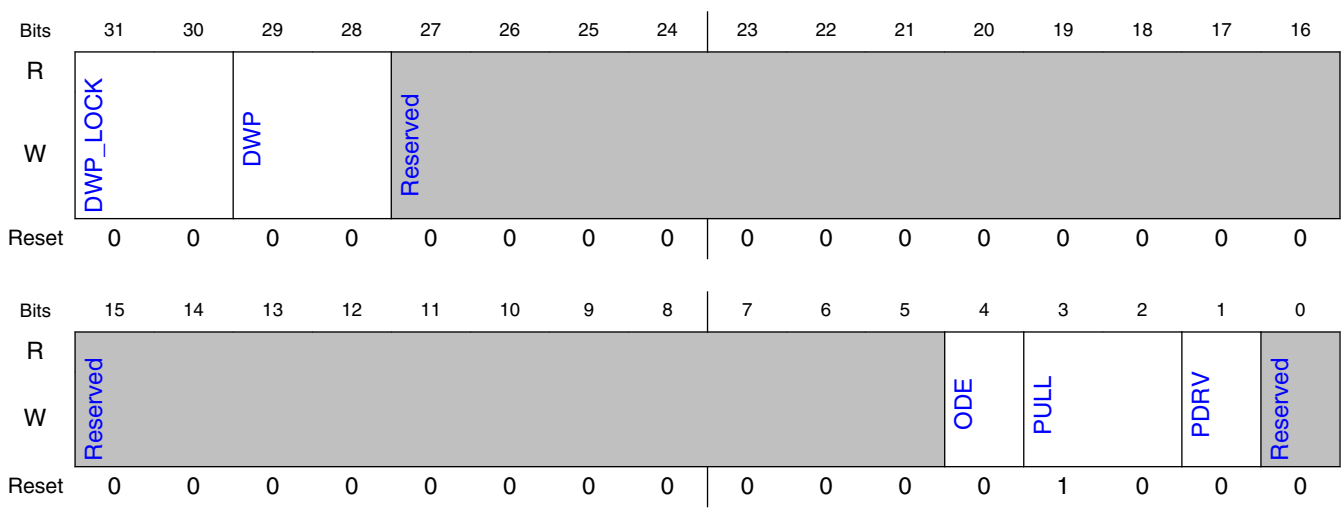
12.4.6.252.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_00	3F8h

12.4.6.252.2 Function

SW_PAD_CTL Register

12.4.6.252.3 Diagram



12.4.6.252.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_00 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_00 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_00 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.253 SW_PAD_CTL_PAD_GPIO_SD_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_01)

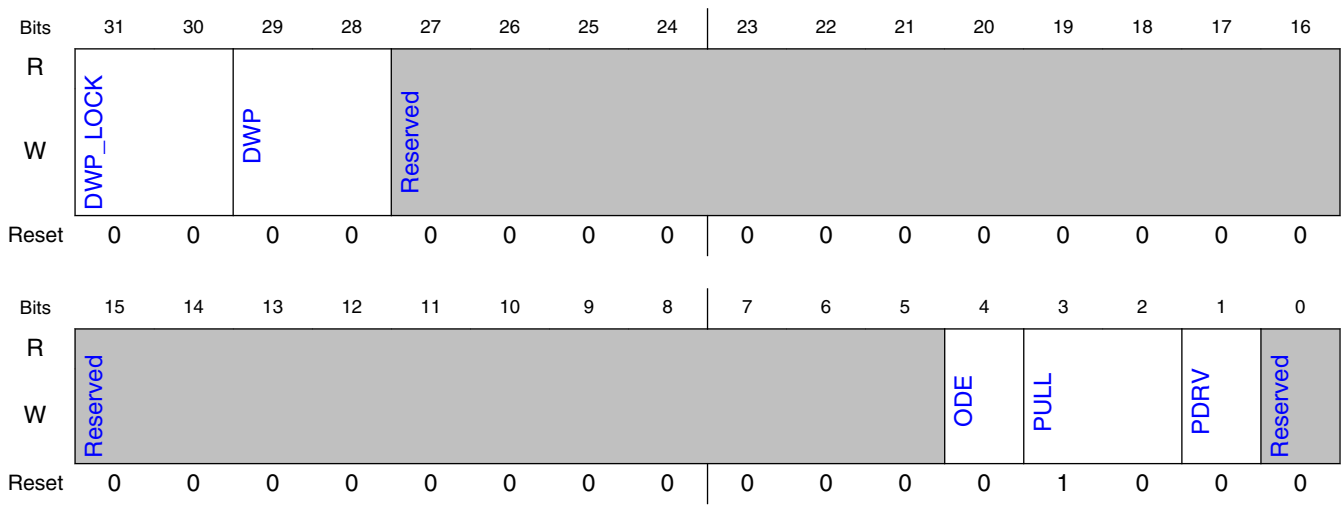
12.4.6.253.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_01	3FCh

12.4.6.253.2 Function

SW_PAD_CTL Register

12.4.6.253.3 Diagram



12.4.6.253.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_01 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_01 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_01 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.254 SW_PAD_CTL_PAD_GPIO_SD_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_02)

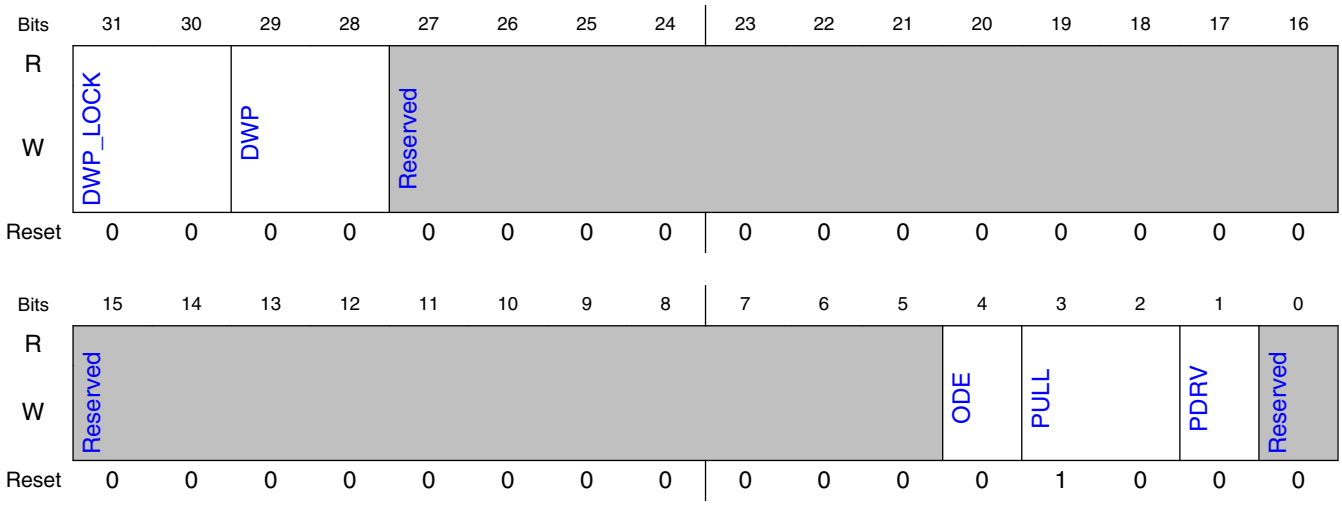
12.4.6.254.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_02	400h

12.4.6.254.2 Function

SW_PAD_CTL Register

12.4.6.254.3 Diagram



12.4.6.254.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_02 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_02 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_02 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.255 SW_PAD_CTL_PAD_GPIO_SD_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_03)

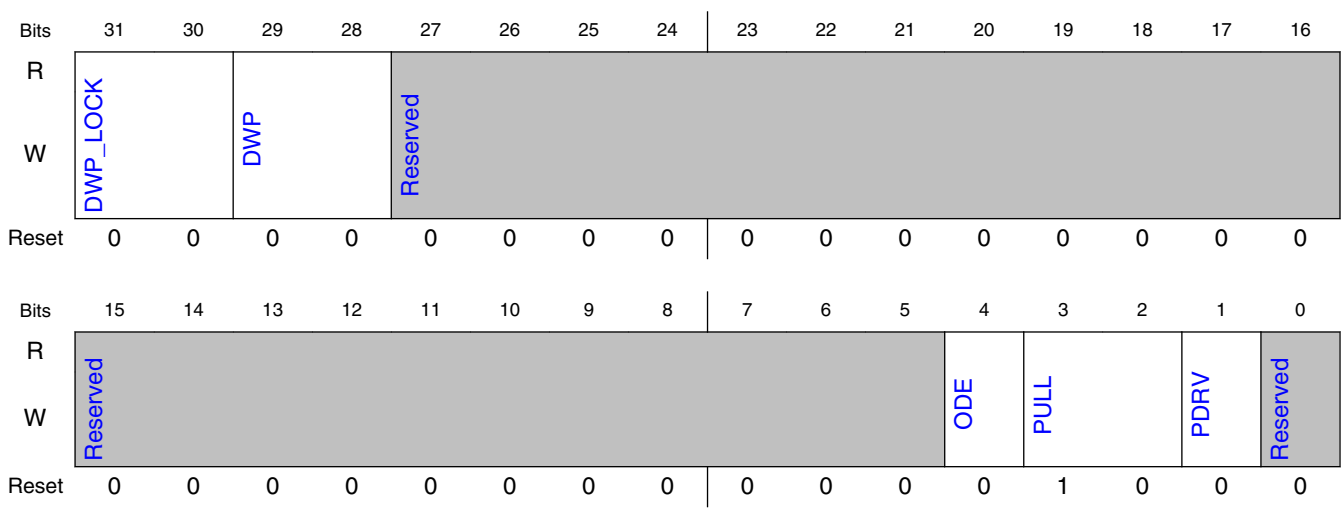
12.4.6.255.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_03	404h

12.4.6.255.2 Function

SW_PAD_CTL Register

12.4.6.255.3 Diagram



12.4.6.255.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_03 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_03 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_03 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.256 SW_PAD_CTL_PAD_GPIO_SD_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_04)

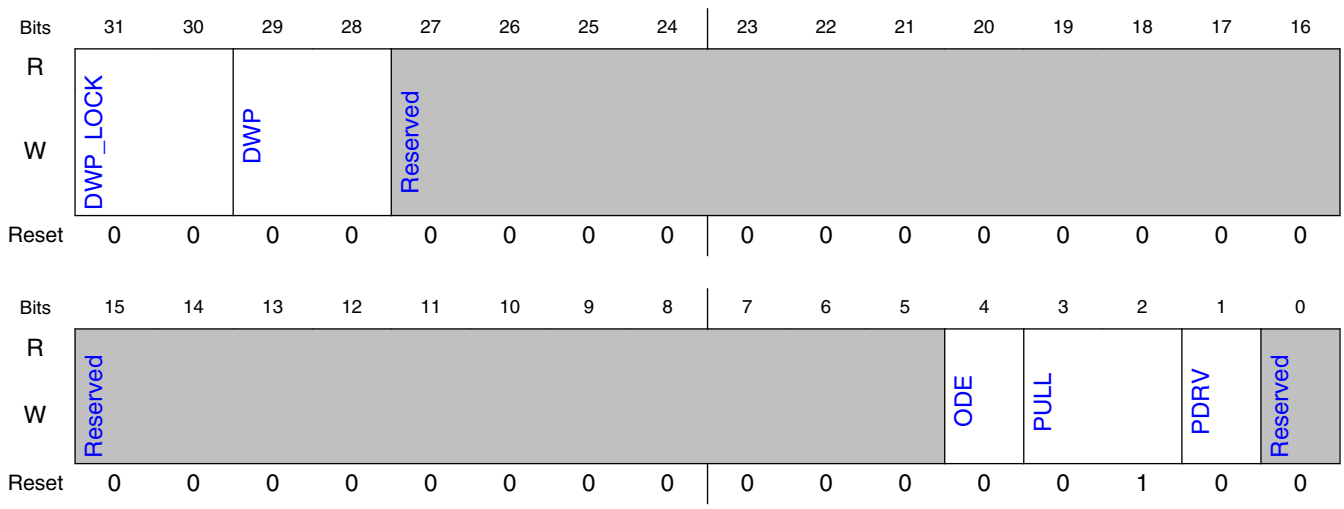
12.4.6.256.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_04	408h

12.4.6.256.2 Function

SW_PAD_CTL Register

12.4.6.256.3 Diagram



12.4.6.256.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_04 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_04 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_04 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.257 SW_PAD_CTL_PAD_GPIO_SD_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_05)

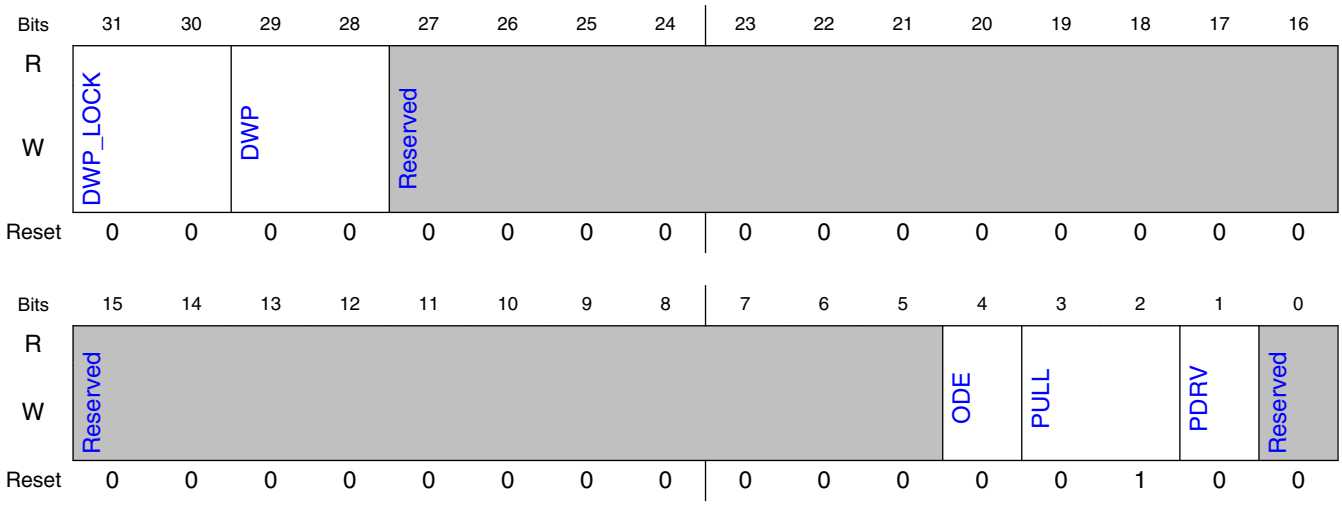
12.4.6.257.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_05	40Ch

12.4.6.257.2 Function

SW_PAD_CTL Register

12.4.6.257.3 Diagram



12.4.6.257.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_05 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_05 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_05 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.258 SW_PAD_CTL_PAD_GPIO_SD_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_06)

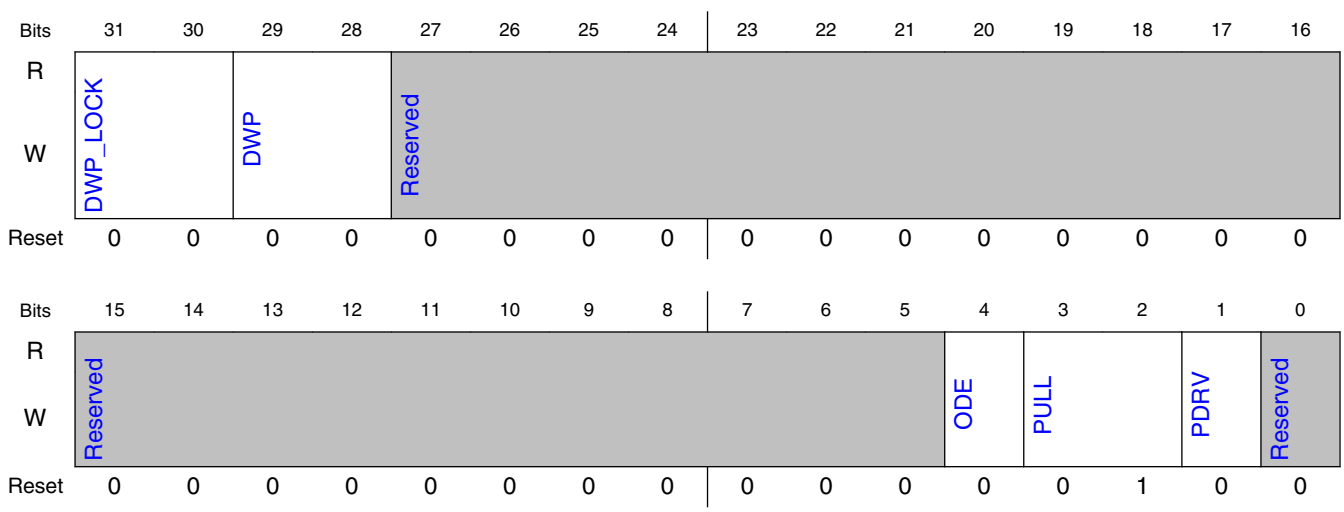
12.4.6.258.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_06	410h

12.4.6.258.2 Function

SW_PAD_CTL Register

12.4.6.258.3 Diagram



12.4.6.258.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_06 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_06 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_06 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.259 SW_PAD_CTL_PAD_GPIO_SD_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_07)

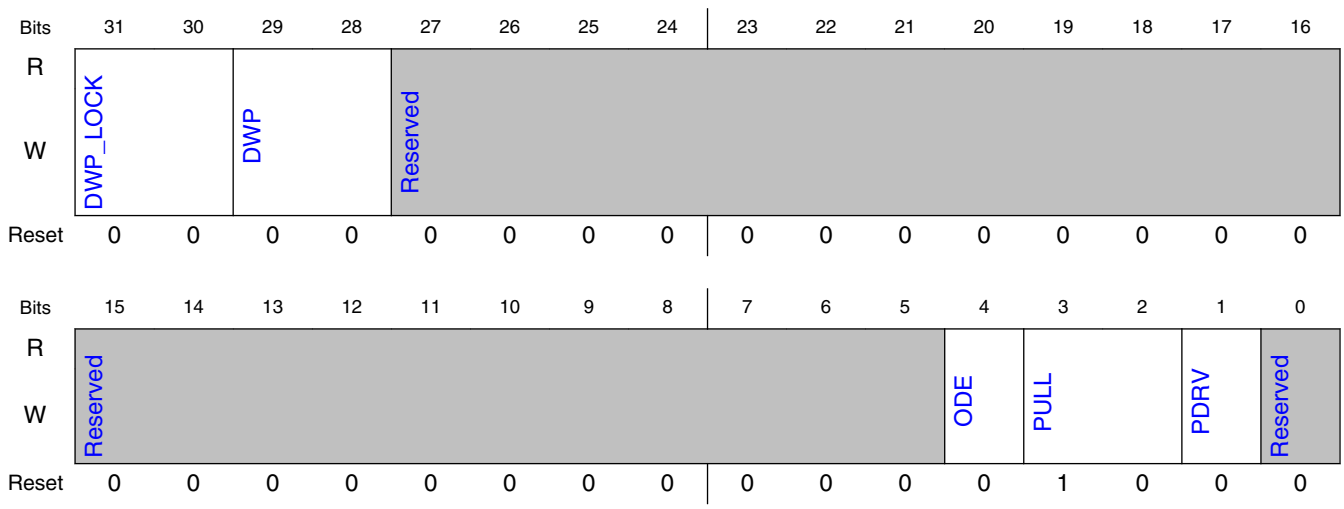
12.4.6.259.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_07	414h

12.4.6.259.2 Function

SW_PAD_CTL Register

12.4.6.259.3 Diagram



12.4.6.259.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_07 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_07 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_07 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.260 SW_PAD_CTL_PAD_GPIO_SD_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_08)

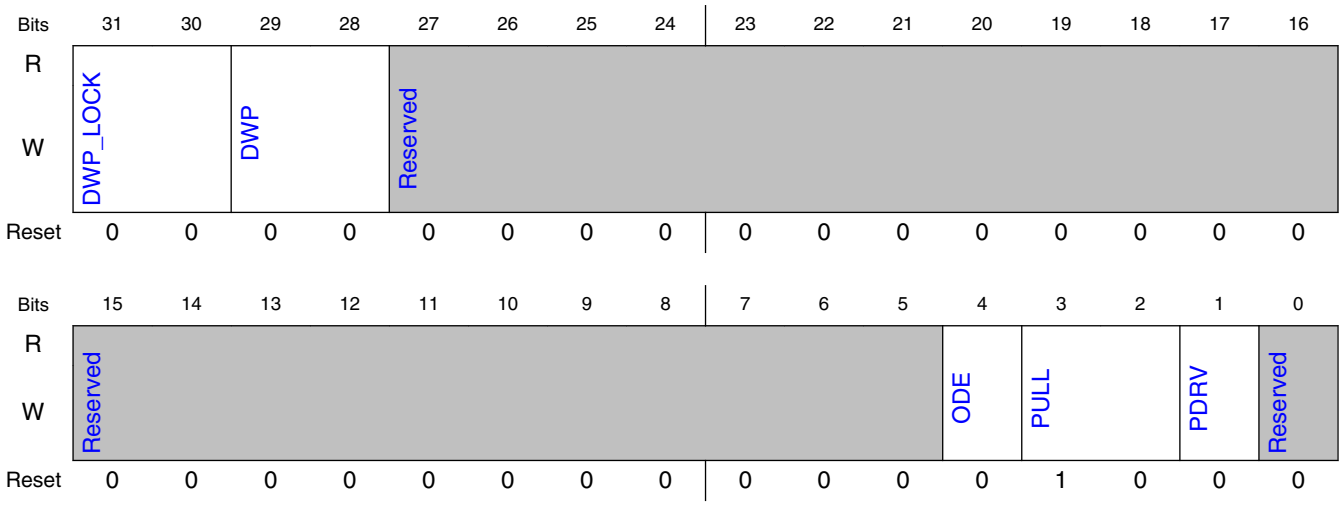
12.4.6.260.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_08	418h

12.4.6.260.2 Function

SW_PAD_CTL Register

12.4.6.260.3 Diagram



12.4.6.260.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_08 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_08 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_08 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.261 SW_PAD_CTL_PAD_GPIO_SD_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_09)

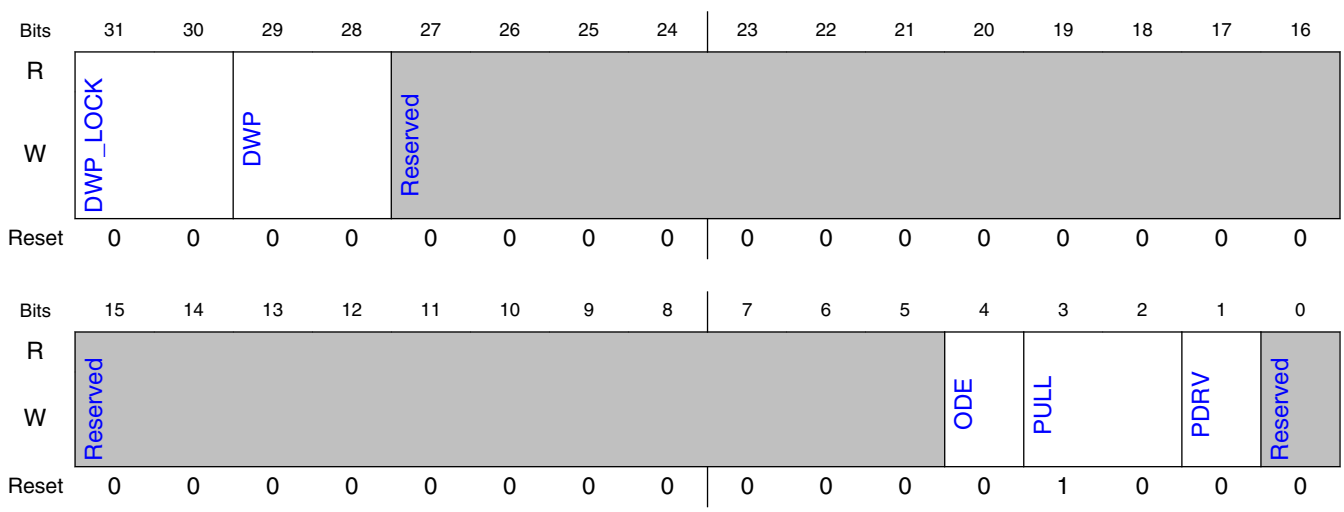
12.4.6.261.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_09	41Ch

12.4.6.261.2 Function

SW_PAD_CTL Register

12.4.6.261.3 Diagram



12.4.6.261.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_09 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_09 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_09 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.262 SW_PAD_CTL_PAD_GPIO_SD_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_10)

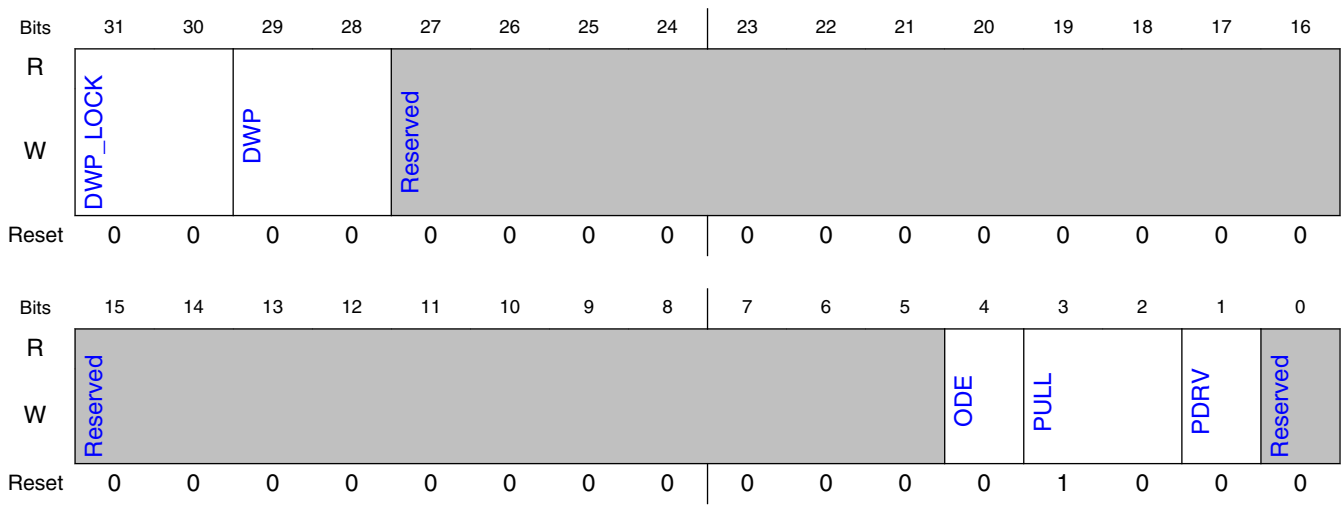
12.4.6.262.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_10	420h

12.4.6.262.2 Function

SW_PAD_CTL Register

12.4.6.262.3 Diagram



12.4.6.262.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_10 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_10 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_10 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.263 SW_PAD_CTL_PAD_GPIO_SD_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_SD_B2_11)

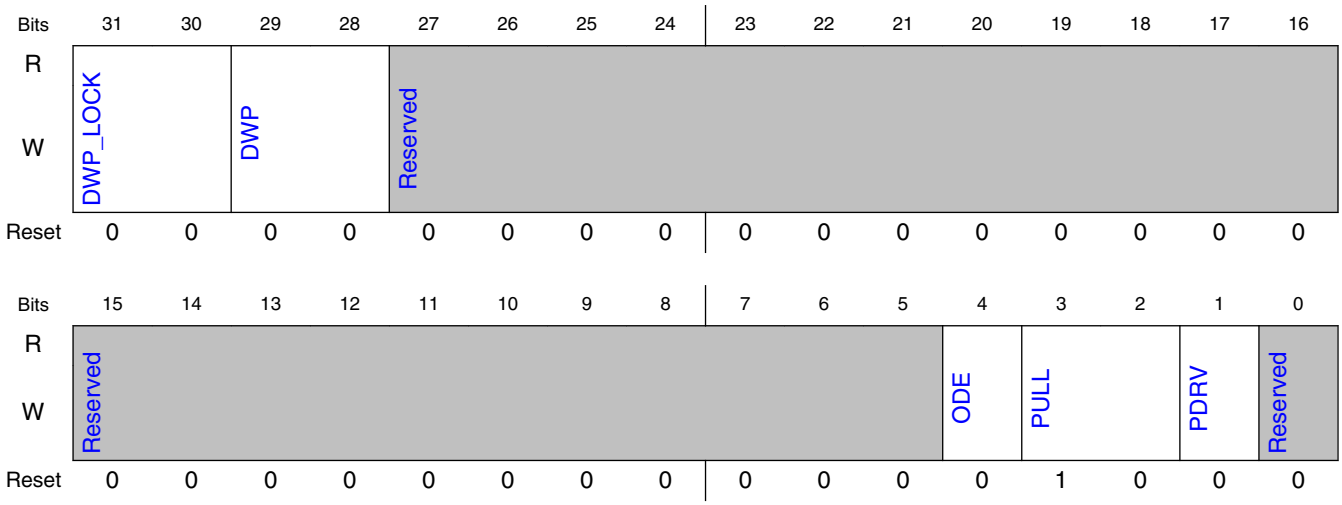
12.4.6.263.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_SD_B2_11	424h

12.4.6.263.2 Function

SW_PAD_CTL Register

12.4.6.263.3 Diagram



12.4.6.263.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_SD_B2_11 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_SD_B2_11 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_SD_B2_11 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.264 SW_PAD_CTL_PAD_GPIO_DISP_B1_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_00)

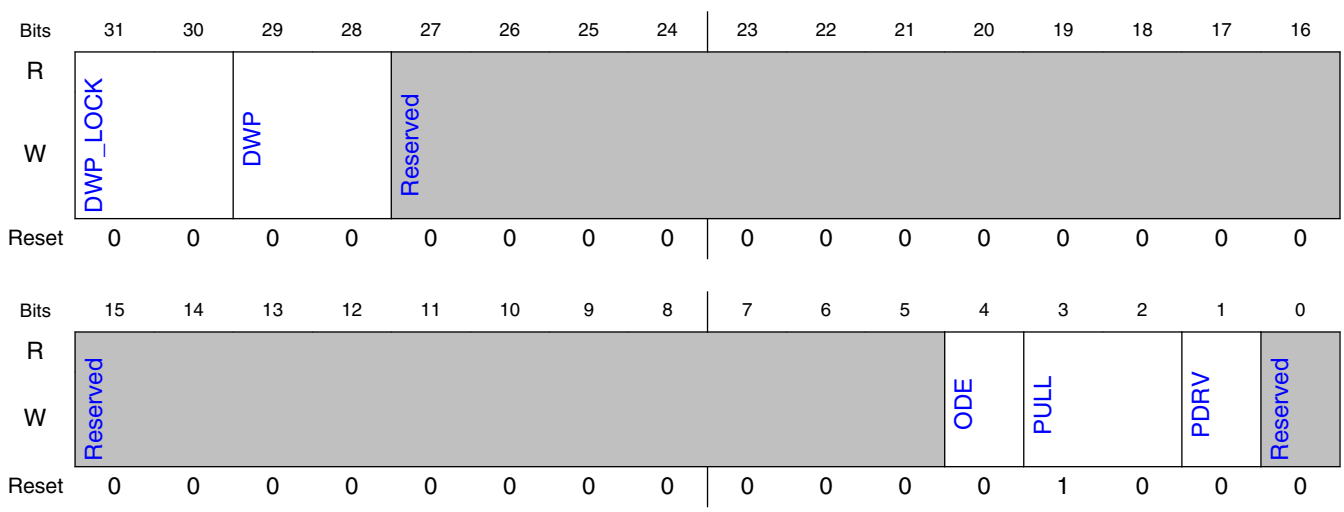
12.4.6.264.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_00	428h

12.4.6.264.2 Function

SW_PAD_CTL Register

12.4.6.264.3 Diagram



12.4.6.264.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_00 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_00 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_00 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.265 SW_PAD_CTL_PAD_GPIO_DISP_B1_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_01)

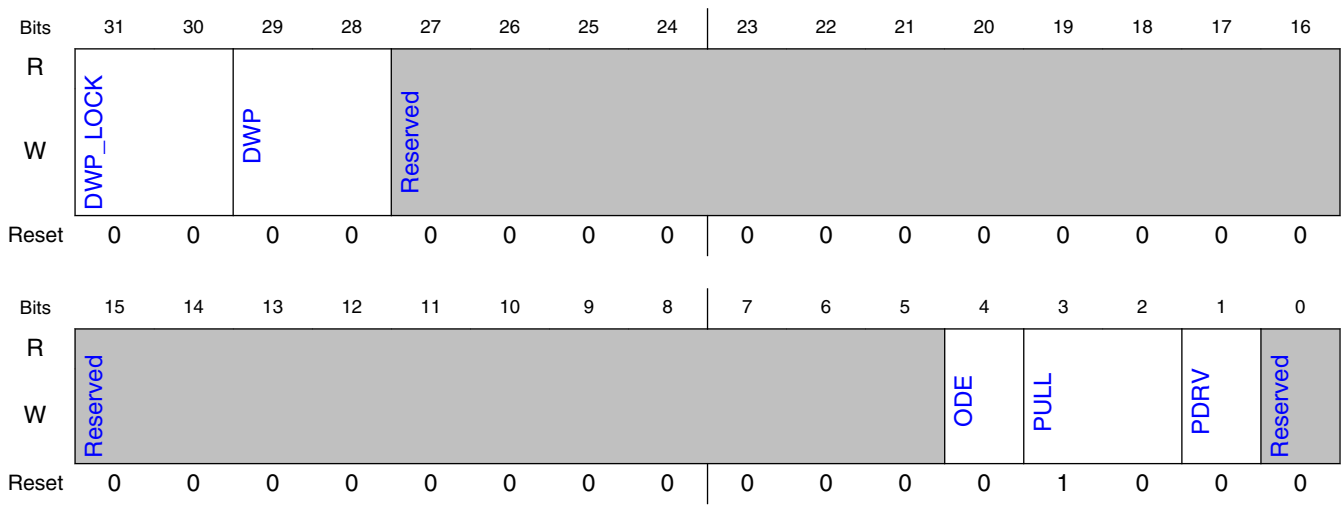
12.4.6.265.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_01	42Ch

12.4.6.265.2 Function

SW_PAD_CTL Register

12.4.6.265.3 Diagram



12.4.6.265.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_01 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_01 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_01 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.266 SW_PAD_CTL_PAD_GPIO_DISP_B1_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_02)

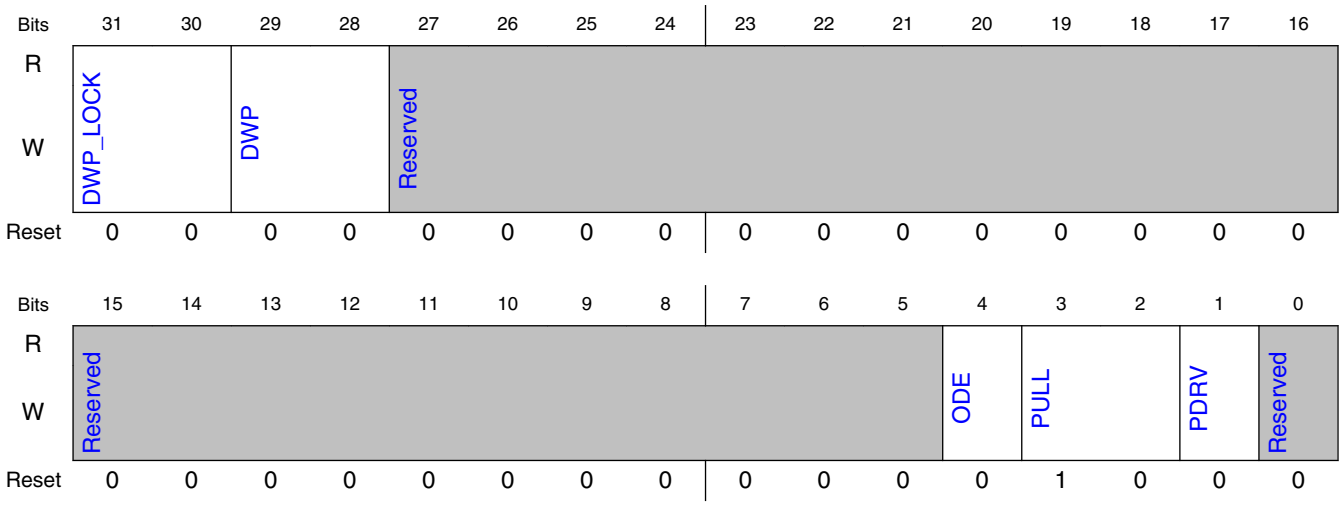
12.4.6.266.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_02	430h

12.4.6.266.2 Function

SW_PAD_CTL Register

12.4.6.266.3 Diagram



12.4.6.266.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_02 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_02 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_02 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.267 SW_PAD_CTL_PAD_GPIO_DISP_B1_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_03)

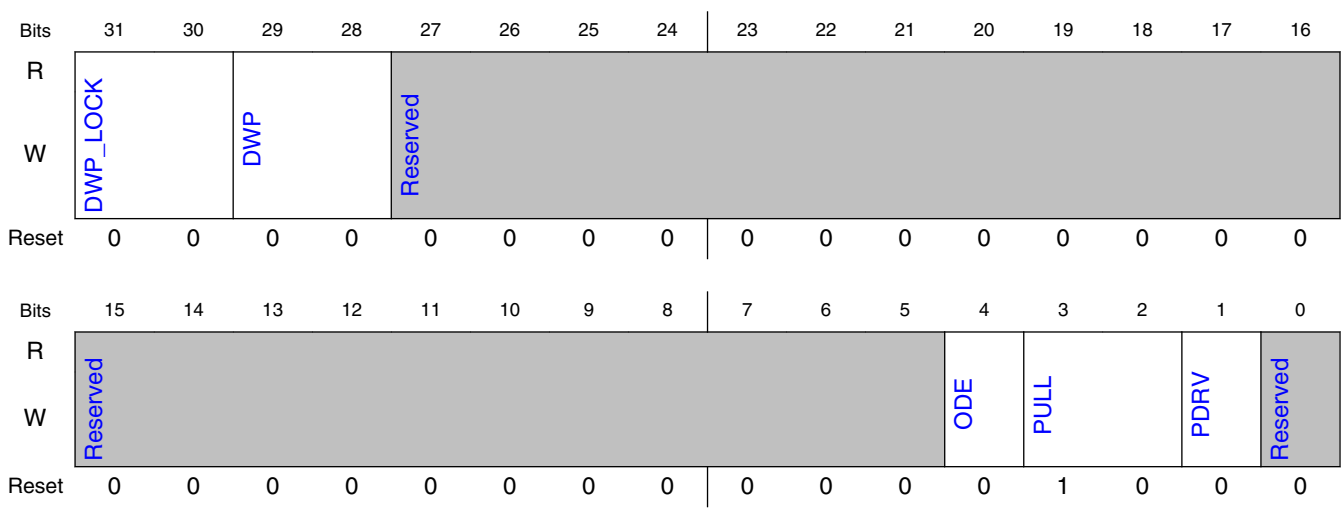
12.4.6.267.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_03	434h

12.4.6.267.2 Function

SW_PAD_CTL Register

12.4.6.267.3 Diagram



12.4.6.267.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_03 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_03 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_03 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.268 SW_PAD_CTL_PAD_GPIO_DISP_B1_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_04)

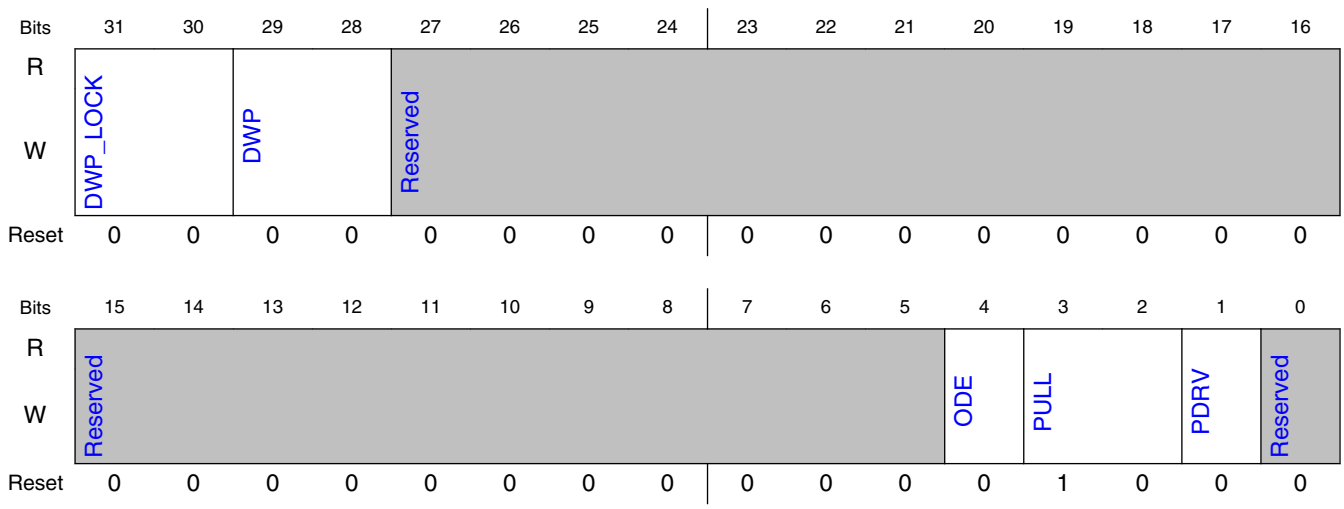
12.4.6.268.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_04	438h

12.4.6.268.2 Function

SW_PAD_CTL Register

12.4.6.268.3 Diagram



12.4.6.268.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_04 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_04 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_04 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.269 SW_PAD_CTL_PAD_GPIO_DISP_B1_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_05)

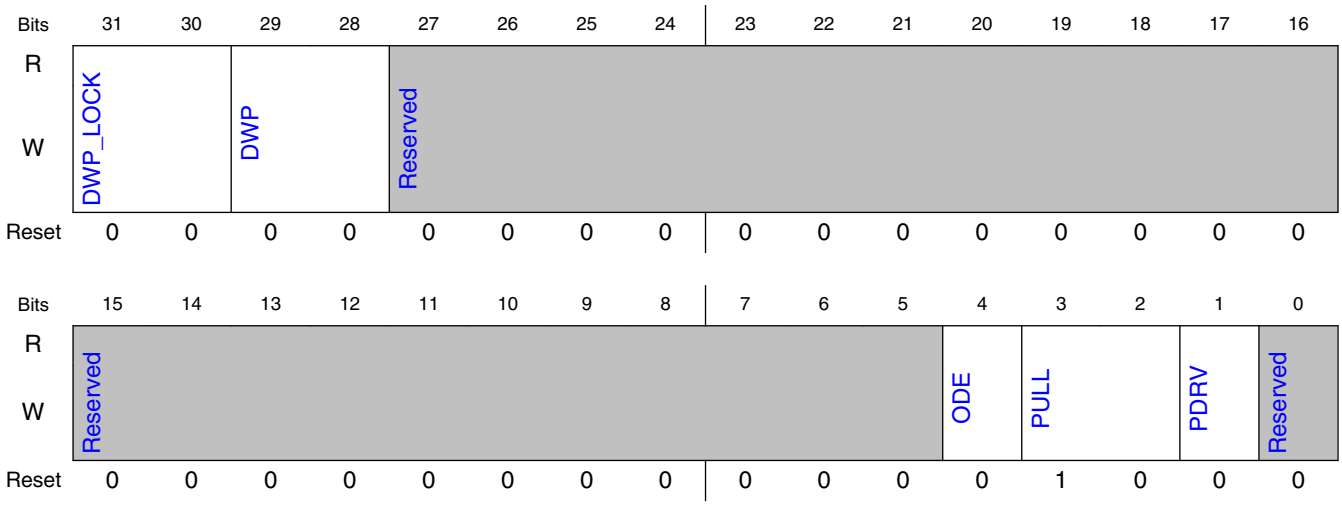
12.4.6.269.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_05	43Ch

12.4.6.269.2 Function

SW_PAD_CTL Register

12.4.6.269.3 Diagram



12.4.6.269.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_05 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_05 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_05 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.270 SW_PAD_CTL_PAD_GPIO_DISP_B1_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_06)

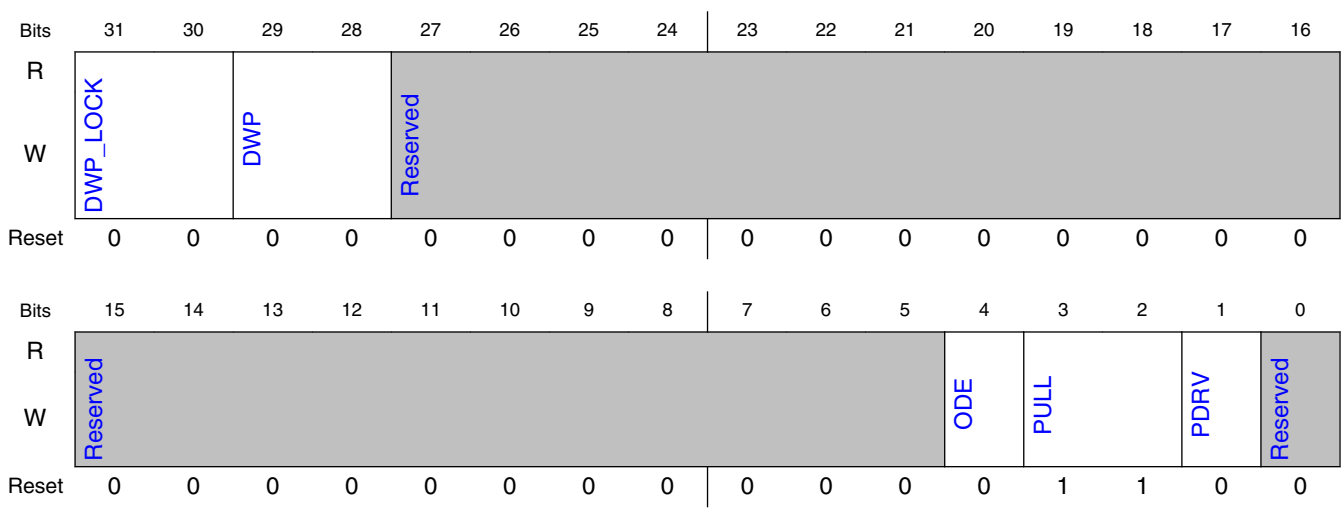
12.4.6.270.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_06	440h

12.4.6.270.2 Function

SW_PAD_CTL Register

12.4.6.270.3 Diagram



12.4.6.270.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_06 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_06 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_06 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.271 SW_PAD_CTL_PAD_GPIO_DISP_B1_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_07)

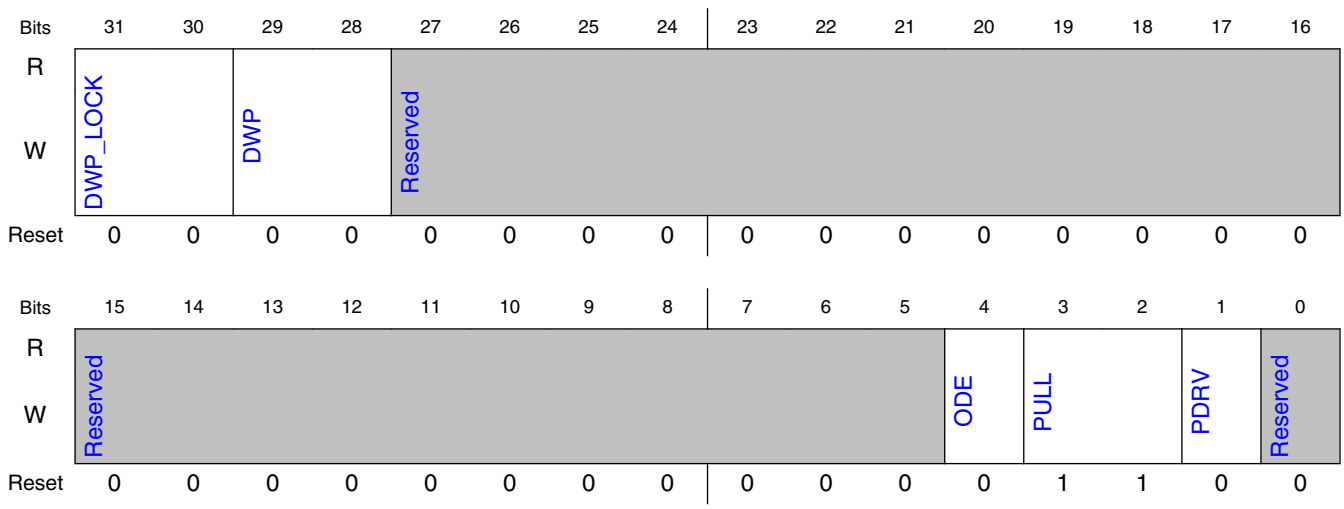
12.4.6.271.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_07	444h

12.4.6.271.2 Function

SW_PAD_CTL Register

12.4.6.271.3 Diagram



12.4.6.271.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_07 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_07 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_07 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.272 SW_PAD_CTL_PAD_GPIO_DISP_B1_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_08)

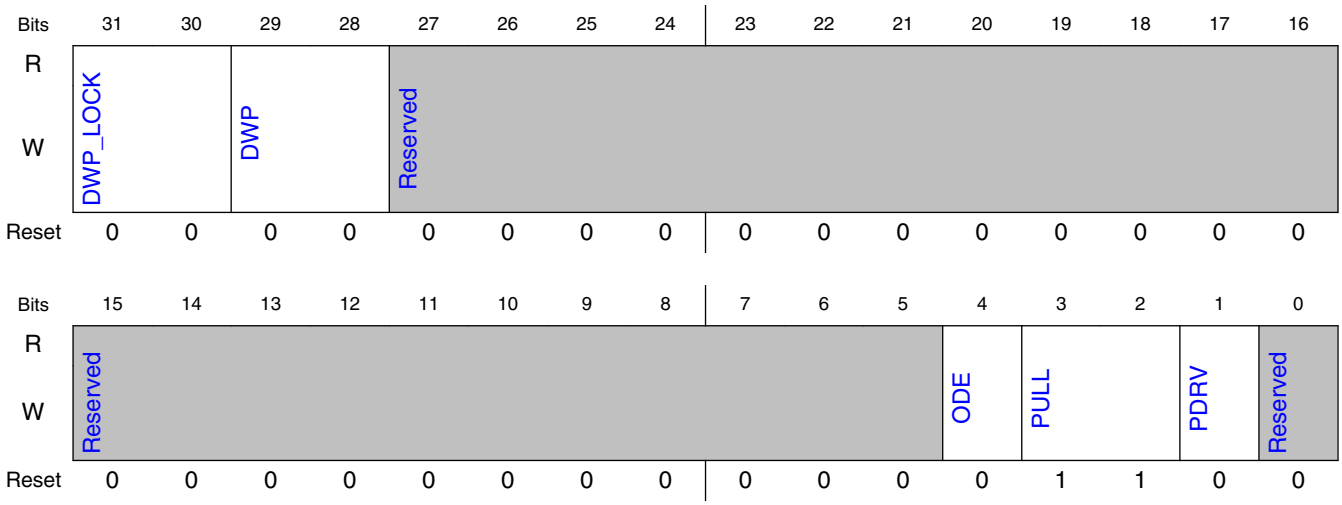
12.4.6.272.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_08	448h

12.4.6.272.2 Function

SW_PAD_CTL Register

12.4.6.272.3 Diagram



12.4.6.272.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_08 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_08 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_08 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.273 SW_PAD_CTL_PAD_GPIO_DISP_B1_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_09)

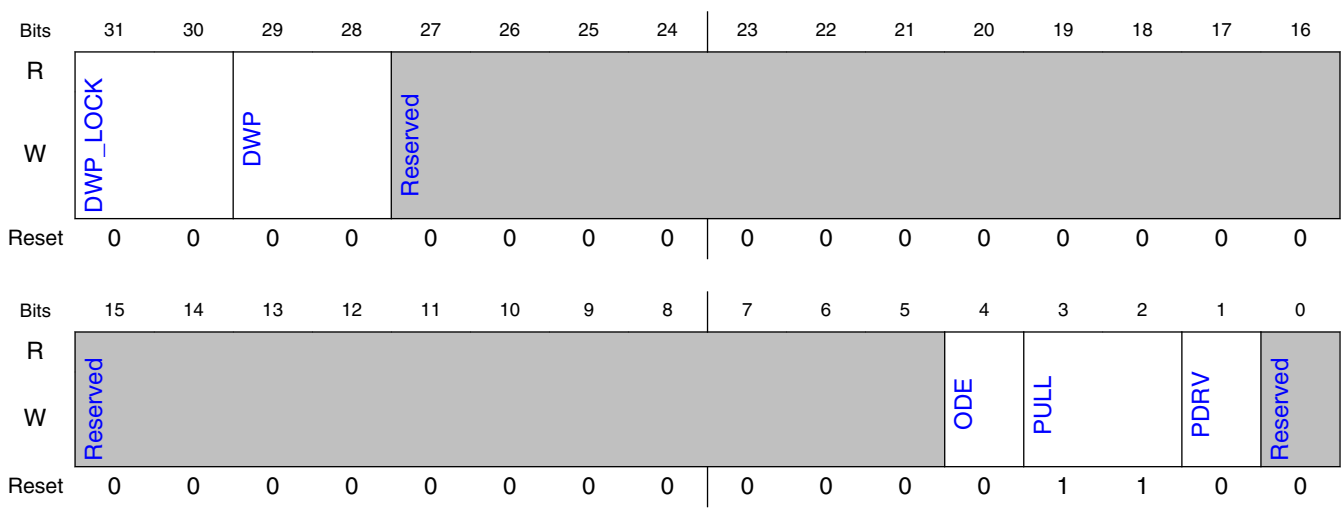
12.4.6.273.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_09	44Ch

12.4.6.273.2 Function

SW_PAD_CTL Register

12.4.6.273.3 Diagram



12.4.6.273.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_09 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_09 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_09 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.274 SW_PAD_CTL_PAD_GPIO_DISP_B1_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_10)

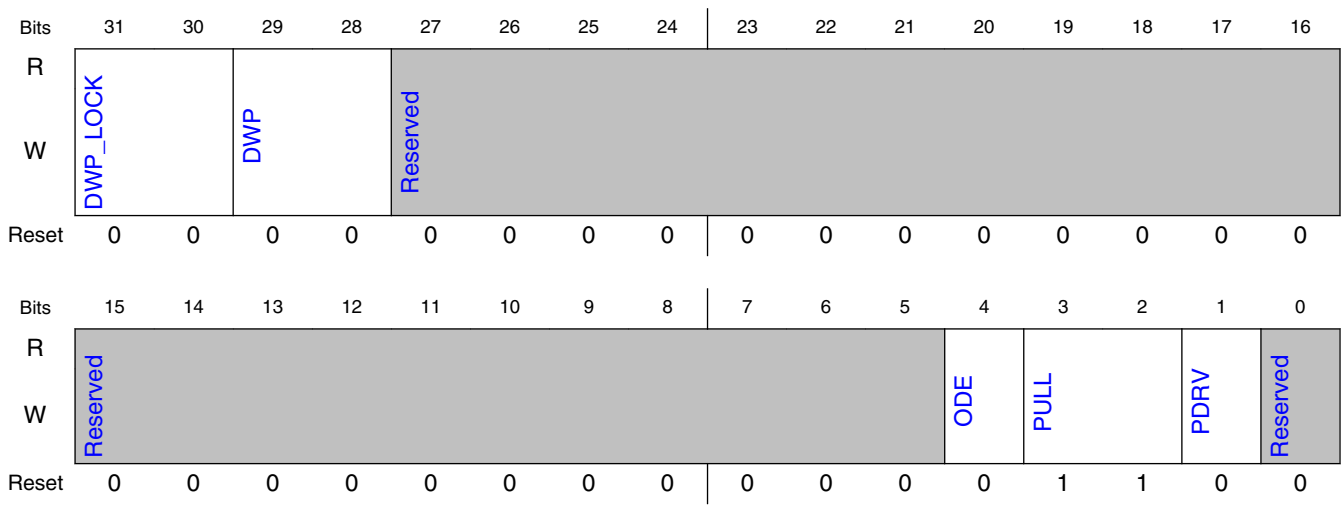
12.4.6.274.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_10	450h

12.4.6.274.2 Function

SW_PAD_CTL Register

12.4.6.274.3 Diagram



12.4.6.274.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5	-

Table continues on the next page...

Field	Function
—	Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_10 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_10 00b - Forbidden 01b - Internal pullup resistor enabled 10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_10 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.275 SW_PAD_CTL_PAD_GPIO_DISP_B1_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B1_11)

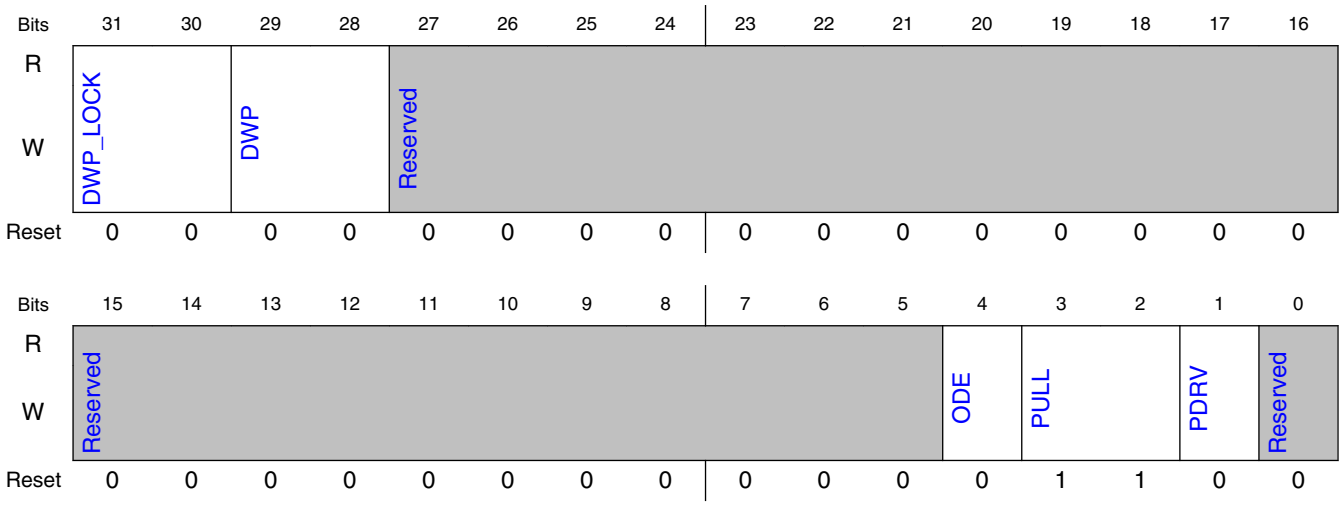
12.4.6.275.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B1_11	454h

12.4.6.275.2 Function

SW_PAD_CTL Register

12.4.6.275.3 Diagram



12.4.6.275.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B1_11 0b - Disabled 1b - Enabled
3-2 PULL	Pull Down Pull Up Field Control signal to enable internal pull-up/down resistors Select one out of next values for pad: GPIO_DISP_B1_11 00b - Forbidden 01b - Internal pullup resistor enabled

Table continues on the next page...

Field	Function
	10b - Internal pulldown resistor enabled 11b - No Pull
1 PDRV	PDRV Field Control signal to configure the pad drive strength. Select one out of next values for pad: GPIO_DISP_B1_11 0b - high drive strength 1b - normal drive strength
0 —	- Reserved

12.4.6.276 SW_PAD_CTL_PAD_GPIO_DISP_B2_00 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_00)

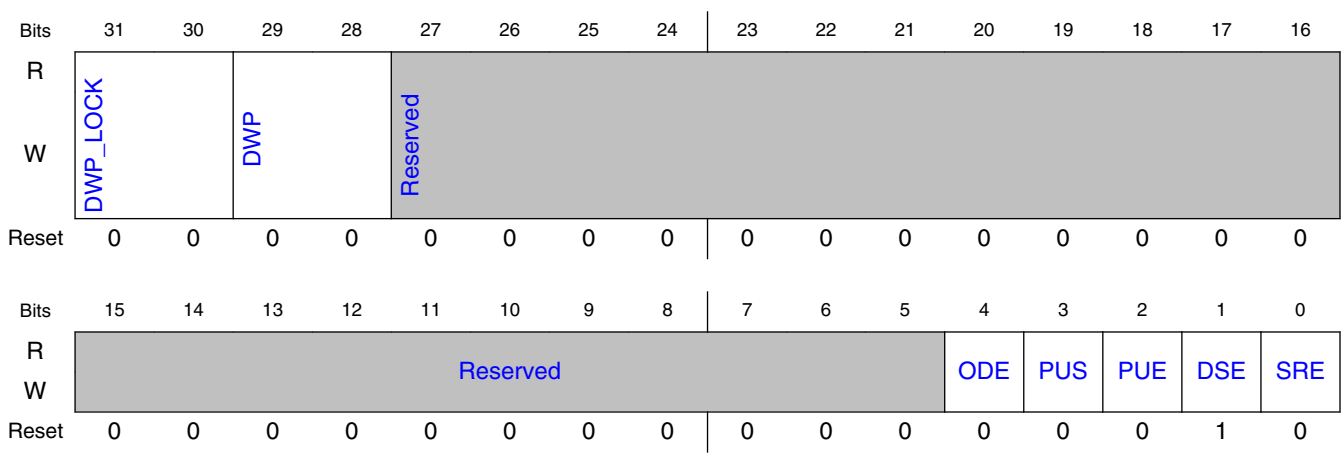
12.4.6.276.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_00	458h

12.4.6.276.2 Function

SW_PAD_CTL Register

12.4.6.276.3 Diagram



12.4.6.276.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_00 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_00 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_00 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_00 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_00 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.277 SW_PAD_CTL_PAD_GPIO_DISP_B2_01 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_01)

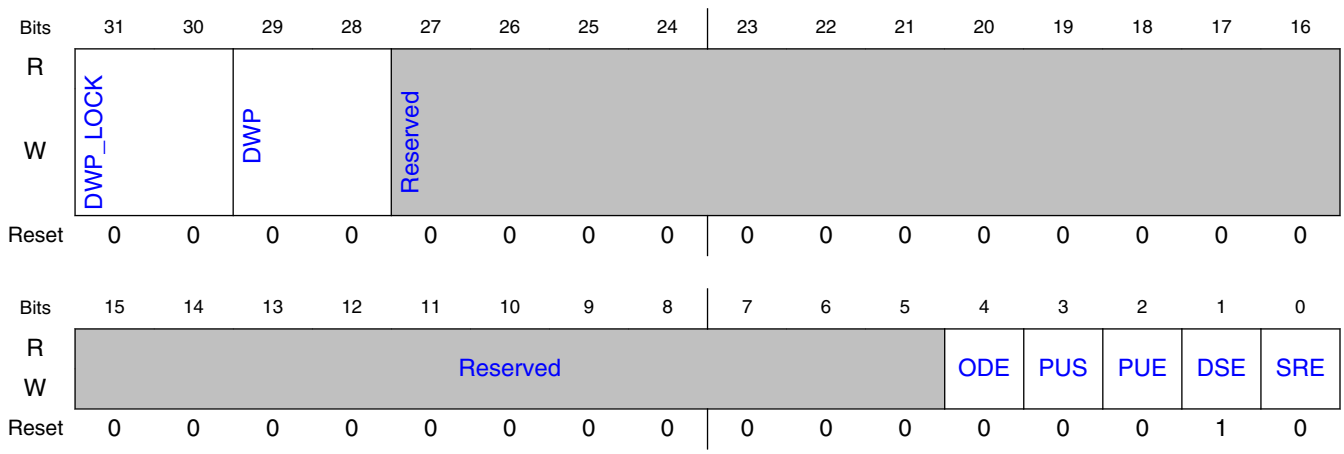
12.4.6.277.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_01	45Ch

12.4.6.277.2 Function

SW_PAD_CTL Register

12.4.6.277.3 Diagram



12.4.6.277.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_01 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_01 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_01 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_01 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_01 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.278 SW_PAD_CTL_PAD_GPIO_DISP_B2_02 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_02)

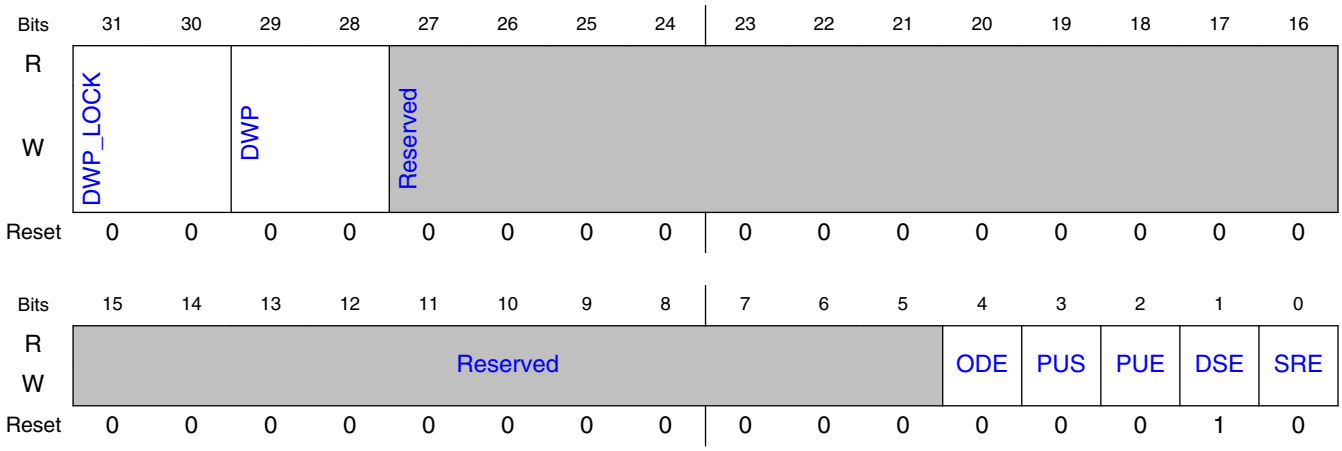
12.4.6.278.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_02	460h

12.4.6.278.2 Function

SW_PAD_CTL Register

12.4.6.278.3 Diagram



12.4.6.278.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_02 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_02 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_02

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_02 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_02 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.279 SW_PAD_CTL_PAD_GPIO_DISP_B2_03 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_03)

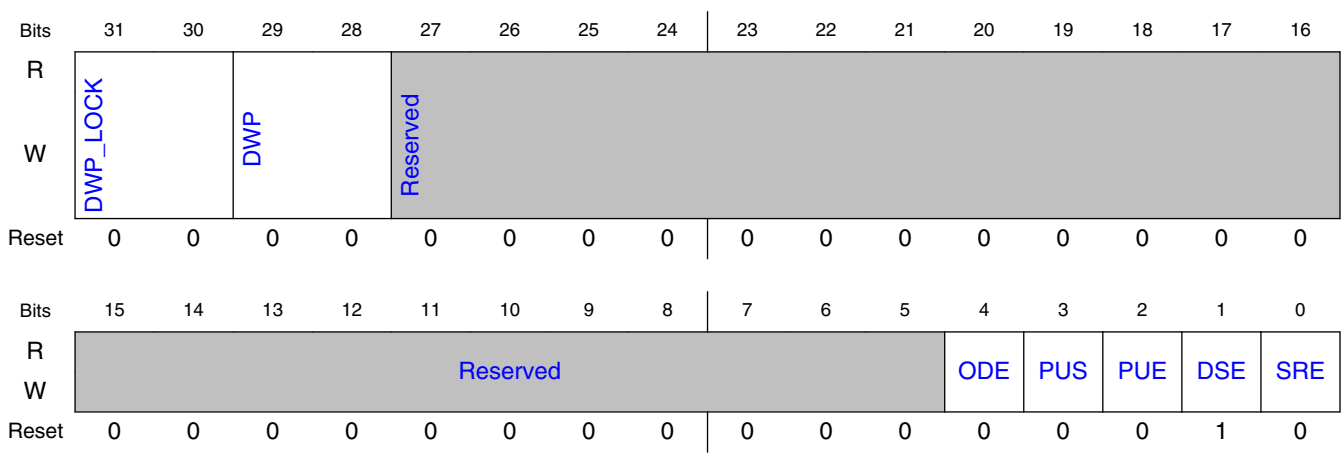
12.4.6.279.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_03	464h

12.4.6.279.2 Function

SW_PAD_CTL Register

12.4.6.279.3 Diagram



12.4.6.279.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_03 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_03 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_03 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_03 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_03 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.280 SW_PAD_CTL_PAD_GPIO_DISP_B2_04 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_04)

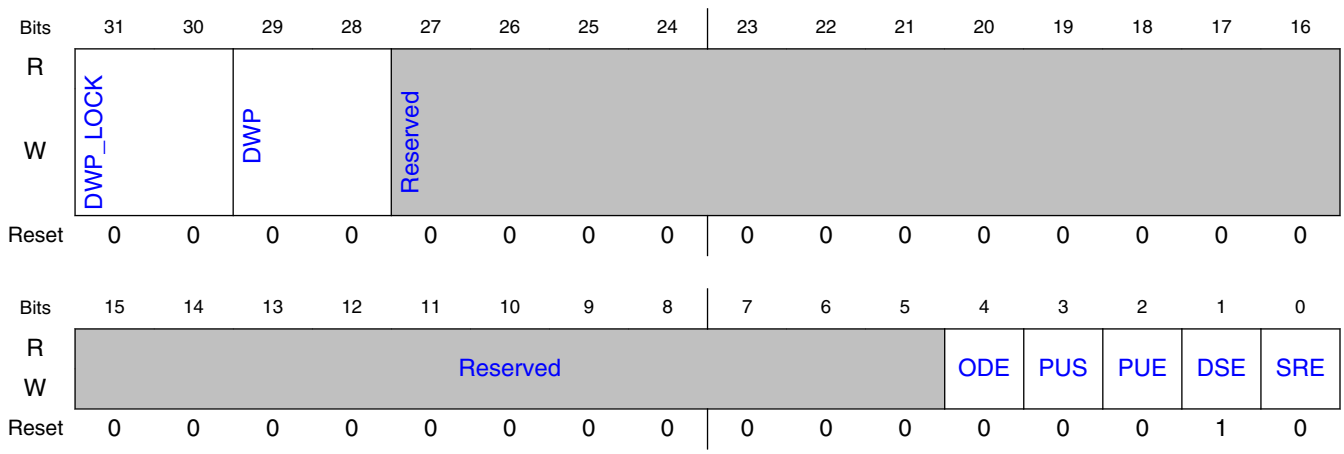
12.4.6.280.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_04	468h

12.4.6.280.2 Function

SW_PAD_CTL Register

12.4.6.280.3 Diagram



12.4.6.280.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_04 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_04 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_04 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_04 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_04 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.281 SW_PAD_CTL_PAD_GPIO_DISP_B2_05 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_05)

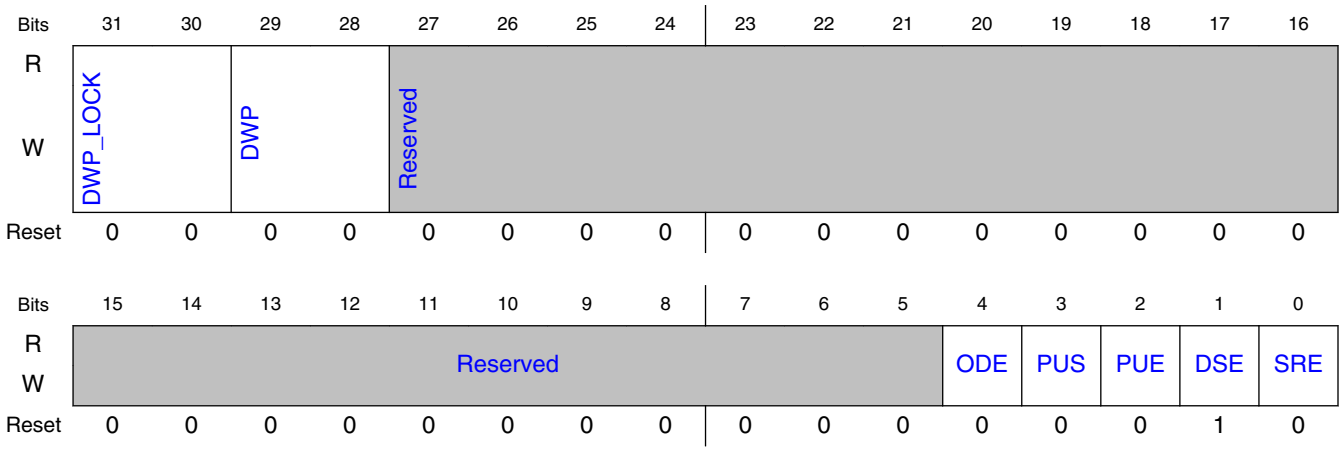
12.4.6.281.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_05	46Ch

12.4.6.281.2 Function

SW_PAD_CTL Register

12.4.6.281.3 Diagram



12.4.6.281.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_05 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_05 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_05

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_05 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_05 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.282 SW_PAD_CTL_PAD_GPIO_DISP_B2_06 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_06)

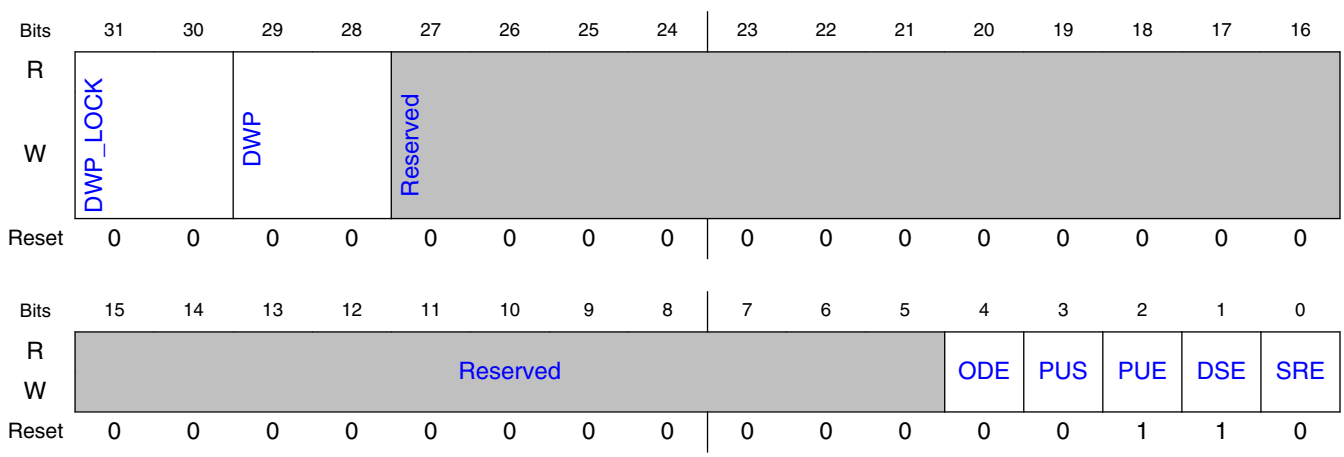
12.4.6.282.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_06	470h

12.4.6.282.2 Function

SW_PAD_CTL Register

12.4.6.282.3 Diagram



12.4.6.282.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_06 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_06 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_06 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_06 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_06 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.283 SW_PAD_CTL_PAD_GPIO_DISP_B2_07 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_07)

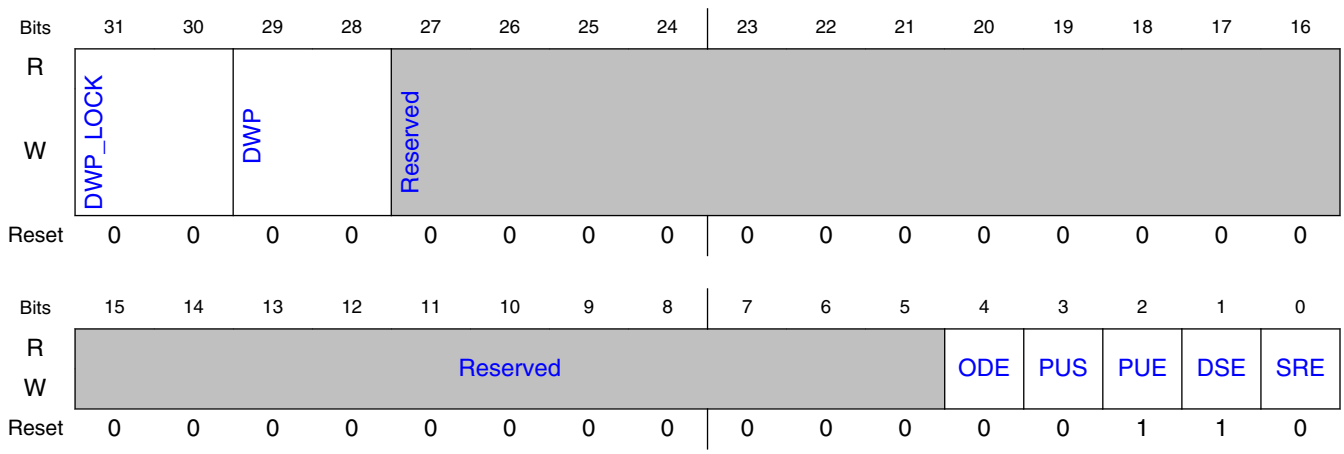
12.4.6.283.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_07	474h

12.4.6.283.2 Function

SW_PAD_CTL Register

12.4.6.283.3 Diagram



12.4.6.283.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_07 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_07 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_07 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_07 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_07 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.284 SW_PAD_CTL_PAD_GPIO_DISP_B2_08 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_08)

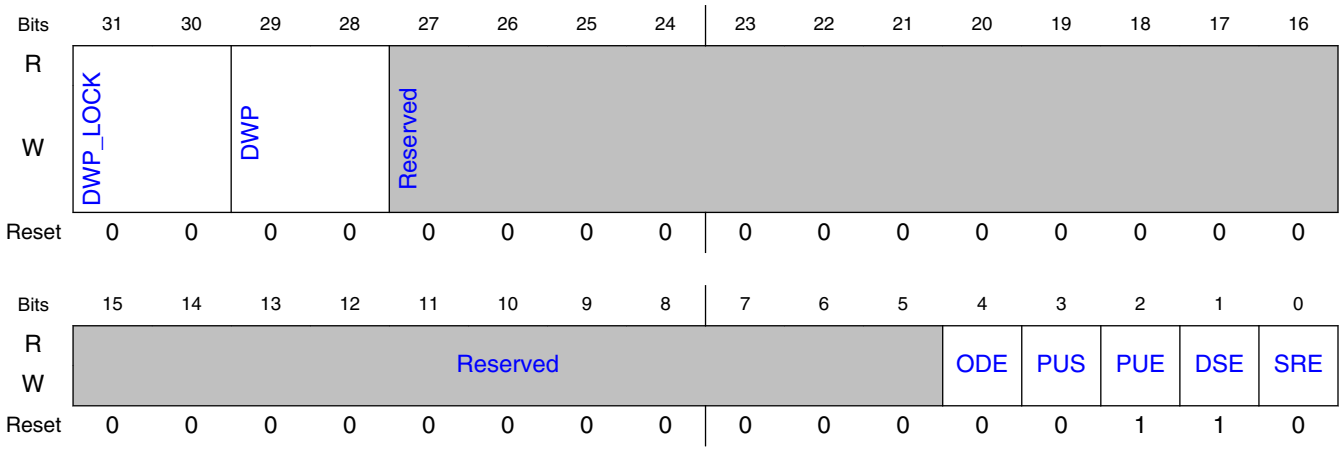
12.4.6.284.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_08	478h

12.4.6.284.2 Function

SW_PAD_CTL Register

12.4.6.284.3 Diagram



12.4.6.284.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_08 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_08 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_08

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_08 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_08 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.285 SW_PAD_CTL_PAD_GPIO_DISP_B2_09 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_09)

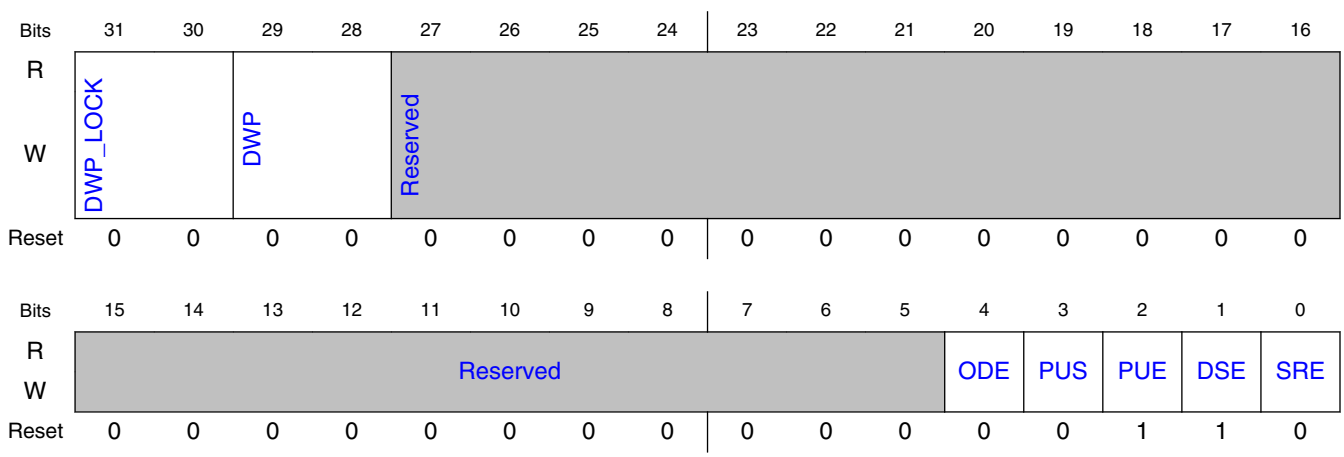
12.4.6.285.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_09	47Ch

12.4.6.285.2 Function

SW_PAD_CTL Register

12.4.6.285.3 Diagram



12.4.6.285.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_09 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_09 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_09 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_09 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_09 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.286 SW_PAD_CTL_PAD_GPIO_DISP_B2_10 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_10)

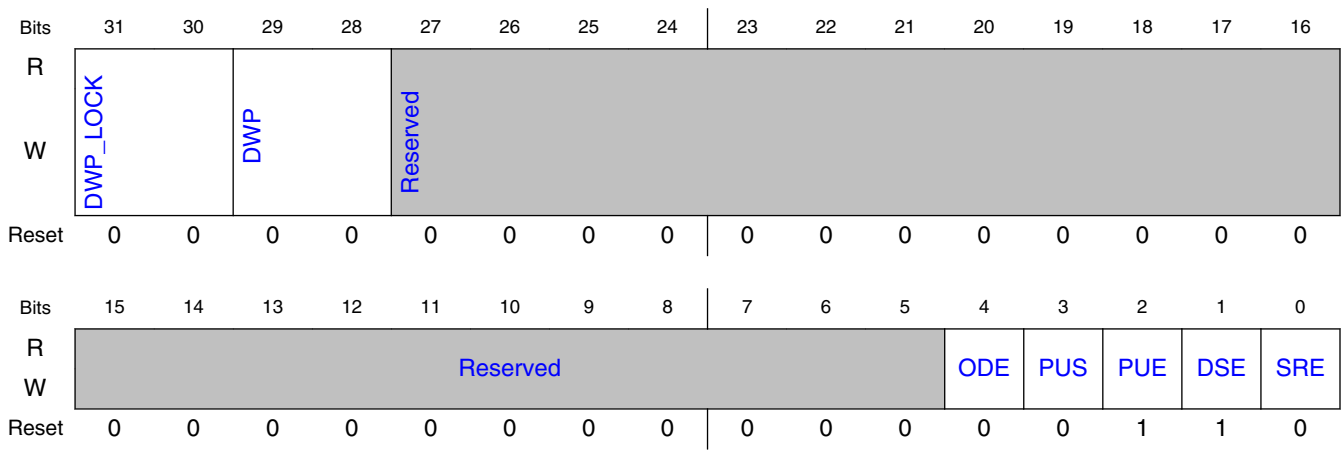
12.4.6.286.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_10	480h

12.4.6.286.2 Function

SW_PAD_CTL Register

12.4.6.286.3 Diagram



12.4.6.286.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_10 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_10 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_10 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_10 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_10 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.287 SW_PAD_CTL_PAD_GPIO_DISP_B2_11 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_11)

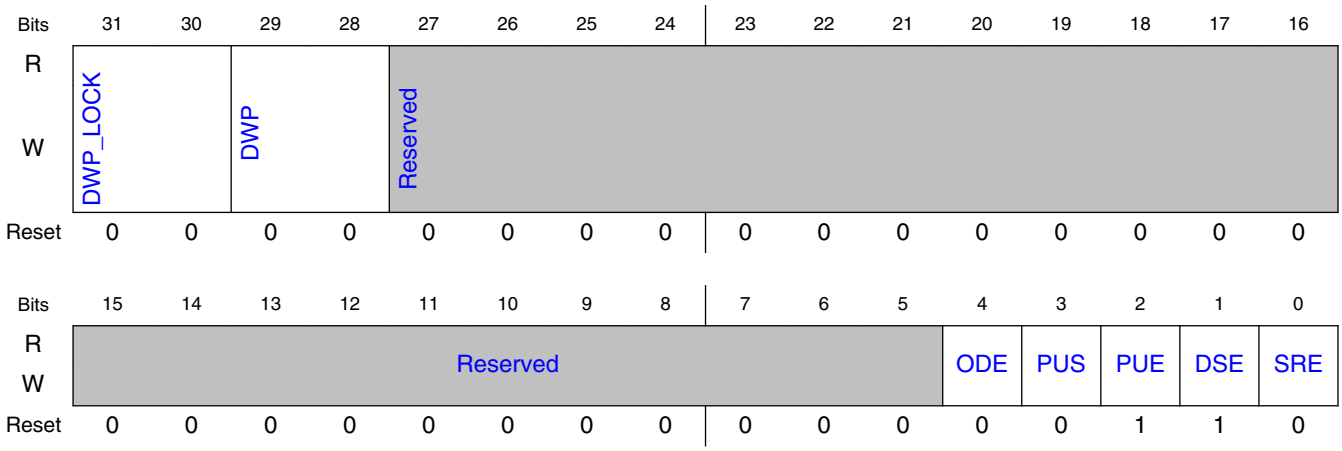
12.4.6.287.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_11	484h

12.4.6.287.2 Function

SW_PAD_CTL Register

12.4.6.287.3 Diagram



12.4.6.287.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_11 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_11 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_11

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_11 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_11 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.288 SW_PAD_CTL_PAD_GPIO_DISP_B2_12 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_12)

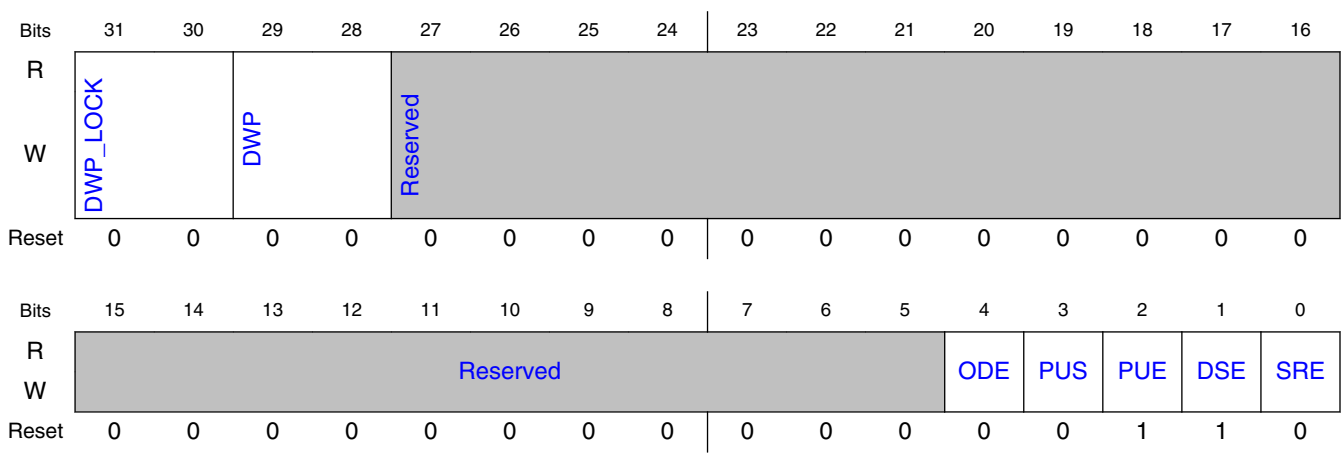
12.4.6.288.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_12	488h

12.4.6.288.2 Function

SW_PAD_CTL Register

12.4.6.288.3 Diagram



12.4.6.288.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_12 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_12 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_12 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_12 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_12 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.289 SW_PAD_CTL_PAD_GPIO_DISP_B2_13 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_13)

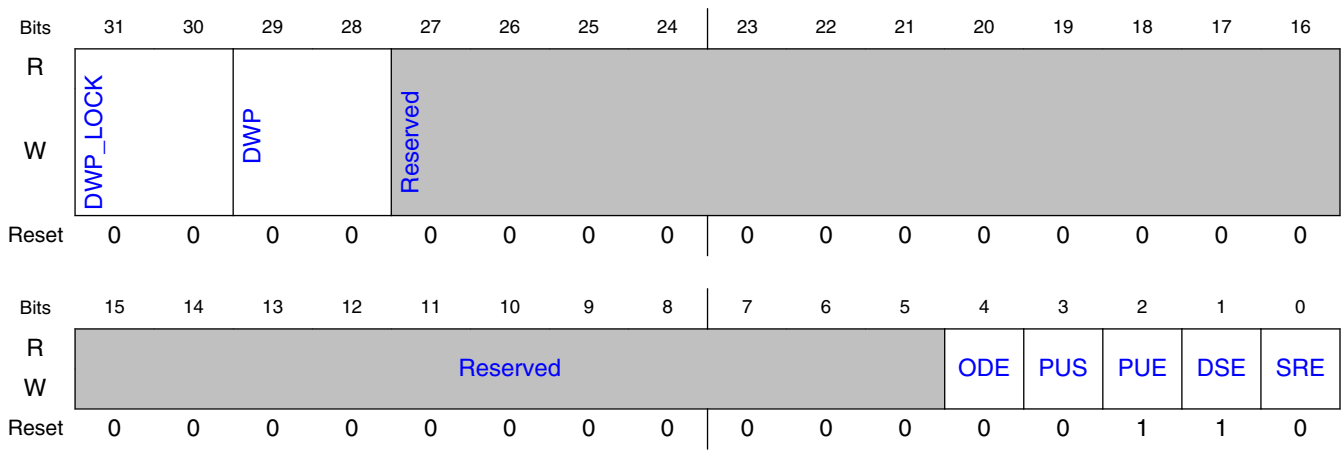
12.4.6.289.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_13	48Ch

12.4.6.289.2 Function

SW_PAD_CTL Register

12.4.6.289.3 Diagram



12.4.6.289.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4	Open Drain Field

Table continues on the next page...

Field	Function
ODE	If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_13 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_13 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_13 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_13 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_13 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.290 SW_PAD_CTL_PAD_GPIO_DISP_B2_14 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_14)

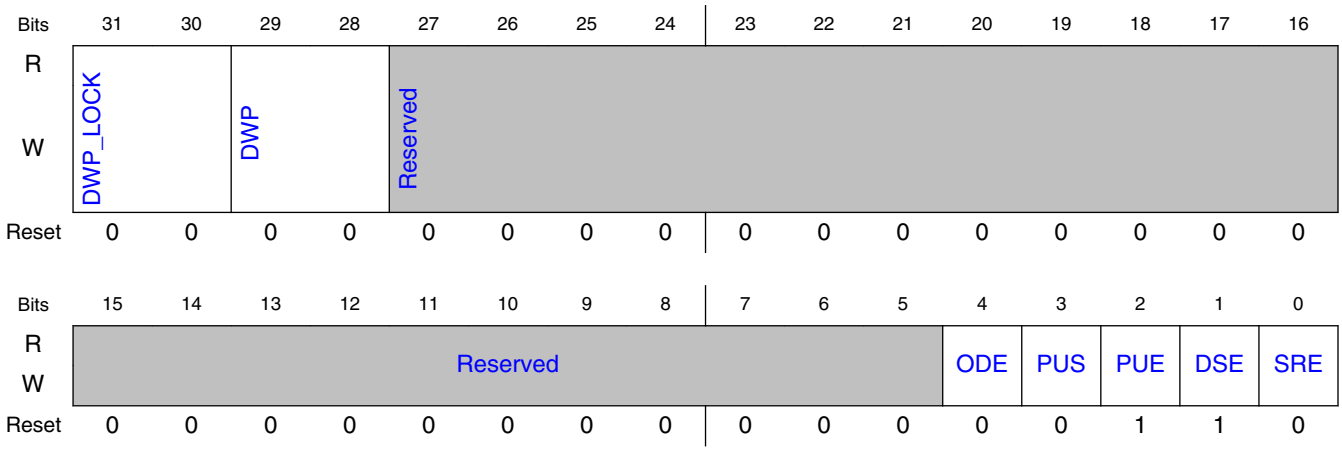
12.4.6.290.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_14	490h

12.4.6.290.2 Function

SW_PAD_CTL Register

12.4.6.290.3 Diagram



12.4.6.290.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_14 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_14 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_14

Table continues on the next page...

Field	Function
	0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_14 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_14 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.291 SW_PAD_CTL_PAD_GPIO_DISP_B2_15 SW PAD Control Register (SW_PAD_CTL_PAD_GPIO_DISP_B2_15)

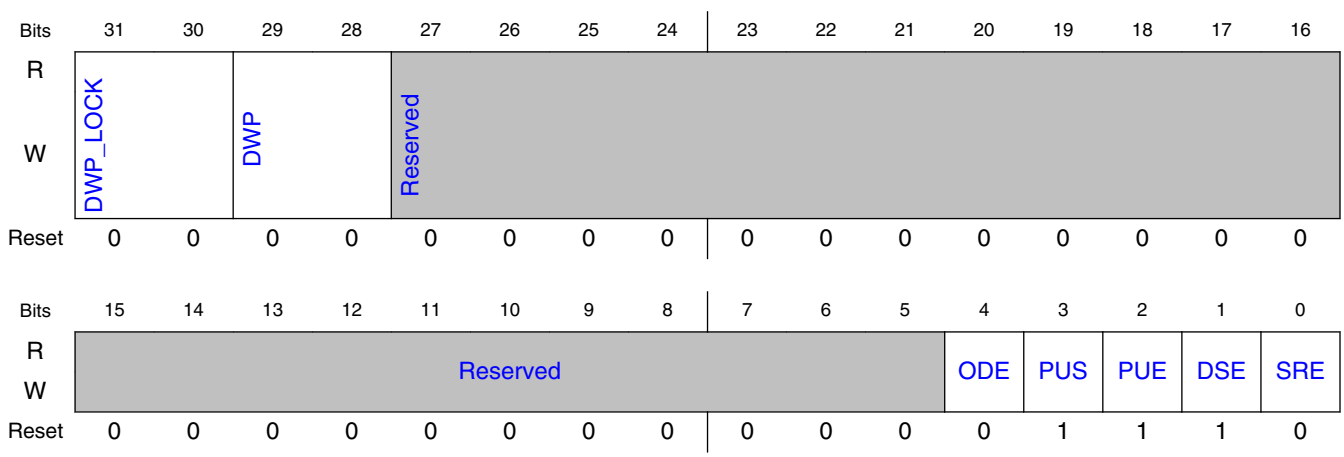
12.4.6.291.1 Offset

Register	Offset
SW_PAD_CTL_PAD_GPIO_DISP_B2_15	494h

12.4.6.291.2 Function

SW_PAD_CTL Register

12.4.6.291.3 Diagram



12.4.6.291.4 Fields

Field	Function
31-30 DWP_LOCK	Domain write protection lock Once DWP_LOCK bit i is set the write to DWP bit i will be blocked. 00b - Neither of DWP bits is locked 01b - The lower DWP bit is locked 10b - The higher DWP bit is locked 11b - Both DWP bits are locked
29-28 DWP	Domain write protection These 2 bits determine which core is forbidden to change bits 6:0. When bit 0 is set, CM7 is forbidden. When bit 1 is set, CM4 is forbidden. 00b - Both cores are allowed 01b - CM7 is forbidden 10b - CM4 is forbidden 11b - Both cores are forbidden
27-5 —	- Reserved
4 ODE	Open Drain Field If set to 1, the output driver drives only logic 0. The drain of the internal transistor is open. It means that logic 1 has to be driven by an external component. This option is essential if connection between the pad and an external component is bi-directional. If ODE = 0, then the output driver drives logic 1 and logic 0. Select one out of next values for pad: GPIO_DISP_B2_15 0b - Disabled 1b - Enabled
3 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_DISP_B2_15 0b - Weak pull down 1b - Weak pull up
2 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_DISP_B2_15 0b - Pull Disable 1b - Pull Enable
1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_DISP_B2_15 0b - normal drive strength 1b - high drive strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_DISP_B2_15 0b - Fast Slew Rate 1b - Slow Slew Rate

12.4.6.292 FLEXCAN1_RX_SELECT_INPUT DAISY Register (FLEXCAN1_RX_SELECT_INPUT)

12.4.6.292.1 Offset

Register	Offset
FLEXCAN1_RX_SELECT_INPUT	498h

12.4.6.292.2 Function

DAISY Register

12.4.6.292.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.292.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: FLEXCAN1, In Pin: canrx 00b - Selecting Pad: GPIO_AD_07 for Mode: ALT1 01b - Selecting Pad: GPIO_DISP_B2_13 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B2_15 for Mode: ALT6

12.4.6.293 FLEXCAN2_RX_SELECT_INPUT DAISY Register (FLEXCAN2_RX_SELECT_INPUT)

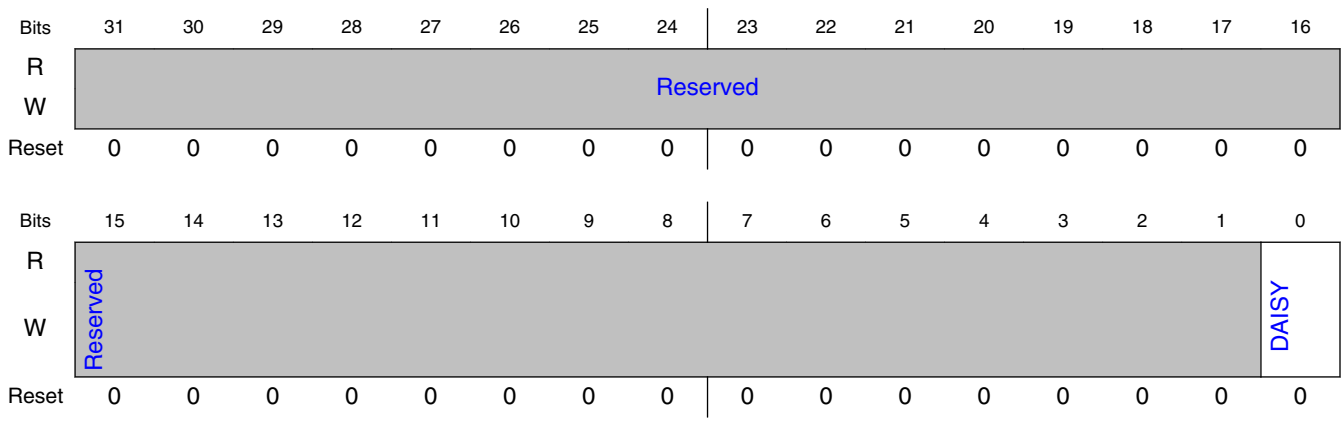
12.4.6.293.1 Offset

Register	Offset
FLEXCAN2_RX_SELECT_INPUT	49Ch

12.4.6.293.2 Function

DAISY Register

12.4.6.293.3 Diagram



12.4.6.293.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: FLEXCAN2, In Pin: canrx 0b - Selecting Pad: GPIO_AD_01 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_31 for Mode: ALT2

12.4.6.294 CCM_ENET_QOS_REF_CLK_SELECT_INPUT DAISY Register (CCM_ENET_QOS_REF_CLK_SELECT_INPUT)

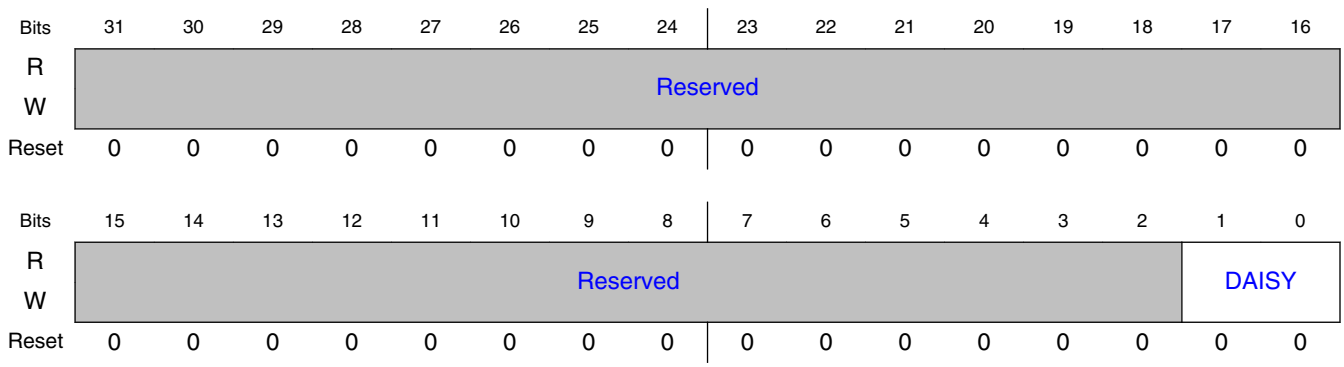
12.4.6.294.1 Offset

Register	Offset
CCM_ENET_QOS_REF_CLK_SELECT_INPUT	4A0h

12.4.6.294.2 Function

DAISY Register

12.4.6.294.3 Diagram



12.4.6.294.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: ref_clk 00b - Selecting Pad: GPIO_EMC_B2_20 for Mode: ALT3 01b - Selecting Pad: GPIO_SD_B2_07 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_11 for Mode: ALT9

12.4.6.295 CCM_ENET_QOS_TX_CLK_SELECT_INPUT DAISY Register (CCM_ENET_QOS_TX_CLK_SELECT_INPUT)

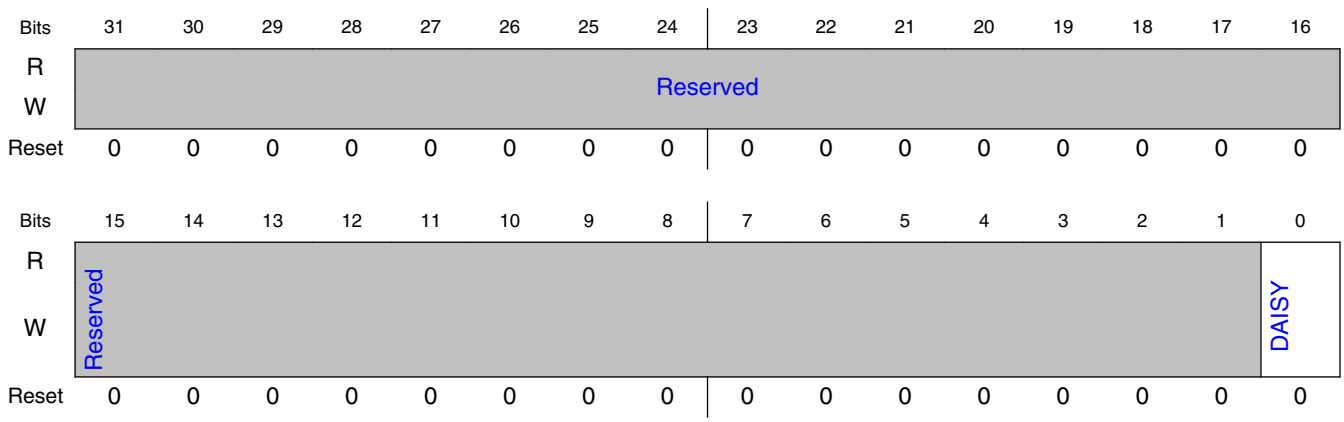
12.4.6.295.1 Offset

Register	Offset
CCM_ENET_QOS_TX_CLK_SELECT_INPUT	4A4h

12.4.6.295.2 Function

DAISY Register

12.4.6.295.3 Diagram



12.4.6.295.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: tx_clk 0b - Selecting Pad: GPIO_DISP_B1_11 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B2_05 for Mode: ALT8

12.4.6.296 ENET_IPG_CLK_RMII_SELECT_INPUT DAISY Register (ENET_IPG_CLK_RMII_SELECT_INPUT)

12.4.6.296.1 Offset

Register	Offset
ENET_IPG_CLK_RMII_SELECT_INPUT	4A8h

12.4.6.296.2 Function

DAISY Register

12.4.6.296.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.296.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: ipg_clk_rmii 00b - Selecting Pad: GPIO_AD_29 for Mode: ALT2 01b - Selecting Pad: GPIO_DISP_B2_05 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B2_13 for Mode: ALT4

12.4.6.297 ENET_MAC0_MDIO_SELECT_INPUT DAISY Register (ENET_MAC0_MDIO_SELECT_INPUT)

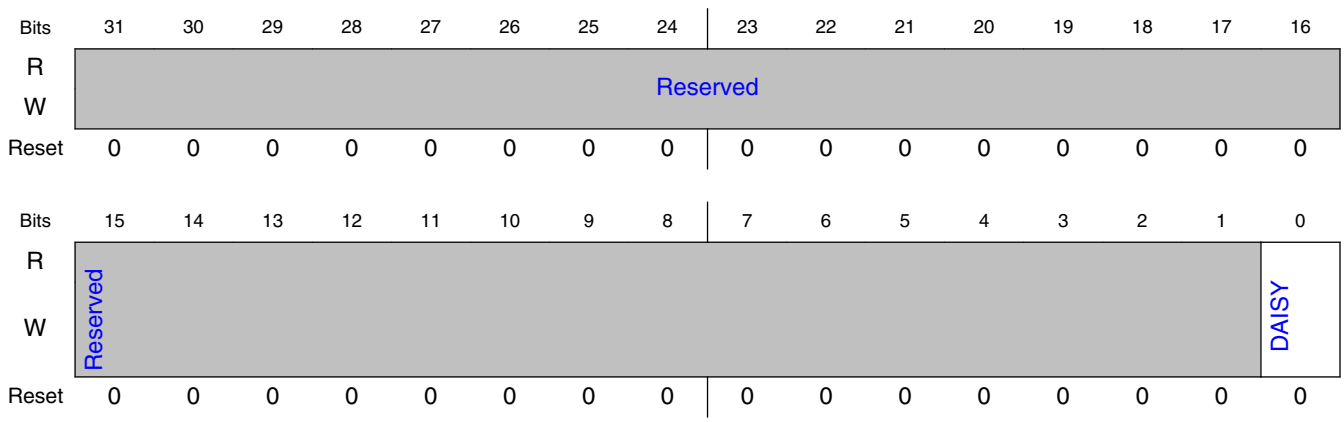
12.4.6.297.1 Offset

Register	Offset
ENET_MAC0_MDIO_SELECT_INPUT	4ACh

12.4.6.297.2 Function

DAISY Register

12.4.6.297.3 Diagram



12.4.6.297.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_mdio 0b - Selecting Pad: GPIO_EMC_B2_20 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_33 for Mode: ALT3

12.4.6.298 ENET_MAC0_RXDATA_SELECT_INPUT_0 DAISY Register (ENET_MAC0_RXDATA_SELECT_INPUT_0)

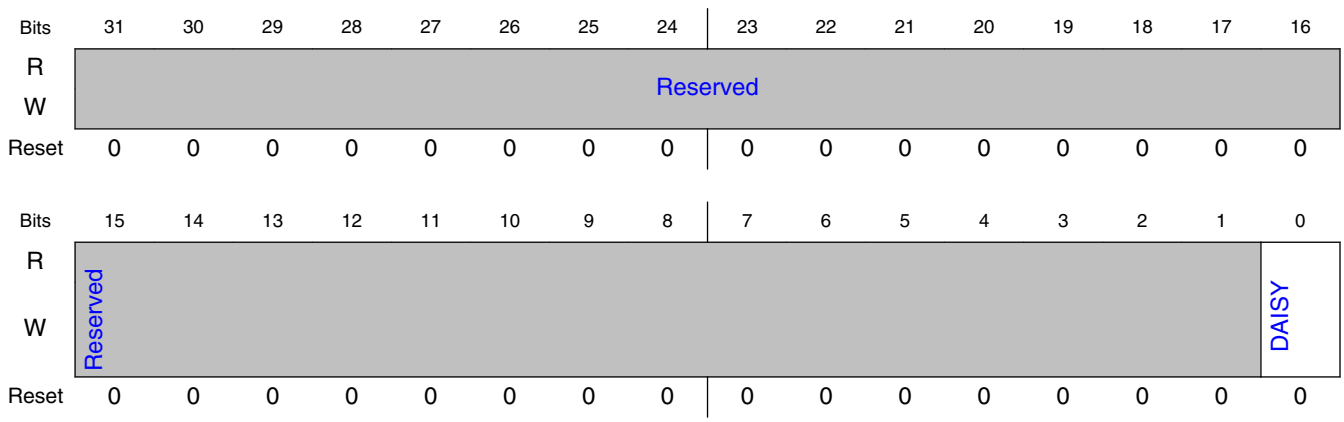
12.4.6.298.1 Offset

Register	Offset
ENET_MAC0_RXDATA_SELECT_INPUT_0	4B0h

12.4.6.298.2 Function

DAISY Register

12.4.6.298.3 Diagram



12.4.6.298.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_rxdata0 0b - Selecting Pad: GPIO_AD_26 for Mode: ALT3 1b - Selecting Pad: GPIO_DISP_B2_06 for Mode: ALT1

12.4.6.299 ENET_MAC0_RXDATA_SELECT_INPUT_1 DAISY Register (ENET_MAC0_RXDATA_SELECT_INPUT_1)

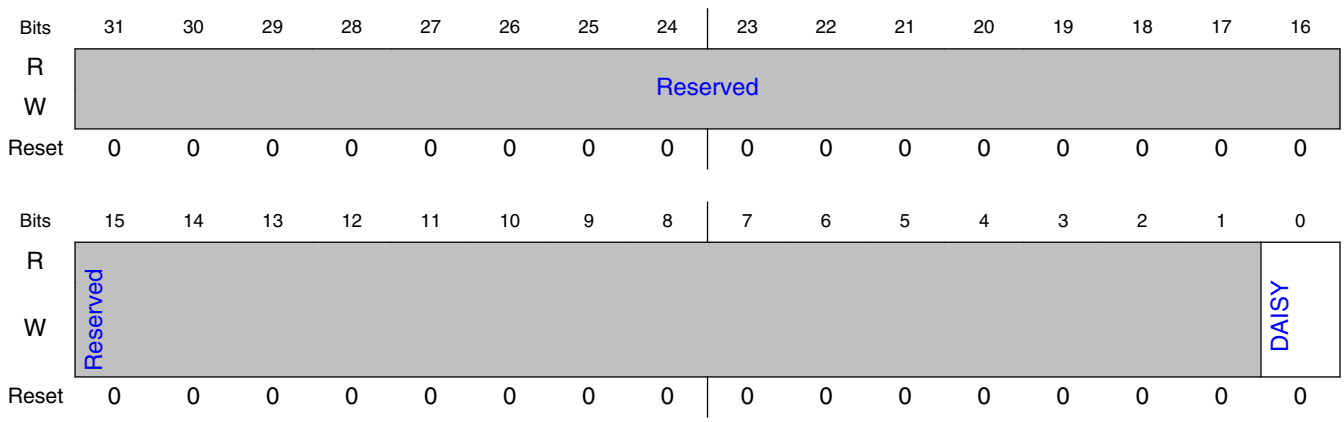
12.4.6.299.1 Offset

Register	Offset
ENET_MAC0_RXDATA_SELECT_INPUT_1	4B4h

12.4.6.299.2 Function

DAISY Register

12.4.6.299.3 Diagram



12.4.6.299.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_rxdata1 0b - Selecting Pad: GPIO_AD_27 for Mode: ALT3 1b - Selecting Pad: GPIO_DISP_B2_07 for Mode: ALT1

12.4.6.300 ENET_MAC0_RXEN_SELECT_INPUT DAISY Register (ENET_MAC0_RXEN_SELECT_INPUT)

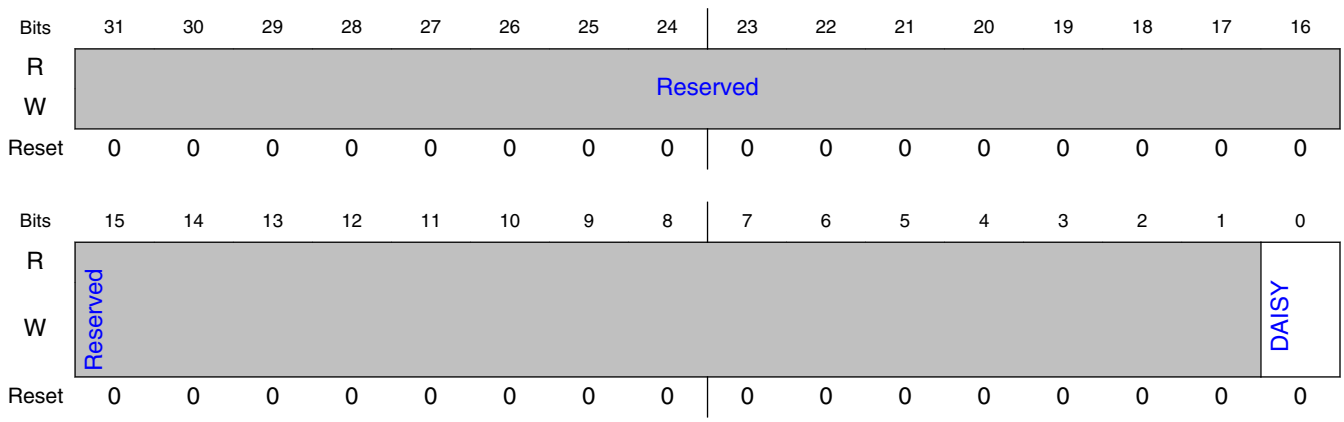
12.4.6.300.1 Offset

Register	Offset
ENET_MAC0_RXEN_SELECT_INPUT	4B8h

12.4.6.300.2 Function

DAISY Register

12.4.6.300.3 Diagram



12.4.6.300.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_rxen 0b - Selecting Pad: GPIO_AD_24 for Mode: ALT3 1b - Selecting Pad: GPIO_DISP_B2_08 for Mode: ALT1

12.4.6.301 ENET_MAC0_RXERR_SELECT_INPUT DAISY Register (ENET_MAC0_RXERR_SELECT_INPUT)

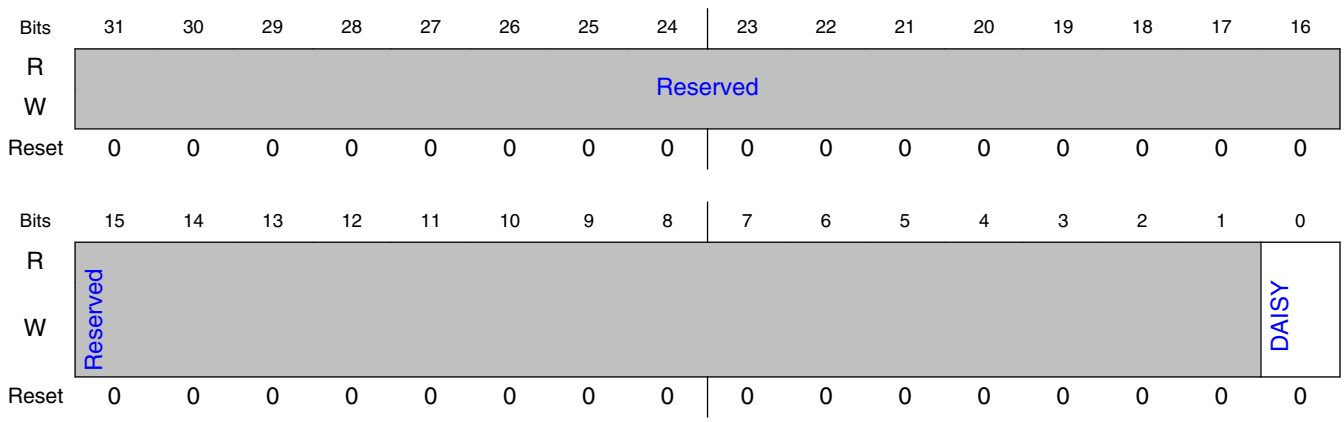
12.4.6.301.1 Offset

Register	Offset
ENET_MAC0_RXERR_SELECT_INPUT	4BCh

12.4.6.301.2 Function

DAISY Register

12.4.6.301.3 Diagram



12.4.6.301.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_rxerr 0b - Selecting Pad: GPIO_AD_25 for Mode: ALT3 1b - Selecting Pad: GPIO_DISP_B2_09 for Mode: ALT1

12.4.6.302 ENET_MAC0_TXCLK_SELECT_INPUT DAISY Register (ENET_MAC0_TXCLK_SELECT_INPUT)

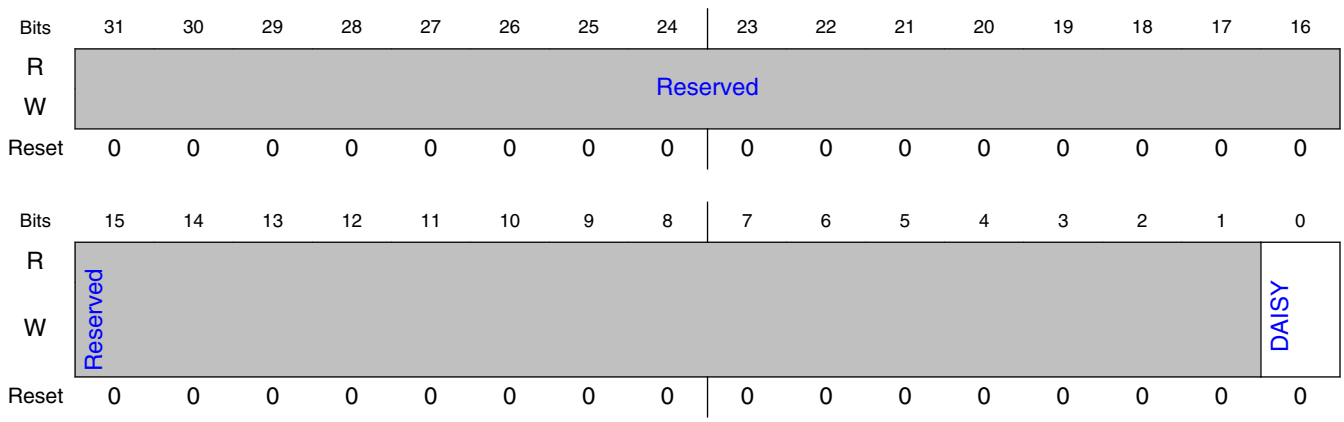
12.4.6.302.1 Offset

Register	Offset
ENET_MAC0_TXCLK_SELECT_INPUT	4C0h

12.4.6.302.2 Function

DAISY Register

12.4.6.302.3 Diagram



12.4.6.302.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet, In Pin: mac0_txclk 0b - Selecting Pad: GPIO_AD_29 for Mode: ALT3 1b - Selecting Pad: GPIO_DISP_B2_05 for Mode: ALT1

12.4.6.303 ENET_1G_IPG_CLK_RMII_SELECT_INPUT DAISY Register (ENET_1G_IPG_CLK_RMII_SELECT_INPUT)

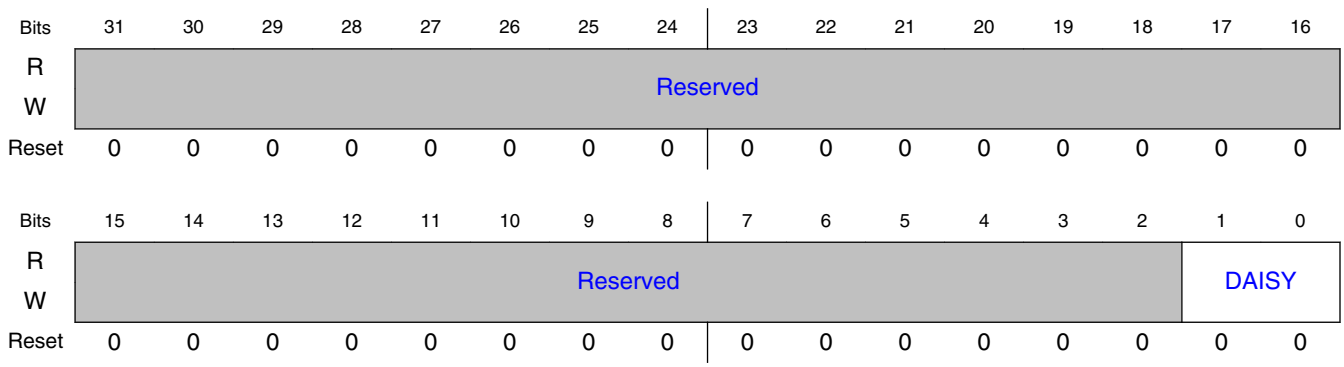
12.4.6.303.1 Offset

Register	Offset
ENET_1G_IPG_CLK_RMII_SELECT_INPUT	4C4h

12.4.6.303.2 Function

DAISY Register

12.4.6.303.3 Diagram



12.4.6.303.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: ipg_clk_rmii 00b - Selecting Pad: GPIO_EMC_B2_19 for Mode: ALT3 01b - Selecting Pad: GPIO_SD_B2_11 for Mode: ALT3 10b - Selecting Pad: GPIO_DISP_B1_11 for Mode: ALT2 11b - Selecting Pad: GPIO_DISP_B2_14 for Mode: ALT4

12.4.6.304 ENET_1G_MAC0_MDIO_SELECT_INPUT DAISY Register (ENET_1G_MAC0_MDIO_SELECT_INPUT)

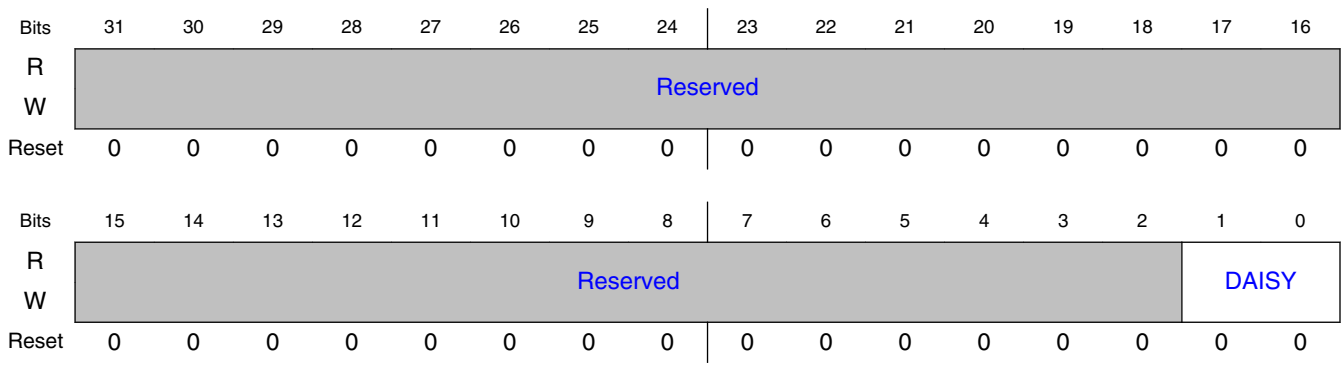
12.4.6.304.1 Offset

Register	Offset
ENET_1G_MAC0_MDIO_SELECT_INPUT	4C8h

12.4.6.304.2 Function

DAISY Register

12.4.6.304.3 Diagram



12.4.6.304.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_mdio 00b - Selecting Pad: GPIO_EMC_B1_41 for Mode: ALT7 01b - Selecting Pad: GPIO_EMC_B2_20 for Mode: ALT2 10b - Selecting Pad: GPIO_AD_17 for Mode: ALT9 11b - Selecting Pad: GPIO_AD_33 for Mode: ALT9

12.4.6.305 ENET_1G_MAC0_RXCLK_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXCLK_SELECT_INPUT)

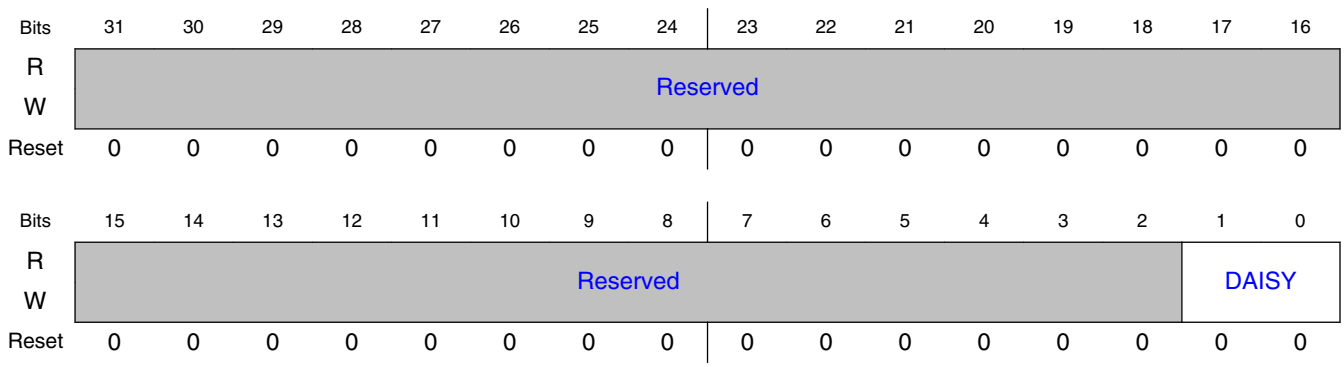
12.4.6.305.1 Offset

Register	Offset
ENET_1G_MAC0_RXCLK_SELECT_INPUT	4CCh

12.4.6.305.2 Function

DAISY Register

12.4.6.305.3 Diagram



12.4.6.305.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxclk 00b - Selecting Pad: GPIO_EMC_B2_05 for Mode: ALT7 01b - Selecting Pad: GPIO_SD_B2_01 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_01 for Mode: ALT1

12.4.6.306 ENET_1G_MAC0_RXDATA_0_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_0_SELECT_INPUT)

12.4.6.306.1 Offset

Register	Offset
ENET_1G_MAC0_RXDATA_0_SELECT_INPUT	4D0h

12.4.6.306.2 Function

DAISY Register

12.4.6.306.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.306.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxdata_0 00b - Selecting Pad: GPIO_EMC_B2_15 for Mode: ALT2 01b - Selecting Pad: GPIO_SD_B2_02 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT1

12.4.6.307 ENET_1G_MAC0_RXDATA_1_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_1_SELECT_INPUT)

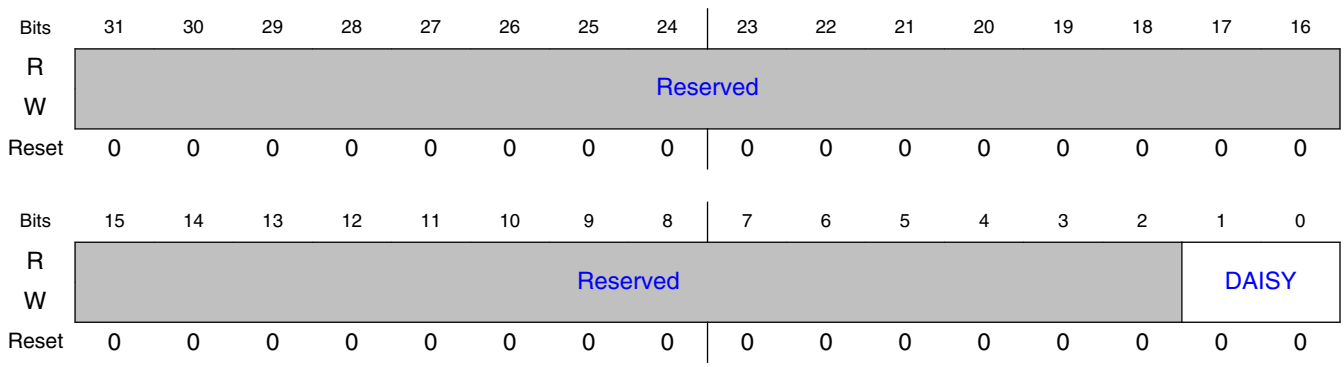
12.4.6.307.1 Offset

Register	Offset
ENET_1G_MAC0_RXDATA_1_SELECT_INPUT	4D4h

12.4.6.307.2 Function

DAISY Register

12.4.6.307.3 Diagram



12.4.6.307.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxdata_1 00b - Selecting Pad: GPIO_EMC_B2_16 for Mode: ALT2 01b - Selecting Pad: GPIO_SD_B2_03 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT1

12.4.6.308 ENET_1G_MAC0_RXDATA_2_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_2_SELECT_INPUT)

12.4.6.308.1 Offset

Register	Offset
ENET_1G_MAC0_RXDATA_2_SELECT_INPUT	4D8h

12.4.6.308.2 Function

DAISY Register

12.4.6.308.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.308.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxdata_2 00b - Selecting Pad: GPIO_EMC_B2_08 for Mode: ALT7 01b - Selecting Pad: GPIO_SD_B2_04 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_04 for Mode: ALT1

12.4.6.309 ENET_1G_MAC0_RXDATA_3_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXDATA_3_SELECT_INPUT)

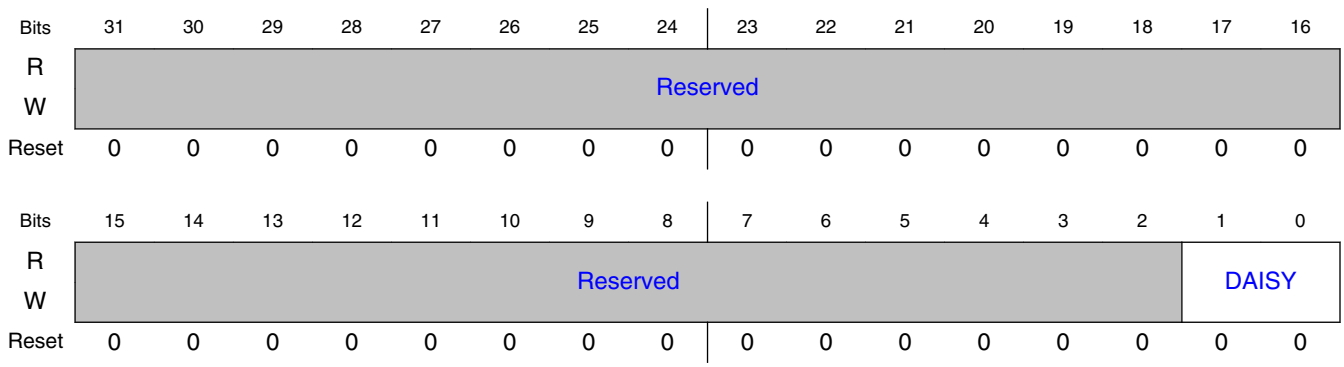
12.4.6.309.1 Offset

Register	Offset
ENET_1G_MAC0_RXDATA_3_SELECT_INPUT	4DCh

12.4.6.309.2 Function

DAISY Register

12.4.6.309.3 Diagram



12.4.6.309.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxdata_3 00b - Selecting Pad: GPIO_EMC_B2_07 for Mode: ALT7 01b - Selecting Pad: GPIO_SD_B2_05 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_05 for Mode: ALT1

12.4.6.310 ENET_1G_MAC0_RXEN_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXEN_SELECT_INPUT)

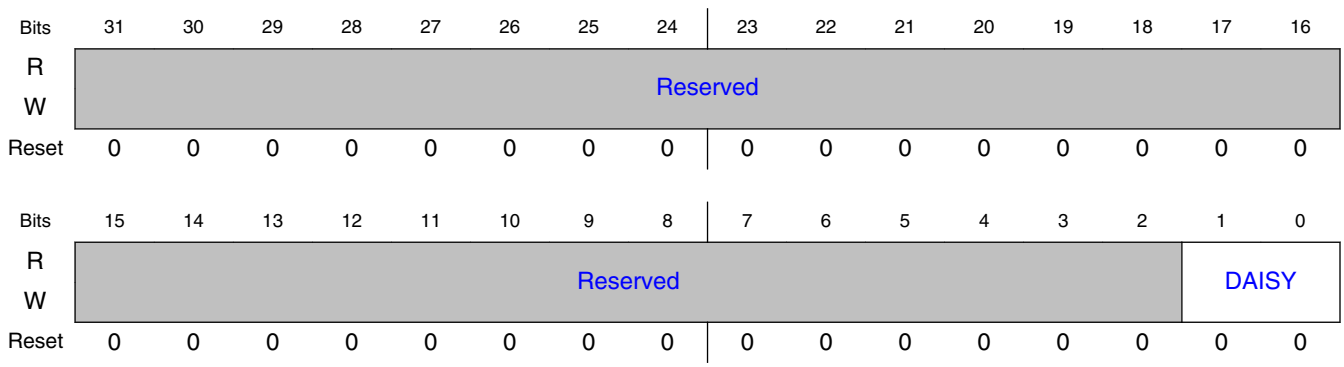
12.4.6.310.1 Offset

Register	Offset
ENET_1G_MAC0_RXEN_SELECT_INPUT	4E0h

12.4.6.310.2 Function

DAISY Register

12.4.6.310.3 Diagram



12.4.6.310.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxen 00b - Selecting Pad: GPIO_EMC_B2_17 for Mode: ALT2 01b - Selecting Pad: GPIO_SD_B2_00 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_00 for Mode: ALT1

12.4.6.311 ENET_1G_MAC0_RXERR_SELECT_INPUT DAISY Register (ENET_1G_MAC0_RXERR_SELECT_INPUT)

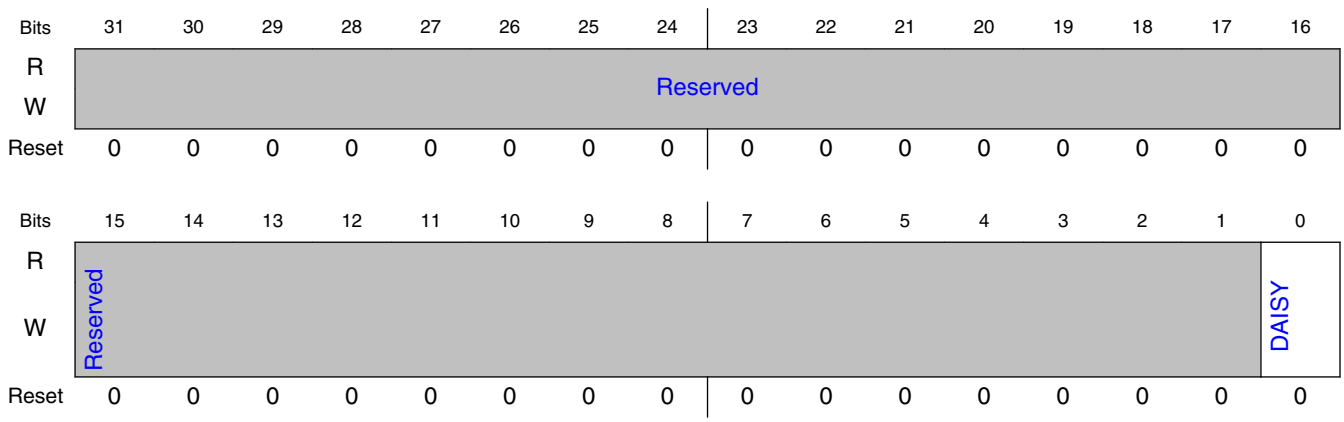
12.4.6.311.1 Offset

Register	Offset
ENET_1G_MAC0_RXERR_SELECT_INPUT	4E4h

12.4.6.311.2 Function

DAISY Register

12.4.6.311.3 Diagram



12.4.6.311.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_rxerr 0b - Selecting Pad: GPIO_EMC_B2_18 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B1_01 for Mode: ALT2

12.4.6.312 ENET_1G_MAC0_TXCLK_SELECT_INPUT DAISY Register (ENET_1G_MAC0_TXCLK_SELECT_INPUT)

12.4.6.312.1 Offset

Register	Offset
ENET_1G_MAC0_TXCLK_SELECT_INPUT	4E8h

12.4.6.312.2 Function

DAISY Register

12.4.6.312.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.312.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_1G, In Pin: mac0_txclk 00b - Selecting Pad: GPIO_EMC_B2_14 for Mode: ALT2 01b - Selecting Pad: GPIO_SD_B2_11 for Mode: ALT2 10b - Selecting Pad: GPIO_DISP_B1_11 for Mode: ALT1

12.4.6.313 ENET_QOS_GMII_MDI_I_SELECT_INPUT DAISY Register (ENET_QOS_GMII_MDI_I_SELECT_INPUT)

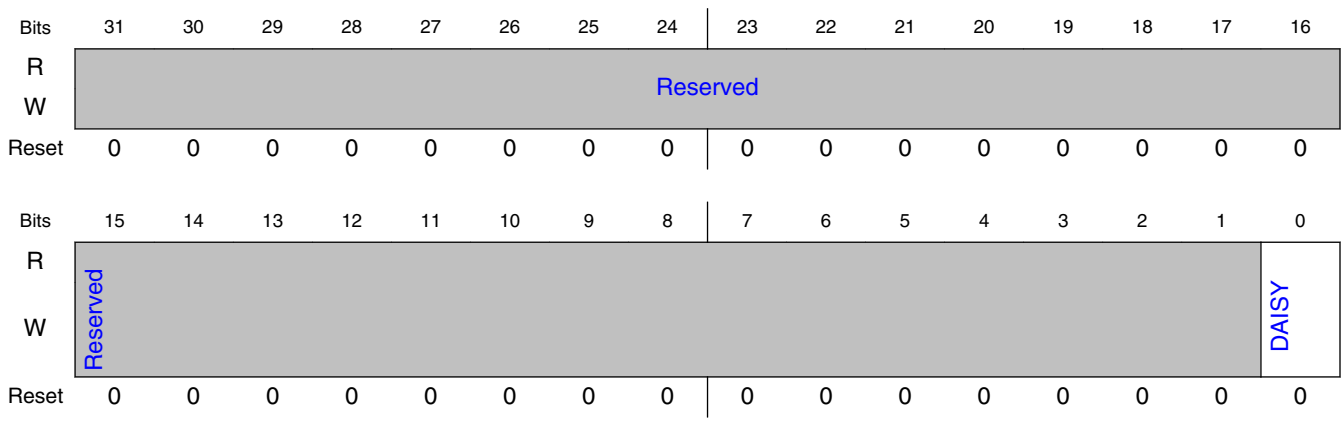
12.4.6.313.1 Offset

Register	Offset
ENET_QOS_GMII_MDI_I_SELECT_INPUT	4ECh

12.4.6.313.2 Function

DAISY Register

12.4.6.313.3 Diagram



12.4.6.313.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: gmii_mdi_i 0b - Selecting Pad: GPIO_EMC_B2_20 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_27 for Mode: ALT9

12.4.6.314 ENET_QOS_PHY_RXD_I_SELECT_INPUT_0 DAISY Register (ENET_QOS_PHY_RXD_I_SELECT_INPUT_0)

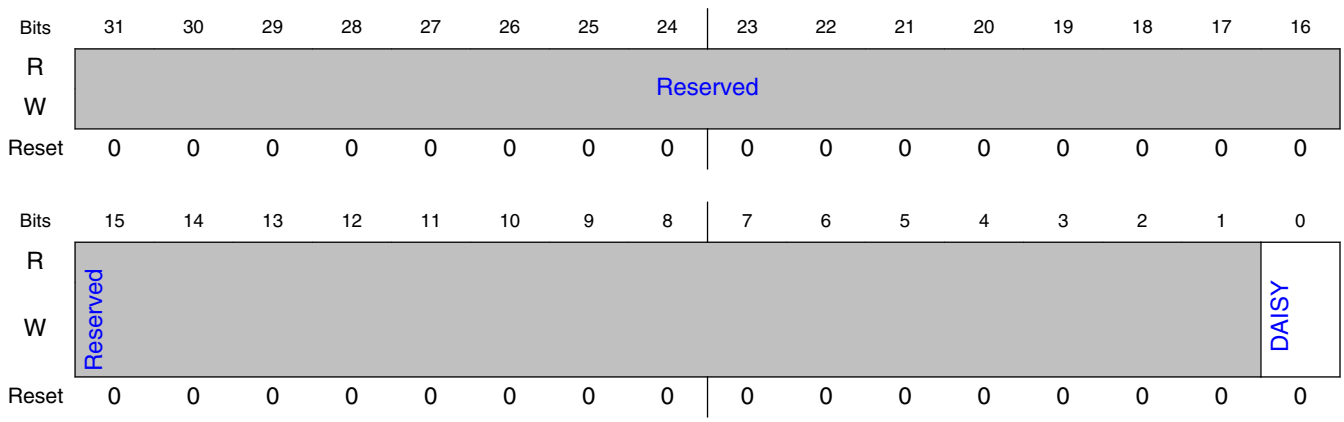
12.4.6.314.1 Offset

Register	Offset
ENET_QOS_PHY_RXD_I_SELECT_INPUT_0	4F0h

12.4.6.314.2 Function

DAISY Register

12.4.6.314.3 Diagram



12.4.6.314.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: phy_rxd_i0 0b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B2_06 for Mode: ALT8

12.4.6.315 ENET_QOS_PHY_RXD_I_SELECT_INPUT_1 DAISY Register (ENET_QOS_PHY_RXD_I_SELECT_INPUT_1)

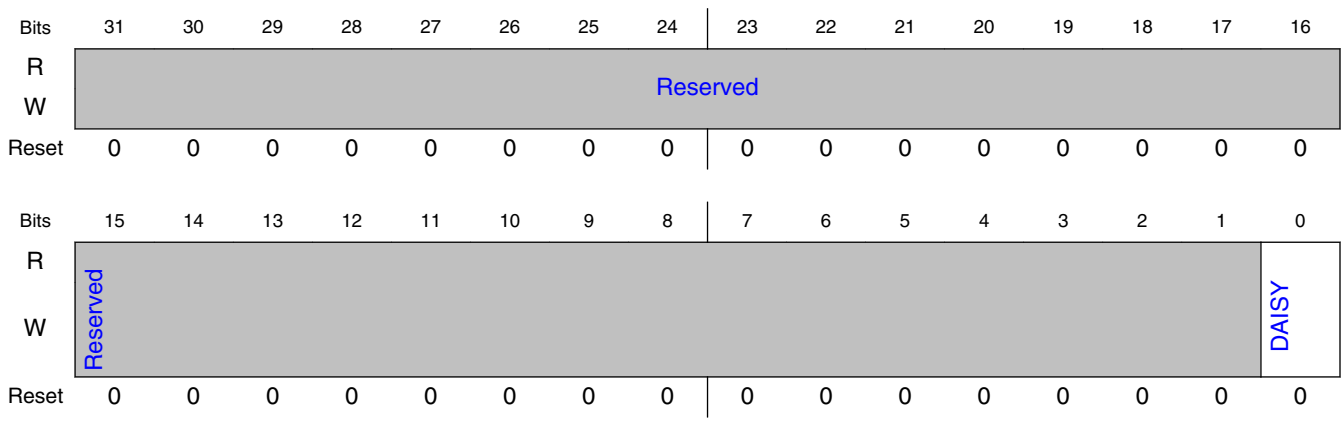
12.4.6.315.1 Offset

Register	Offset
ENET_QOS_PHY_RXD_I_SELECT_INPUT_1	4F4h

12.4.6.315.2 Function

DAISY Register

12.4.6.315.3 Diagram



12.4.6.315.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: phy_rxd_i1 0b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B2_07 for Mode: ALT8

12.4.6.316 ENET_QOS_PHY_RXDV_I_SELECT_INPUT DAISY Register (ENET_QOS_PHY_RXDV_I_SELECT_INPUT)

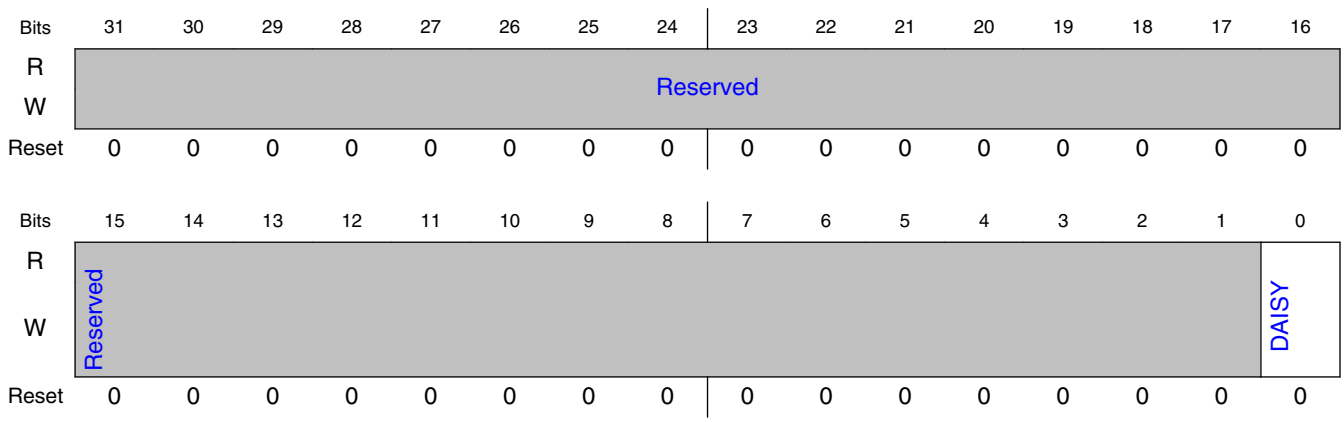
12.4.6.316.1 Offset

Register	Offset
ENET_QOS_PHY_RXDV_I_SELECT_INPUT	4F8h

12.4.6.316.2 Function

DAISY Register

12.4.6.316.3 Diagram



12.4.6.316.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: phy_rxdv_i 0b - Selecting Pad: GPIO_DISP_B1_00 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B2_08 for Mode: ALT8

12.4.6.317 ENET_QOS_PHY_RXER_I_SELECT_INPUT DAISY Register (ENET_QOS_PHY_RXER_I_SELECT_INPUT)

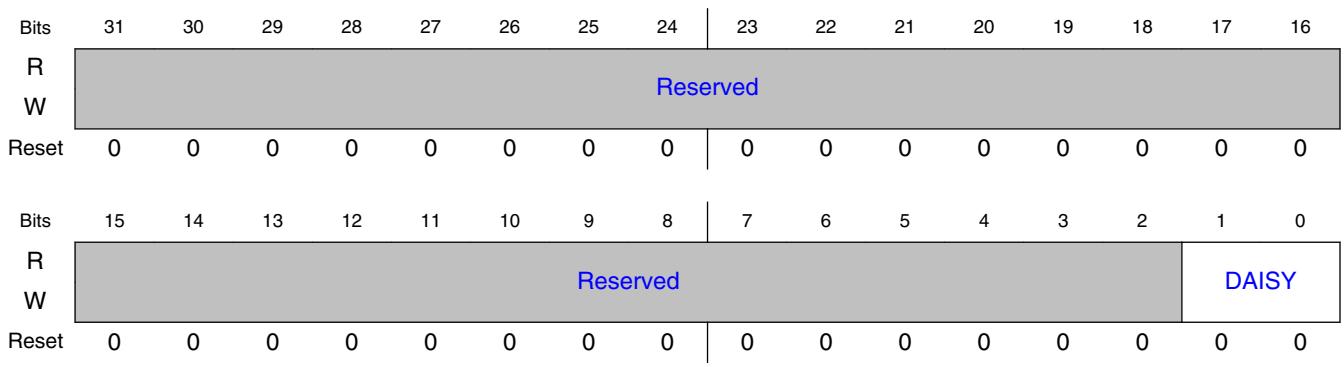
12.4.6.317.1 Offset

Register	Offset
ENET_QOS_PHY_RXER_I_SELECT_INPUT	4FCh

12.4.6.317.2 Function

DAISY Register

12.4.6.317.3 Diagram



12.4.6.317.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. instance: ENET_QOS, In Pin: phy_rxer_i 00b - Selecting Pad: GPIO_DISP_B1_01 for Mode: ALT9 01b - Selecting Pad: GPIO_DISP_B2_09 for Mode: ALT8 10b - Selecting Pad: GPIO_DISP_B2_10 for Mode: ALT8

12.4.6.318 FLEXPWM1_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_0)

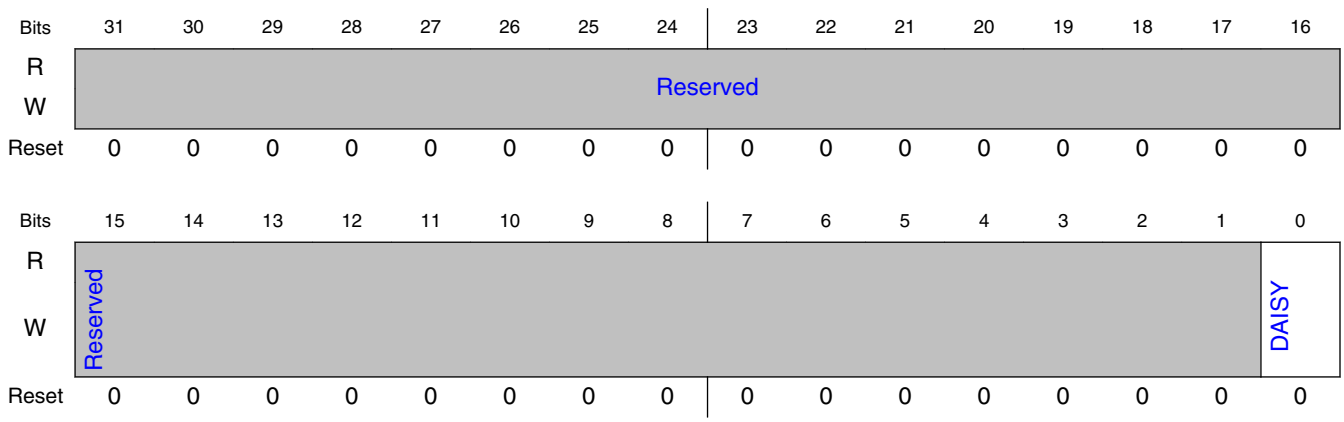
12.4.6.318.1 Offset

Register	Offset
FLEXPWM1_PWMA_SELECT_INPUT_0	500h

12.4.6.318.2 Function

DAISY Register

12.4.6.318.3 Diagram



12.4.6.318.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwma0 0b - Selecting Pad: GPIO_EMC_B1_23 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_00 for Mode: ALT4

12.4.6.319 FLEXPWM1_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_1)

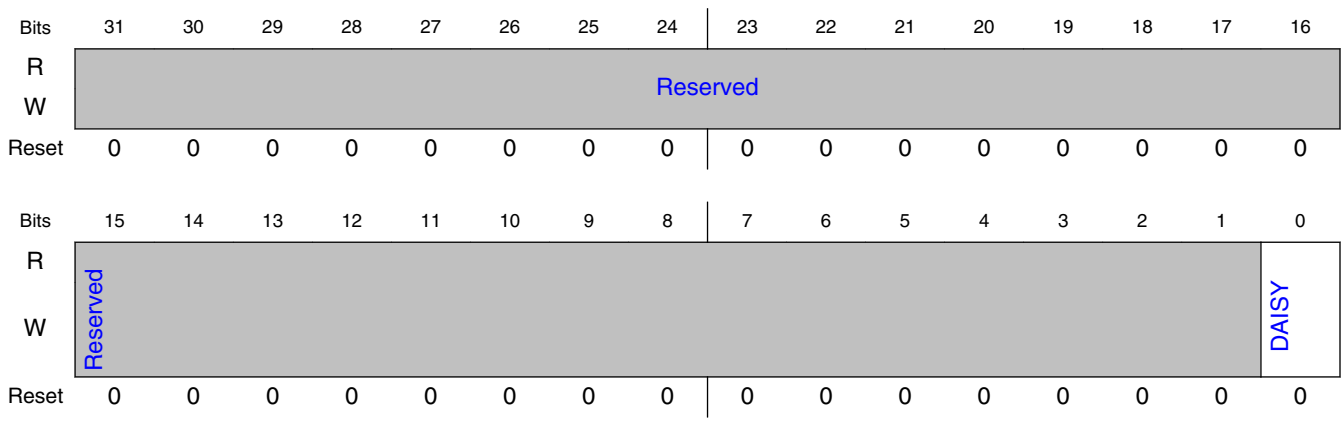
12.4.6.319.1 Offset

Register	Offset
FLEXPWM1_PWMA_SELECT_INPUT_1	504h

12.4.6.319.2 Function

DAISY Register

12.4.6.319.3 Diagram



12.4.6.319.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwma1 0b - Selecting Pad: GPIO_EMC_B1_25 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_02 for Mode: ALT4

12.4.6.320 FLEXPWM1_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM1_PWMA_SELECT_INPUT_2)

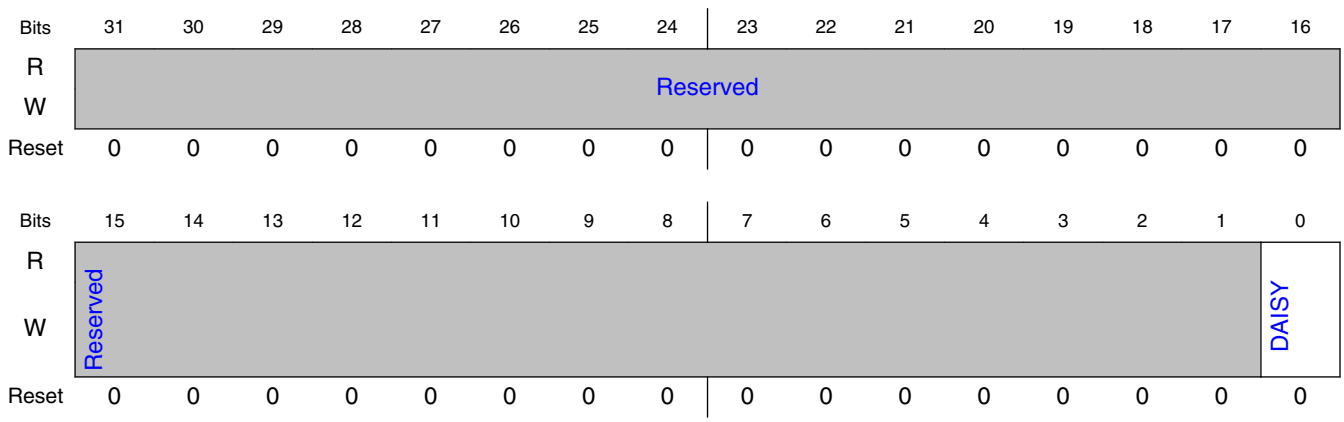
12.4.6.320.1 Offset

Register	Offset
FLEXPWM1_PWMA_SELECT_INPUT_2	508h

12.4.6.320.2 Function

DAISY Register

12.4.6.320.3 Diagram



12.4.6.320.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwma2 0b - Selecting Pad: GPIO_EMC_B1_27 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_04 for Mode: ALT4

12.4.6.321 FLEXPWM1_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_0)

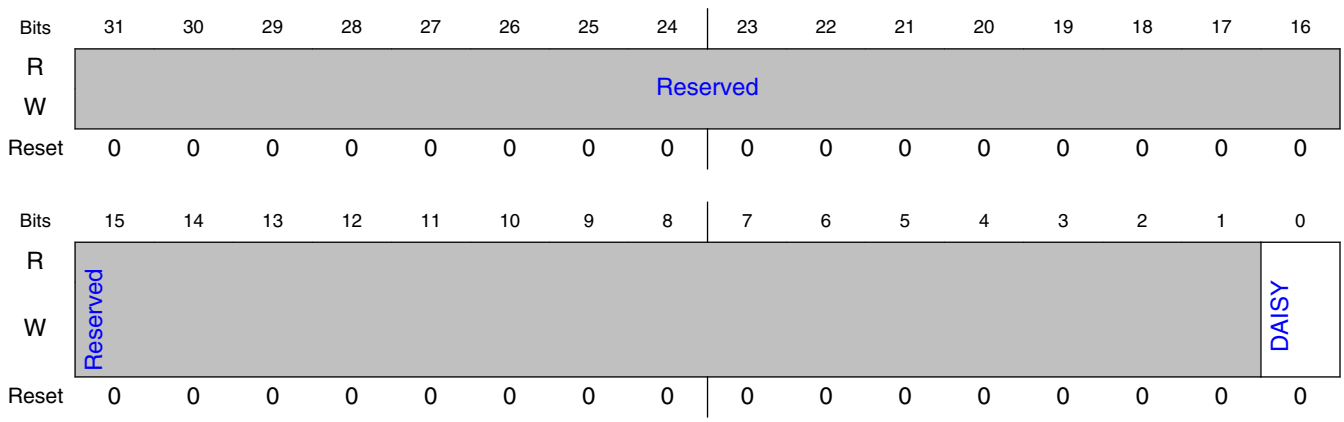
12.4.6.321.1 Offset

Register	Offset
FLEXPWM1_PWMB_SELECT_INPUT_0	50Ch

12.4.6.321.2 Function

DAISY Register

12.4.6.321.3 Diagram



12.4.6.321.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwmb0 0b - Selecting Pad: GPIO_EMC_B1_24 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_01 for Mode: ALT4

12.4.6.322 FLEXPWM1_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_1)

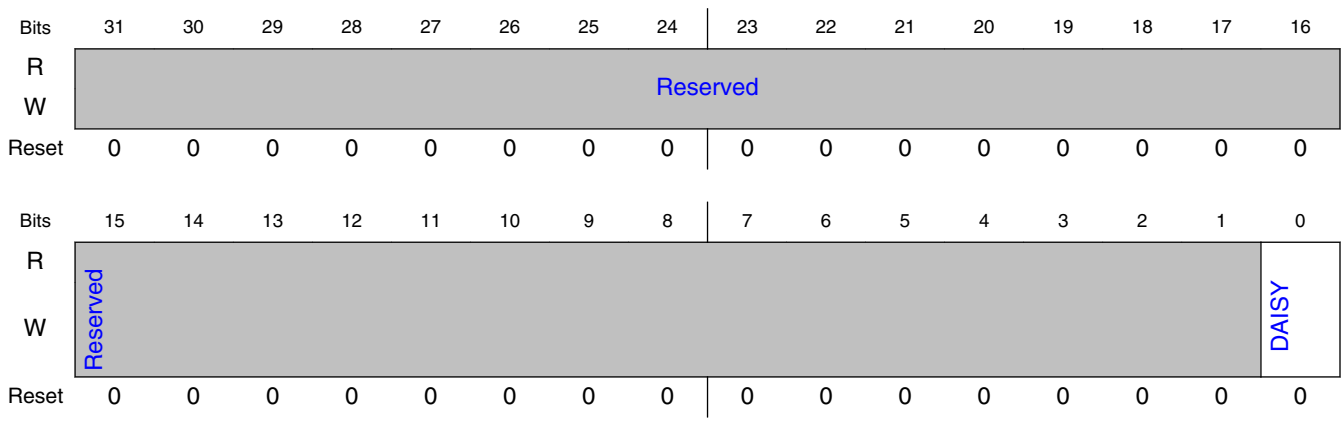
12.4.6.322.1 Offset

Register	Offset
FLEXPWM1_PWMB_SELECT_INPUT_1	510h

12.4.6.322.2 Function

DAISY Register

12.4.6.322.3 Diagram



12.4.6.322.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwmb1 0b - Selecting Pad: GPIO_EMC_B1_26 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_03 for Mode: ALT4

12.4.6.323 FLEXPWM1_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM1_PWMB_SELECT_INPUT_2)

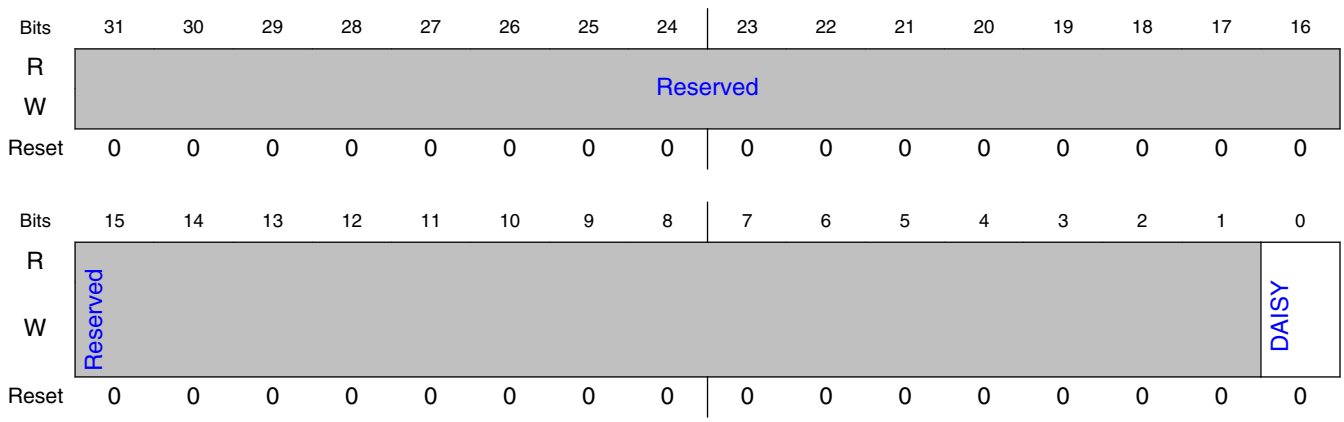
12.4.6.323.1 Offset

Register	Offset
FLEXPWM1_PWMB_SELECT_INPUT_2	514h

12.4.6.323.2 Function

DAISY Register

12.4.6.323.3 Diagram



12.4.6.323.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm1, In Pin: pwmb2 0b - Selecting Pad: GPIO_EMC_B1_28 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_05 for Mode: ALT4

12.4.6.324 FLEXPWM2_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_0)

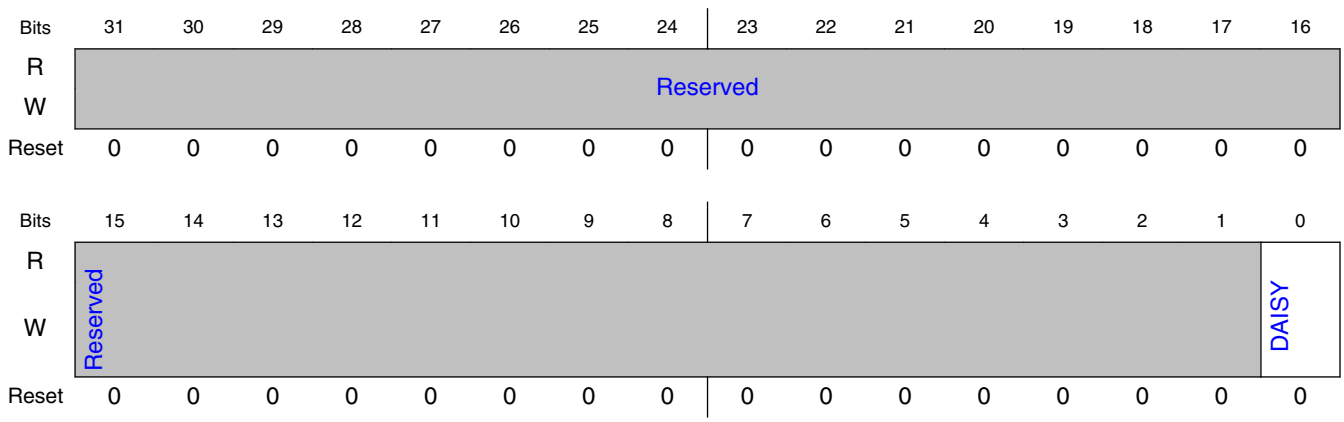
12.4.6.324.1 Offset

Register	Offset
FLEXPWM2_PWMA_SELECT_INPUT_0	518h

12.4.6.324.2 Function

DAISY Register

12.4.6.324.3 Diagram



12.4.6.324.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwma0 0b - Selecting Pad: GPIO_EMC_B1_06 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_24 for Mode: ALT4

12.4.6.325 FLEXPWM2_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_1)

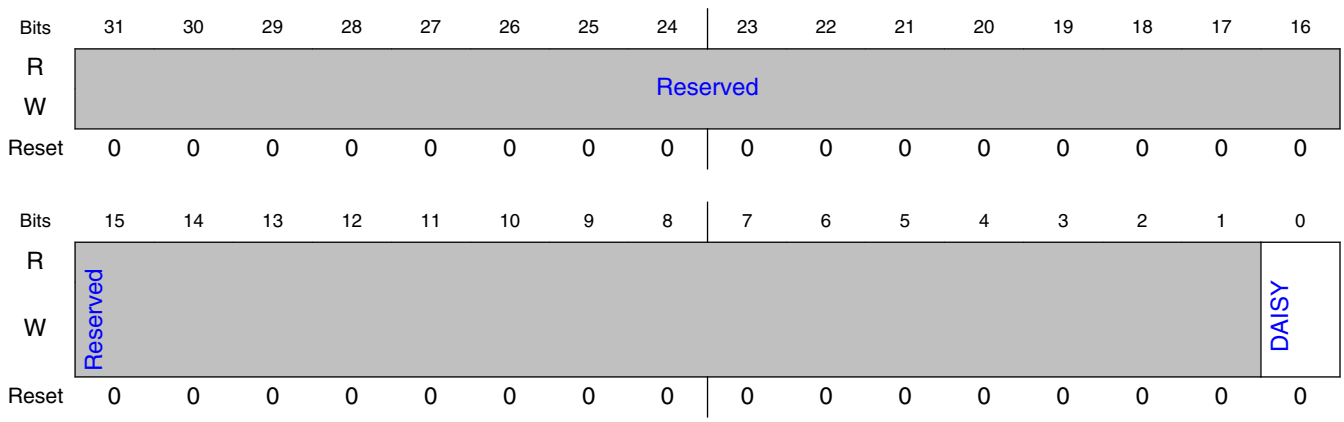
12.4.6.325.1 Offset

Register	Offset
FLEXPWM2_PWMA_SELECT_INPUT_1	51Ch

12.4.6.325.2 Function

DAISY Register

12.4.6.325.3 Diagram



12.4.6.325.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwma1 0b - Selecting Pad: GPIO_EMC_B1_08 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_26 for Mode: ALT4

12.4.6.326 FLEXPWM2_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM2_PWMA_SELECT_INPUT_2)

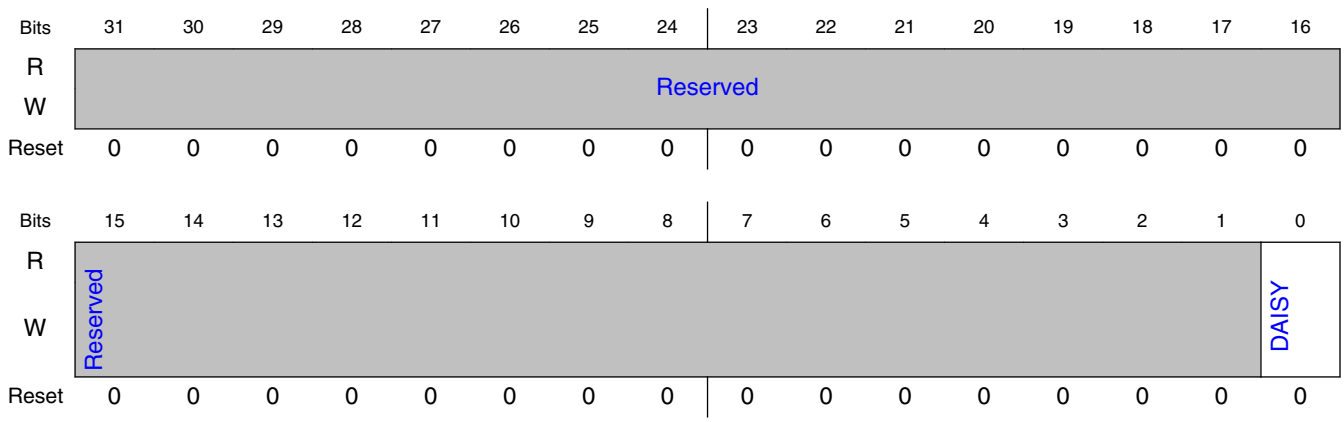
12.4.6.326.1 Offset

Register	Offset
FLEXPWM2_PWMA_SELECT_INPUT_2	520h

12.4.6.326.2 Function

DAISY Register

12.4.6.326.3 Diagram



12.4.6.326.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwma2 0b - Selecting Pad: GPIO_EMC_B1_10 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_28 for Mode: ALT4

12.4.6.327 FLEXPWM2_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_0)

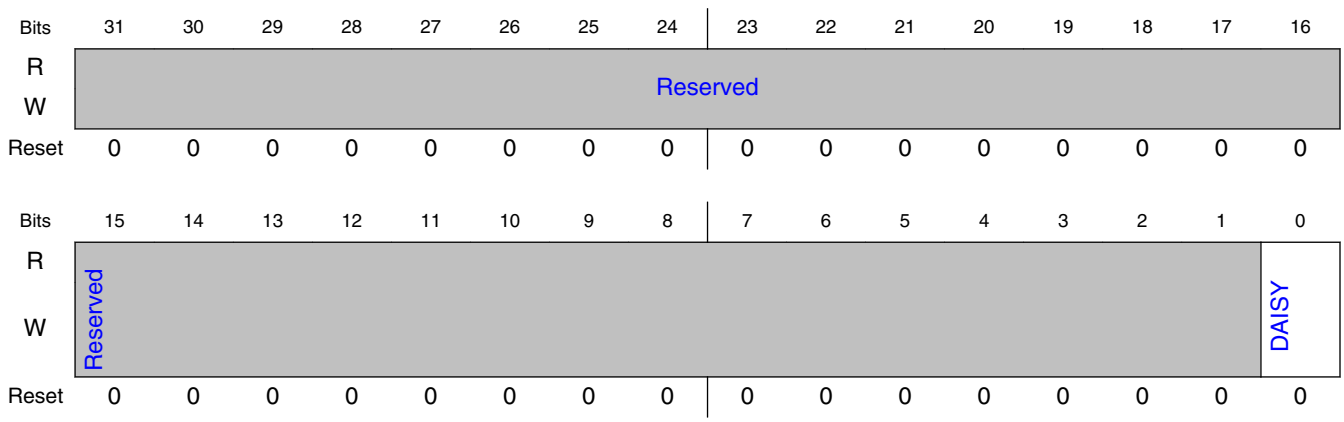
12.4.6.327.1 Offset

Register	Offset
FLEXPWM2_PWMB_SELECT_INPUT_0	524h

12.4.6.327.2 Function

DAISY Register

12.4.6.327.3 Diagram



12.4.6.327.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwmb0 0b - Selecting Pad: GPIO_EMC_B1_07 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_25 for Mode: ALT4

12.4.6.328 FLEXPWM2_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_1)

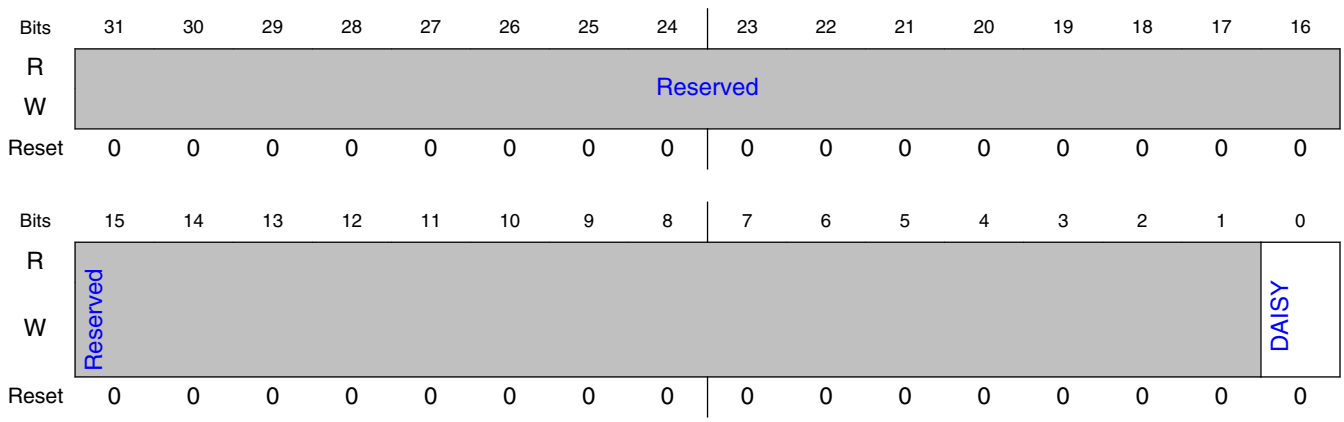
12.4.6.328.1 Offset

Register	Offset
FLEXPWM2_PWMB_SELECT_INPUT_1	528h

12.4.6.328.2 Function

DAISY Register

12.4.6.328.3 Diagram



12.4.6.328.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwmb1 0b - Selecting Pad: GPIO_EMC_B1_09 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_27 for Mode: ALT4

12.4.6.329 FLEXPWM2_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM2_PWMB_SELECT_INPUT_2)

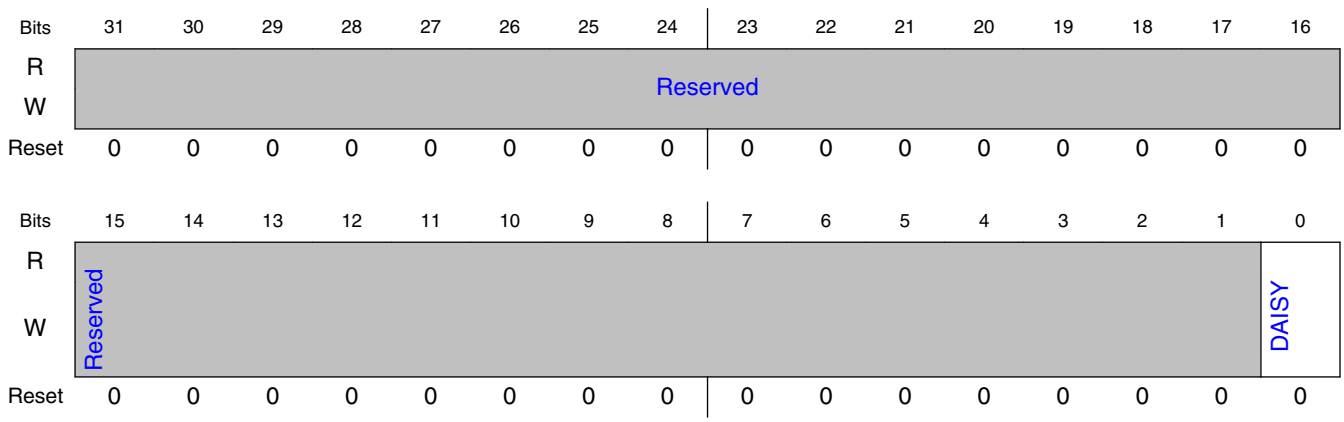
12.4.6.329.1 Offset

Register	Offset
FLEXPWM2_PWMB_SELECT_INPUT_2	52Ch

12.4.6.329.2 Function

DAISY Register

12.4.6.329.3 Diagram



12.4.6.329.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm2, In Pin: pwmb2 0b - Selecting Pad: GPIO_EMC_B1_11 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_29 for Mode: ALT4

12.4.6.330 FLEXPWM3_PWMA_SELECT_INPUT_0 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_0)

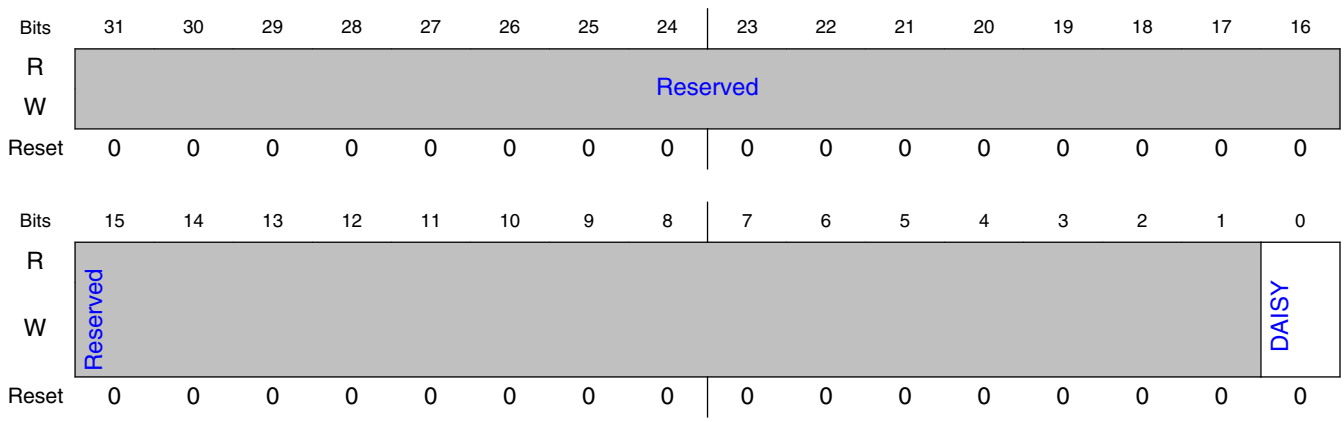
12.4.6.330.1 Offset

Register	Offset
FLEXPWM3_PWMA_SELECT_INPUT_0	530h

12.4.6.330.2 Function

DAISY Register

12.4.6.330.3 Diagram



12.4.6.330.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwma0 0b - Selecting Pad: GPIO_EMC_B1_29 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_00 for Mode: ALT11

12.4.6.331 FLEXPWM3_PWMA_SELECT_INPUT_1 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_1)

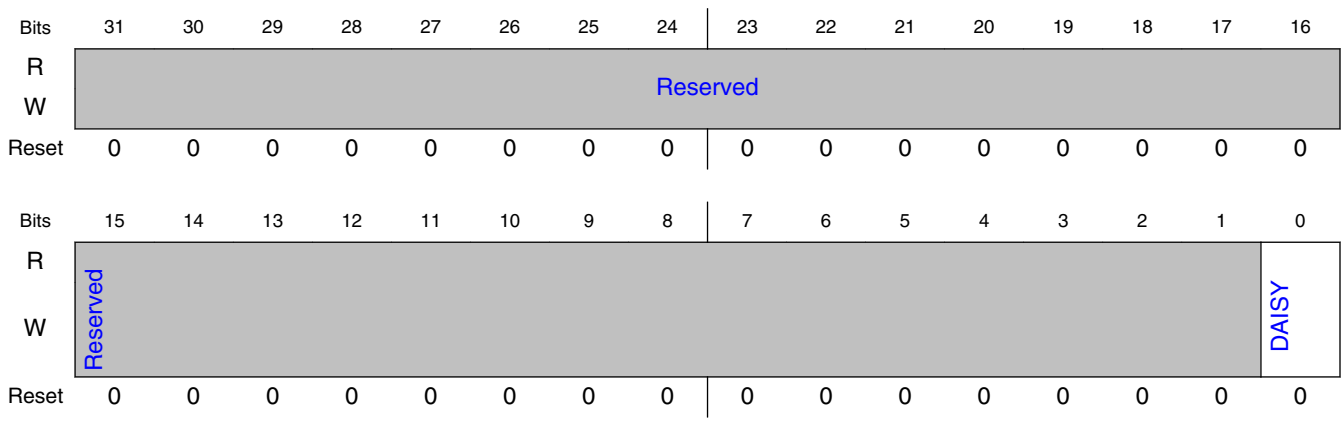
12.4.6.331.1 Offset

Register	Offset
FLEXPWM3_PWMA_SELECT_INPUT_1	534h

12.4.6.331.2 Function

DAISY Register

12.4.6.331.3 Diagram



12.4.6.331.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwma1 0b - Selecting Pad: GPIO_EMC_B1_31 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_02 for Mode: ALT11

12.4.6.332 FLEXPWM3_PWMA_SELECT_INPUT_2 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_2)

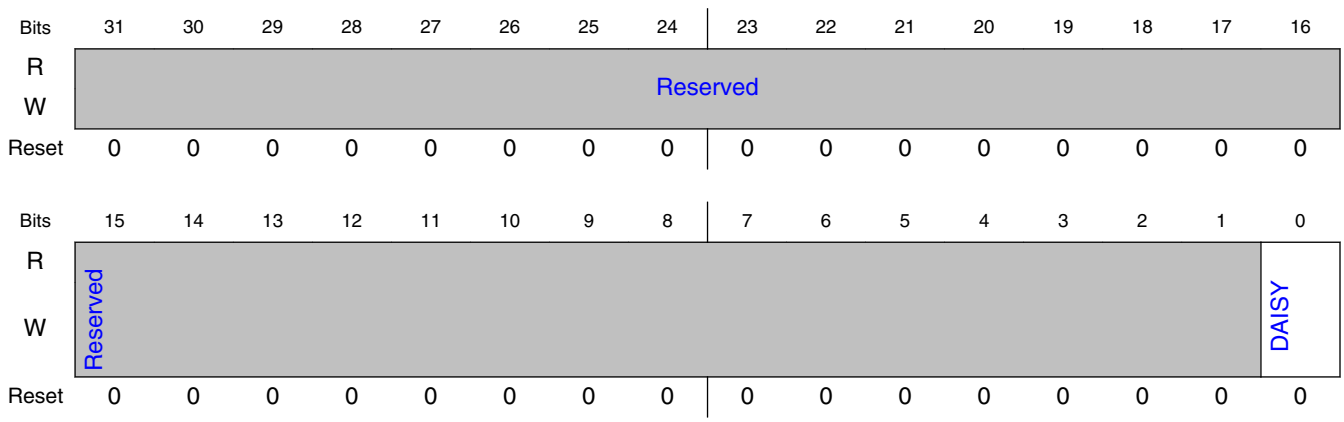
12.4.6.332.1 Offset

Register	Offset
FLEXPWM3_PWMA_SELECT_INPUT_2	538h

12.4.6.332.2 Function

DAISY Register

12.4.6.332.3 Diagram



12.4.6.332.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwma2 0b - Selecting Pad: GPIO_EMC_B1_33 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_04 for Mode: ALT11

12.4.6.333 FLEXPWM3_PWMA_SELECT_INPUT_3 DAISY Register (FLEXPWM3_PWMA_SELECT_INPUT_3)

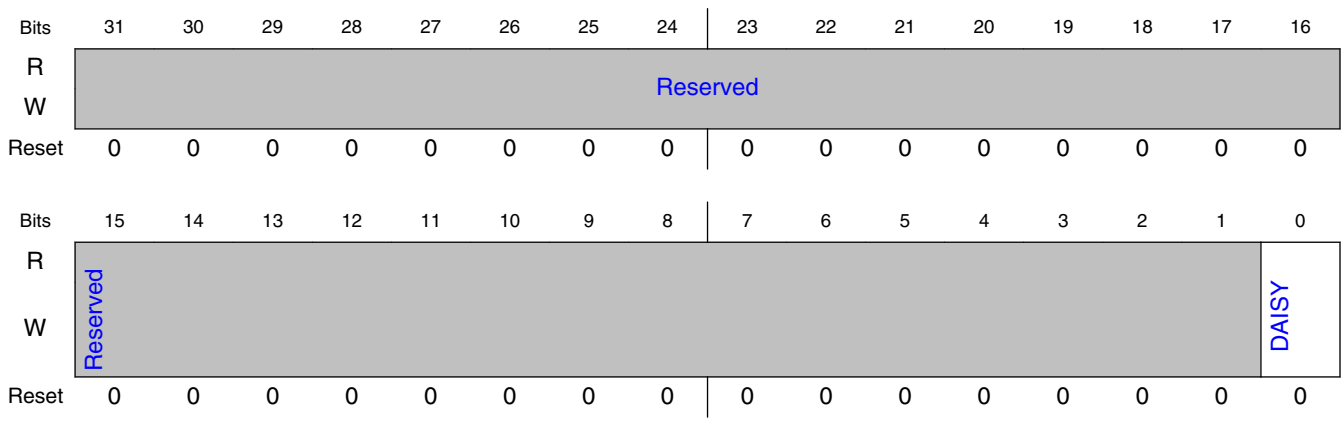
12.4.6.333.1 Offset

Register	Offset
FLEXPWM3_PWMA_SELECT_INPUT_3	53Ch

12.4.6.333.2 Function

DAISY Register

12.4.6.333.3 Diagram



12.4.6.333.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwma3 0b - Selecting Pad: GPIO_EMC_B1_21 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_06 for Mode: ALT11

12.4.6.334 FLEXPWM3_PWMB_SELECT_INPUT_0 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_0)

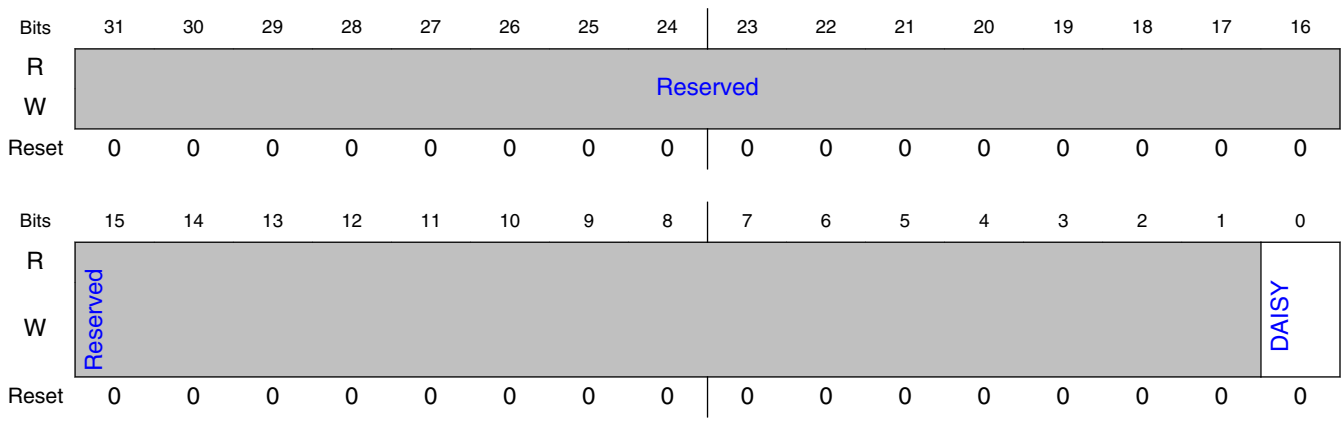
12.4.6.334.1 Offset

Register	Offset
FLEXPWM3_PWMB_SELECT_INPUT_0	540h

12.4.6.334.2 Function

DAISY Register

12.4.6.334.3 Diagram



12.4.6.334.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwmb0 0b - Selecting Pad: GPIO_EMC_B1_30 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT11

12.4.6.335 FLEXPWM3_PWMB_SELECT_INPUT_1 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_1)

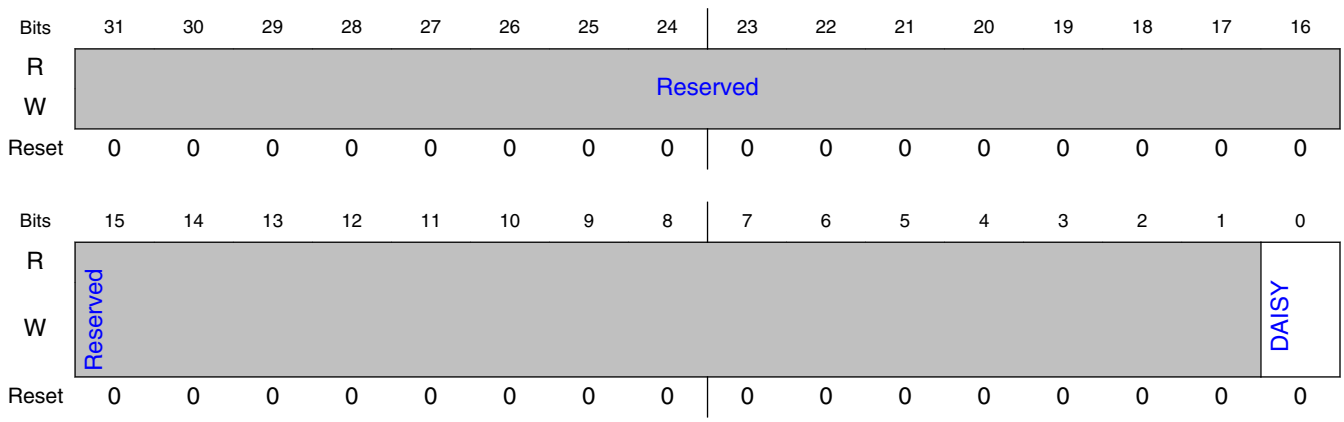
12.4.6.335.1 Offset

Register	Offset
FLEXPWM3_PWMB_SELECT_INPUT_1	544h

12.4.6.335.2 Function

DAISY Register

12.4.6.335.3 Diagram



12.4.6.335.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwmb1 0b - Selecting Pad: GPIO_EMC_B1_32 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_03 for Mode: ALT11

12.4.6.336 FLEXPWM3_PWMB_SELECT_INPUT_2 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_2)

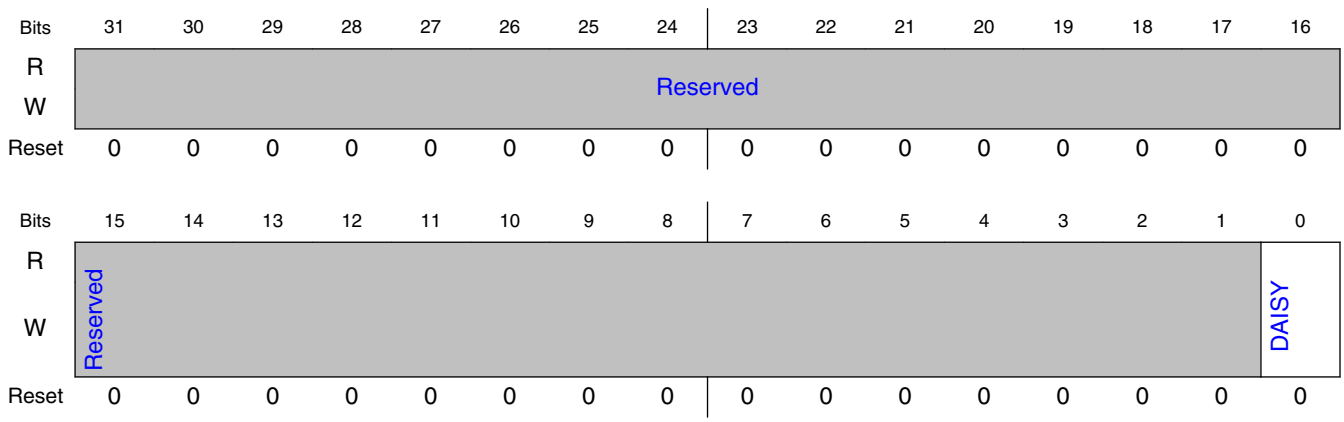
12.4.6.336.1 Offset

Register	Offset
FLEXPWM3_PWMB_SELECT_INPUT_2	548h

12.4.6.336.2 Function

DAISY Register

12.4.6.336.3 Diagram



12.4.6.336.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwmb2 0b - Selecting Pad: GPIO_EMC_B1_34 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_05 for Mode: ALT11

12.4.6.337 FLEXPWM3_PWMB_SELECT_INPUT_3 DAISY Register (FLEXPWM3_PWMB_SELECT_INPUT_3)

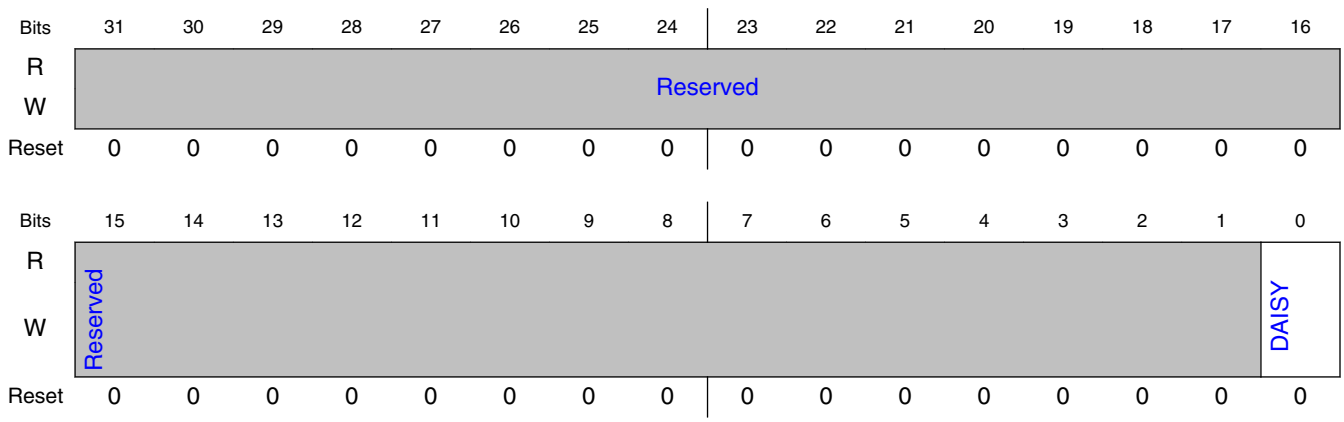
12.4.6.337.1 Offset

Register	Offset
FLEXPWM3_PWMB_SELECT_INPUT_3	54Ch

12.4.6.337.2 Function

DAISY Register

12.4.6.337.3 Diagram



12.4.6.337.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexpwm3, In Pin: pwmb3 0b - Selecting Pad: GPIO_EMC_B1_22 for Mode: ALT1 1b - Selecting Pad: GPIO_EMC_B2_07 for Mode: ALT11

12.4.6.338 FLEXSPI1_I_DQS_FA_SELECT_INPUT DAISY Register (FLEXSPI1_I_DQS_FA_SELECT_INPUT)

12.4.6.338.1 Offset

Register	Offset
FLEXSPI1_I_DQS_FA_SELECT_INPUT	550h

12.4.6.338.2 Function

DAISY Register

12.4.6.338.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.338.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_dqs_fa 00b - Selecting Pad: GPIO_EMC_B2_18 for Mode: ALT6 01b - Selecting Pad: GPIO_AD_17 for Mode: ALT3 10b - Selecting Pad: GPIO_SD_B2_05 for Mode: ALT1

12.4.6.339 FLEXSPI1_I_IO_FA_SELECT_INPUT_0 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_0)

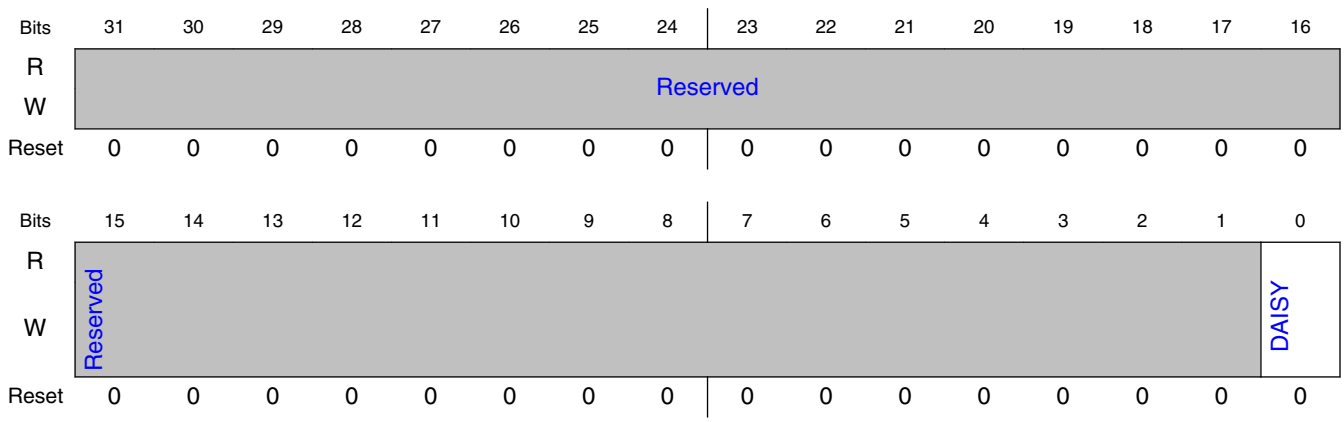
12.4.6.339.1 Offset

Register	Offset
FLEXSPI1_I_IO_FA_SELECT_INPUT_0	554h

12.4.6.339.2 Function

DAISY Register

12.4.6.339.3 Diagram



12.4.6.339.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fa0 0b - Selecting Pad: GPIO_AD_20 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_08 for Mode: ALT1

12.4.6.340 FLEXSPI1_I_IO_FA_SELECT_INPUT_1 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_1)

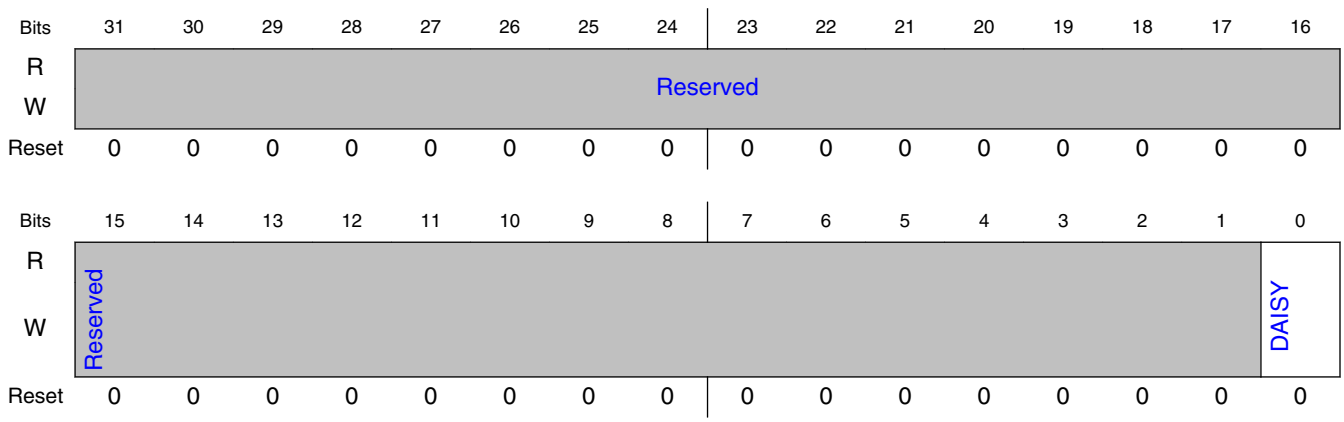
12.4.6.340.1 Offset

Register	Offset
FLEXSPI1_I_IO_FA_SELECT_INPUT_1	558h

12.4.6.340.2 Function

DAISY Register

12.4.6.340.3 Diagram



12.4.6.340.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fa1 0b - Selecting Pad: GPIO_AD_21 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_09 for Mode: ALT1

12.4.6.341 FLEXSPI1_I_IO_FA_SELECT_INPUT_2 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_2)

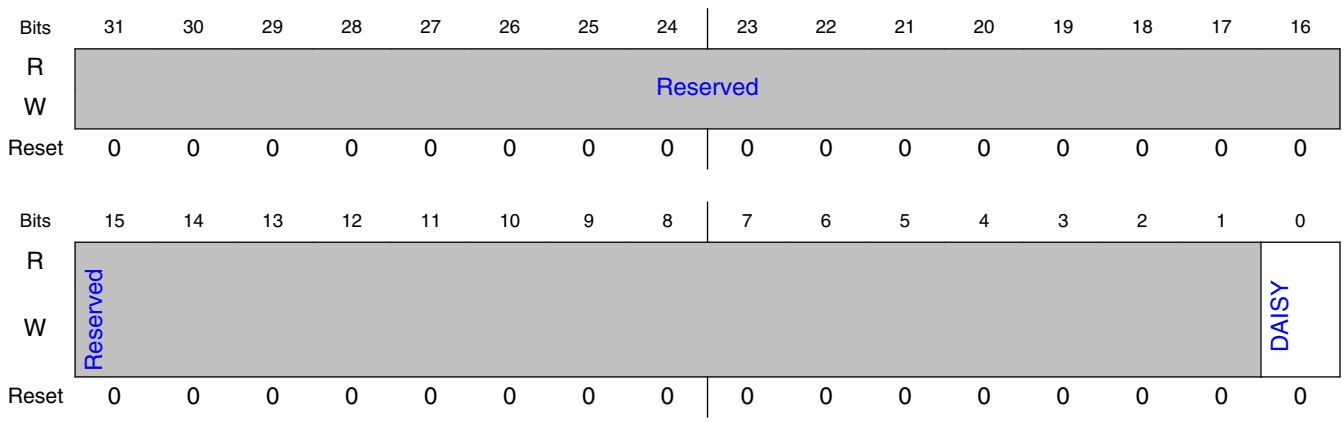
12.4.6.341.1 Offset

Register	Offset
FLEXSPI1_I_IO_FA_SELECT_INPUT_2	55Ch

12.4.6.341.2 Function

DAISY Register

12.4.6.341.3 Diagram



12.4.6.341.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fa2 0b - Selecting Pad: GPIO_AD_22 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_10 for Mode: ALT1

12.4.6.342 FLEXSPI1_I_IO_FA_SELECT_INPUT_3 DAISY Register (FLEXSPI1_I_IO_FA_SELECT_INPUT_3)

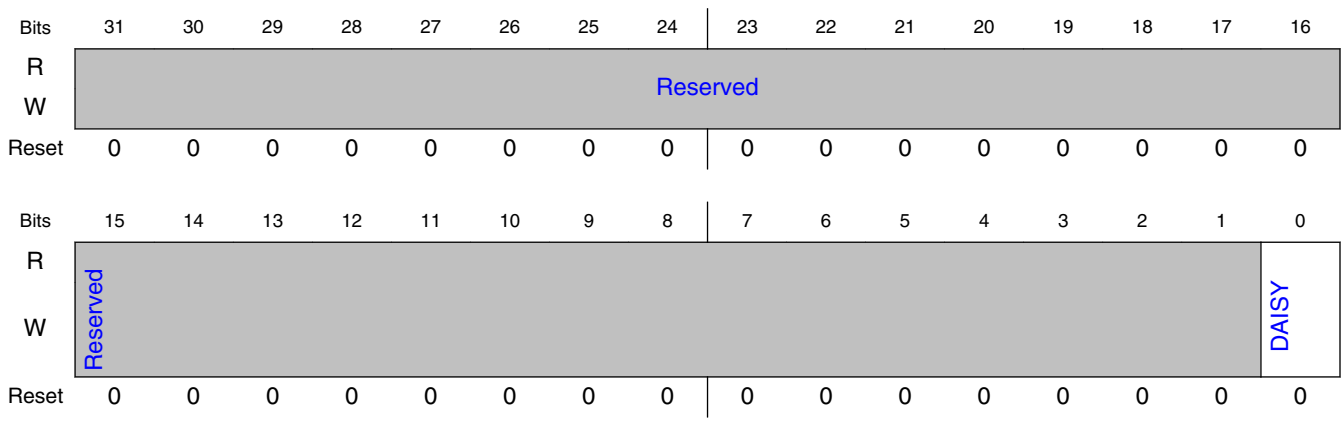
12.4.6.342.1 Offset

Register	Offset
FLEXSPI1_I_IO_FA_SELECT_INPUT_3	560h

12.4.6.342.2 Function

DAISY Register

12.4.6.342.3 Diagram



12.4.6.342.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fa3 0b - Selecting Pad: GPIO_AD_23 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_11 for Mode: ALT1

12.4.6.343 FLEXSPI1_I_IO_FB_SELECT_INPUT_0 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_0)

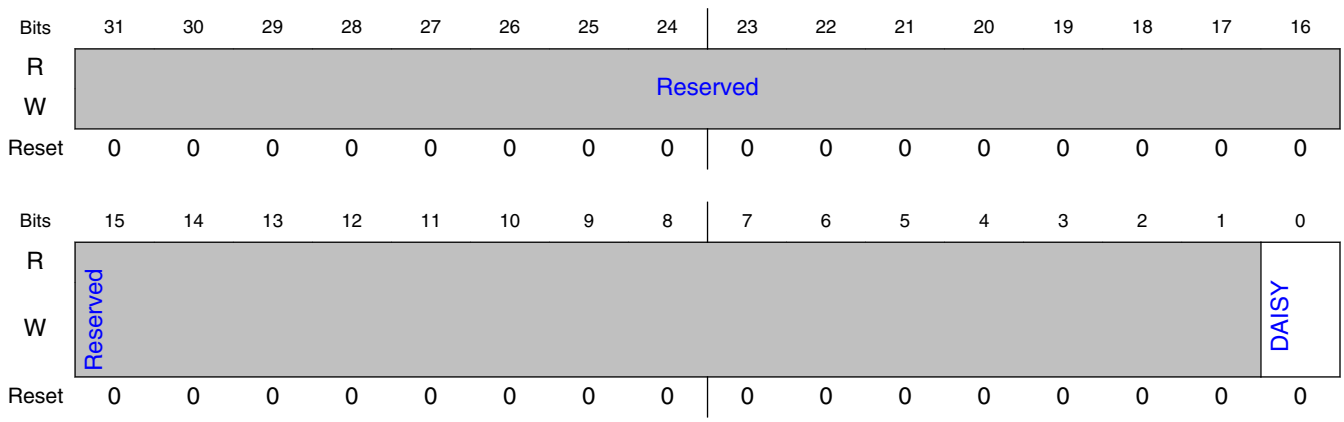
12.4.6.343.1 Offset

Register	Offset
FLEXSPI1_I_IO_FB_SELECT_INPUT_0	564h

12.4.6.343.2 Function

DAISY Register

12.4.6.343.3 Diagram



12.4.6.343.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fb0 0b - Selecting Pad: GPIO_AD_15 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_03 for Mode: ALT1

12.4.6.344 FLEXSPI1_I_IO_FB_SELECT_INPUT_1 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_1)

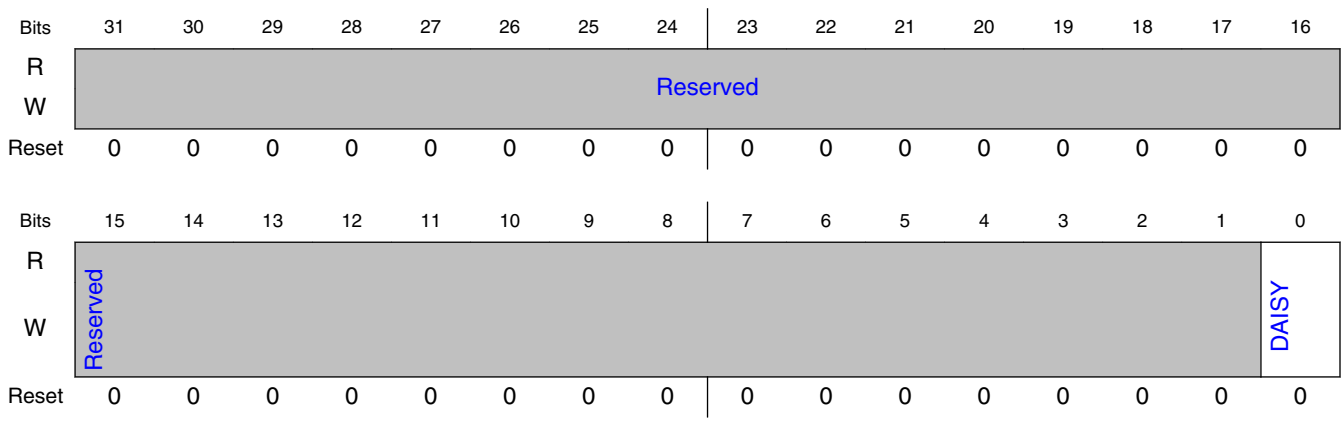
12.4.6.344.1 Offset

Register	Offset
FLEXSPI1_I_IO_FB_SELECT_INPUT_1	568h

12.4.6.344.2 Function

DAISY Register

12.4.6.344.3 Diagram



12.4.6.344.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fb1 0b - Selecting Pad: GPIO_AD_14 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_02 for Mode: ALT1

12.4.6.345 FLEXSPI1_I_IO_FB_SELECT_INPUT_2 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_2)

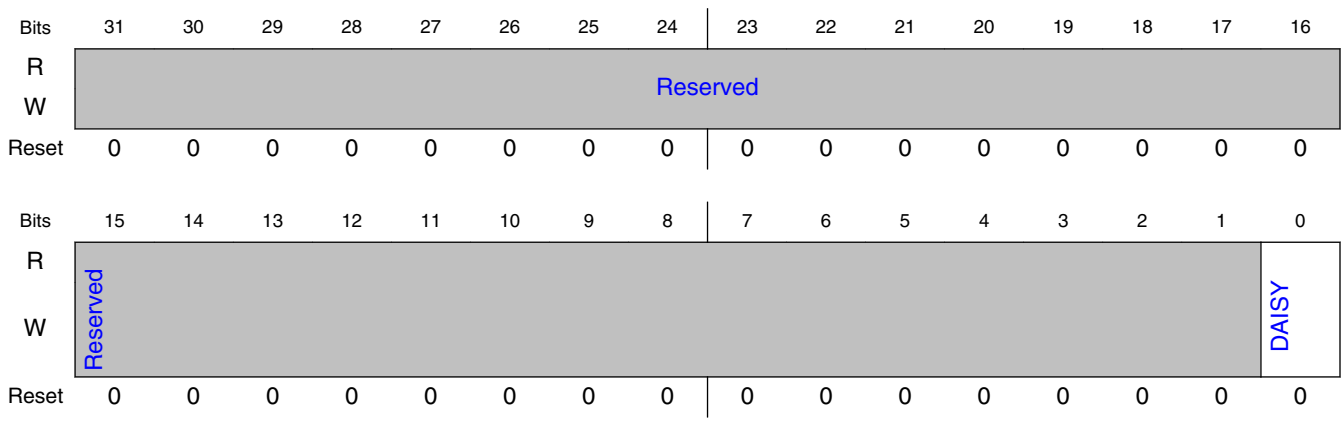
12.4.6.345.1 Offset

Register	Offset
FLEXSPI1_I_IO_FB_SELECT_INPUT_2	56Ch

12.4.6.345.2 Function

DAISY Register

12.4.6.345.3 Diagram



12.4.6.345.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fb2 0b - Selecting Pad: GPIO_AD_13 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_01 for Mode: ALT1

12.4.6.346 FLEXSPI1_I_IO_FB_SELECT_INPUT_3 DAISY Register (FLEXSPI1_I_IO_FB_SELECT_INPUT_3)

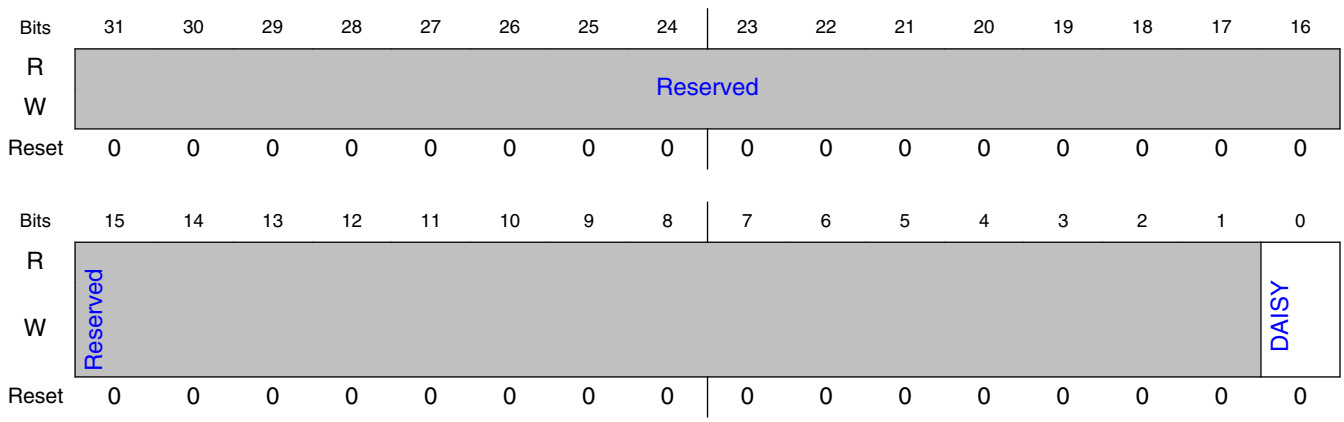
12.4.6.346.1 Offset

Register	Offset
FLEXSPI1_I_IO_FB_SELECT_INPUT_3	570h

12.4.6.346.2 Function

DAISY Register

12.4.6.346.3 Diagram



12.4.6.346.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_io_fb3 0b - Selecting Pad: GPIO_AD_12 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_00 for Mode: ALT1

12.4.6.347 FLEXSPI1_I_SCK_FA_SELECT_INPUT DAISY Register (FLEXSPI1_I_SCK_FA_SELECT_INPUT)

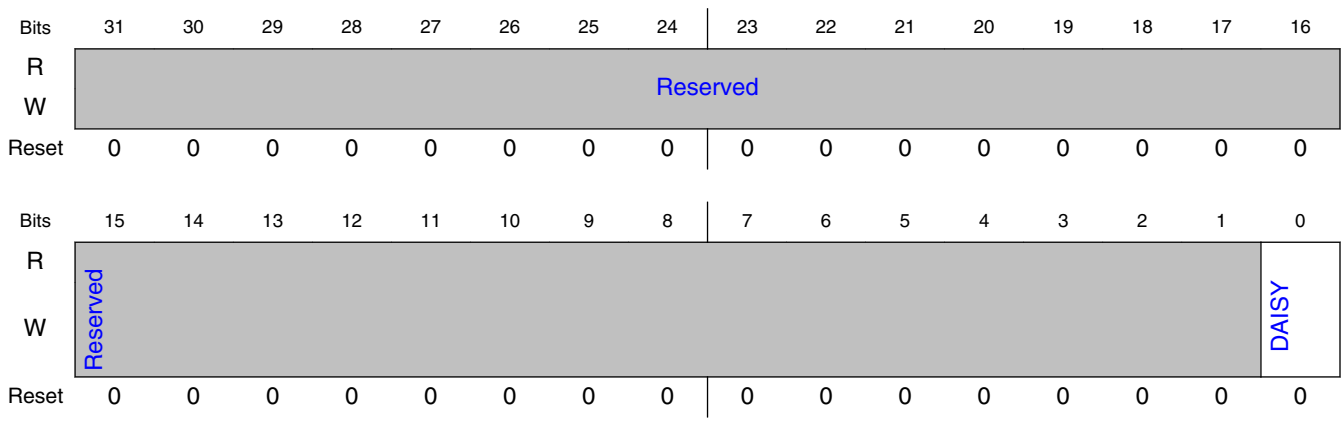
12.4.6.347.1 Offset

Register	Offset
FLEXSPI1_I_SCK_FA_SELECT_INPUT	574h

12.4.6.347.2 Function

DAISY Register

12.4.6.347.3 Diagram



12.4.6.347.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_sck_fa 0b - Selecting Pad: GPIO_AD_19 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_07 for Mode: ALT1

12.4.6.348 FLEXSPI1_I_SCK_FB_SELECT_INPUT DAISY Register (FLEXSPI1_I_SCK_FB_SELECT_INPUT)

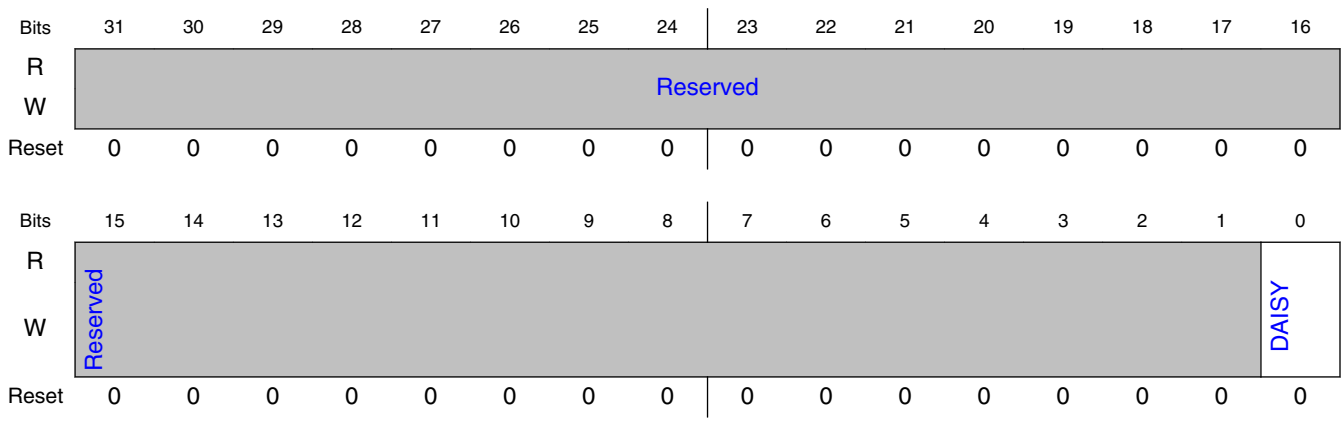
12.4.6.348.1 Offset

Register	Offset
FLEXSPI1_I_SCK_FB_SELECT_INPUT	578h

12.4.6.348.2 Function

DAISY Register

12.4.6.348.3 Diagram



12.4.6.348.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi1, In Pin: i_sck_fb 0b - Selecting Pad: GPIO_AD_16 for Mode: ALT3 1b - Selecting Pad: GPIO_SD_B2_04 for Mode: ALT1

12.4.6.349 FLEXSPI2_I_IO_FA_SELECT_INPUT_0 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_0)

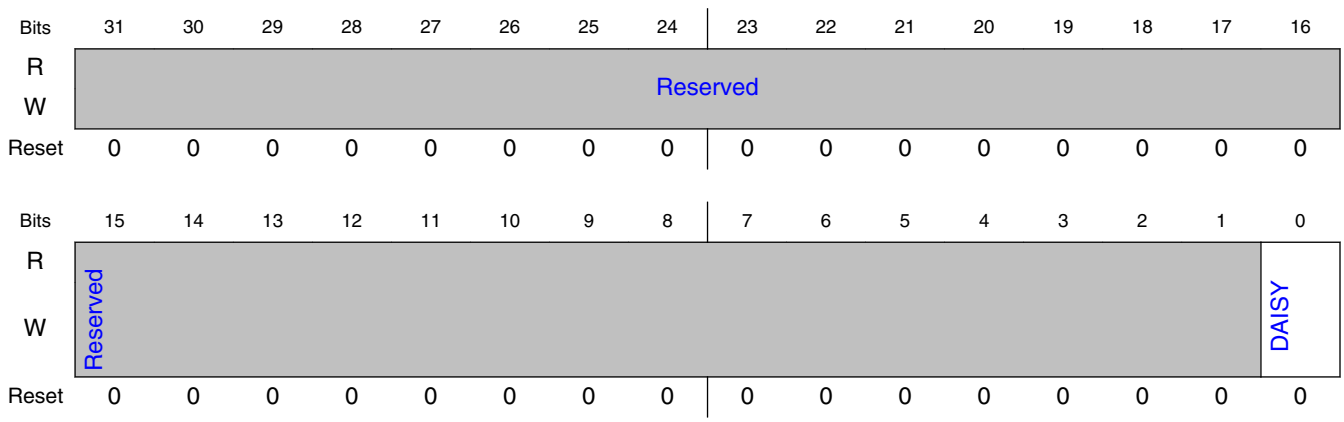
12.4.6.349.1 Offset

Register	Offset
FLEXSPI2_I_IO_FA_SELECT_INPUT_0	57Ch

12.4.6.349.2 Function

DAISY Register

12.4.6.349.3 Diagram



12.4.6.349.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi2, In Pin: i_io_fa0 0b - Selecting Pad: GPIO_EMC_B2_13 for Mode: ALT4 1b - Selecting Pad: GPIO_SD_B1_02 for Mode: ALT6

12.4.6.350 FLEXSPI2_I_IO_FA_SELECT_INPUT_1 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_1)

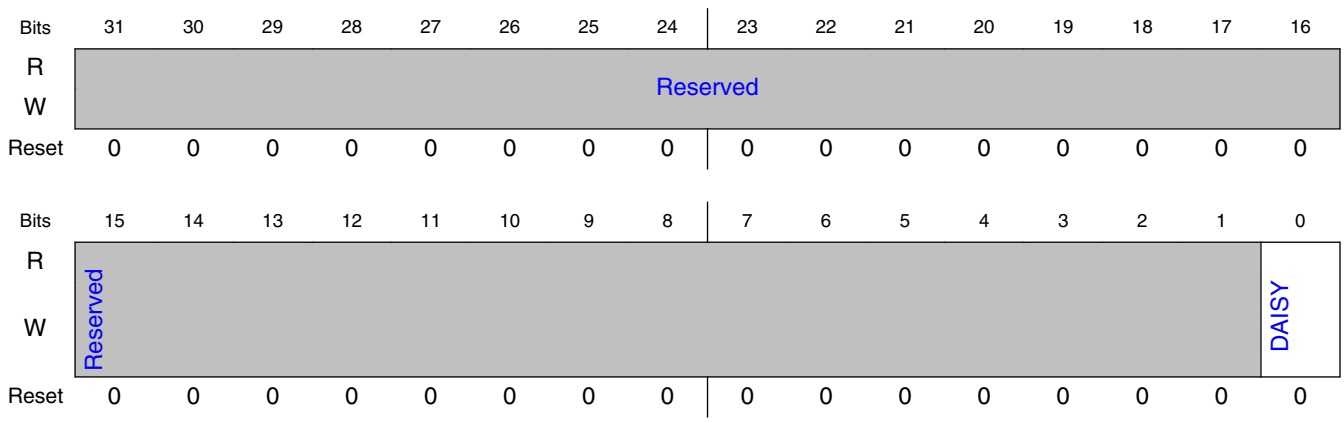
12.4.6.350.1 Offset

Register	Offset
FLEXSPI2_I_IO_FA_SELECT_INPUT_1	580h

12.4.6.350.2 Function

DAISY Register

12.4.6.350.3 Diagram



12.4.6.350.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi2, In Pin: i_io_fa1 0b - Selecting Pad: GPIO_EMC_B2_14 for Mode: ALT4 1b - Selecting Pad: GPIO_SD_B1_03 for Mode: ALT6

12.4.6.351 FLEXSPI2_I_IO_FA_SELECT_INPUT_2 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_2)

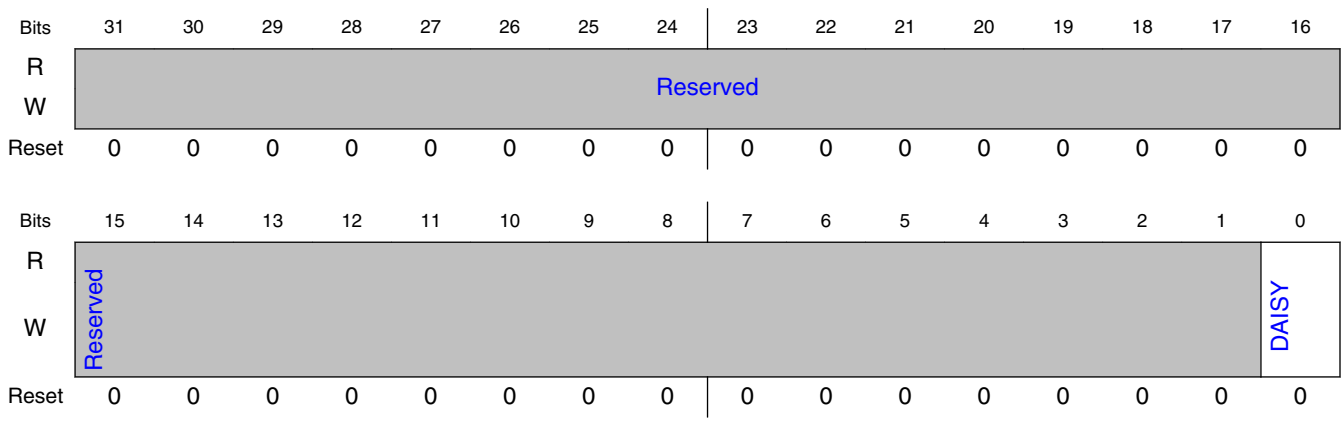
12.4.6.351.1 Offset

Register	Offset
FLEXSPI2_I_IO_FA_SELECT_INPUT_2	584h

12.4.6.351.2 Function

DAISY Register

12.4.6.351.3 Diagram



12.4.6.351.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi2, In Pin: i_io_fa2 0b - Selecting Pad: GPIO_EMC_B2_15 for Mode: ALT4 1b - Selecting Pad: GPIO_SD_B1_04 for Mode: ALT6

12.4.6.352 FLEXSPI2_I_IO_FA_SELECT_INPUT_3 DAISY Register (FLEXSPI2_I_IO_FA_SELECT_INPUT_3)

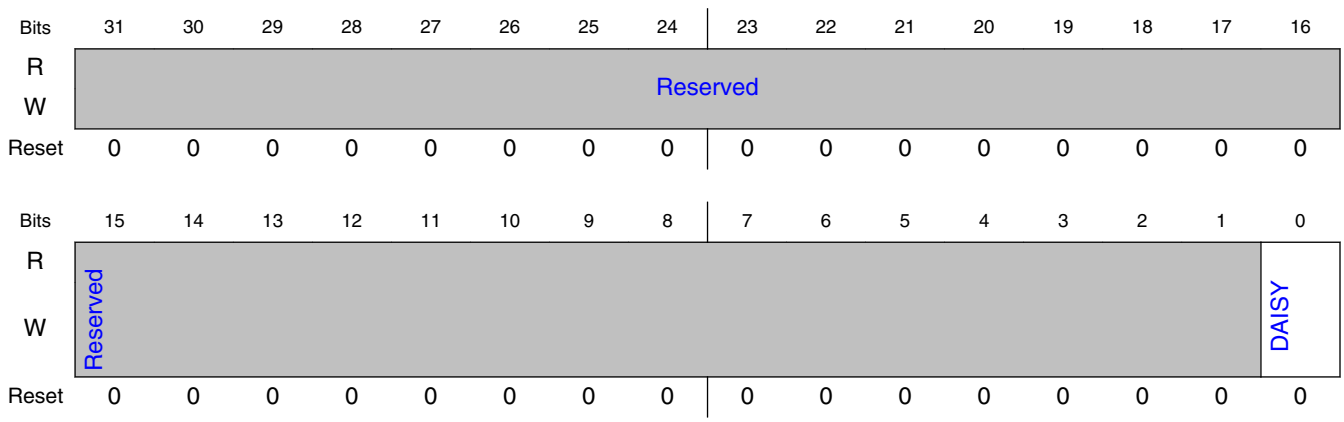
12.4.6.352.1 Offset

Register	Offset
FLEXSPI2_I_IO_FA_SELECT_INPUT_3	588h

12.4.6.352.2 Function

DAISY Register

12.4.6.352.3 Diagram



12.4.6.352.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi2, In Pin: i_io_fa3 0b - Selecting Pad: GPIO_EMC_B2_16 for Mode: ALT4 1b - Selecting Pad: GPIO_SD_B1_05 for Mode: ALT6

12.4.6.353 FLEXSPI2_I_SCK_FA_SELECT_INPUT DAISY Register (FLEXSPI2_I_SCK_FA_SELECT_INPUT)

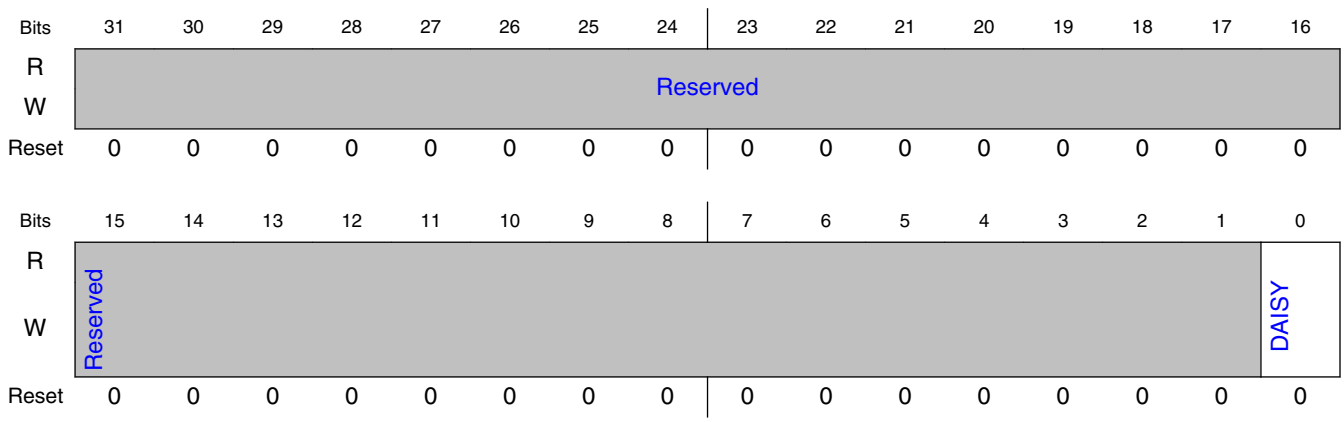
12.4.6.353.1 Offset

Register	Offset
FLEXSPI2_I_SCK_FA_SELECT_INPUT	58Ch

12.4.6.353.2 Function

DAISY Register

12.4.6.353.3 Diagram



12.4.6.353.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flexspi2, In Pin: i_sck_fa 0b - Selecting Pad: GPIO_EMC_B2_10 for Mode: ALT4 1b - Selecting Pad: GPIO_SD_B1_01 for Mode: ALT6

12.4.6.354 GPT3_CAPIN1_SELECT_INPUT DAISY Register (GPT3_CAPIN1_SELECT_INPUT)

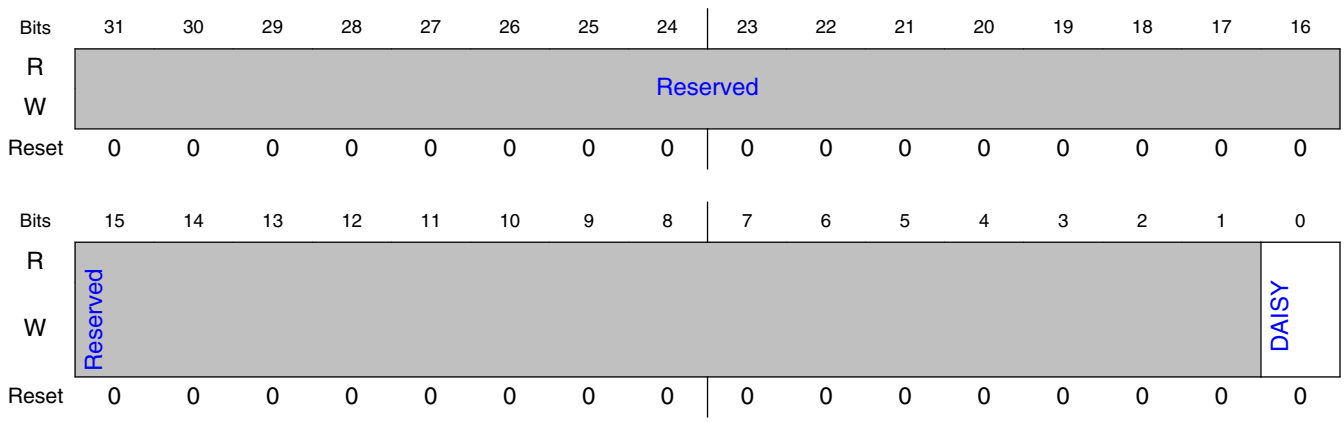
12.4.6.354.1 Offset

Register	Offset
GPT3_CAPIN1_SELECT_INPUT	590h

12.4.6.354.2 Function

DAISY Register

12.4.6.354.3 Diagram



12.4.6.354.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt3, In Pin: capin1 0b - Selecting Pad: GPIO_EMC_B2_06 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_06 for Mode: ALT3

12.4.6.355 GPT3_CAPIN2_SELECT_INPUT DAISY Register (GPT3_CAPIN2_SELECT_INPUT)

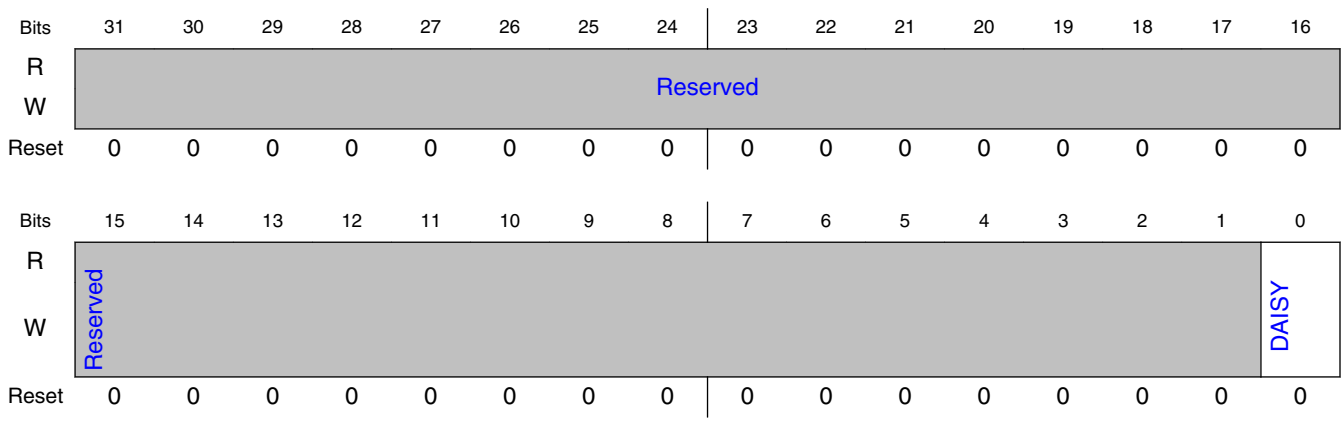
12.4.6.355.1 Offset

Register	Offset
GPT3_CAPIN2_SELECT_INPUT	594h

12.4.6.355.2 Function

DAISY Register

12.4.6.355.3 Diagram



12.4.6.355.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt3, In Pin: capin2 0b - Selecting Pad: GPIO_EMC_B2_07 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_07 for Mode: ALT3

12.4.6.356 GPT3_CLKIN_SELECT_INPUT DAISY Register (GPT3_CLKIN_SELECT_INPUT)

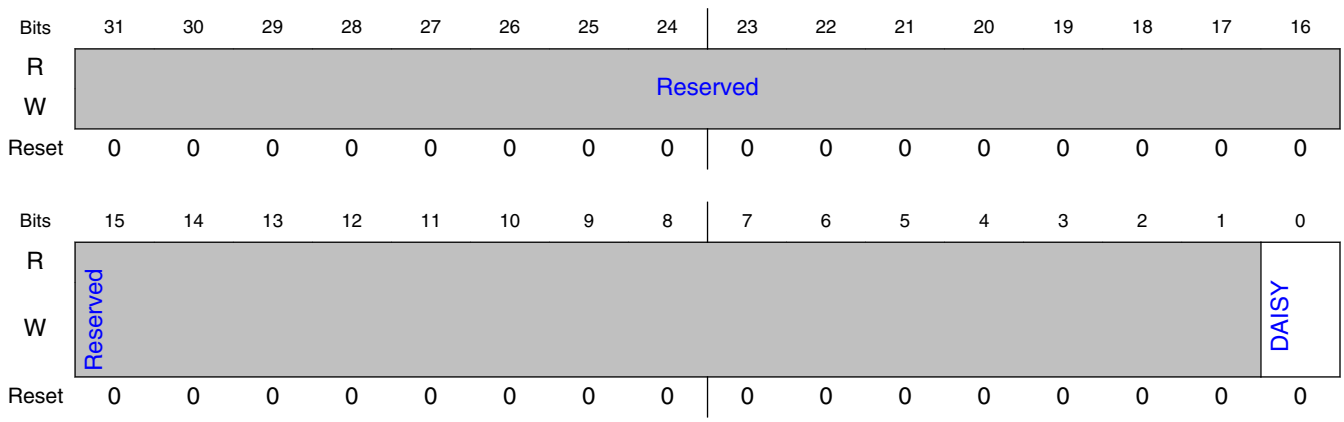
12.4.6.356.1 Offset

Register	Offset
GPT3_CLKIN_SELECT_INPUT	598h

12.4.6.356.2 Function

DAISY Register

12.4.6.356.3 Diagram



12.4.6.356.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpt3, In Pin: clkin 0b - Selecting Pad: GPIO_EMC_B2_05 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_11 for Mode: ALT3

12.4.6.357 KPP_COL_SELECT_INPUT_6 DAISY Register (KPP_COL_SELECT_INPUT_6)

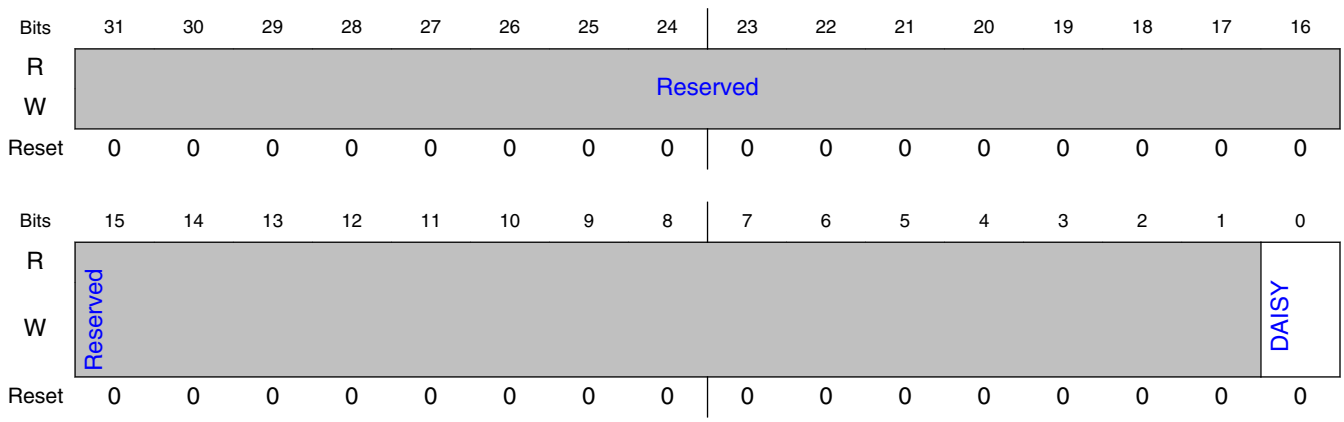
12.4.6.357.1 Offset

Register	Offset
KPP_COL_SELECT_INPUT_6	59Ch

12.4.6.357.2 Function

DAISY Register

12.4.6.357.3 Diagram



12.4.6.357.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: col6 0b - Selecting Pad: GPIO_AD_23 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_03 for Mode: ALT8

12.4.6.358 KPP_COL_SELECT_INPUT_7 DAISY Register (KPP_COL_SELECT_INPUT_7)

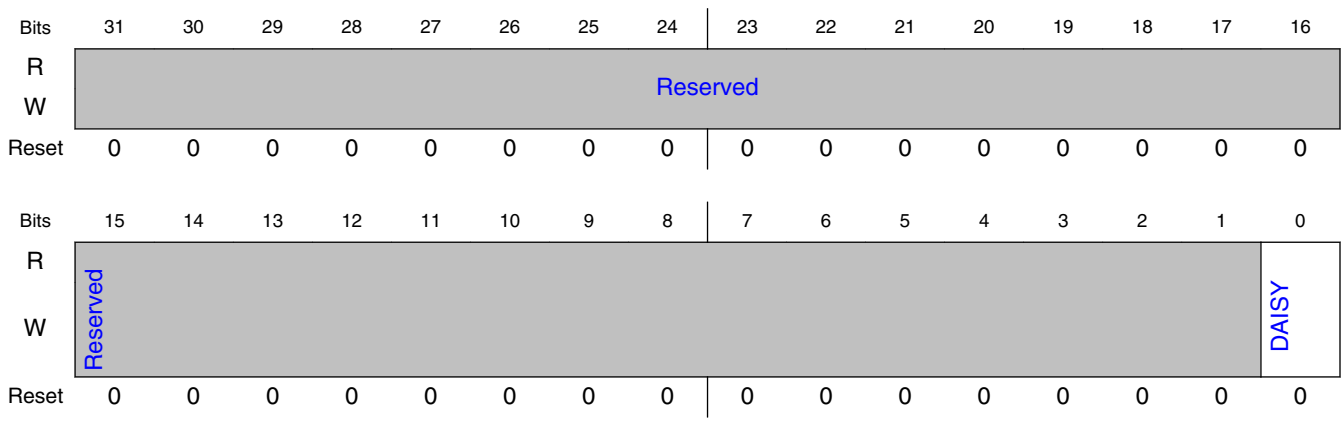
12.4.6.358.1 Offset

Register	Offset
KPP_COL_SELECT_INPUT_7	5A0h

12.4.6.358.2 Function

DAISY Register

12.4.6.358.3 Diagram



12.4.6.358.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: col7 0b - Selecting Pad: GPIO_AD_21 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_01 for Mode: ALT8

12.4.6.359 KPP_ROW_SELECT_INPUT_6 DAISY Register (KPP_ROW_SELECT_INPUT_6)

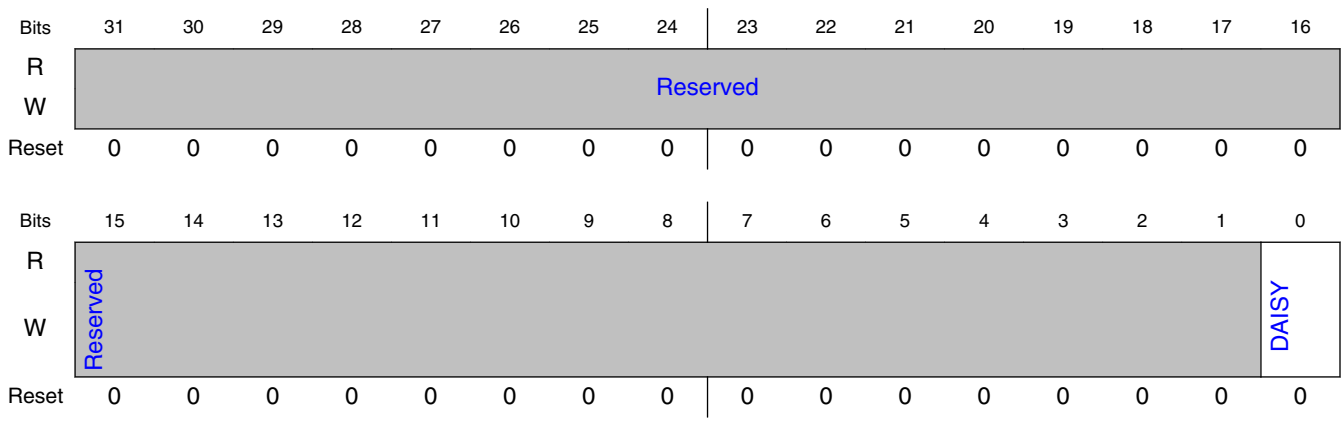
12.4.6.359.1 Offset

Register	Offset
KPP_ROW_SELECT_INPUT_6	5A4h

12.4.6.359.2 Function

DAISY Register

12.4.6.359.3 Diagram



12.4.6.359.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: row6 0b - Selecting Pad: GPIO_AD_22 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_02 for Mode: ALT8

12.4.6.360 KPP_ROW_SELECT_INPUT_7 DAISY Register (KPP_ROW_SELECT_INPUT_7)

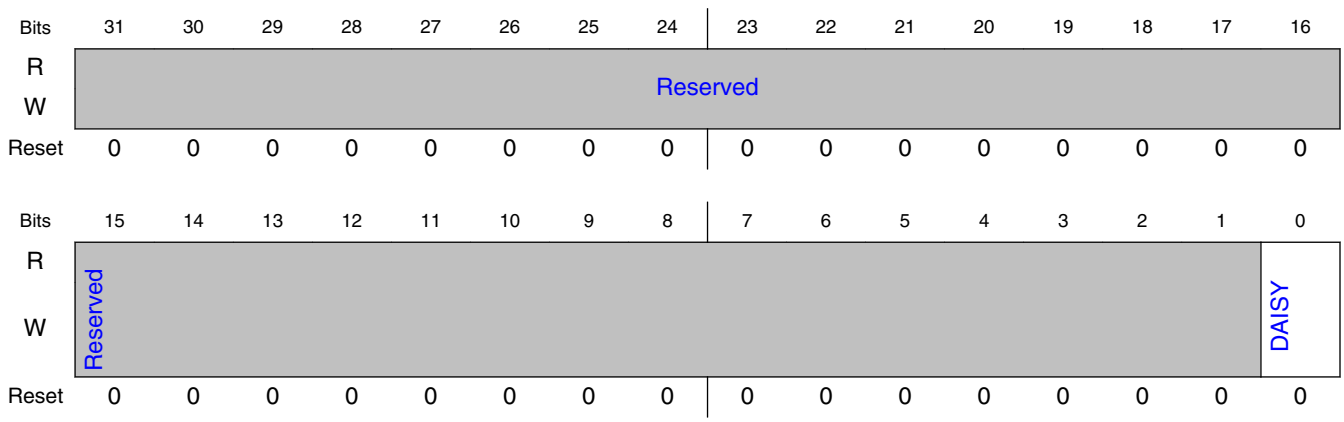
12.4.6.360.1 Offset

Register	Offset
KPP_ROW_SELECT_INPUT_7	5A8h

12.4.6.360.2 Function

DAISY Register

12.4.6.360.3 Diagram



12.4.6.360.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: row7 0b - Selecting Pad: GPIO_AD_20 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_00 for Mode: ALT8

12.4.6.361 LPI2C1_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C1_LPI2C_SCL_SELECT_INPUT)

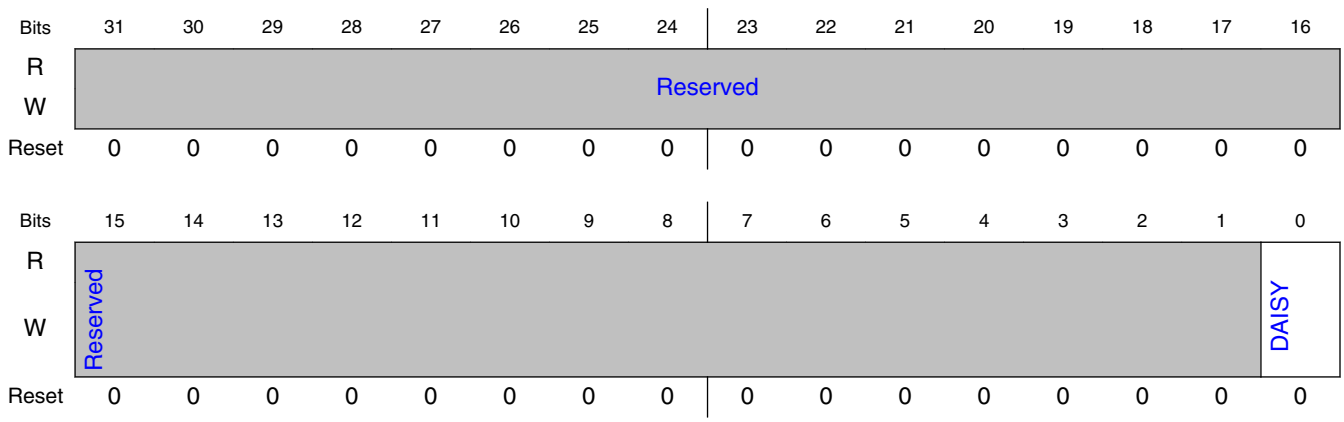
12.4.6.361.1 Offset

Register	Offset
LPI2C1_LPI2C_SCL_SELECT_INPUT	5ACh

12.4.6.361.2 Function

DAISY Register

12.4.6.361.3 Diagram



12.4.6.361.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c1, In Pin: lpi2c_scl 0b - Selecting Pad: GPIO_AD_08 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_32 for Mode: ALT0

12.4.6.362 LPI2C1_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C1_LPI2C_SDA_SELECT_INPUT)

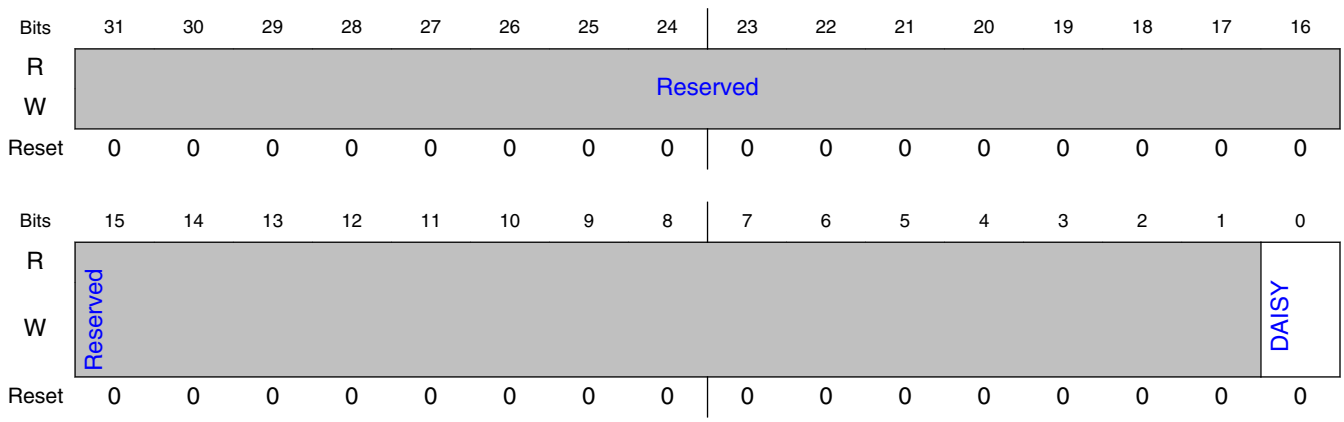
12.4.6.362.1 Offset

Register	Offset
LPI2C1_LPI2C_SDA_SELECT_INPUT	5B0h

12.4.6.362.2 Function

DAISY Register

12.4.6.362.3 Diagram



12.4.6.362.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c1, In Pin: lpi2c_sda 0b - Selecting Pad: GPIO_AD_09 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_33 for Mode: ALT0

12.4.6.363 LPI2C2_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C2_LPI2C_SCL_SELECT_INPUT)

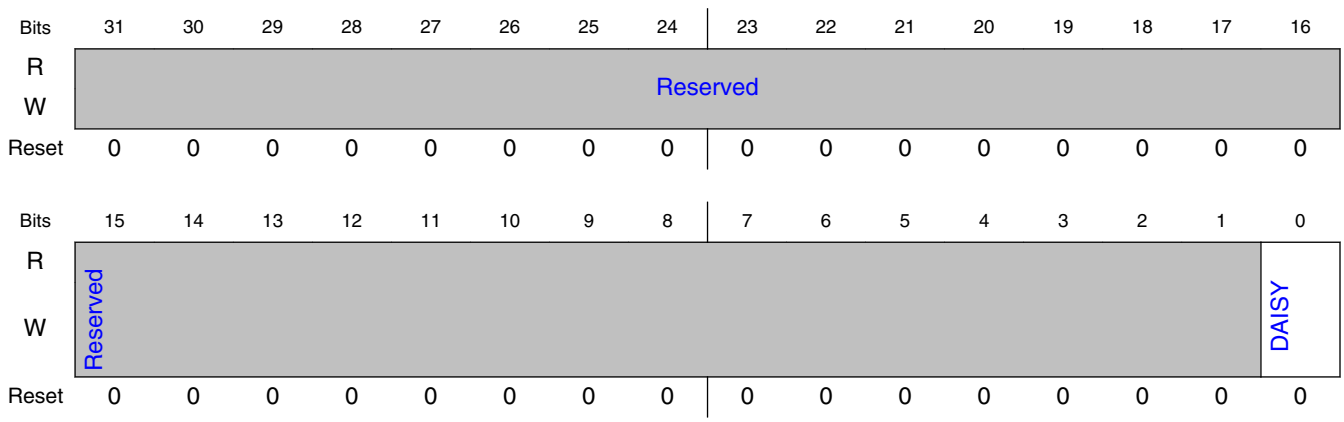
12.4.6.363.1 Offset

Register	Offset
LPI2C2_LPI2C_SCL_SELECT_INPUT	5B4h

12.4.6.363.2 Function

DAISY Register

12.4.6.363.3 Diagram



12.4.6.363.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c2, In Pin: lpi2c_scl 0b - Selecting Pad: GPIO_EMC_B2_00 for Mode: ALT9 1b - Selecting Pad: GPIO_AD_18 for Mode: ALT9

12.4.6.364 LPI2C2_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C2_LPI2C_SDA_SELECT_INPUT)

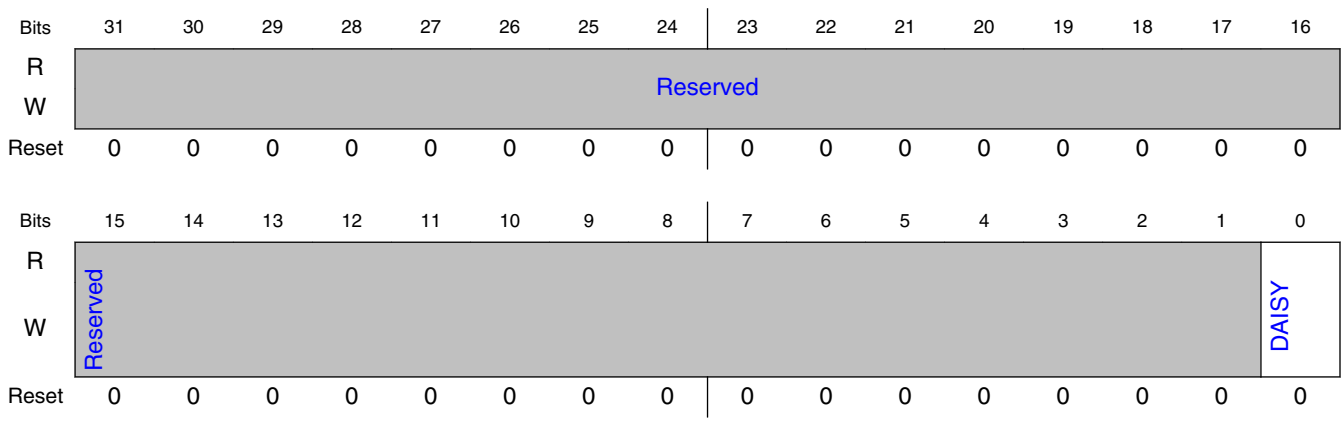
12.4.6.364.1 Offset

Register	Offset
LPI2C2_LPI2C_SDA_SELECT_INPUT	5B8h

12.4.6.364.2 Function

DAISY Register

12.4.6.364.3 Diagram



12.4.6.364.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c2, In Pin: lpi2c_sda 0b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT9 1b - Selecting Pad: GPIO_AD_19 for Mode: ALT9

12.4.6.365 LPI2C3_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C3_LPI2C_SCL_SELECT_INPUT)

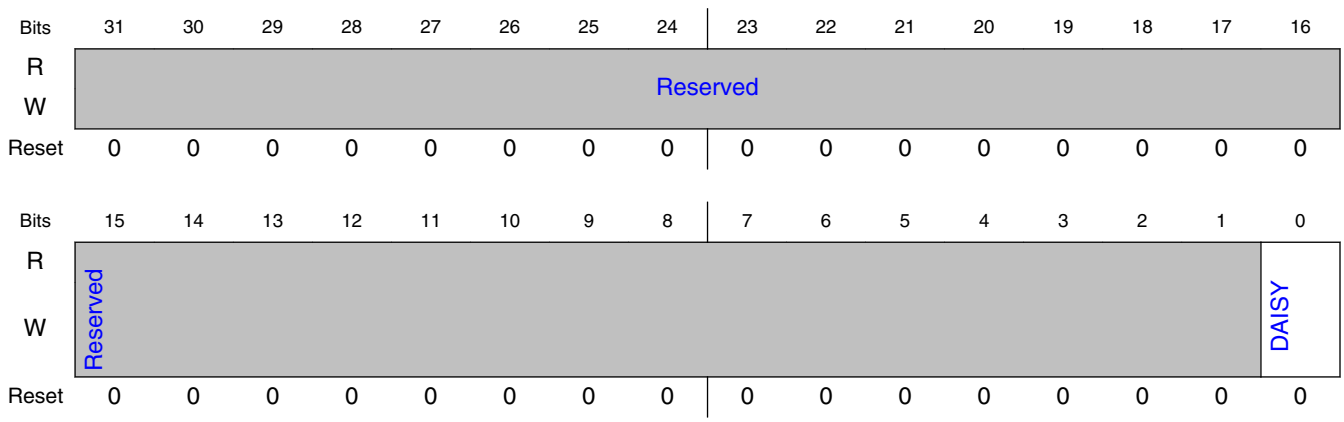
12.4.6.365.1 Offset

Register	Offset
LPI2C3_LPI2C_SCL_SELECT_INPUT	5BCh

12.4.6.365.2 Function

DAISY Register

12.4.6.365.3 Diagram



12.4.6.365.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c3, In Pin: lpi2c_scl 0b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B2_10 for Mode: ALT6

12.4.6.366 LPI2C3_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C3_LPI2C_SDA_SELECT_INPUT)

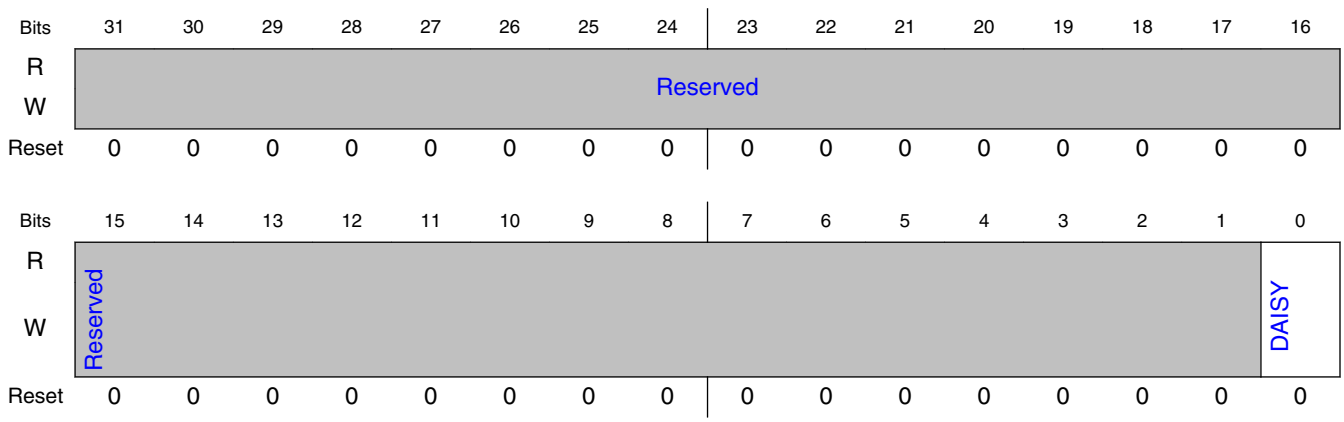
12.4.6.366.1 Offset

Register	Offset
LPI2C3_LPI2C_SDA_SELECT_INPUT	5C0h

12.4.6.366.2 Function

DAISY Register

12.4.6.366.3 Diagram



12.4.6.366.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c3, In Pin: lpi2c_sda 0b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B2_11 for Mode: ALT6

12.4.6.367 LPI2C4_LPI2C_SCL_SELECT_INPUT DAISY Register (LPI2C4_LPI2C_SCL_SELECT_INPUT)

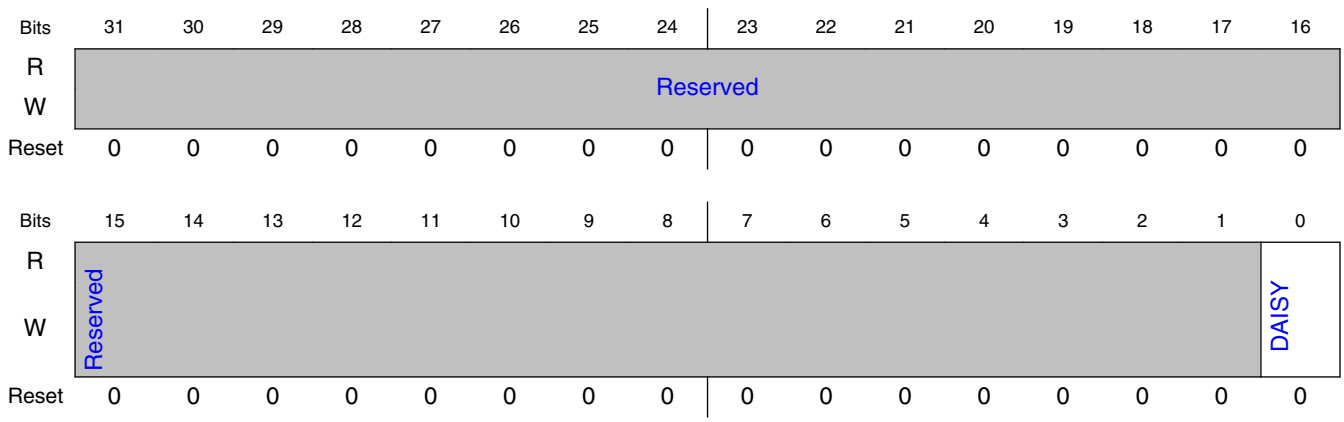
12.4.6.367.1 Offset

Register	Offset
LPI2C4_LPI2C_SCL_SELECT_INPUT	5C4h

12.4.6.367.2 Function

DAISY Register

12.4.6.367.3 Diagram



12.4.6.367.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c4, In Pin: lpi2c_scl 0b - Selecting Pad: GPIO_AD_24 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B2_12 for Mode: ALT6

12.4.6.368 LPI2C4_LPI2C_SDA_SELECT_INPUT DAISY Register (LPI2C4_LPI2C_SDA_SELECT_INPUT)

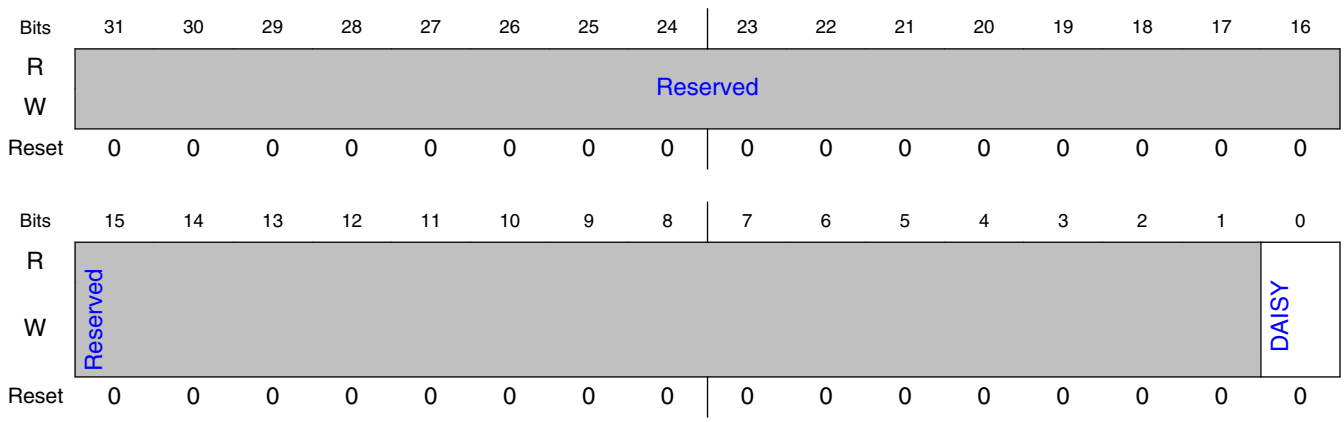
12.4.6.368.1 Offset

Register	Offset
LPI2C4_LPI2C_SDA_SELECT_INPUT	5C8h

12.4.6.368.2 Function

DAISY Register

12.4.6.368.3 Diagram



12.4.6.368.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpi2c4, In Pin: lpi2c_sda 0b - Selecting Pad: GPIO_AD_25 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B2_13 for Mode: ALT6

12.4.6.369 LPSPI1_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI1_LPSPI_PCS_SELECT_INPUT_0)

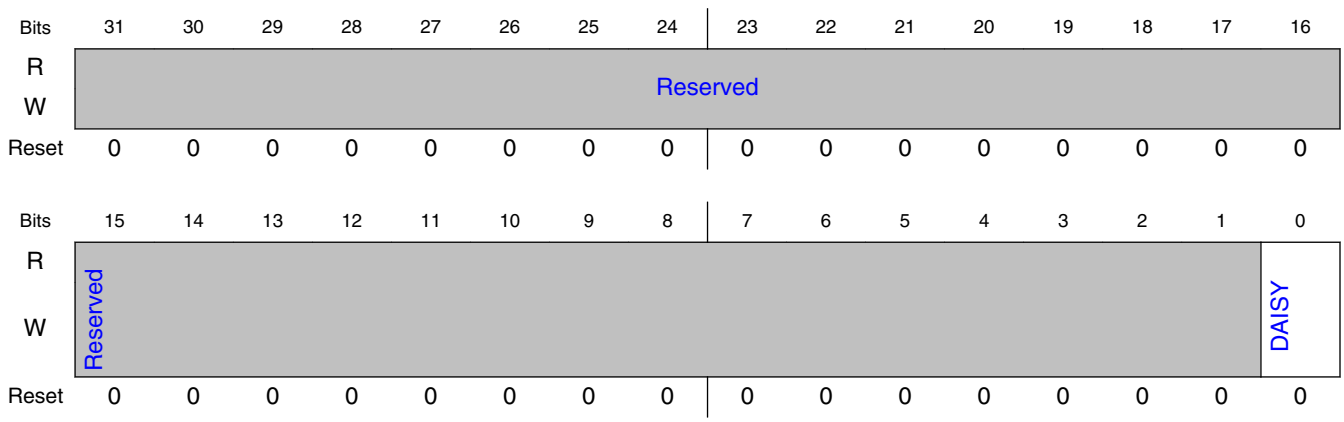
12.4.6.369.1 Offset

Register	Offset
LPSPI1_LPSPI_PCS_SELECT_INPUT_0	5CCh

12.4.6.369.2 Function

DAISY Register

12.4.6.369.3 Diagram



12.4.6.369.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi1, In Pin: lpspi_pcs0 0b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_29 for Mode: ALT0

12.4.6.370 LPSPI1_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SCK_SELECT_INPUT)

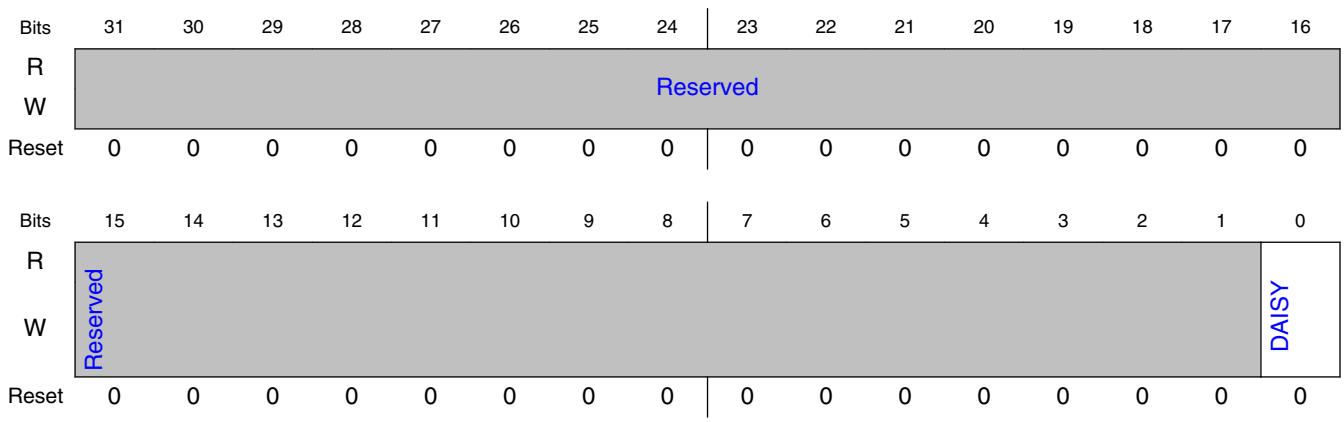
12.4.6.370.1 Offset

Register	Offset
LPSPI1_LPSPI_SCK_SELECT_INPUT	5D0h

12.4.6.370.2 Function

DAISY Register

12.4.6.370.3 Diagram



12.4.6.370.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi1, In Pin: lpspi_sck 0b - Selecting Pad: GPIO_EMC_B2_00 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_28 for Mode: ALT0

12.4.6.371 LPSPI1_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SDI_SELECT_INPUT)

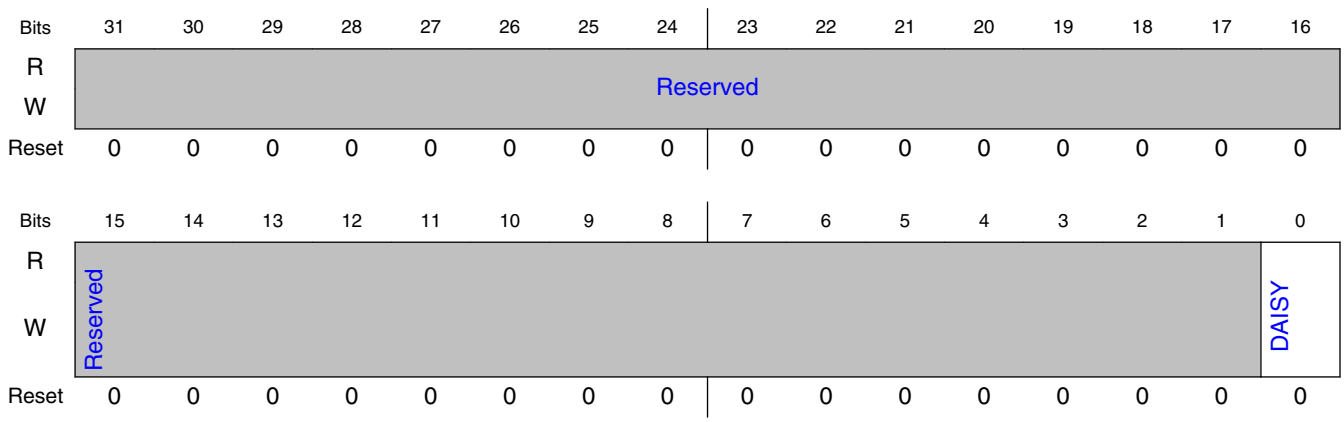
12.4.6.371.1 Offset

Register	Offset
LPSPI1_LPSPI_SDI_SELECT_INPUT	5D4h

12.4.6.371.2 Function

DAISY Register

12.4.6.371.3 Diagram



12.4.6.371.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi1, In Pin: lpspi_sdi 0b - Selecting Pad: GPIO_EMC_B2_03 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_31 for Mode: ALT0

12.4.6.372 LPSPI1_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI1_LPSPI_SDO_SELECT_INPUT)

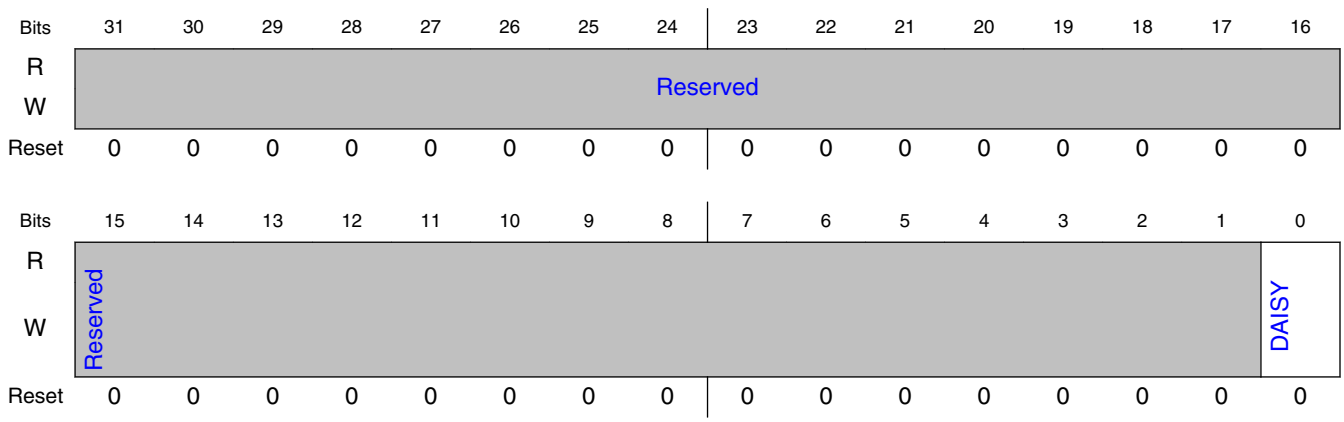
12.4.6.372.1 Offset

Register	Offset
LPSPI1_LPSPI_SDO_SELECT_INPUT	5D8h

12.4.6.372.2 Function

DAISY Register

12.4.6.372.3 Diagram



12.4.6.372.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi1, In Pin: lpspi_sdo 0b - Selecting Pad: GPIO_EMC_B2_02 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_30 for Mode: ALT0

12.4.6.373 LPSPI2_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI2_LPSPI_PCS_SELECT_INPUT_0)

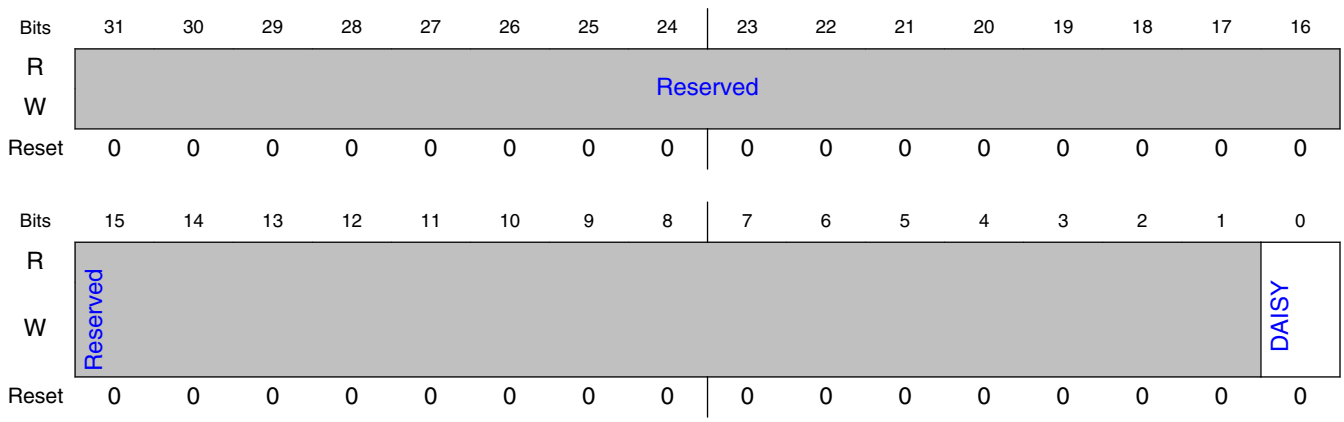
12.4.6.373.1 Offset

Register	Offset
LPSPI2_LPSPI_PCS_SELECT_INPUT_0	5DCh

12.4.6.373.2 Function

DAISY Register

12.4.6.373.3 Diagram



12.4.6.373.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi2, In Pin: lpspi_pcs0 0b - Selecting Pad: GPIO_AD_25 for Mode: ALT1 1b - Selecting Pad: GPIO_SD_B2_08 for Mode: ALT6

12.4.6.374 LPSPI2_LPSPI_PCS_SELECT_INPUT_1 DAISY Register (LPSPI2_LPSPI_PCS_SELECT_INPUT_1)

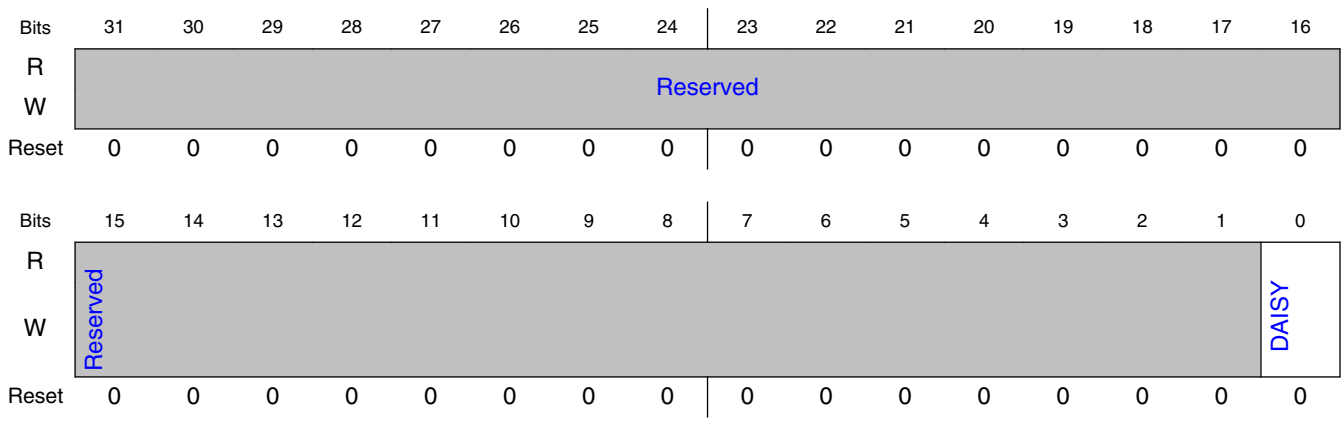
12.4.6.374.1 Offset

Register	Offset
LPSPI2_LPSPI_PCS_SELECT_INPUT_1	5E0h

12.4.6.374.2 Function

DAISY Register

12.4.6.374.3 Diagram



12.4.6.374.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi2, In Pin: lpspi_pcs1 0b - Selecting Pad: GPIO_AD_21 for Mode: ALT2 1b - Selecting Pad: GPIO_SD_B2_11 for Mode: ALT6

12.4.6.375 LPSPI2_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SCK_SELECT_INPUT)

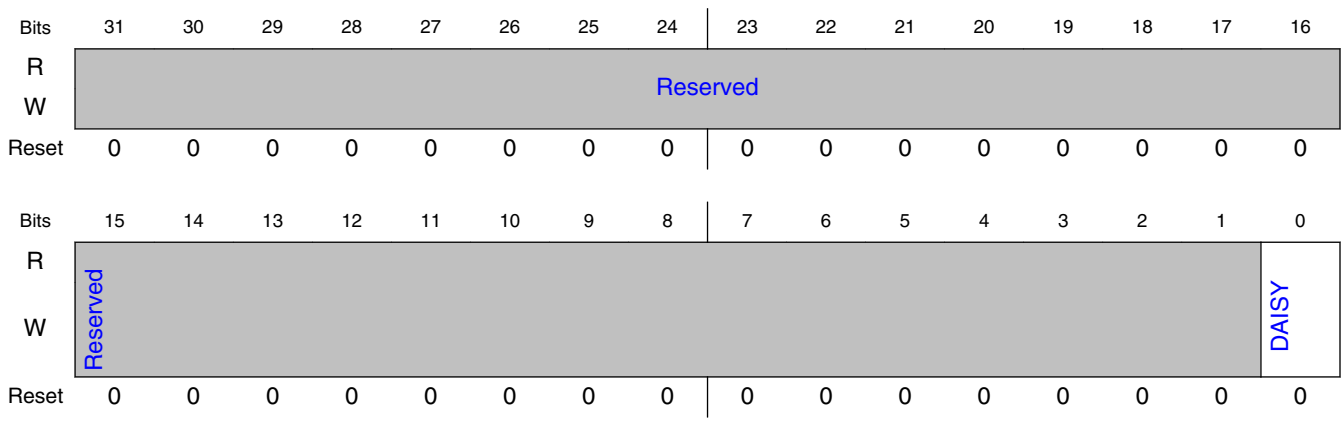
12.4.6.375.1 Offset

Register	Offset
LPSPI2_LPSPI_SCK_SELECT_INPUT	5E4h

12.4.6.375.2 Function

DAISY Register

12.4.6.375.3 Diagram



12.4.6.375.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi2, In Pin: lpspi_sck 0b - Selecting Pad: GPIO_AD_24 for Mode: ALT1 1b - Selecting Pad: GPIO_SD_B2_07 for Mode: ALT6

12.4.6.376 LPSPI2_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SDI_SELECT_INPUT)

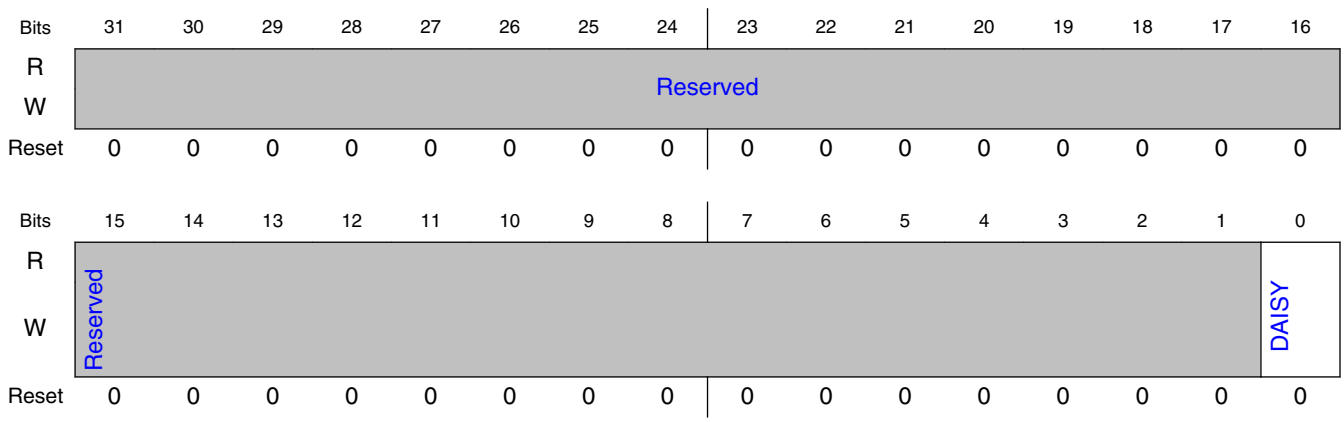
12.4.6.376.1 Offset

Register	Offset
LPSPI2_LPSPI_SDI_SELECT_INPUT	5E8h

12.4.6.376.2 Function

DAISY Register

12.4.6.376.3 Diagram



12.4.6.376.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi2, In Pin: lpspi_sdi 0b - Selecting Pad: GPIO_AD_27 for Mode: ALT1 1b - Selecting Pad: GPIO_SD_B2_10 for Mode: ALT6

12.4.6.377 LPSPI2_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI2_LPSPI_SDO_SELECT_INPUT)

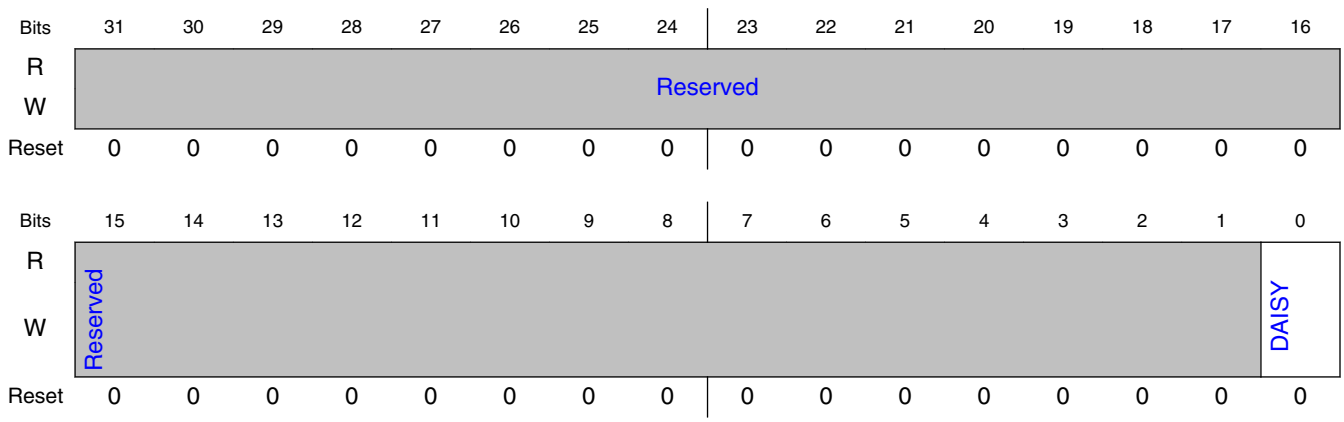
12.4.6.377.1 Offset

Register	Offset
LPSPI2_LPSPI_SDO_SELECT_INPUT	5ECh

12.4.6.377.2 Function

DAISY Register

12.4.6.377.3 Diagram



12.4.6.377.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi2, In Pin: lpspi_sdo 0b - Selecting Pad: GPIO_AD_26 for Mode: ALT1 1b - Selecting Pad: GPIO_SD_B2_09 for Mode: ALT6

12.4.6.378 LPSPI3_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_0)

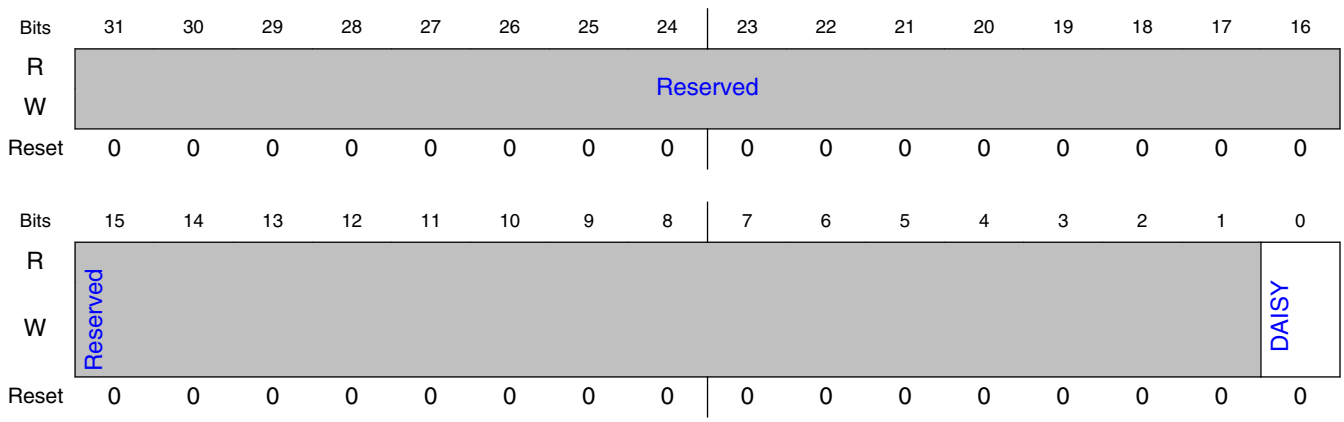
12.4.6.378.1 Offset

Register	Offset
LPSPI3_LPSPI_PCS_SELECT_INPUT_0	5F0h

12.4.6.378.2 Function

DAISY Register

12.4.6.378.3 Diagram



12.4.6.378.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_pcs0 0b - Selecting Pad: GPIO_EMC_B2_05 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_07 for Mode: ALT9

12.4.6.379 LPSPI3_LPSPI_PCS_SELECT_INPUT_1 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_1)

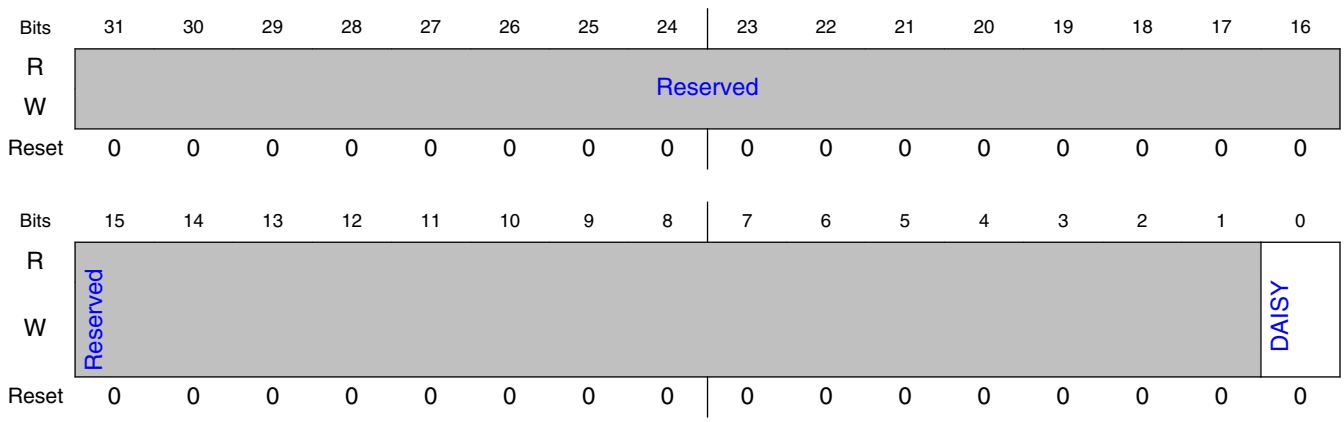
12.4.6.379.1 Offset

Register	Offset
LPSPI3_LPSPI_PCS_SELECT_INPUT_1	5F4h

12.4.6.379.2 Function

DAISY Register

12.4.6.379.3 Diagram



12.4.6.379.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_pcs1 0b - Selecting Pad: GPIO_EMC_B2_08 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_08 for Mode: ALT9

12.4.6.380 LPSPI3_LPSPI_PCS_SELECT_INPUT_2 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_2)

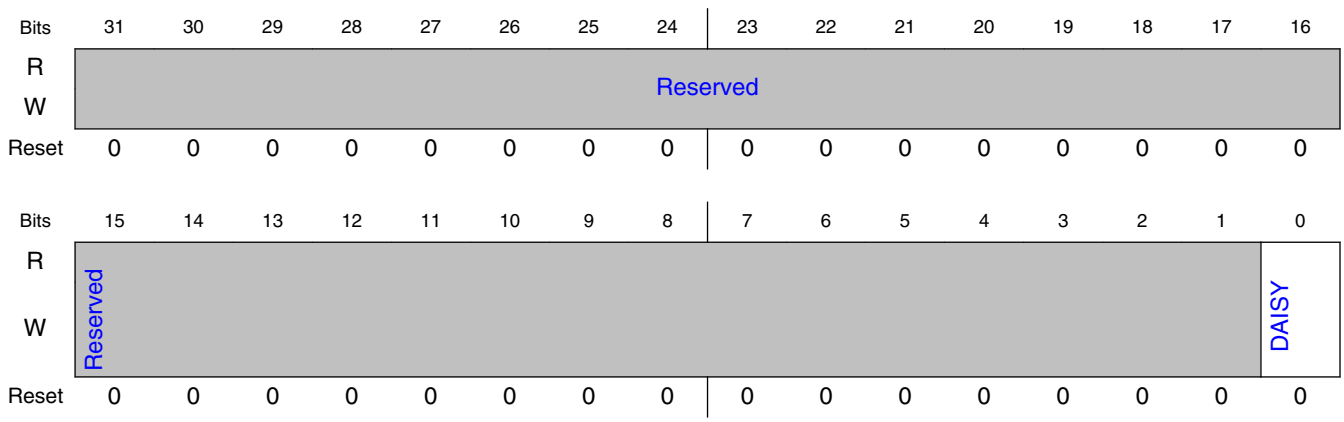
12.4.6.380.1 Offset

Register	Offset
LPSPI3_LPSPI_PCS_SELECT_INPUT_2	5F8h

12.4.6.380.2 Function

DAISY Register

12.4.6.380.3 Diagram



12.4.6.380.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_pcs2 0b - Selecting Pad: GPIO_EMC_B2_09 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_09 for Mode: ALT9

12.4.6.381 LPSPI3_LPSPI_PCS_SELECT_INPUT_3 DAISY Register (LPSPI3_LPSPI_PCS_SELECT_INPUT_3)

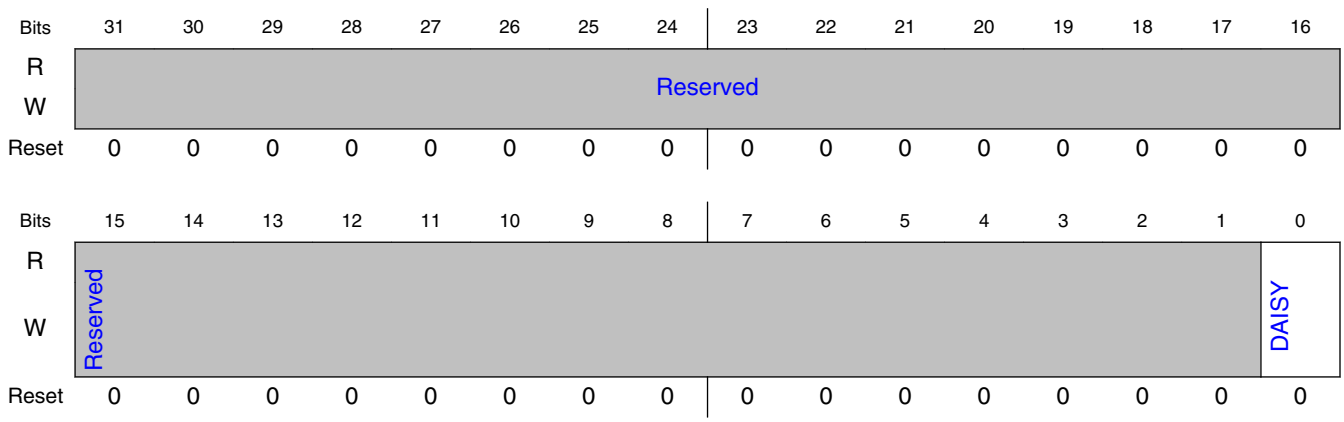
12.4.6.381.1 Offset

Register	Offset
LPSPI3_LPSPI_PCS_SELECT_INPUT_3	5FCh

12.4.6.381.2 Function

DAISY Register

12.4.6.381.3 Diagram



12.4.6.381.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_pcs3 0b - Selecting Pad: GPIO_EMC_B2_10 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_10 for Mode: ALT9

12.4.6.382 LPSPI3_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SCK_SELECT_INPUT)

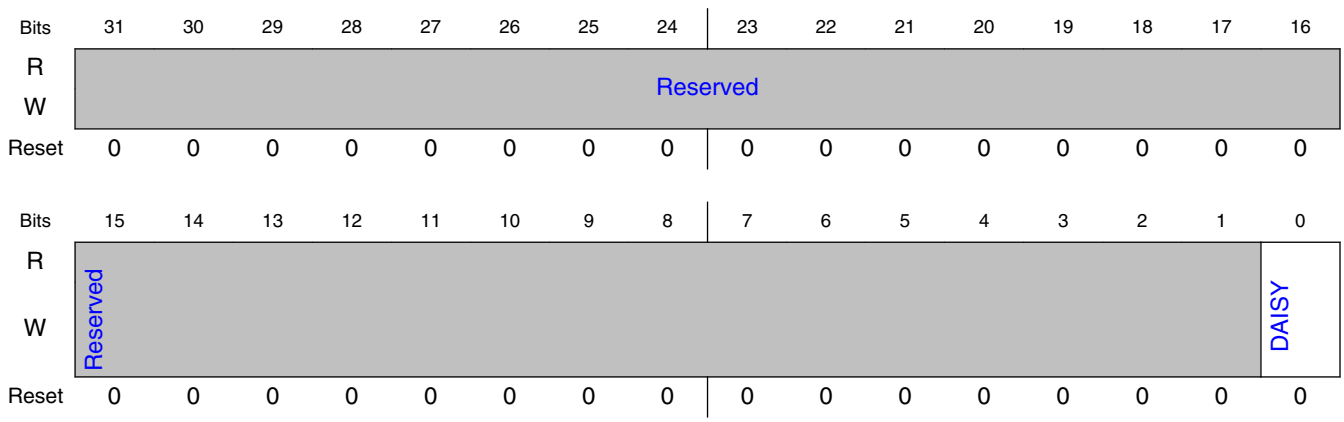
12.4.6.382.1 Offset

Register	Offset
LPSPI3_LPSPI_SCK_SELECT_INPUT	600h

12.4.6.382.2 Function

DAISY Register

12.4.6.382.3 Diagram



12.4.6.382.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_sck 0b - Selecting Pad: GPIO_EMC_B2_04 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_04 for Mode: ALT9

12.4.6.383 LPSPI3_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SDI_SELECT_INPUT)

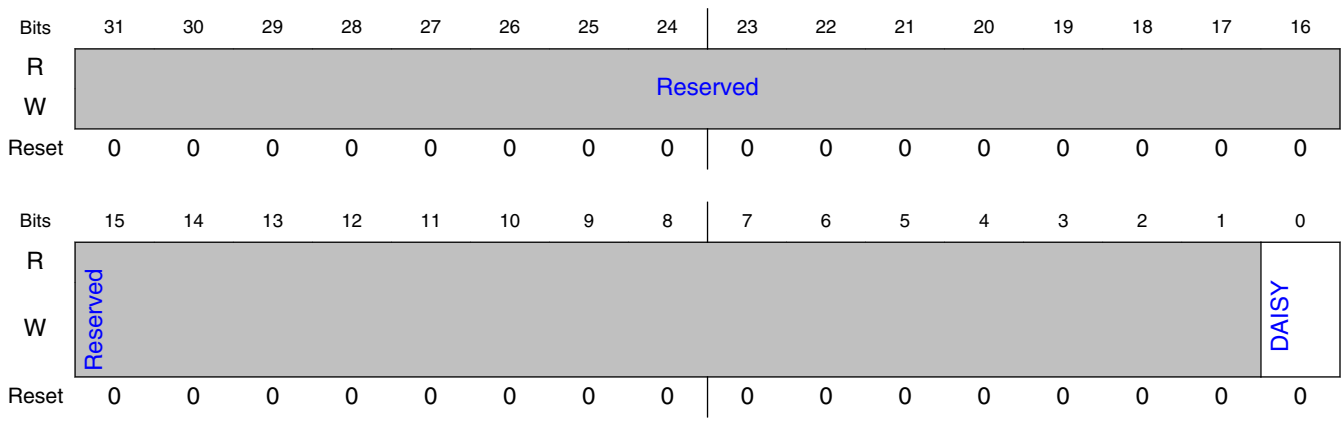
12.4.6.383.1 Offset

Register	Offset
LPSPI3_LPSPI_SDI_SELECT_INPUT	604h

12.4.6.383.2 Function

DAISY Register

12.4.6.383.3 Diagram



12.4.6.383.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_sdi 0b - Selecting Pad: GPIO_EMC_B2_07 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_05 for Mode: ALT9

12.4.6.384 LPSPI3_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI3_LPSPI_SDO_SELECT_INPUT)

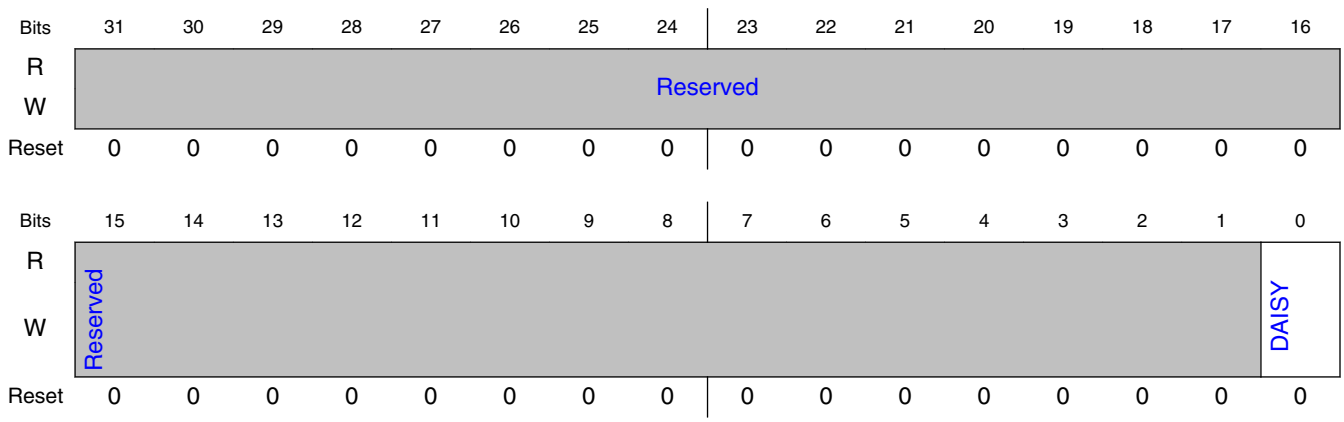
12.4.6.384.1 Offset

Register	Offset
LPSPI3_LPSPI_SDO_SELECT_INPUT	608h

12.4.6.384.2 Function

DAISY Register

12.4.6.384.3 Diagram



12.4.6.384.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi3, In Pin: lpspi_sdo 0b - Selecting Pad: GPIO_EMC_B2_06 for Mode: ALT8 1b - Selecting Pad: GPIO_DISP_B1_06 for Mode: ALT9

12.4.6.385 LPSPI4_LPSPI_PCS_SELECT_INPUT_0 DAISY Register (LPSPI4_LPSPI_PCS_SELECT_INPUT_0)

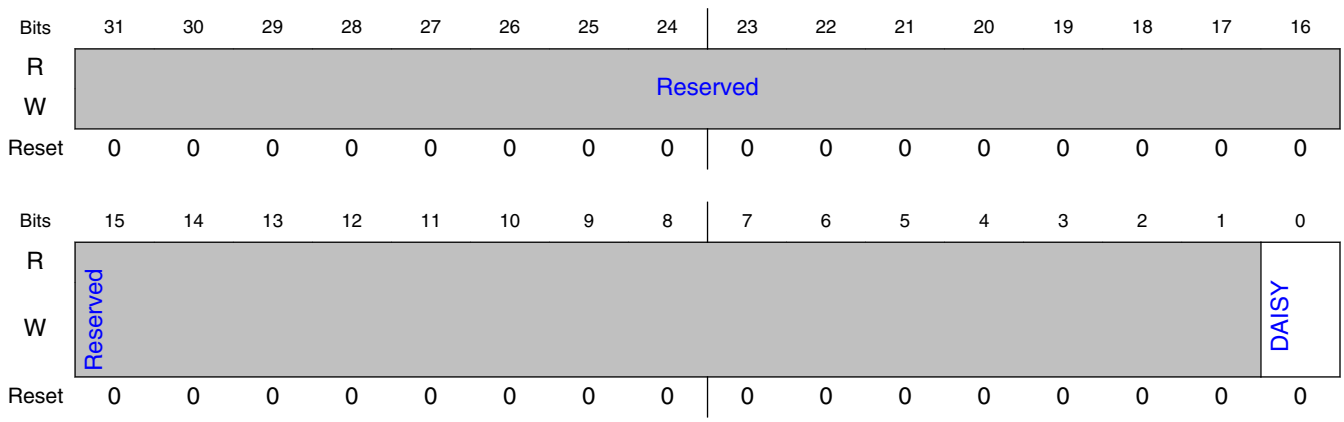
12.4.6.385.1 Offset

Register	Offset
LPSPI4_LPSPI_PCS_SELECT_INPUT_0	60Ch

12.4.6.385.2 Function

DAISY Register

12.4.6.385.3 Diagram



12.4.6.385.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi4, In Pin: lpspi_pcs0 0b - Selecting Pad: GPIO_SD_B2_01 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B2_15 for Mode: ALT9

12.4.6.386 LPSPI4_LPSPI_SCK_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SCK_SELECT_INPUT)

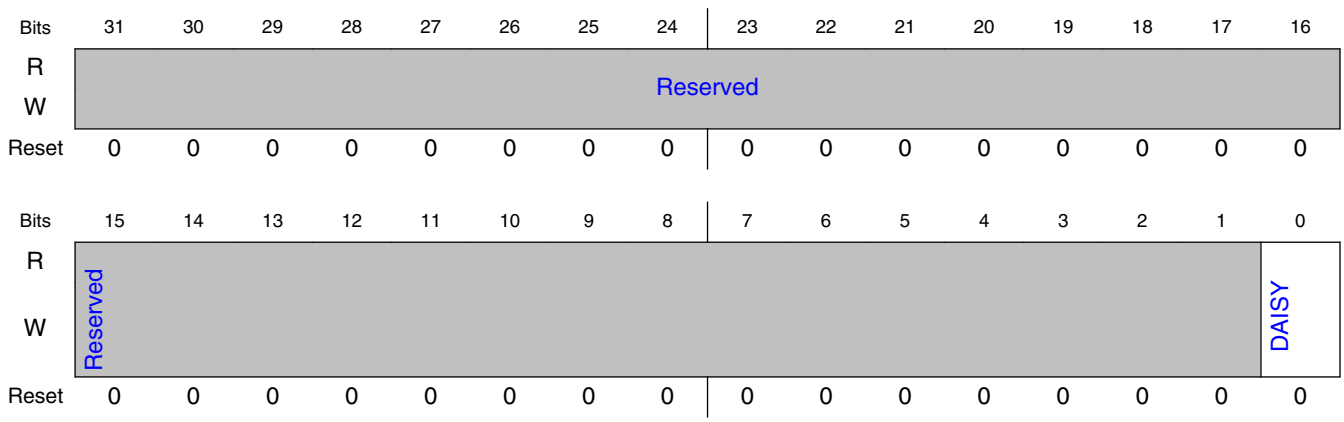
12.4.6.386.1 Offset

Register	Offset
LPSPI4_LPSPI_SCK_SELECT_INPUT	610h

12.4.6.386.2 Function

DAISY Register

12.4.6.386.3 Diagram



12.4.6.386.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi4, In Pin: lpspi_sck 0b - Selecting Pad: GPIO_SD_B2_00 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B2_12 for Mode: ALT9

12.4.6.387 LPSPI4_LPSPI_SDI_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SDI_SELECT_INPUT)

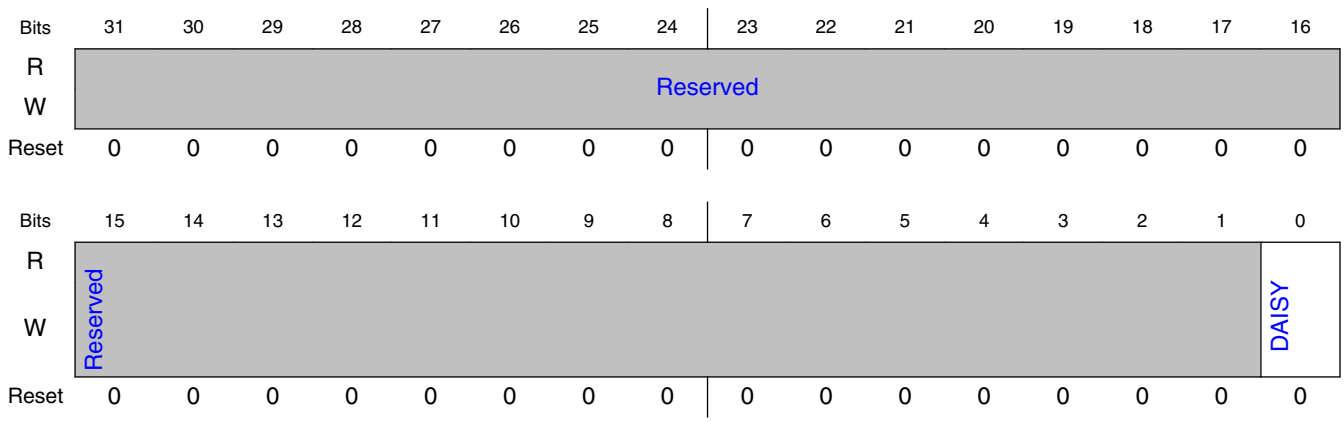
12.4.6.387.1 Offset

Register	Offset
LPSPI4_LPSPI_SDI_SELECT_INPUT	614h

12.4.6.387.2 Function

DAISY Register

12.4.6.387.3 Diagram



12.4.6.387.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi4, In Pin: lpspi_sdi 0b - Selecting Pad: GPIO_SD_B2_03 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B2_13 for Mode: ALT9

12.4.6.388 LPSPI4_LPSPI_SDO_SELECT_INPUT DAISY Register (LPSPI4_LPSPI_SDO_SELECT_INPUT)

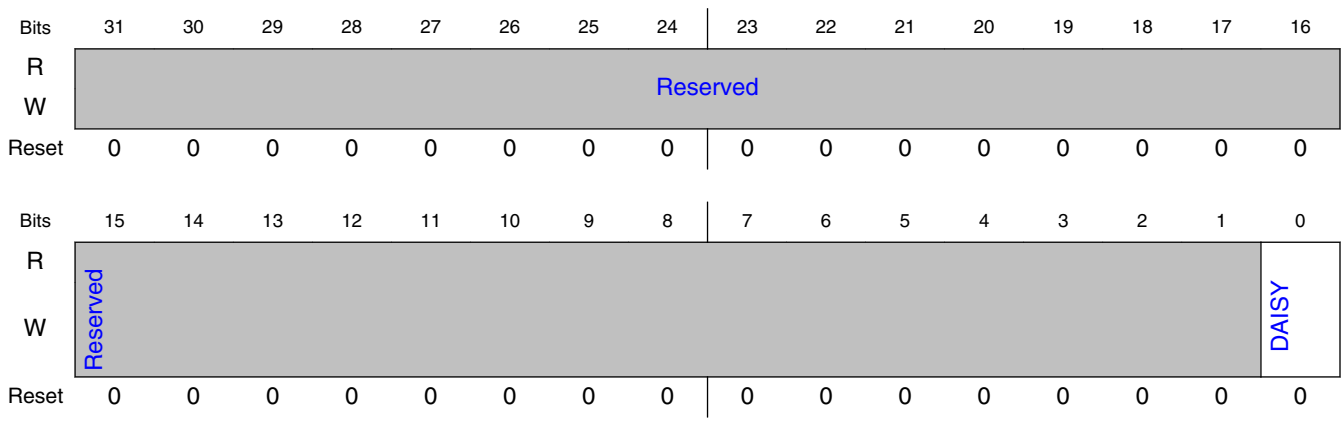
12.4.6.388.1 Offset

Register	Offset
LPSPI4_LPSPI_SDO_SELECT_INPUT	618h

12.4.6.388.2 Function

DAISY Register

12.4.6.388.3 Diagram



12.4.6.388.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpspi4, In Pin: lpspi_sdo 0b - Selecting Pad: GPIO_SD_B2_02 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B2_14 for Mode: ALT9

12.4.6.389 LPUART1_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART1_LPUART_RXD_SELECT_INPUT)

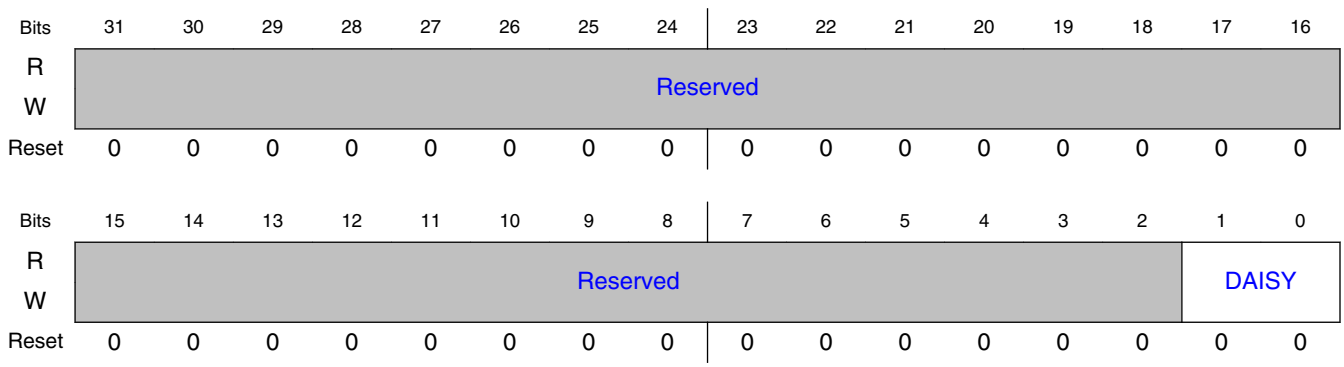
12.4.6.389.1 Offset

Register	Offset
LPUART1_LPUART_RXD_SELECT_INPUT	61Ch

12.4.6.389.2 Function

DAISY Register

12.4.6.389.3 Diagram



12.4.6.389.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart1, In Pin: lpuart_rxd 00b - Selecting Pad: GPIO_AD_25 for Mode: ALT0 01b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B2_09 for Mode: ALT9

12.4.6.390 LPUART1_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART1_LPUART_TXD_SELECT_INPUT)

12.4.6.390.1 Offset

Register	Offset
LPUART1_LPUART_TXD_SELECT_INPUT	620h

12.4.6.390.2 Function

DAISY Register

12.4.6.390.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.390.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart1, In Pin: lpuart_txd 00b - Selecting Pad: GPIO_AD_24 for Mode: ALT0 01b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B2_08 for Mode: ALT9

12.4.6.391 LPUART10_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART10_LPUART_RXD_SELECT_INPUT)

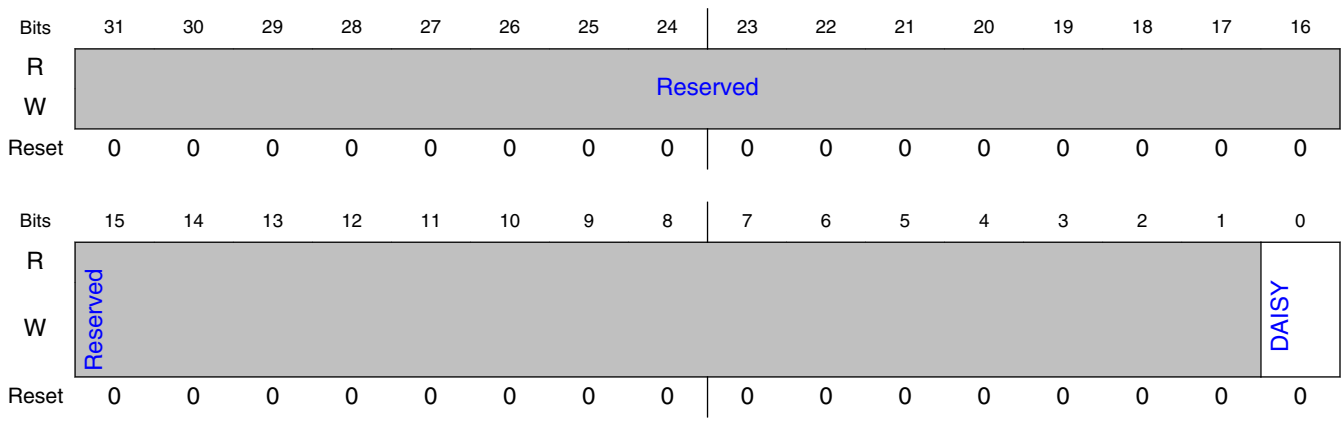
12.4.6.391.1 Offset

Register	Offset
LPUART10_LPUART_RXD_SELECT_INPUT	624h

12.4.6.391.2 Function

DAISY Register

12.4.6.391.3 Diagram



12.4.6.391.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart10, In Pin: lpuart_rxd 0b - Selecting Pad: GPIO_AD_16 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_33 for Mode: ALT8

12.4.6.392 LPUART10_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART10_LPUART_TXD_SELECT_INPUT)

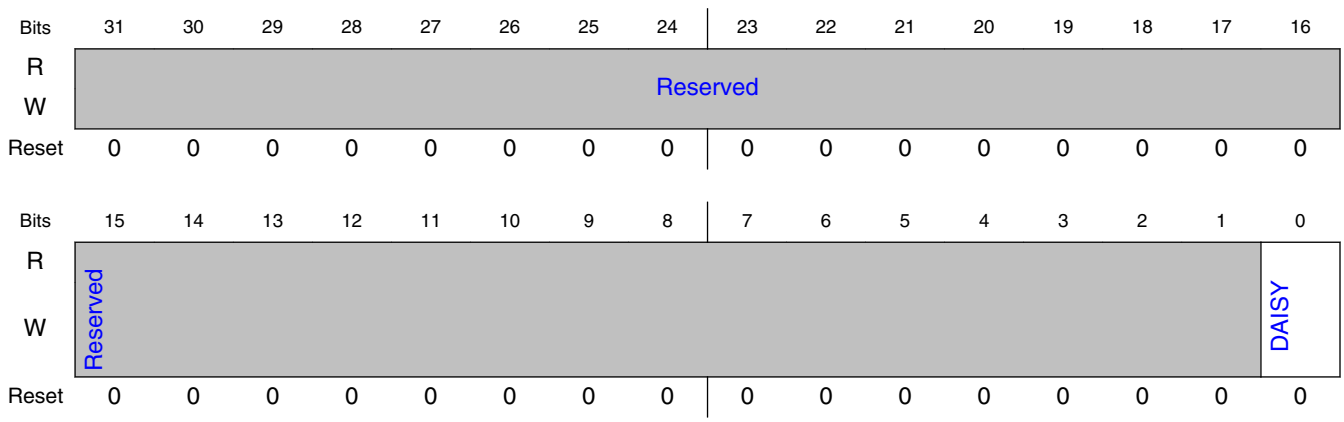
12.4.6.392.1 Offset

Register	Offset
LPUART10_LPUART_TXD_SELECT_INPUT	628h

12.4.6.392.2 Function

DAISY Register

12.4.6.392.3 Diagram



12.4.6.392.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart10, In Pin: lpuart_txd 0b - Selecting Pad: GPIO_AD_15 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_32 for Mode: ALT8

12.4.6.393 LPUART7_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART7_LPUART_RXD_SELECT_INPUT)

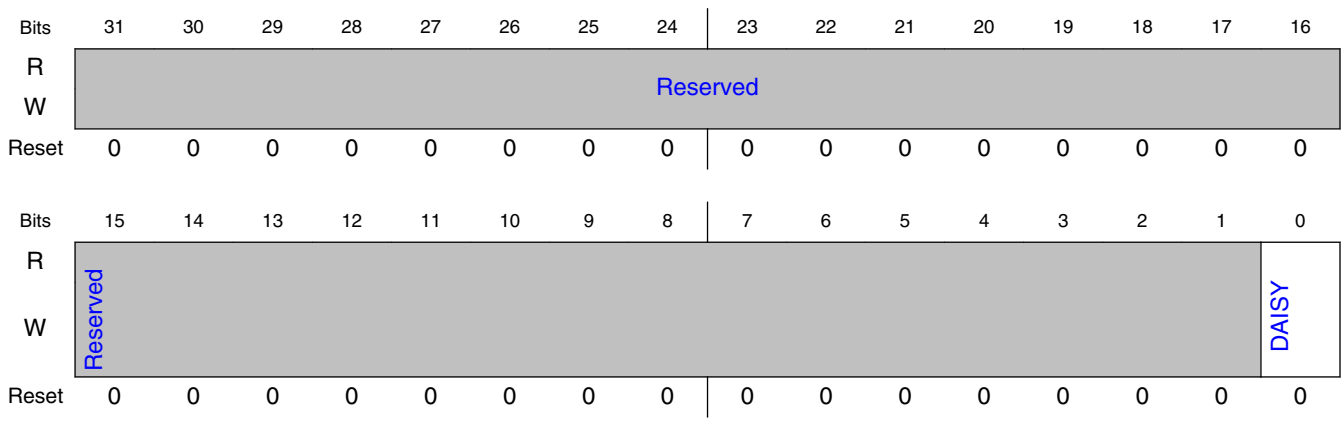
12.4.6.393.1 Offset

Register	Offset
LPUART7_LPUART_RXD_SELECT_INPUT	62Ch

12.4.6.393.2 Function

DAISY Register

12.4.6.393.3 Diagram



12.4.6.393.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart7, In Pin: lpuart_rxd 0b - Selecting Pad: GPIO_AD_01 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B2_07 for Mode: ALT2

12.4.6.394 LPUART7_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART7_LPUART_TXD_SELECT_INPUT)

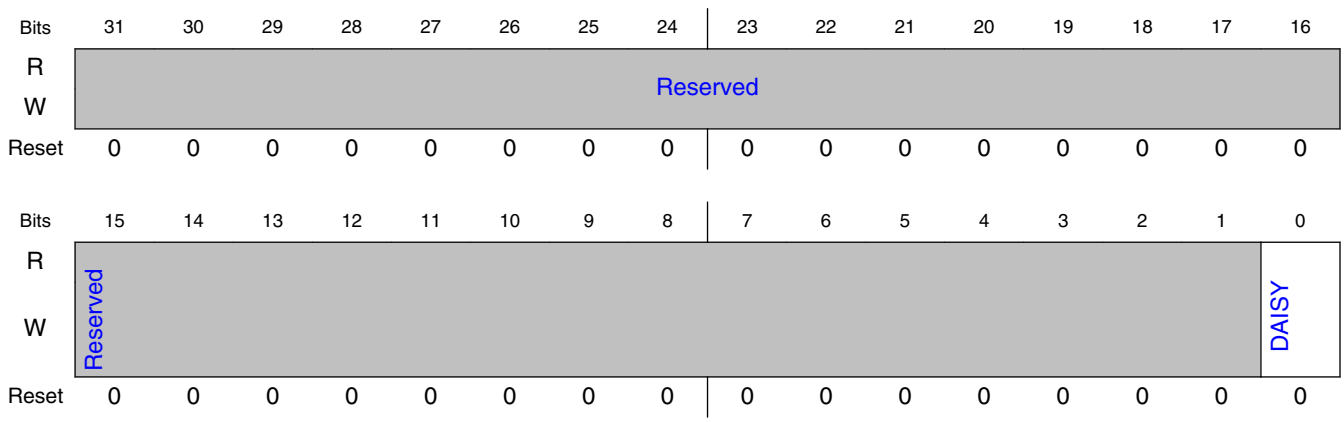
12.4.6.394.1 Offset

Register	Offset
LPUART7_LPUART_TXD_SELECT_INPUT	630h

12.4.6.394.2 Function

DAISY Register

12.4.6.394.3 Diagram



12.4.6.394.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart7, In Pin: lpuart_txd 0b - Selecting Pad: GPIO_AD_00 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B2_06 for Mode: ALT2

12.4.6.395 LPUART8_LPUART_RXD_SELECT_INPUT DAISY Register (LPUART8_LPUART_RXD_SELECT_INPUT)

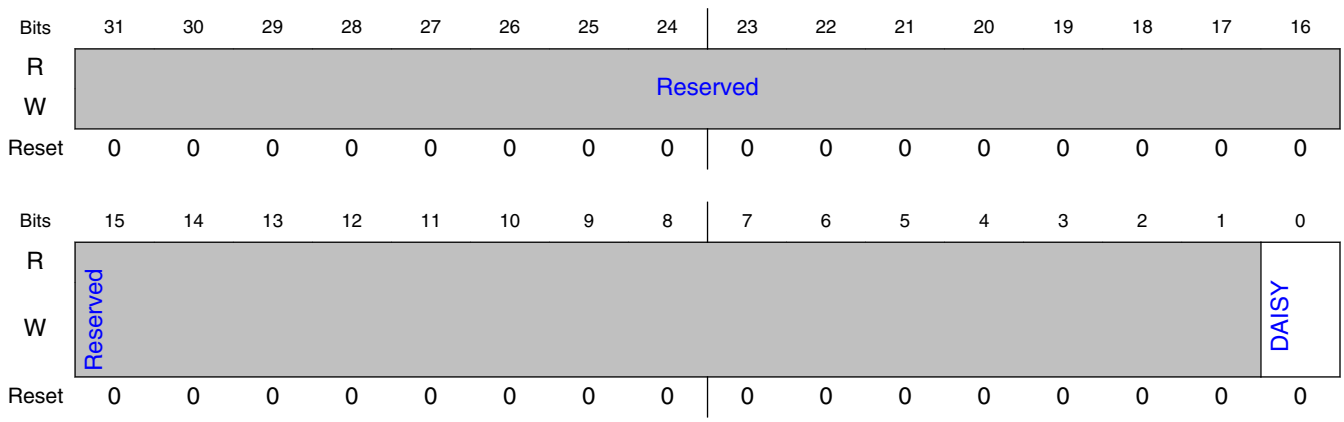
12.4.6.395.1 Offset

Register	Offset
LPUART8_LPUART_RXD_SELECT_INPUT	634h

12.4.6.395.2 Function

DAISY Register

12.4.6.395.3 Diagram



12.4.6.395.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart8, In Pin: lpuart_rxd 0b - Selecting Pad: GPIO_AD_03 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B2_09 for Mode: ALT2

12.4.6.396 LPUART8_LPUART_TXD_SELECT_INPUT DAISY Register (LPUART8_LPUART_TXD_SELECT_INPUT)

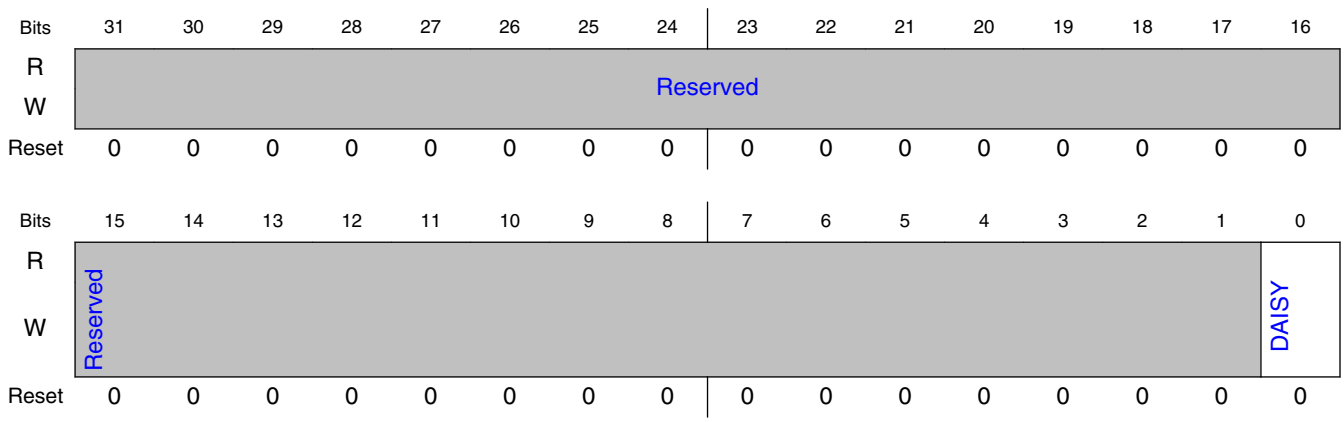
12.4.6.396.1 Offset

Register	Offset
LPUART8_LPUART_TXD_SELECT_INPUT	638h

12.4.6.396.2 Function

DAISY Register

12.4.6.396.3 Diagram



12.4.6.396.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: lpuart8, In Pin: lpuart_txd 0b - Selecting Pad: GPIO_AD_02 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B2_08 for Mode: ALT2

12.4.6.397 QTIMER1_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR0_INPUT_SELECT_INPUT)

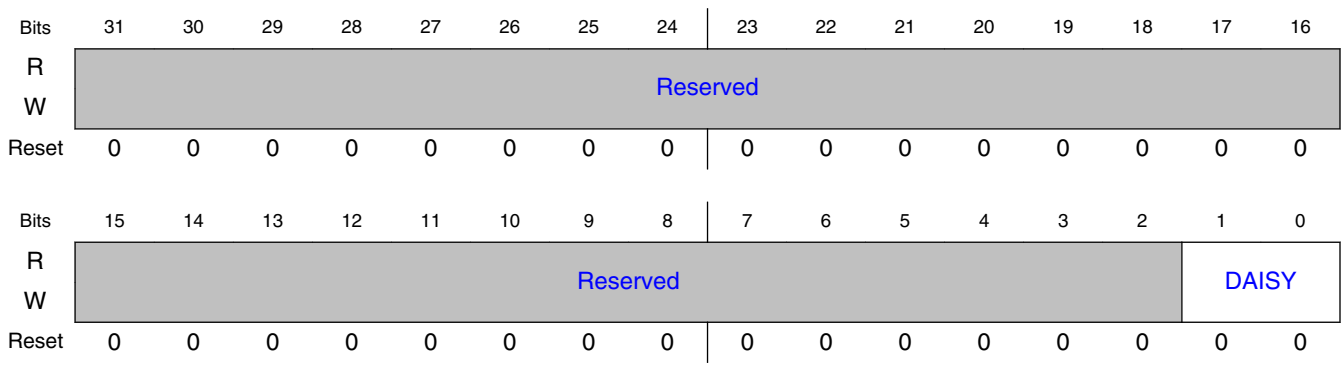
12.4.6.397.1 Offset

Register	Offset
QTIMER1_TMR0_INPUT_SELECT_INPUT	63Ch

12.4.6.397.2 Function

DAISY Register

12.4.6.397.3 Diagram



12.4.6.397.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer1, In Pin: tmr0_input 00b - Selecting Pad: GPIO_EMC_B1_17 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_09 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_00 for Mode: ALT3

12.4.6.398 QTIMER1_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR1_INPUT_SELECT_INPUT)

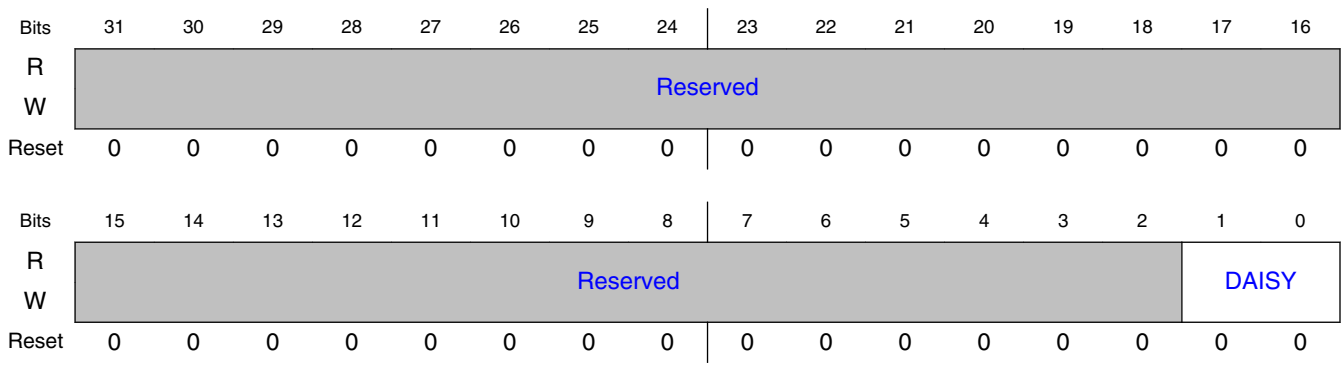
12.4.6.398.1 Offset

Register	Offset
QTIMER1_TMR1_INPUT_SELECT_INPUT	640h

12.4.6.398.2 Function

DAISY Register

12.4.6.398.3 Diagram



12.4.6.398.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer1, In Pin: tmr1_input 00b - Selecting Pad: GPIO_EMC_B1_38 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_10 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_01 for Mode: ALT3

12.4.6.399 QTIMER1_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER1_TMR2_INPUT_SELECT_INPUT)

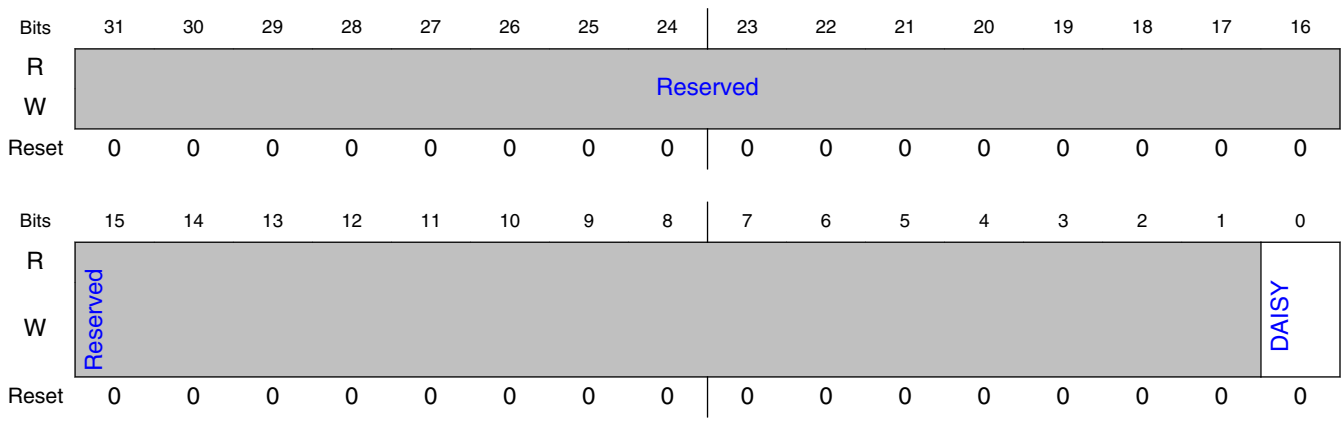
12.4.6.399.1 Offset

Register	Offset
QTIMER1_TMR2_INPUT_SELECT_INPUT	644h

12.4.6.399.2 Function

DAISY Register

12.4.6.399.3 Diagram



12.4.6.399.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer1, In Pin: tmr2_input 0b - Selecting Pad: GPIO_EMC_B2_11 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT3

12.4.6.400 QTIMER2_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR0_INPUT_SELECT_INPUT)

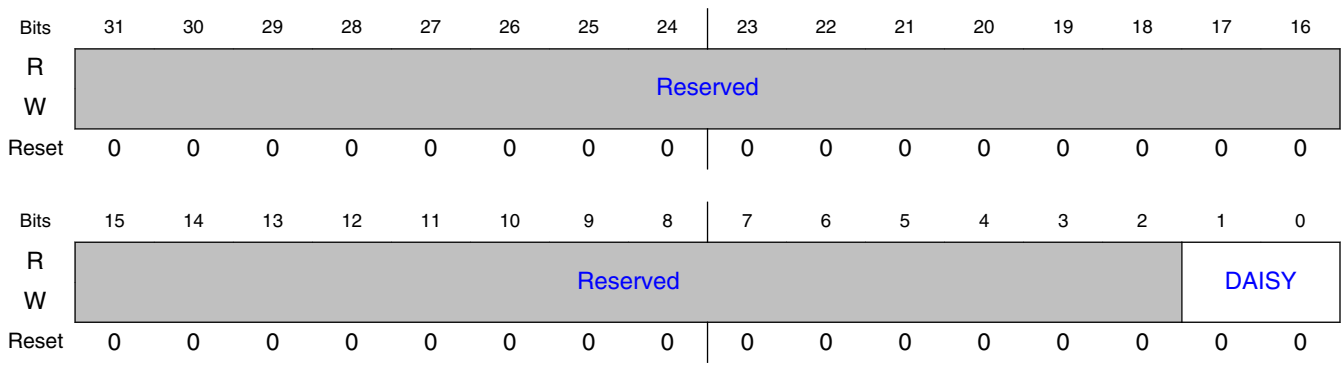
12.4.6.400.1 Offset

Register	Offset
QTIMER2_TMR0_INPUT_SELECT_INPUT	648h

12.4.6.400.2 Function

DAISY Register

12.4.6.400.3 Diagram



12.4.6.400.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer2, In Pin: tmr0_input 00b - Selecting Pad: GPIO_EMC_B1_18 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_13 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT3

12.4.6.401 QTIMER2_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR1_INPUT_SELECT_INPUT)

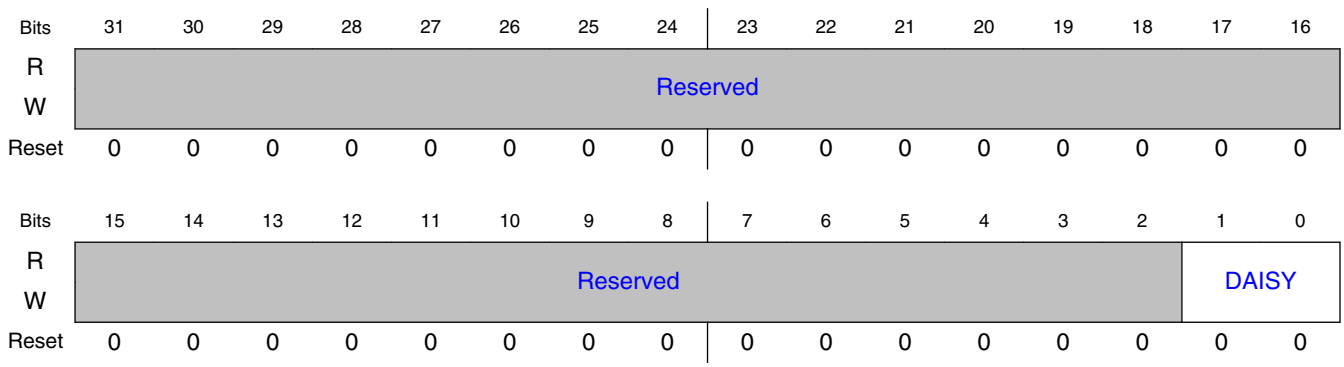
12.4.6.401.1 Offset

Register	Offset
QTIMER2_TMR1_INPUT_SELECT_INPUT	64Ch

12.4.6.401.2 Function

DAISY Register

12.4.6.401.3 Diagram



12.4.6.401.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer2, In Pin: tmr1_input 00b - Selecting Pad: GPIO_EMC_B1_39 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_14 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_04 for Mode: ALT3

12.4.6.402 QTIMER2_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER2_TMR2_INPUT_SELECT_INPUT)

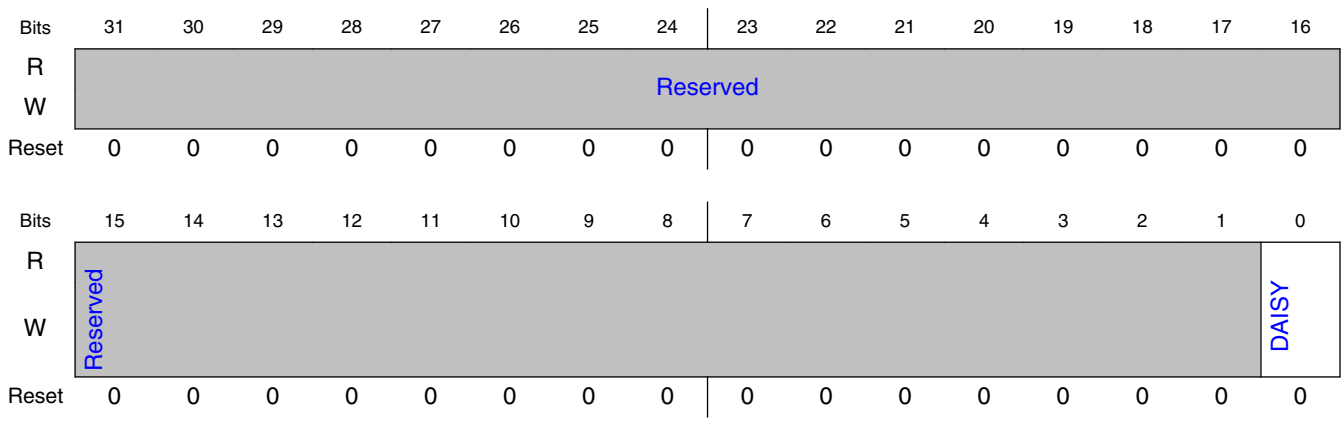
12.4.6.402.1 Offset

Register	Offset
QTIMER2_TMR2_INPUT_SELECT_INPUT	650h

12.4.6.402.2 Function

DAISY Register

12.4.6.402.3 Diagram



12.4.6.402.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer2, In Pin: tmr2_input 0b - Selecting Pad: GPIO_EMC_B2_15 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B1_05 for Mode: ALT3

12.4.6.403 QTIMER3_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR0_INPUT_SELECT_INPUT)

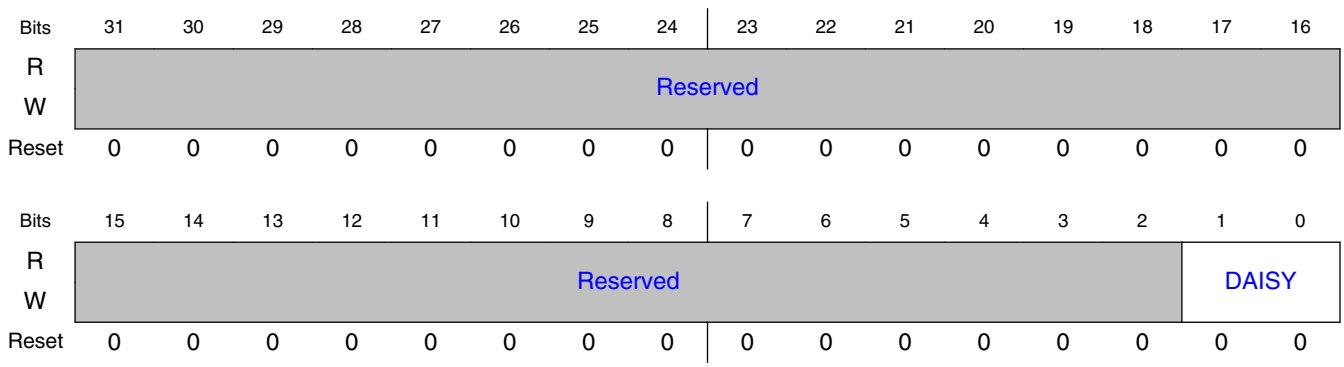
12.4.6.403.1 Offset

Register	Offset
QTIMER3_TMR0_INPUT_SELECT_INPUT	654h

12.4.6.403.2 Function

DAISY Register

12.4.6.403.3 Diagram



12.4.6.403.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer3, In Pin: tmr0_input 00b - Selecting Pad: GPIO_EMC_B1_19 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_17 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_06 for Mode: ALT3

12.4.6.404 QTIMER3_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR1_INPUT_SELECT_INPUT)

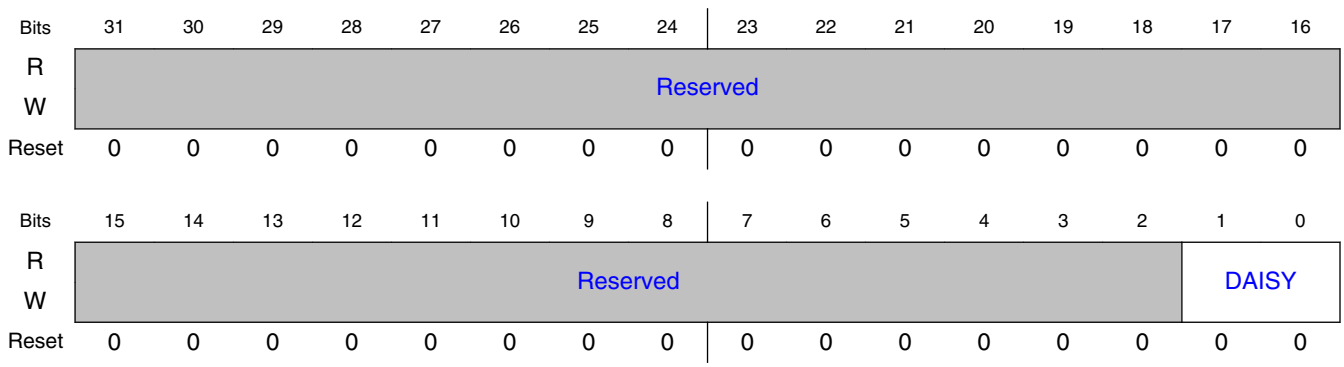
12.4.6.404.1 Offset

Register	Offset
QTIMER3_TMR1_INPUT_SELECT_INPUT	658h

12.4.6.404.2 Function

DAISY Register

12.4.6.404.3 Diagram



12.4.6.404.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer3, In Pin: tmr1_input 00b - Selecting Pad: GPIO_EMC_B2_00 for Mode: ALT2 01b - Selecting Pad: GPIO_EMC_B2_18 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_07 for Mode: ALT3

12.4.6.405 QTIMER3_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER3_TMR2_INPUT_SELECT_INPUT)

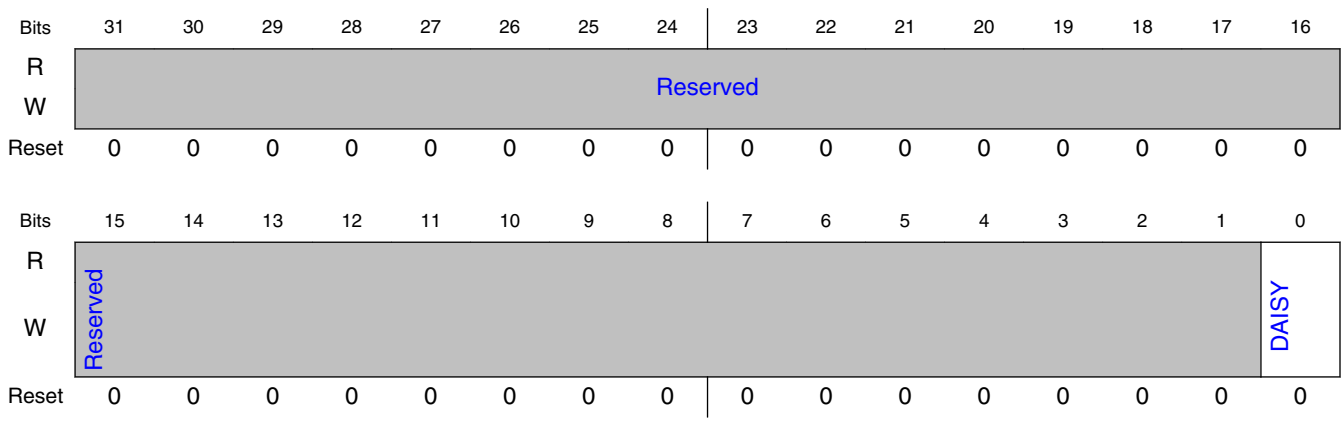
12.4.6.405.1 Offset

Register	Offset
QTIMER3_TMR2_INPUT_SELECT_INPUT	65Ch

12.4.6.405.2 Function

DAISY Register

12.4.6.405.3 Diagram



12.4.6.405.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer3, In Pin: tmr2_input 0b - Selecting Pad: GPIO_EMC_B2_19 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B1_08 for Mode: ALT3

12.4.6.406 QTIMER4_TMR0_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR0_INPUT_SELECT_INPUT)

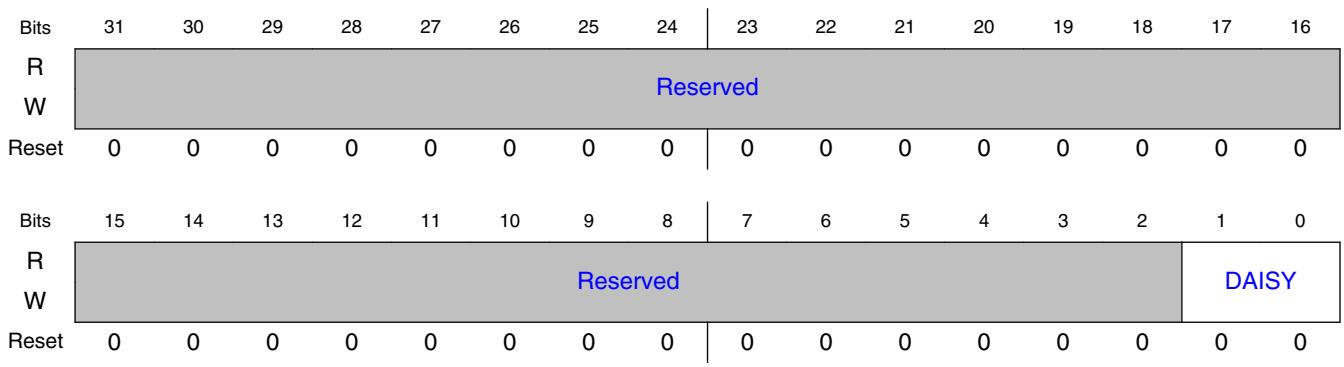
12.4.6.406.1 Offset

Register	Offset
QTIMER4_TMR0_INPUT_SELECT_INPUT	660h

12.4.6.406.2 Function

DAISY Register

12.4.6.406.3 Diagram



12.4.6.406.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer4, In Pin: tmr0_input 00b - Selecting Pad: GPIO_EMC_B1_20 for Mode: ALT2 01b - Selecting Pad: GPIO_AD_04 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_09 for Mode: ALT3

12.4.6.407 QTIMER4_TMR1_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR1_INPUT_SELECT_INPUT)

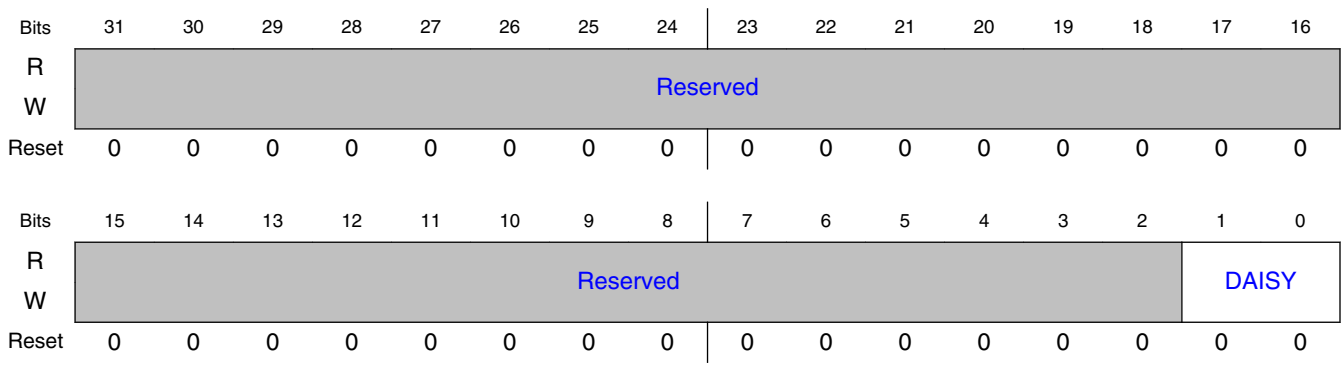
12.4.6.407.1 Offset

Register	Offset
QTIMER4_TMR1_INPUT_SELECT_INPUT	664h

12.4.6.407.2 Function

DAISY Register

12.4.6.407.3 Diagram



12.4.6.407.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer4, In Pin: tmr1_input 00b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT2 01b - Selecting Pad: GPIO_AD_05 for Mode: ALT9 10b - Selecting Pad: GPIO_DISP_B1_10 for Mode: ALT3

12.4.6.408 QTIMER4_TMR2_INPUT_SELECT_INPUT DAISY Register (QTIMER4_TMR2_INPUT_SELECT_INPUT)

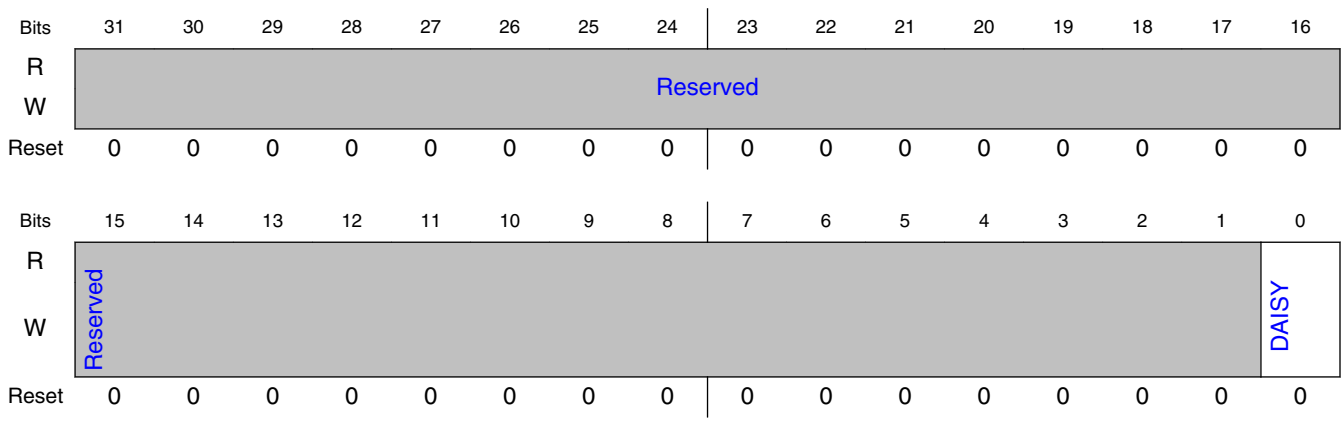
12.4.6.408.1 Offset

Register	Offset
QTIMER4_TMR2_INPUT_SELECT_INPUT	668h

12.4.6.408.2 Function

DAISY Register

12.4.6.408.3 Diagram



12.4.6.408.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: qtimer4, In Pin: tmr2_input 0b - Selecting Pad: GPIO_AD_06 for Mode: ALT9 1b - Selecting Pad: GPIO_DISP_B1_11 for Mode: ALT3

12.4.6.409 SAI1_IPG_CLK_SAI_MCLK_SELECT_INPUT DAISY Register (SAI1_IPG_CLK_SAI_MCLK_SELECT_INPUT)

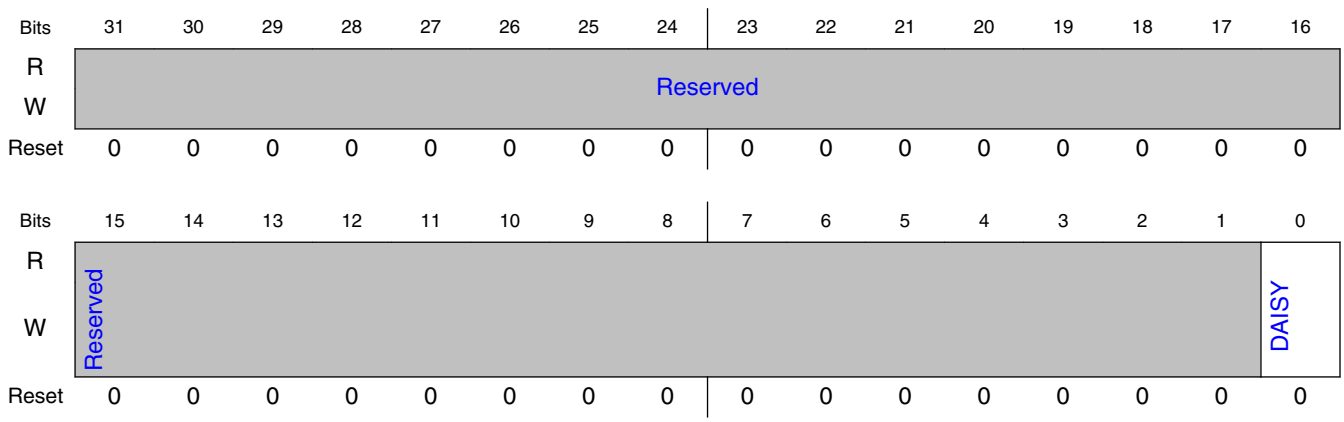
12.4.6.409.1 Offset

Register	Offset
SAI1_IPG_CLK_SAI_MCLK_SELECT_INPUT	66Ch

12.4.6.409.2 Function

DAISY Register

12.4.6.409.3 Diagram



12.4.6.409.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: ipg_clk_sai_mclk 0b - Selecting Pad: GPIO_AD_17 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_03 for Mode: ALT4

12.4.6.410 SAI1_SAI_RXBCLK_SELECT_INPUT DAISY Register (SAI1_SAI_RXBCLK_SELECT_INPUT)

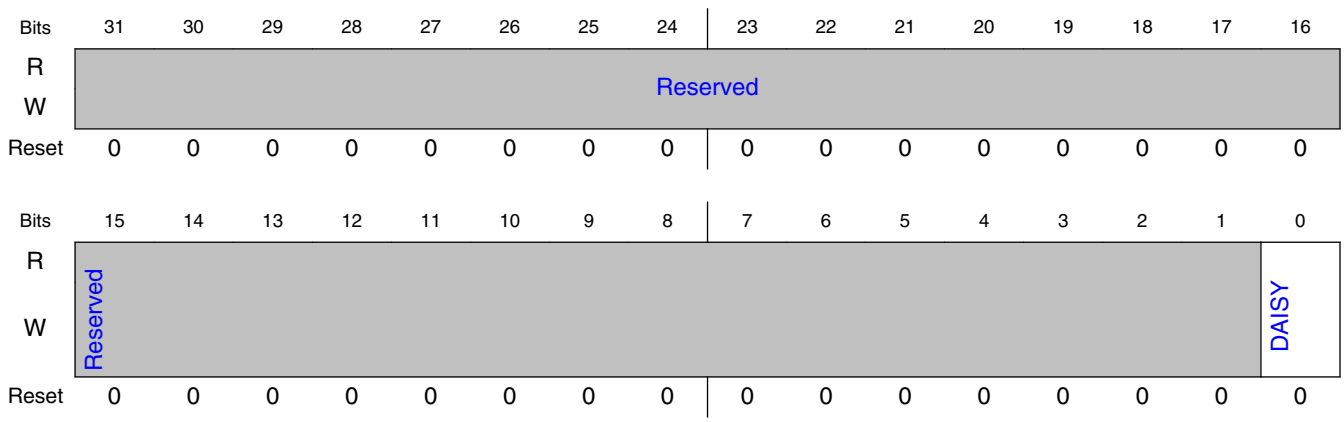
12.4.6.410.1 Offset

Register	Offset
SAI1_SAI_RXBCLK_SELECT_INPUT	670h

12.4.6.410.2 Function

DAISY Register

12.4.6.410.3 Diagram



12.4.6.410.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: sai_rxbclk 0b - Selecting Pad: GPIO_AD_19 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_05 for Mode: ALT4

12.4.6.411 SAI1_SAI_RXDATA_SELECT_INPUT_0 DAISY Register (SAI1_SAI_RXDATA_SELECT_INPUT_0)

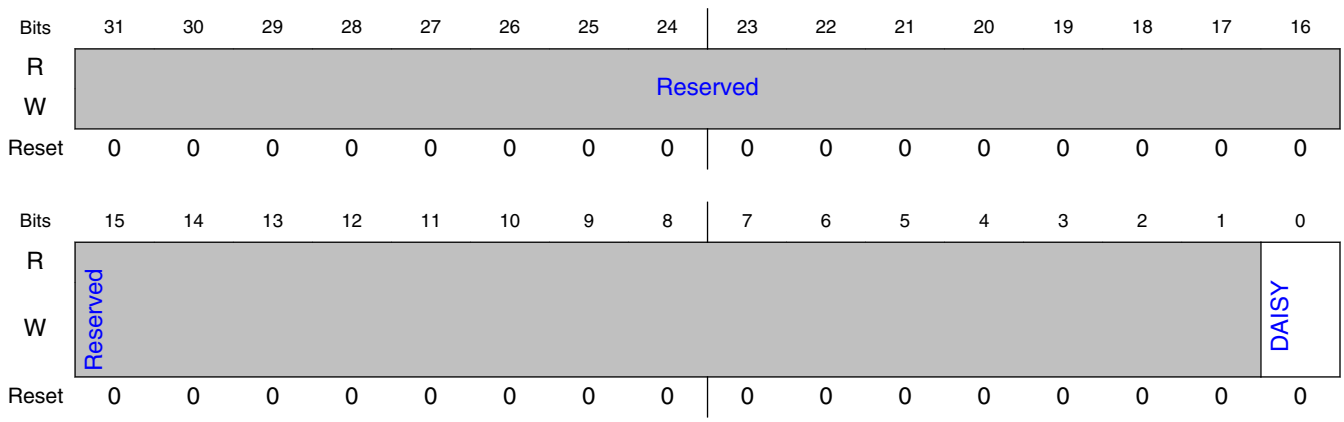
12.4.6.411.1 Offset

Register	Offset
SAI1_SAI_RXDATA_SELECT_INPUT_0	674h

12.4.6.411.2 Function

DAISY Register

12.4.6.411.3 Diagram



12.4.6.411.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: sai_rxdata0 0b - Selecting Pad: GPIO_AD_20 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_06 for Mode: ALT4

12.4.6.412 SAI1_SAI_RXSYNC_SELECT_INPUT DAISY Register (SAI1_SAI_RXSYNC_SELECT_INPUT)

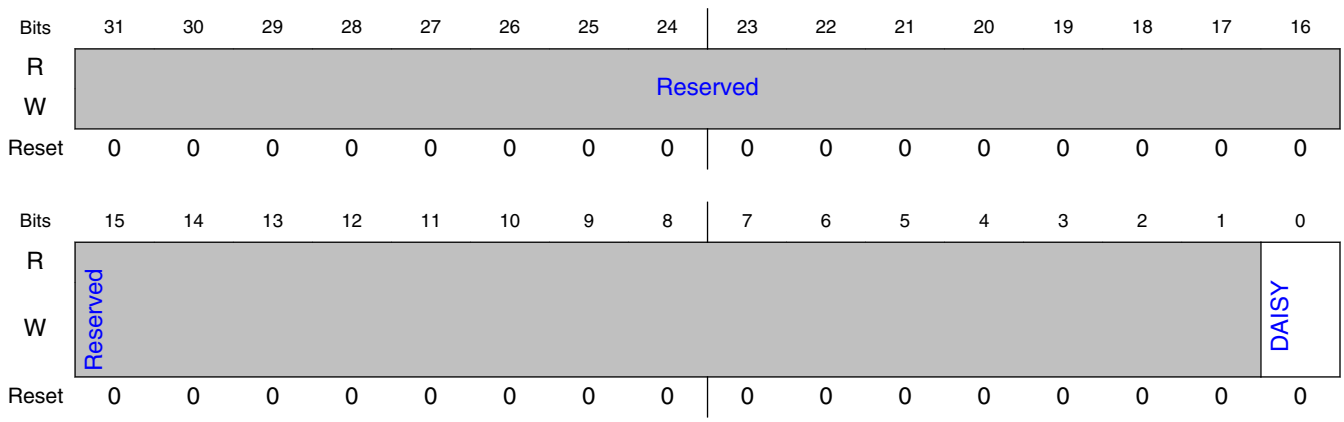
12.4.6.412.1 Offset

Register	Offset
SAI1_SAI_RXSYNC_SELECT_INPUT	678h

12.4.6.412.2 Function

DAISY Register

12.4.6.412.3 Diagram



12.4.6.412.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: sai_rxsync 0b - Selecting Pad: GPIO_AD_18 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_04 for Mode: ALT4

12.4.6.413 SAI1_SAI_TXBCLK_SELECT_INPUT DAISY Register (SAI1_SAI_TXBCLK_SELECT_INPUT)

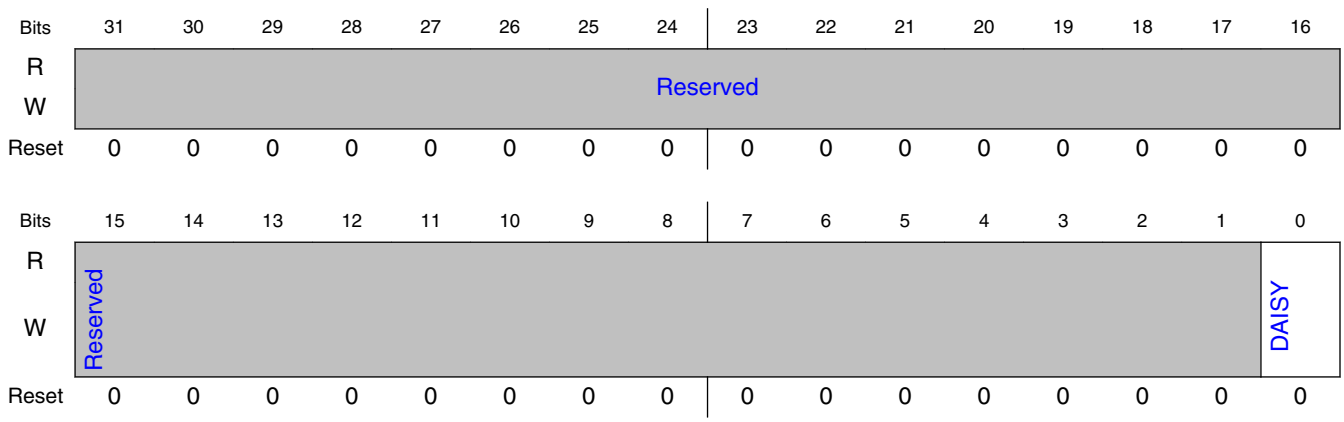
12.4.6.413.1 Offset

Register	Offset
SAI1_SAI_TXBCLK_SELECT_INPUT	67Ch

12.4.6.413.2 Function

DAISY Register

12.4.6.413.3 Diagram



12.4.6.413.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: sai_txbclk 0b - Selecting Pad: GPIO_AD_22 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_08 for Mode: ALT4

12.4.6.414 SAI1_SAI_TXSYNC_SELECT_INPUT DAISY Register (SAI1_SAI_TXSYNC_SELECT_INPUT)

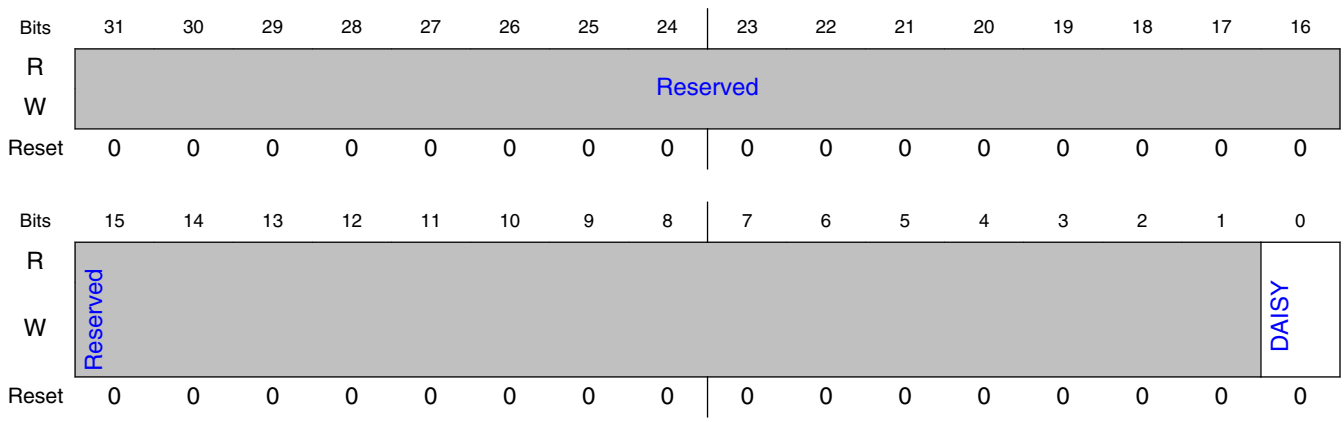
12.4.6.414.1 Offset

Register	Offset
SAI1_SAI_TXSYNC_SELECT_INPUT	680h

12.4.6.414.2 Function

DAISY Register

12.4.6.414.3 Diagram



12.4.6.414.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: sai_txsync 0b - Selecting Pad: GPIO_AD_23 for Mode: ALT0 1b - Selecting Pad: GPIO_DISP_B2_09 for Mode: ALT4

12.4.6.415 EMVSIM1_SIO_SELECT_INPUT DAISY Register (EMVSIM1_SIO_SELECT_INPUT)

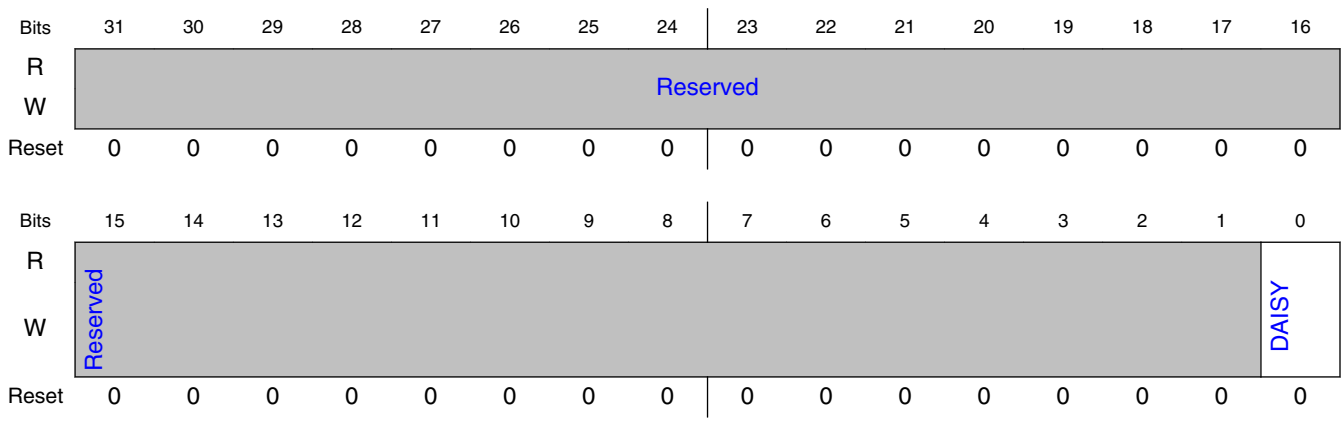
12.4.6.415.1 Offset

Register	Offset
EMVSIM1_SIO_SELECT_INPUT	69Ch

12.4.6.415.2 Function

DAISY Register

12.4.6.415.3 Diagram



12.4.6.415.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM1, In Pin: sio 0b - Selecting Pad: GPIO_EMC_B2_11 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_00 for Mode: ALT0

12.4.6.416 EMVSIM1_IPP_SIMPD_SELECT_INPUT DAISY Register (EMVSIM1_IPP_SIMPD_SELECT_INPUT)

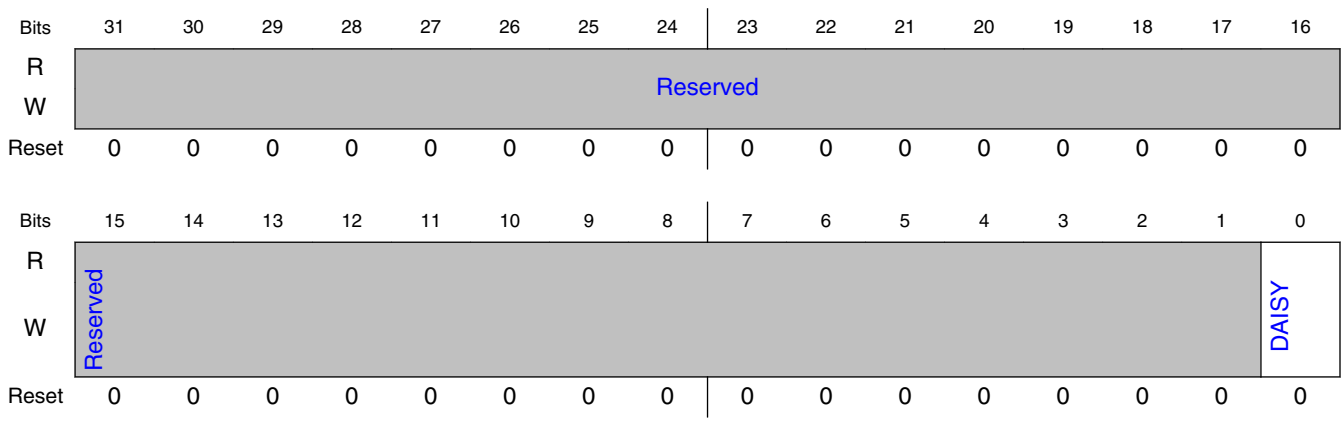
12.4.6.416.1 Offset

Register	Offset
EMVSIM1_IPP_SIMPD_SELECT_INPUT	6A0h

12.4.6.416.2 Function

DAISY Register

12.4.6.416.3 Diagram



12.4.6.416.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM1, In Pin: ipp_simpd 0b - Selecting Pad: GPIO_EMC_B2_15 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_04 for Mode: ALT0

12.4.6.417 EMVSIM1_POWER_FAIL_SELECT_INPUT DAISY Register (EMVSIM1_POWER_FAIL_SELECT_INPUT)

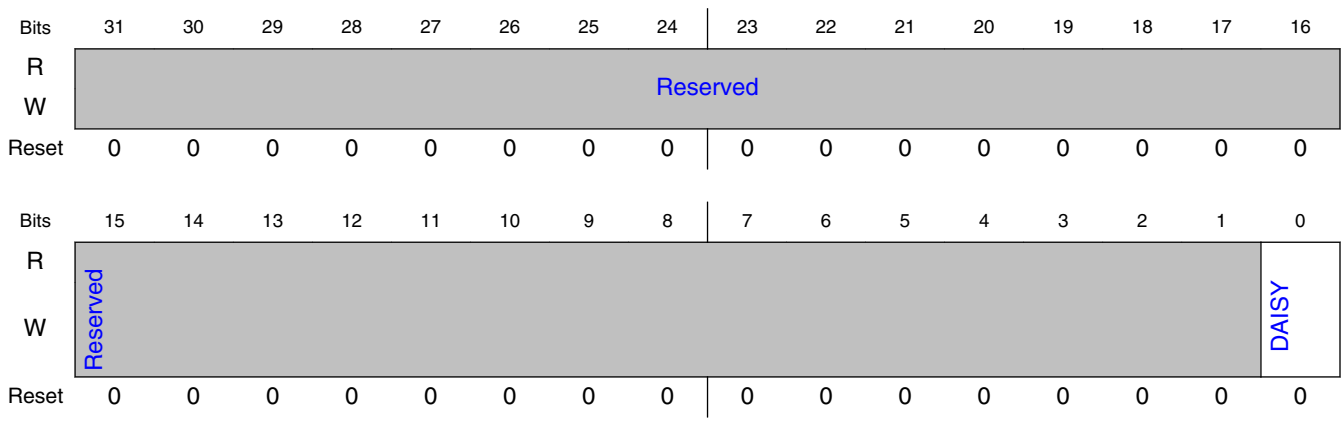
12.4.6.417.1 Offset

Register	Offset
EMVSIM1_POWER_FAIL_SELECT_INPUT	6A4h

12.4.6.417.2 Function

DAISY Register

12.4.6.417.3 Diagram



12.4.6.417.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM1, In Pin: power_fail 0b - Selecting Pad: GPIO_EMC_B2_16 for Mode: ALT8 1b - Selecting Pad: GPIO_AD_05 for Mode: ALT0

12.4.6.418 EMVSIM2_SIO_SELECT_INPUT DAISY Register (EMVSIM2_SIO_SELECT_INPUT)

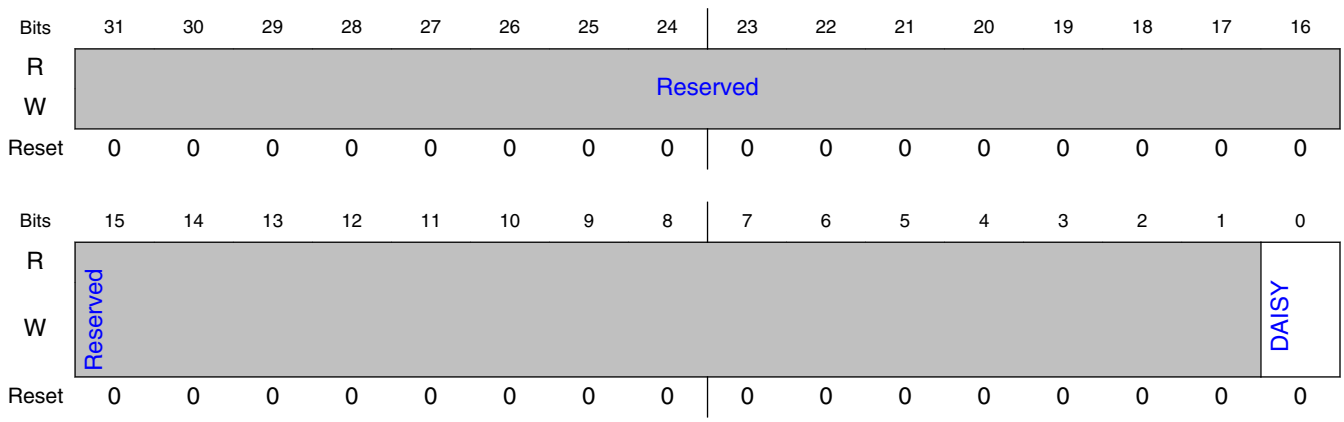
12.4.6.418.1 Offset

Register	Offset
EMVSIM2_SIO_SELECT_INPUT	6A8h

12.4.6.418.2 Function

DAISY Register

12.4.6.418.3 Diagram



12.4.6.418.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM2, In Pin: sio 0b - Selecting Pad: GPIO_AD_06 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B2_10 for Mode: ALT1

12.4.6.419 EMVSIM2_IPP_SIMPD_SELECT_INPUT DAISY Register (EMVSIM2_IPP_SIMPD_SELECT_INPUT)

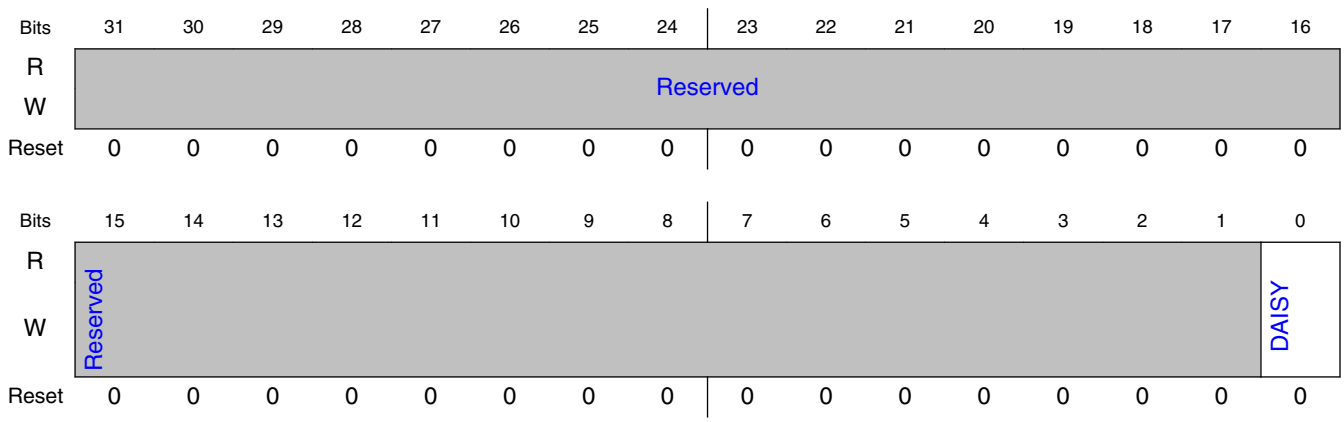
12.4.6.419.1 Offset

Register	Offset
EMVSIM2_IPP_SIMPD_SELECT_INPUT	6ACh

12.4.6.419.2 Function

DAISY Register

12.4.6.419.3 Diagram



12.4.6.419.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM2, In Pin: ipp_simpd 0b - Selecting Pad: GPIO_AD_10 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B2_14 for Mode: ALT1

12.4.6.420 EMVSIM2_POWER_FAIL_SELECT_INPUT DAISY Register (EMVSIM2_POWER_FAIL_SELECT_INPUT)

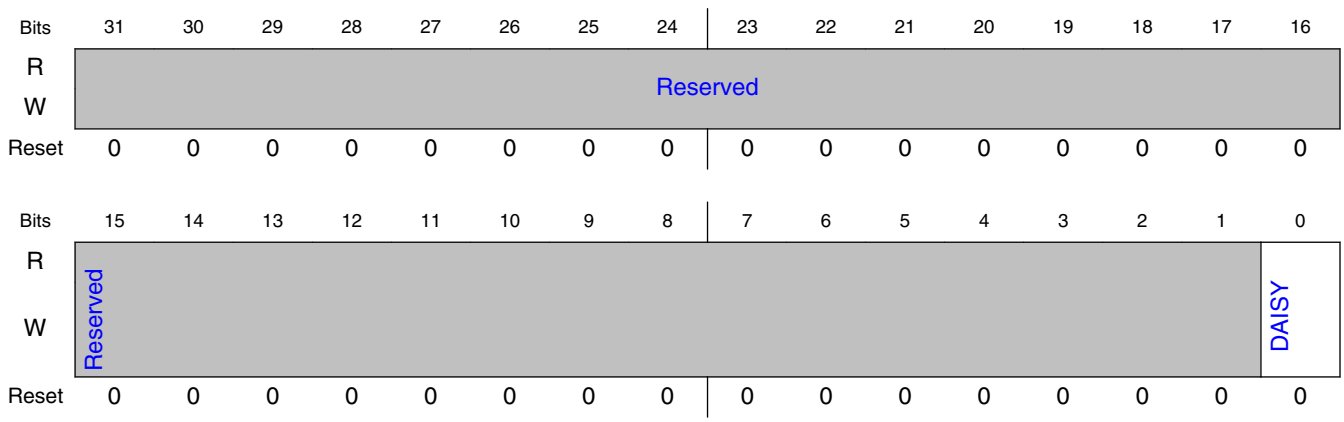
12.4.6.420.1 Offset

Register	Offset
EMVSIM2_POWER_FAIL_SELECT_INPUT	6B0h

12.4.6.420.2 Function

DAISY Register

12.4.6.420.3 Diagram



12.4.6.420.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: EMVSIM2, In Pin: power_fail 0b - Selecting Pad: GPIO_AD_11 for Mode: ALT2 1b - Selecting Pad: GPIO_DISP_B2_15 for Mode: ALT1

12.4.6.421 SPDIF_SPDIF_IN1_SELECT_INPUT DAISY Register (SPDIF_SPDIF_IN1_SELECT_INPUT)

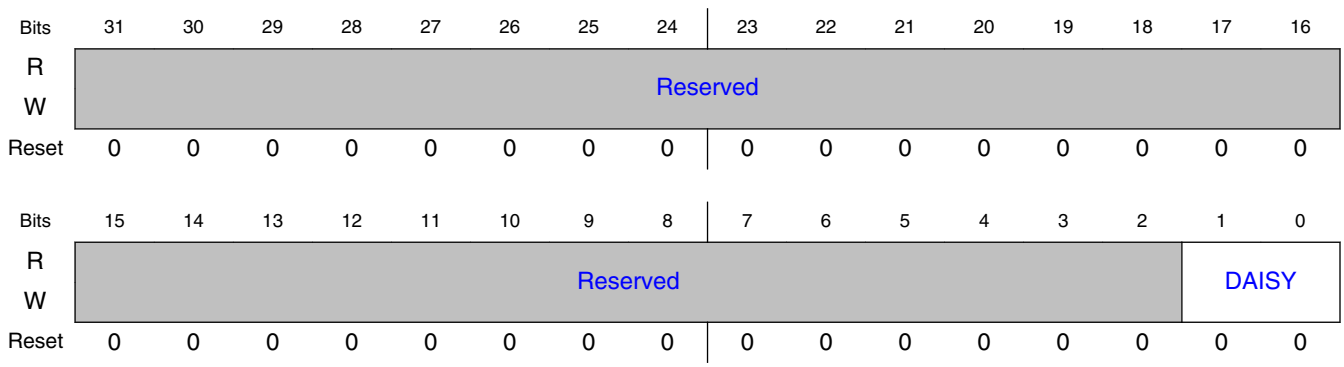
12.4.6.421.1 Offset

Register	Offset
SPDIF_SPDIF_IN1_SELECT_INPUT	6B4h

12.4.6.421.2 Function

DAISY Register

12.4.6.421.3 Diagram



12.4.6.421.4 Fields

Field	Function
31-2 —	- Reserved
1-0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: spdif, In Pin: spdif_in1 00b - Selecting Pad: GPIO_EMC_B2_11 for Mode: ALT1 01b - Selecting Pad: GPIO_AD_15 for Mode: ALT0 10b - Selecting Pad: GPIO_DISP_B2_10 for Mode: ALT9

12.4.6.422 USB_OTG2_OC_SELECT_INPUT DAISY Register (USB_OTG2_OC_SELECT_INPUT)

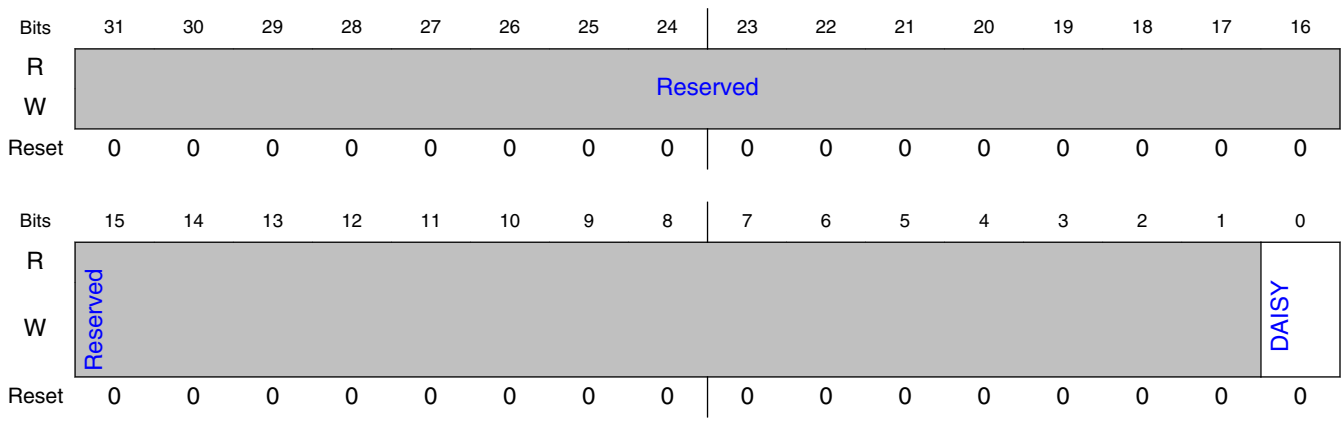
12.4.6.422.1 Offset

Register	Offset
USB_OTG2_OC_SELECT_INPUT	6B8h

12.4.6.422.2 Function

DAISY Register

12.4.6.422.3 Diagram



12.4.6.422.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usb, In Pin: otg2_oc 0b - Selecting Pad: GPIO_AD_06 for Mode: ALT0 1b - Selecting Pad: GPIO_AD_30 for Mode: ALT1

12.4.6.423 USB_OTG_OC_SELECT_INPUT DAISY Register (USB_OTG_OC_SELECT_INPUT)

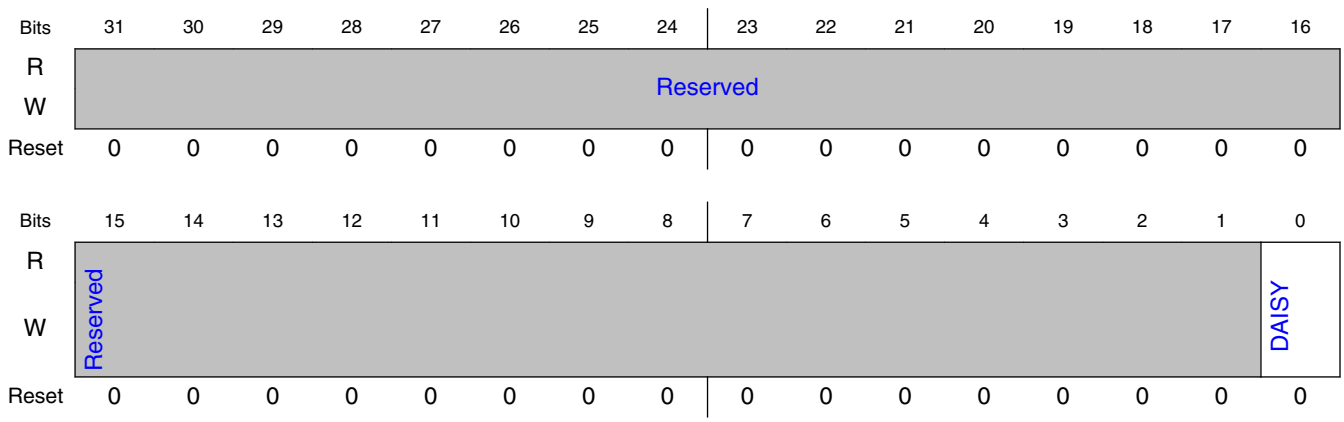
12.4.6.423.1 Offset

Register	Offset
USB_OTG_OC_SELECT_INPUT	6BCh

12.4.6.423.2 Function

DAISY Register

12.4.6.423.3 Diagram



12.4.6.423.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usb, In Pin: otg_oc 0b - Selecting Pad: GPIO_AD_11 for Mode: ALT0 1b - Selecting Pad: GPIO_AD_35 for Mode: ALT1

12.4.6.424 USBPHY1_USB_ID_SELECT_INPUT DAISY Register (USBPHY1_USB_ID_SELECT_INPUT)

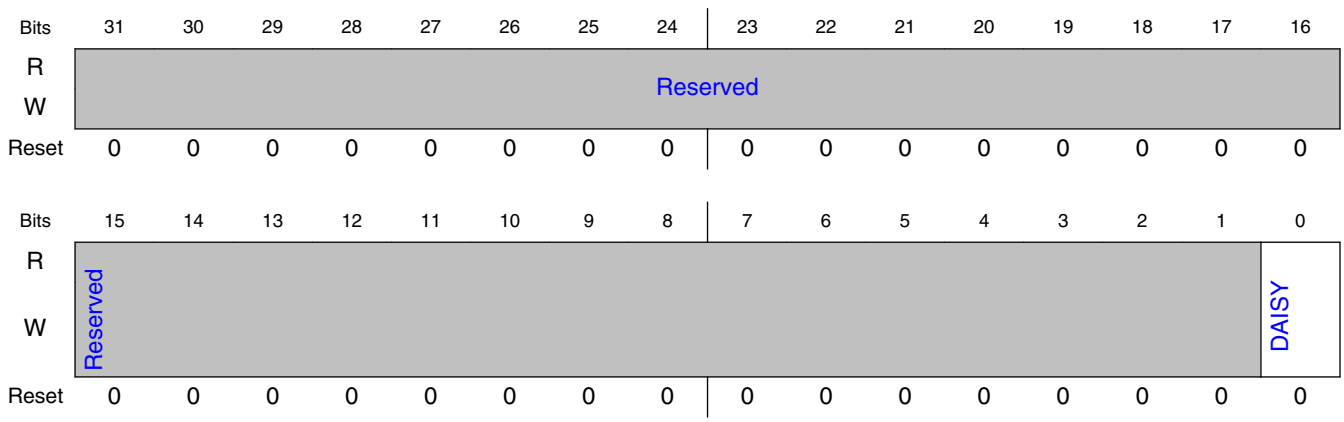
12.4.6.424.1 Offset

Register	Offset
USBPHY1_USB_ID_SELECT_INPUT	6C0h

12.4.6.424.2 Function

DAISY Register

12.4.6.424.3 Diagram



12.4.6.424.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usbphy1, In Pin: usb_id 0b - Selecting Pad: GPIO_AD_09 for Mode: ALT0 1b - Selecting Pad: GPIO_AD_33 for Mode: ALT1

12.4.6.425 USBPHY2_USB_ID_SELECT_INPUT DAISY Register (USBPHY2_USB_ID_SELECT_INPUT)

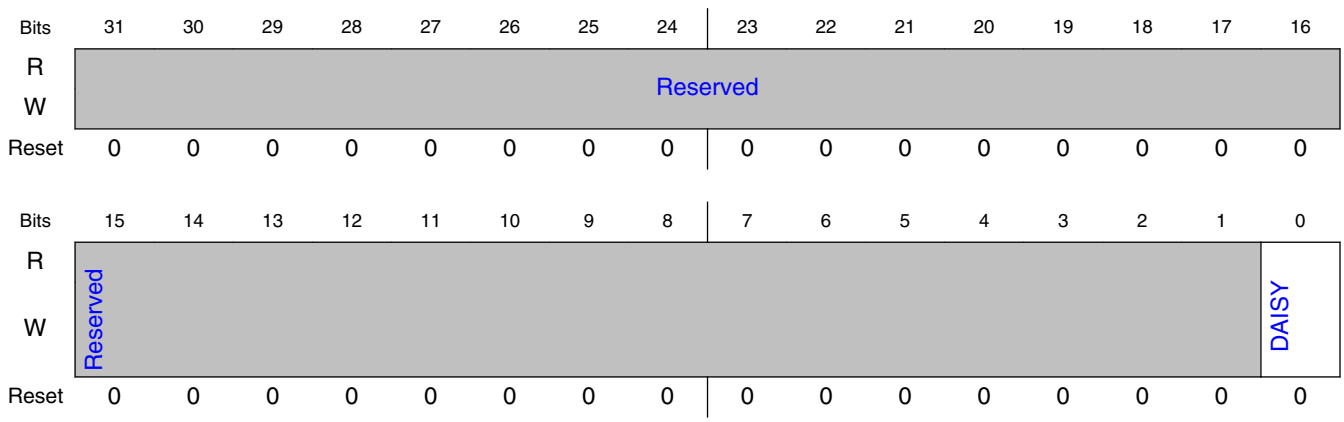
12.4.6.425.1 Offset

Register	Offset
USBPHY2_USB_ID_SELECT_INPUT	6C4h

12.4.6.425.2 Function

DAISY Register

12.4.6.425.3 Diagram



12.4.6.425.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usbphy2, In Pin: usb_id 0b - Selecting Pad: GPIO_AD_08 for Mode: ALT0 1b - Selecting Pad: GPIO_AD_32 for Mode: ALT1

12.4.6.426 USDHC1_IPP_CARD_DET_SELECT_INPUT DAISY Register (USDHC1_IPP_CARD_DET_SELECT_INPUT)

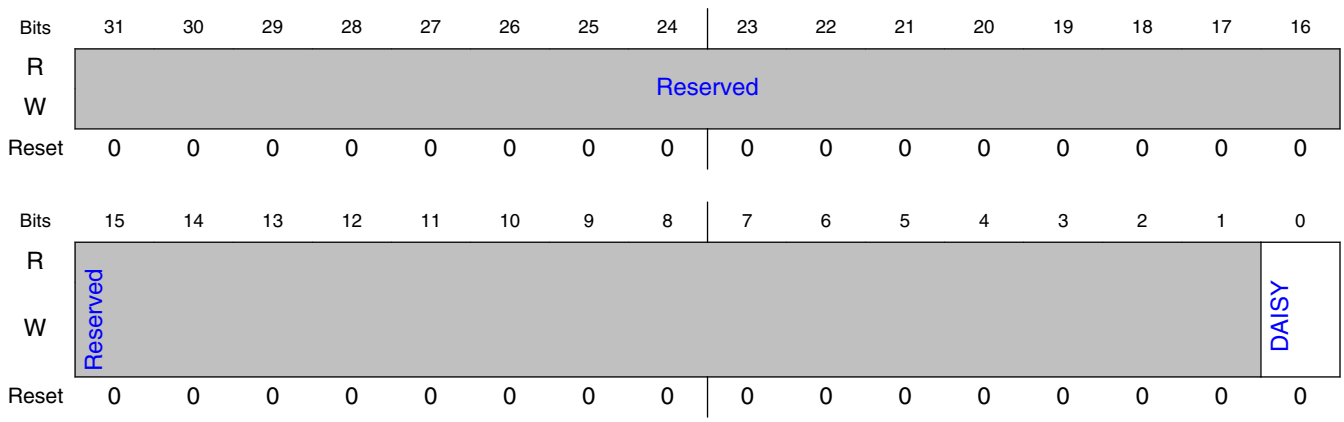
12.4.6.426.1 Offset

Register	Offset
USDHC1_IPP_CARD_DET_SELECT_INPUT	6C8h

12.4.6.426.2 Function

DAISY Register

12.4.6.426.3 Diagram



12.4.6.426.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc1, In Pin: ipp_card_det 0b - Selecting Pad: GPIO_AD_32 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B1_08 for Mode: ALT2

12.4.6.427 USDHC1_IPP_WP_ON_SELECT_INPUT DAISY Register (USDHC1_IPP_WP_ON_SELECT_INPUT)

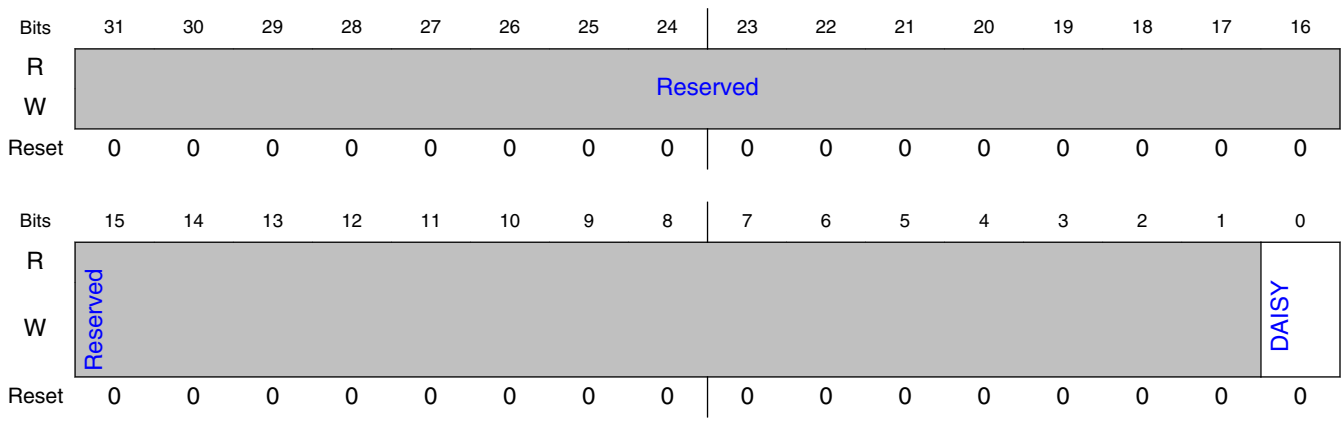
12.4.6.427.1 Offset

Register	Offset
USDHC1_IPP_WP_ON_SELECT_INPUT	6CCh

12.4.6.427.2 Function

DAISY Register

12.4.6.427.3 Diagram



12.4.6.427.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc1, In Pin: ipp_wp_on 0b - Selecting Pad: GPIO_AD_33 for Mode: ALT4 1b - Selecting Pad: GPIO_DISP_B1_09 for Mode: ALT2

12.4.6.428 USDHC2_IPP_CARD_DET_SELECT_INPUT DAISY Register (USDHC2_IPP_CARD_DET_SELECT_INPUT)

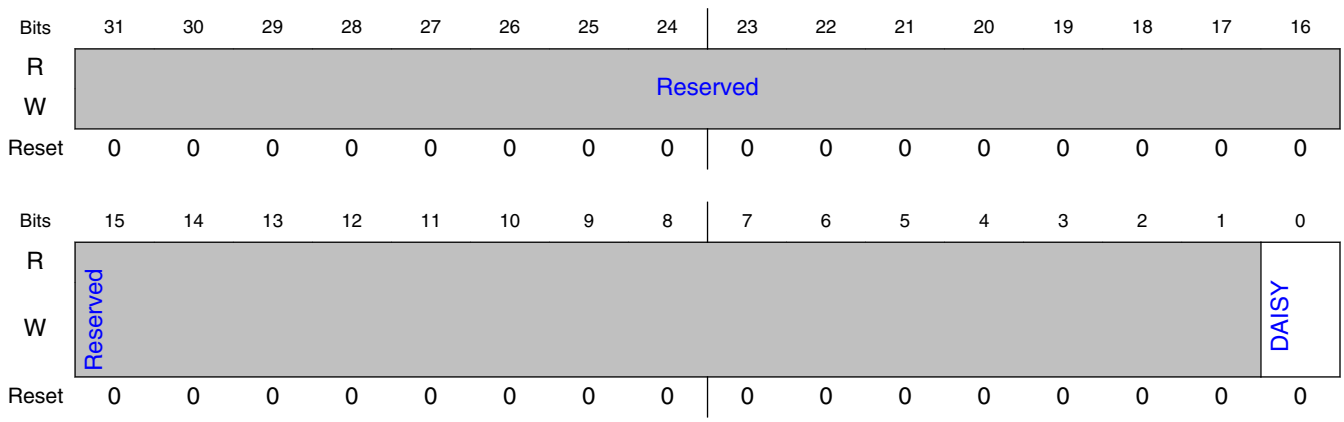
12.4.6.428.1 Offset

Register	Offset
USDHC2_IPP_CARD_DET_SELECT_INPUT	6D0h

12.4.6.428.2 Function

DAISY Register

12.4.6.428.3 Diagram



12.4.6.428.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc2, In Pin: ipp_card_det 0b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_26 for Mode: ALT11

12.4.6.429 USDHC2_IPP_WP_ON_SELECT_INPUT DAISY Register (USDHC2_IPP_WP_ON_SELECT_INPUT)

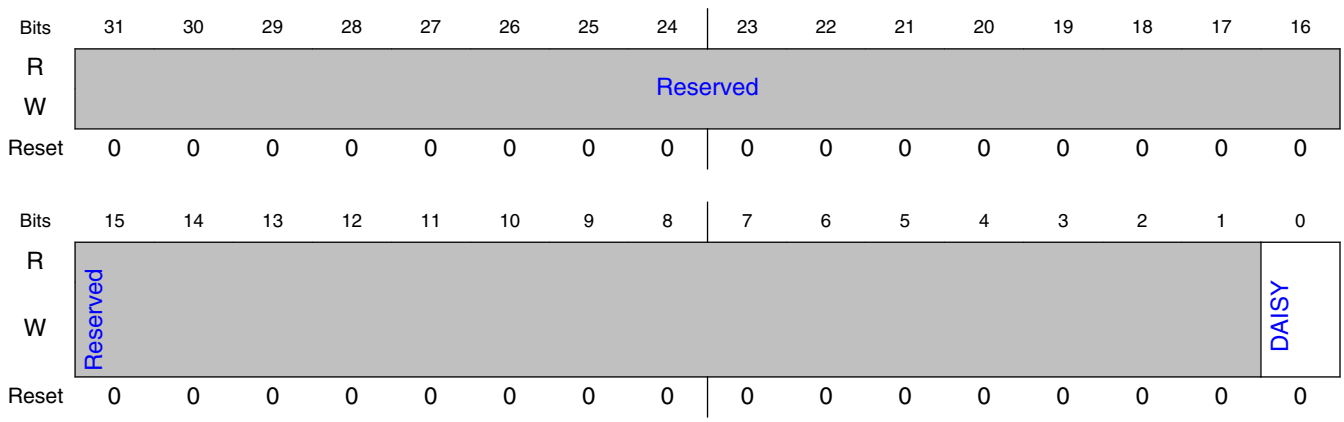
12.4.6.429.1 Offset

Register	Offset
USDHC2_IPP_WP_ON_SELECT_INPUT	6D4h

12.4.6.429.2 Function

DAISY Register

12.4.6.429.3 Diagram



12.4.6.429.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usdhc2, In Pin: ipp_wp_on 0b - Selecting Pad: GPIO_EMC_B2_02 for Mode: ALT1 1b - Selecting Pad: GPIO_AD_27 for Mode: ALT11

12.4.6.430 XBAR1_IN_SELECT_INPUT_20 DAISY Register (XBAR1_IN_SELECT_INPUT_20)

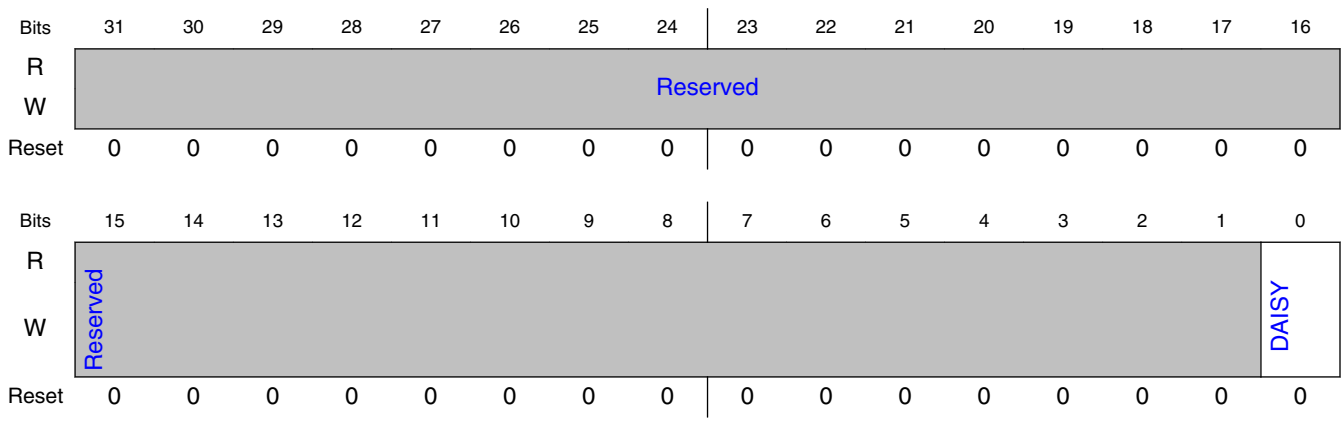
12.4.6.430.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_20	6D8h

12.4.6.430.2 Function

DAISY Register

12.4.6.430.3 Diagram



12.4.6.430.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in20 0b - Selecting Pad: GPIO_EMC_B2_00 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_00 for Mode: ALT2

12.4.6.431 XBAR1_IN_SELECT_INPUT_21 DAISY Register (XBAR1_IN_SELECT_INPUT_21)

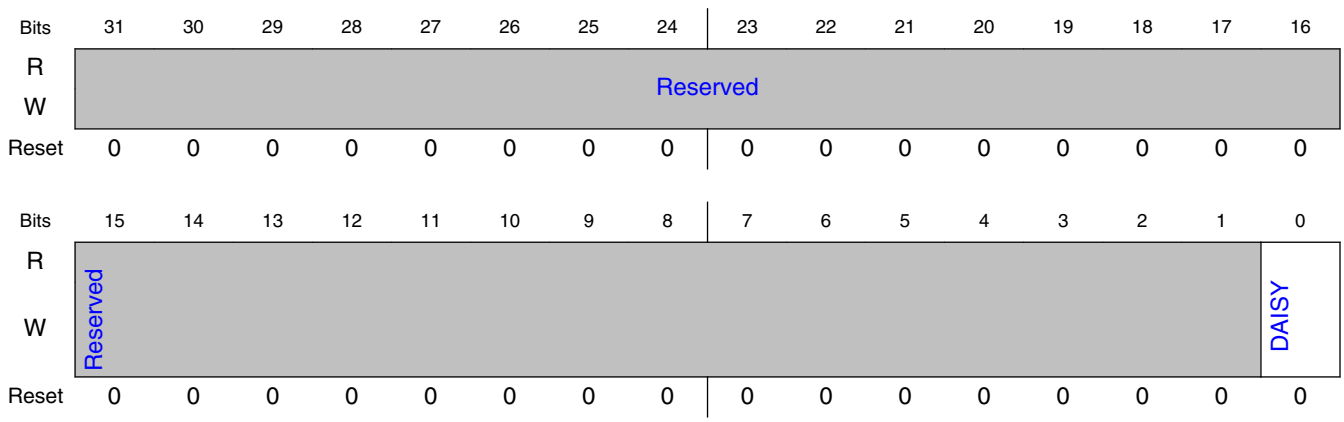
12.4.6.431.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_21	6DCh

12.4.6.431.2 Function

DAISY Register

12.4.6.431.3 Diagram



12.4.6.431.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in21 0b - Selecting Pad: GPIO_EMC_B2_01 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_01 for Mode: ALT2

12.4.6.432 XBAR1_IN_SELECT_INPUT_22 DAISY Register (XBAR1_IN_SELECT_INPUT_22)

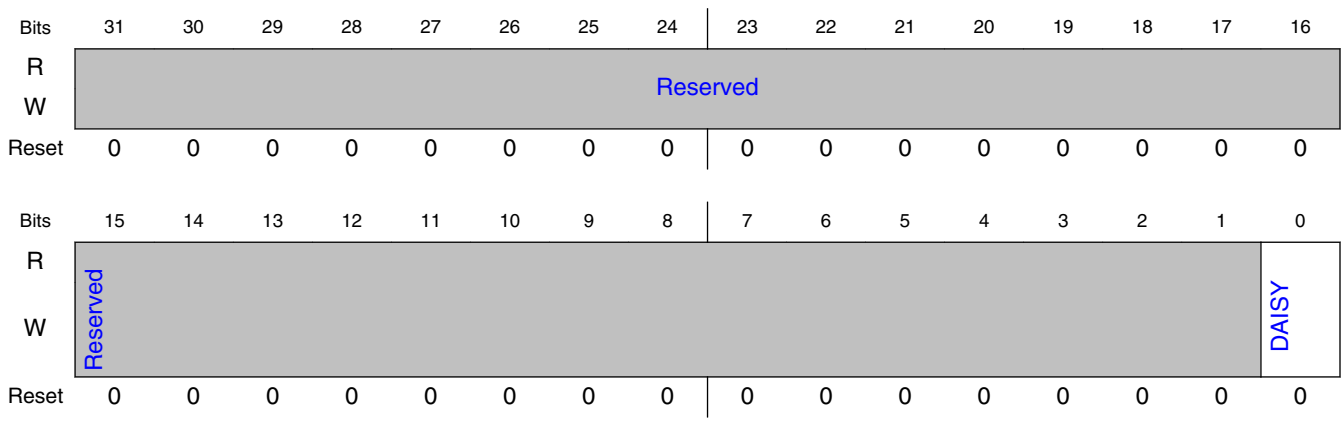
12.4.6.432.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_22	6E0h

12.4.6.432.2 Function

DAISY Register

12.4.6.432.3 Diagram



12.4.6.432.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in22 0b - Selecting Pad: GPIO_EMC_B2_02 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_02 for Mode: ALT2

12.4.6.433 XBAR1_IN_SELECT_INPUT_23 DAISY Register (XBAR1_IN_SELECT_INPUT_23)

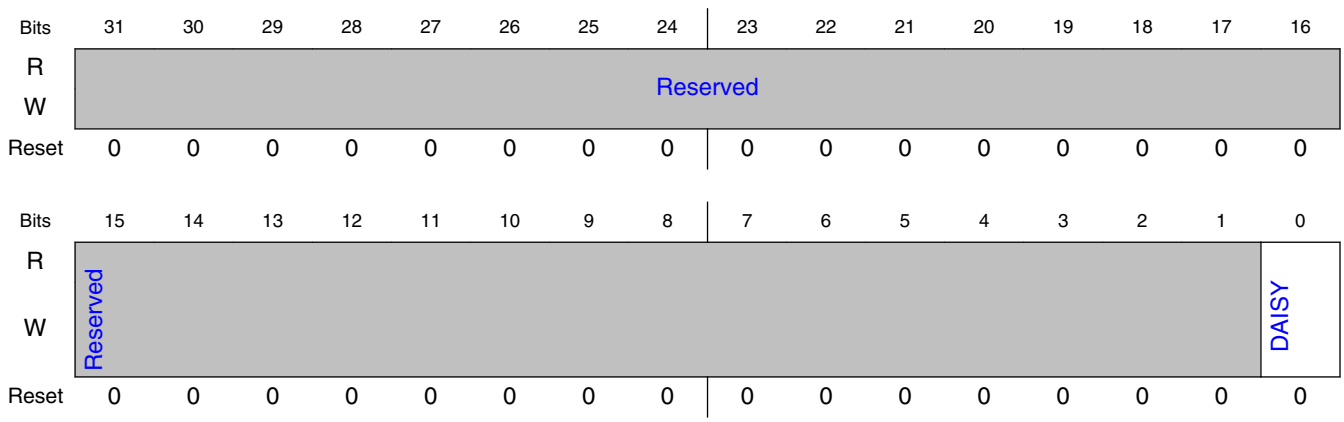
12.4.6.433.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_23	6E4h

12.4.6.433.2 Function

DAISY Register

12.4.6.433.3 Diagram



12.4.6.433.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in23 0b - Selecting Pad: GPIO_EMC_B2_03 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_03 for Mode: ALT2

12.4.6.434 XBAR1_IN_SELECT_INPUT_24 DAISY Register (XBAR1_IN_SELECT_INPUT_24)

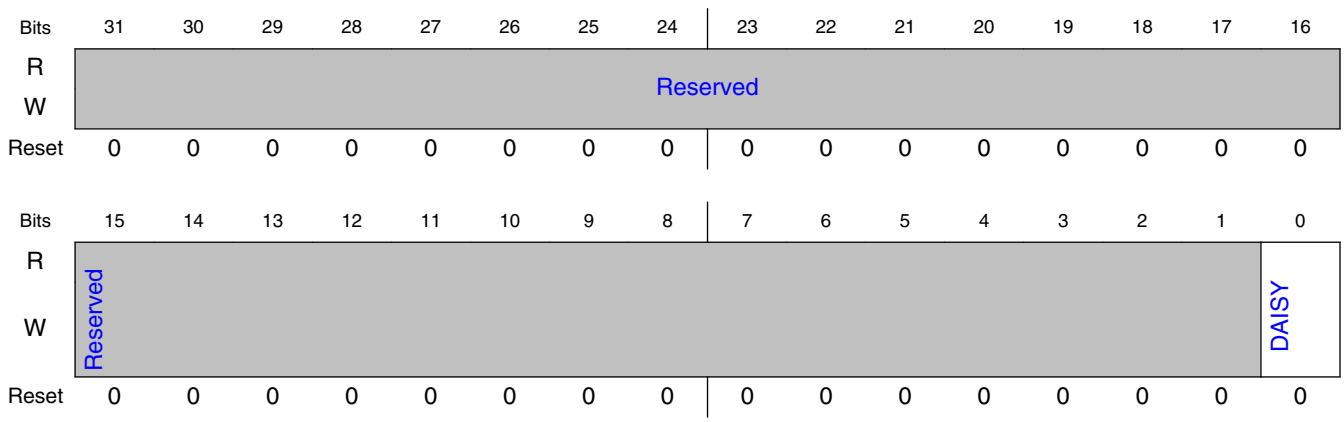
12.4.6.434.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_24	6E8h

12.4.6.434.2 Function

DAISY Register

12.4.6.434.3 Diagram



12.4.6.434.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in24 0b - Selecting Pad: GPIO_EMC_B2_04 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_04 for Mode: ALT2

12.4.6.435 XBAR1_IN_SELECT_INPUT_25 DAISY Register (XBAR1_IN_SELECT_INPUT_25)

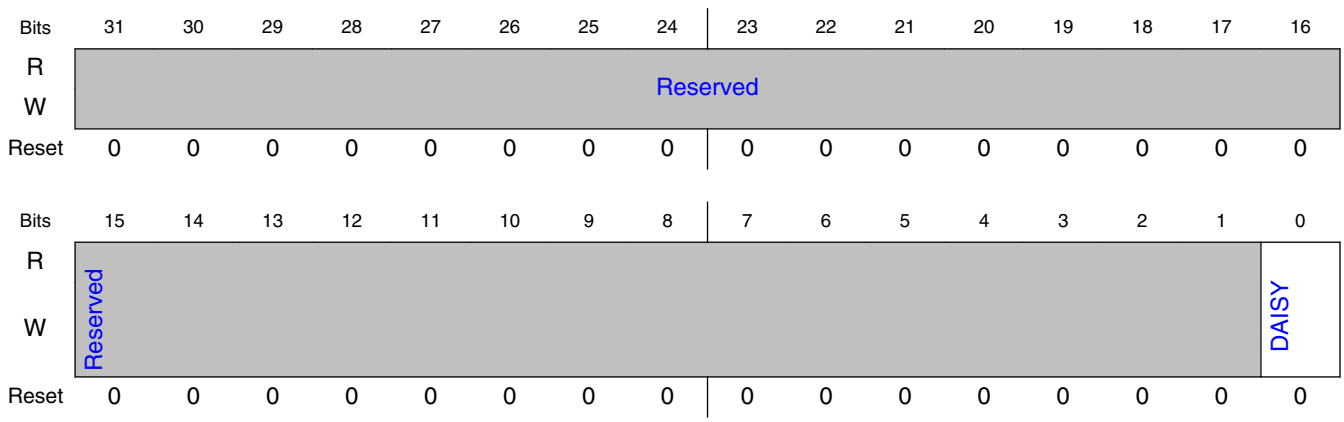
12.4.6.435.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_25	6ECh

12.4.6.435.2 Function

DAISY Register

12.4.6.435.3 Diagram



12.4.6.435.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in25 0b - Selecting Pad: GPIO_EMC_B2_05 for Mode: ALT6 1b - Selecting Pad: GPIO_SD_B1_05 for Mode: ALT2

12.4.6.436 XBAR1_IN_SELECT_INPUT_26 DAISY Register (XBAR1_IN_SELECT_INPUT_26)

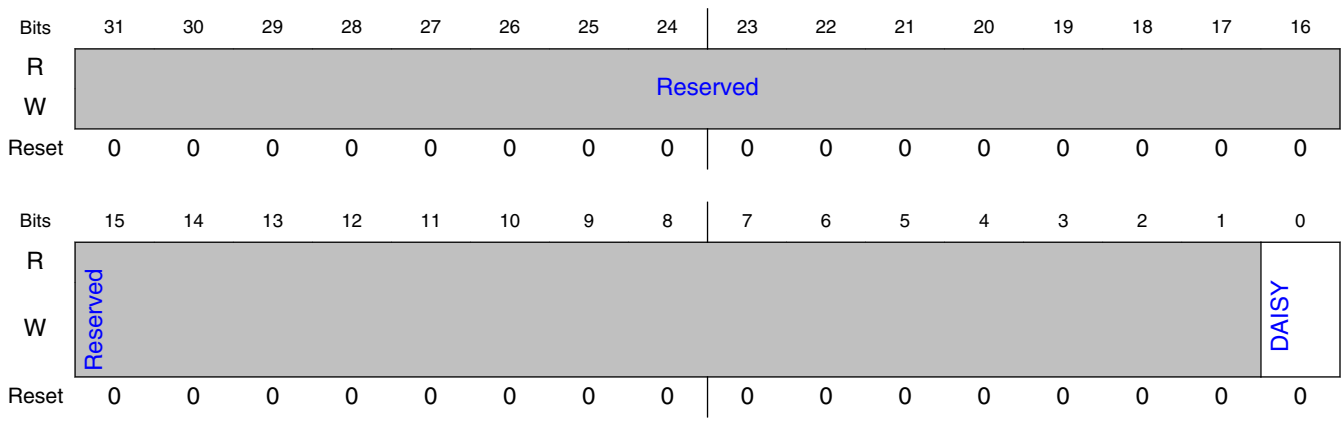
12.4.6.436.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_26	6F0h

12.4.6.436.2 Function

DAISY Register

12.4.6.436.3 Diagram



12.4.6.436.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in26 0b - Selecting Pad: GPIO_EMC_B2_06 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_00 for Mode: ALT4

12.4.6.437 XBAR1_IN_SELECT_INPUT_27 DAISY Register (XBAR1_IN_SELECT_INPUT_27)

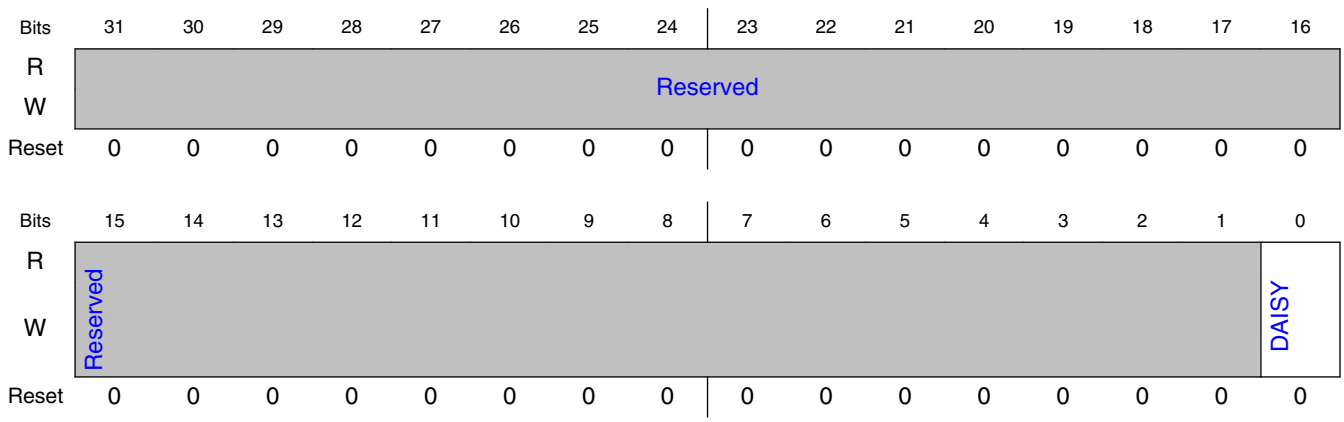
12.4.6.437.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_27	6F4h

12.4.6.437.2 Function

DAISY Register

12.4.6.437.3 Diagram



12.4.6.437.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in27 0b - Selecting Pad: GPIO_EMC_B2_07 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_01 for Mode: ALT4

12.4.6.438 XBAR1_IN_SELECT_INPUT_28 DAISY Register (XBAR1_IN_SELECT_INPUT_28)

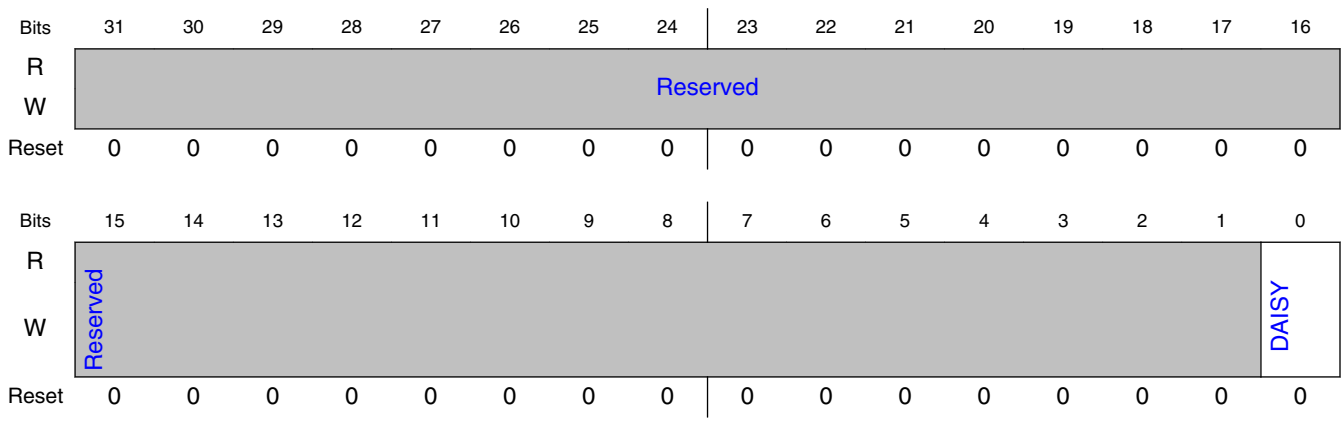
12.4.6.438.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_28	6F8h

12.4.6.438.2 Function

DAISY Register

12.4.6.438.3 Diagram



12.4.6.438.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in28 0b - Selecting Pad: GPIO_EMC_B2_08 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_02 for Mode: ALT4

12.4.6.439 XBAR1_IN_SELECT_INPUT_29 DAISY Register (XBAR1_IN_SELECT_INPUT_29)

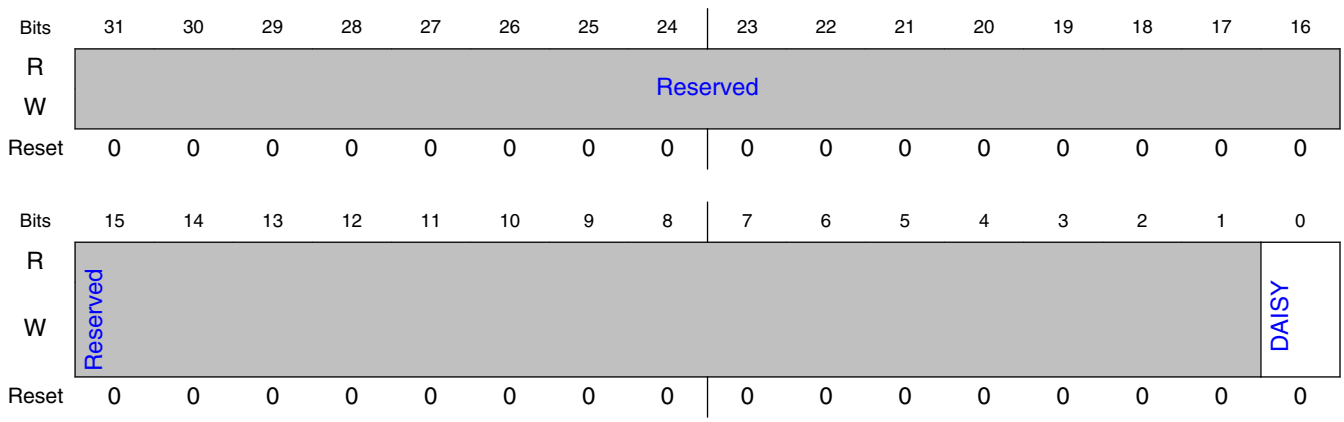
12.4.6.439.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_29	6FCh

12.4.6.439.2 Function

DAISY Register

12.4.6.439.3 Diagram



12.4.6.439.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in29 0b - Selecting Pad: GPIO_EMC_B2_09 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_03 for Mode: ALT4

12.4.6.440 XBAR1_IN_SELECT_INPUT_30 DAISY Register (XBAR1_IN_SELECT_INPUT_30)

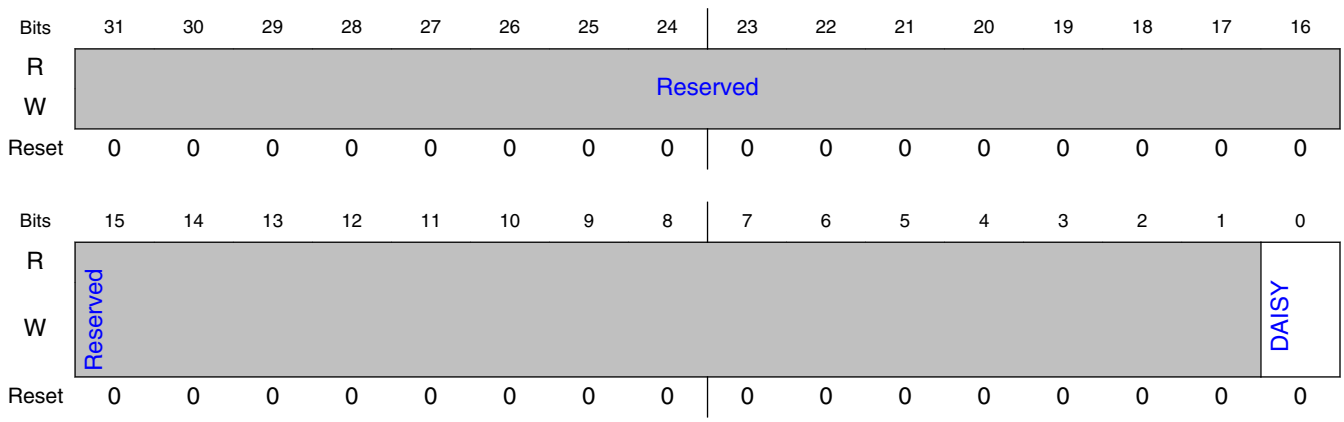
12.4.6.440.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_30	700h

12.4.6.440.2 Function

DAISY Register

12.4.6.440.3 Diagram



12.4.6.440.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in30 0b - Selecting Pad: GPIO_EMC_B2_10 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_04 for Mode: ALT4

12.4.6.441 XBAR1_IN_SELECT_INPUT_31 DAISY Register (XBAR1_IN_SELECT_INPUT_31)

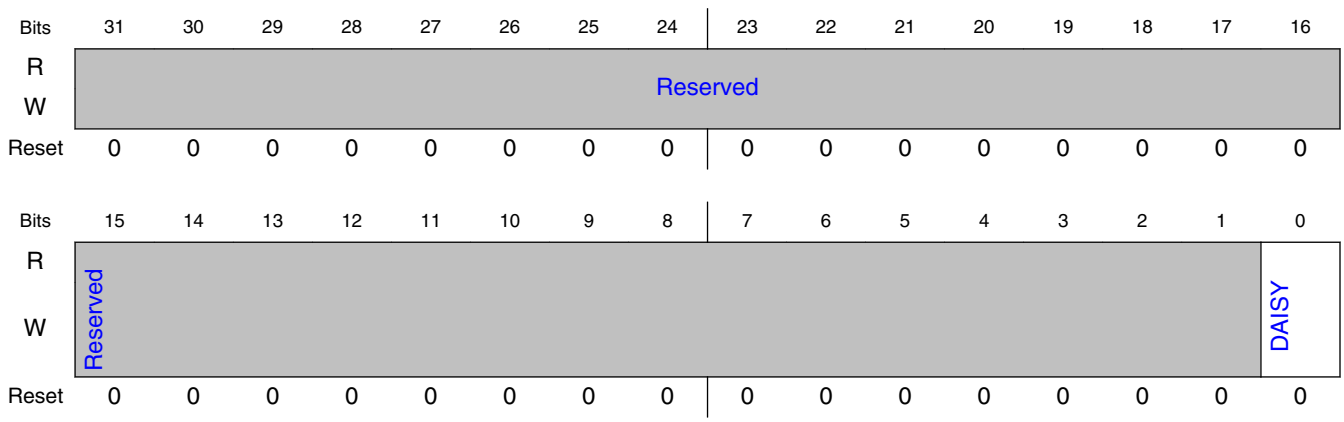
12.4.6.441.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_31	704h

12.4.6.441.2 Function

DAISY Register

12.4.6.441.3 Diagram



12.4.6.441.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in31 0b - Selecting Pad: GPIO_EMC_B2_11 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_05 for Mode: ALT4

12.4.6.442 XBAR1_IN_SELECT_INPUT_32 DAISY Register (XBAR1_IN_SELECT_INPUT_32)

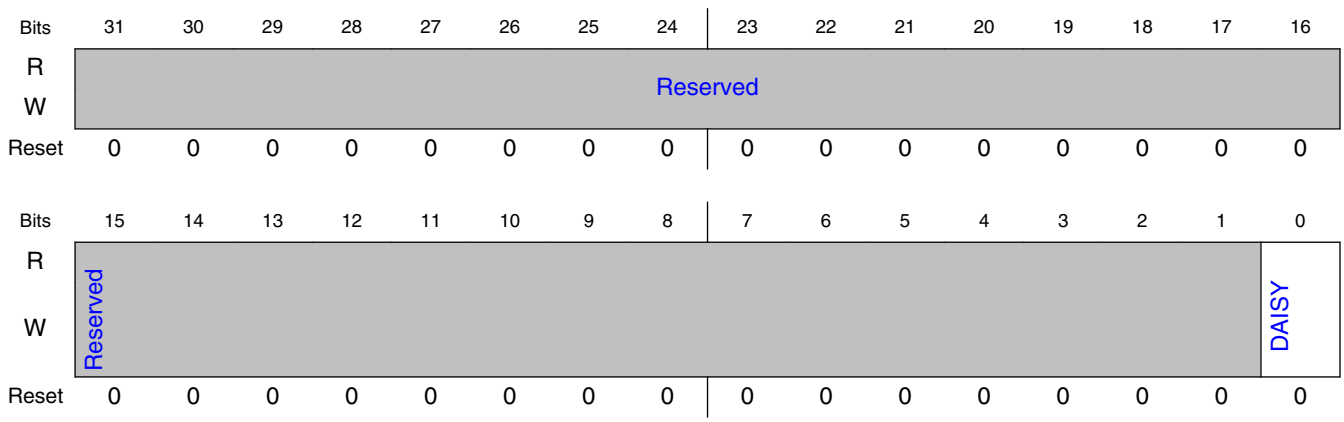
12.4.6.442.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_32	708h

12.4.6.442.2 Function

DAISY Register

12.4.6.442.3 Diagram



12.4.6.442.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in32 0b - Selecting Pad: GPIO_EMC_B2_12 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_06 for Mode: ALT4

12.4.6.443 XBAR1_IN_SELECT_INPUT_33 DAISY Register (XBAR1_IN_SELECT_INPUT_33)

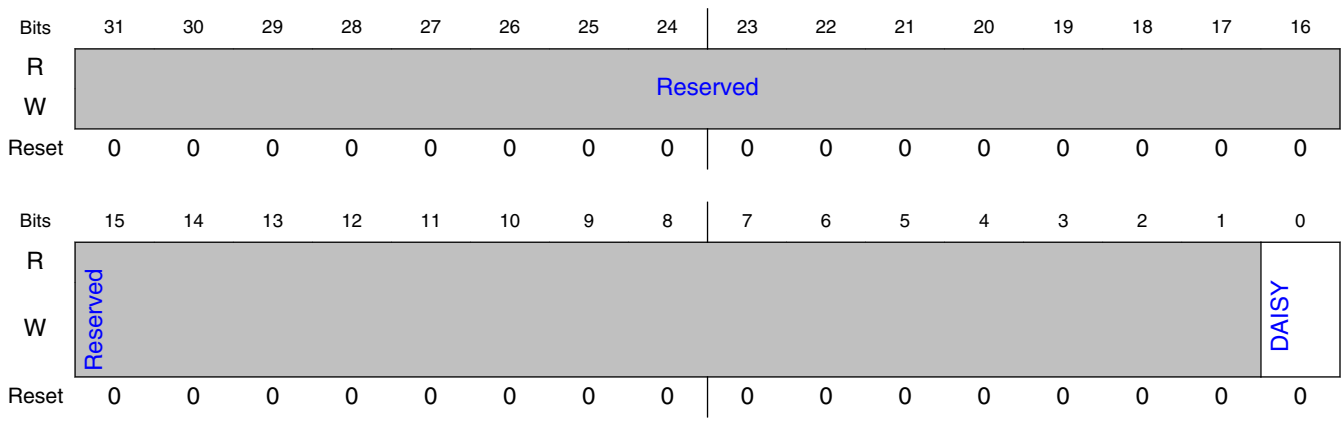
12.4.6.443.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_33	70Ch

12.4.6.443.2 Function

DAISY Register

12.4.6.443.3 Diagram



12.4.6.443.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in33 0b - Selecting Pad: GPIO_EMC_B2_13 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_07 for Mode: ALT4

12.4.6.444 XBAR1_IN_SELECT_INPUT_34 DAISY Register (XBAR1_IN_SELECT_INPUT_34)

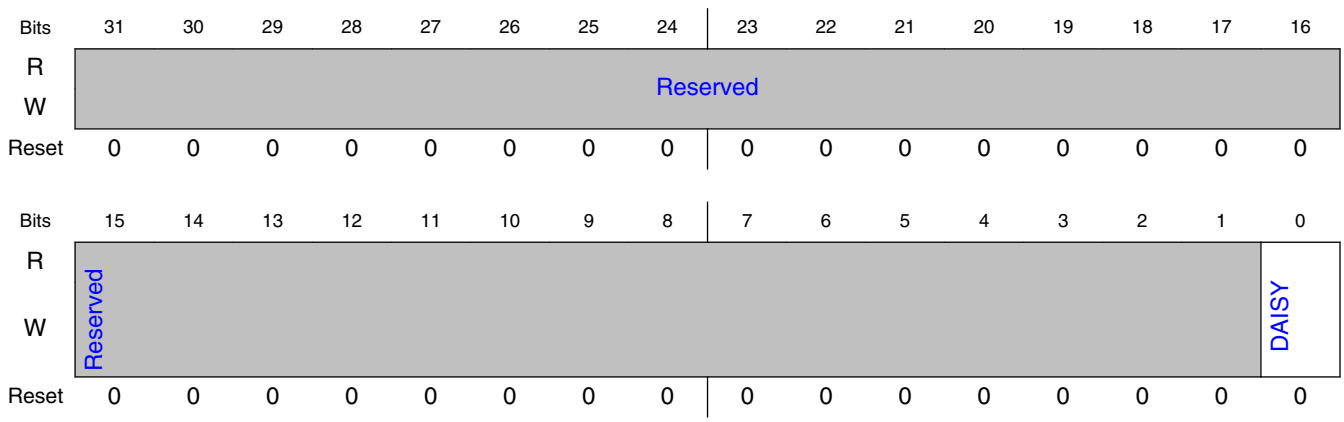
12.4.6.444.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_34	710h

12.4.6.444.2 Function

DAISY Register

12.4.6.444.3 Diagram



12.4.6.444.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in34 0b - Selecting Pad: GPIO_EMC_B2_14 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_08 for Mode: ALT4

12.4.6.445 XBAR1_IN_SELECT_INPUT_35 DAISY Register (XBAR1_IN_SELECT_INPUT_35)

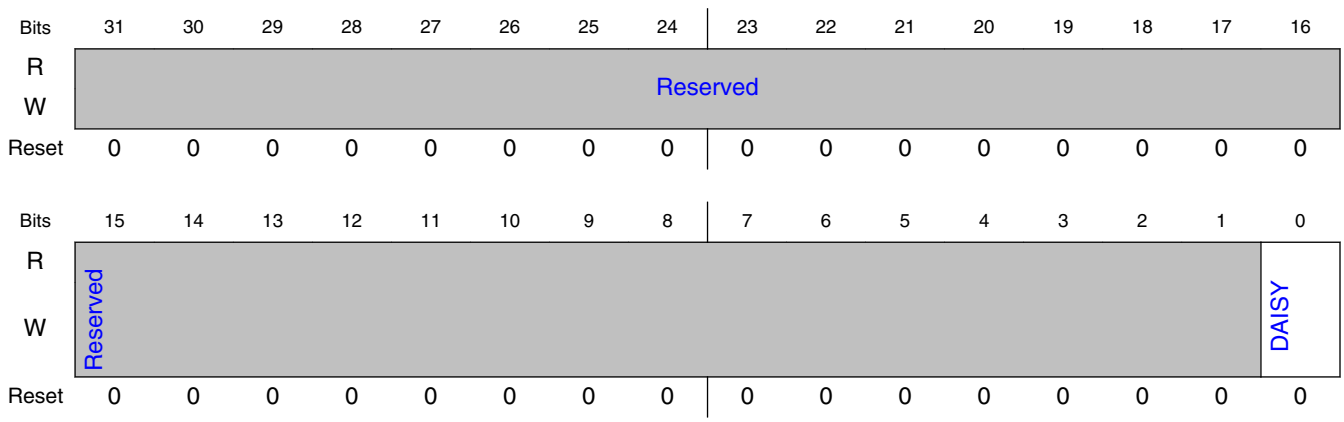
12.4.6.445.1 Offset

Register	Offset
XBAR1_IN_SELECT_INPUT_35	714h

12.4.6.445.2 Function

DAISY Register

12.4.6.445.3 Diagram



12.4.6.445.4 Fields

Field	Function
31-1 —	- Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: xbar1, In Pin: xbar_in35 0b - Selecting Pad: GPIO_EMC_B2_15 for Mode: ALT6 1b - Selecting Pad: GPIO_DISP_B1_09 for Mode: ALT4

Chapter 13
General Purpose Input/Output (GPIO)

13.1 Chip-specific GPIO information

Table 13-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

GPIO_MUX2_IOn and GPIO_MUX3_IOn signals can be routed to the standard GPIO modules (GPIO2 and GPIO3), or they can be routed to high speed GPIO modules (CM7_GPIO2 and CM7_GPIO3). By default, all of the GPIO_MUX2 and GPIO_MUX3 signals route to the standard GPIO2 and GPIO3 modules. To use high speed GPIOs, the IOMUXC_GPR40 - IOMUC_GPR43 registers must be reconfigured to route the desired signals to the CM7_GPIO2 and/or CM7_GPIO3 modules.

When LPSRMIX master is accessing GPIO13, XB will hold WAKEUPMIX master requests to any AIPS4 peripheral until current LPSRMIX master access is completed. Because GPIO13 access is based on 32 K clock as it's in SNVS domain to save power, these WAKEUPMIX master requests could be held for over 31µs.

13.2 Overview

The General-Purpose Input/Output (GPIO) peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

When configured as an output, it is possible to write to an internal register to control the state driven on the output pin. When configured as an input, it is possible to detect the state of the input by reading the state of an internal register. In addition, the GPIO peripheral can produce CORE interrupts.

13.2.1 Block Diagram

The GPIO subsystem contains multiple GPIO blocks, which can generate and control up to 32 signals for general purpose.

The block diagram for the GPIO block is provided below.

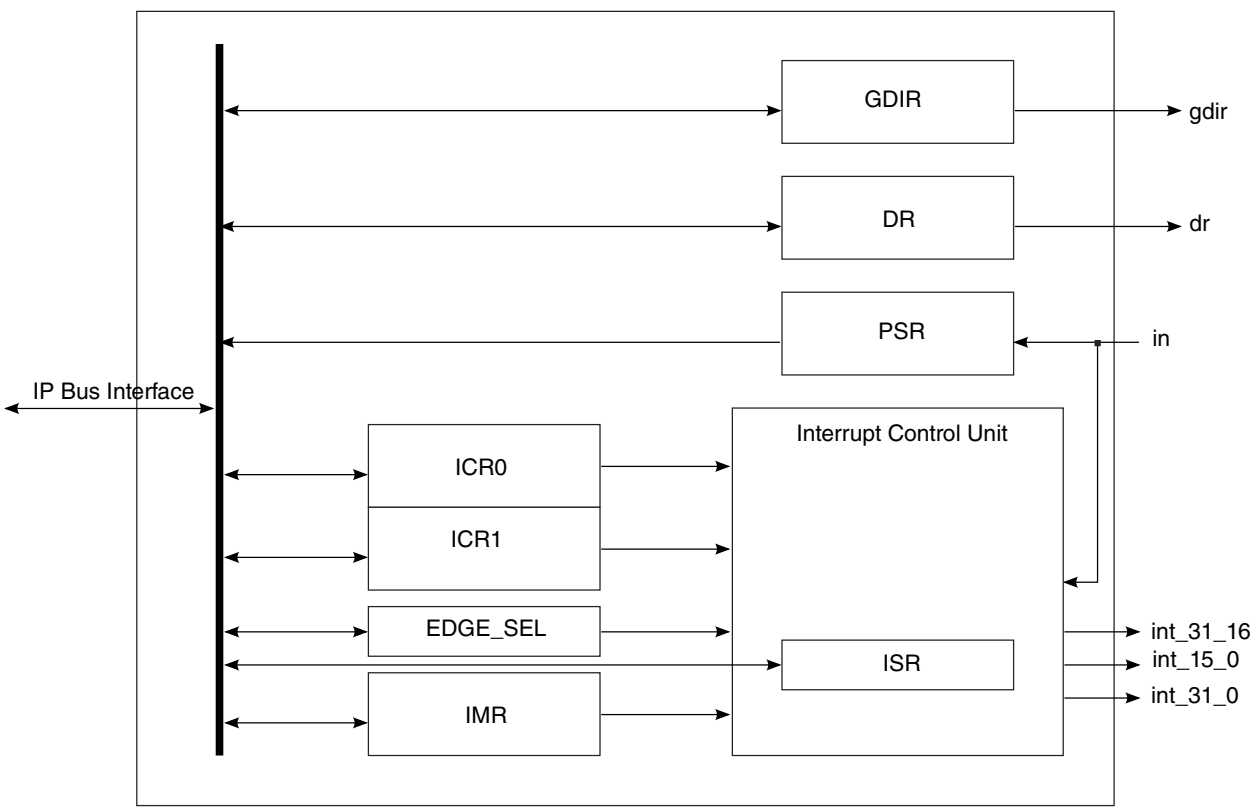


Figure 13-1. GPIO Block Diagram

13.2.2 Features

The GPIO includes the following features:

- General purpose input/output logic capabilities:
  - Drives specific data to output using the data register (DR)
  - Controls the direction of the signal using the GPIO direction register (GDIR)
  - Enables the core to sample the status of the corresponding inputs by reading the pad sample register (PSR).
- GPIO interrupt capabilities:
  - Supports 32 interrupts
  - Identifies interrupt edges
  - Generates three active-high interrupts to the SoC interrupt controller

13.3 Functional Description

The GPIO is one of the blocks controlling the IOMUX of the chip.

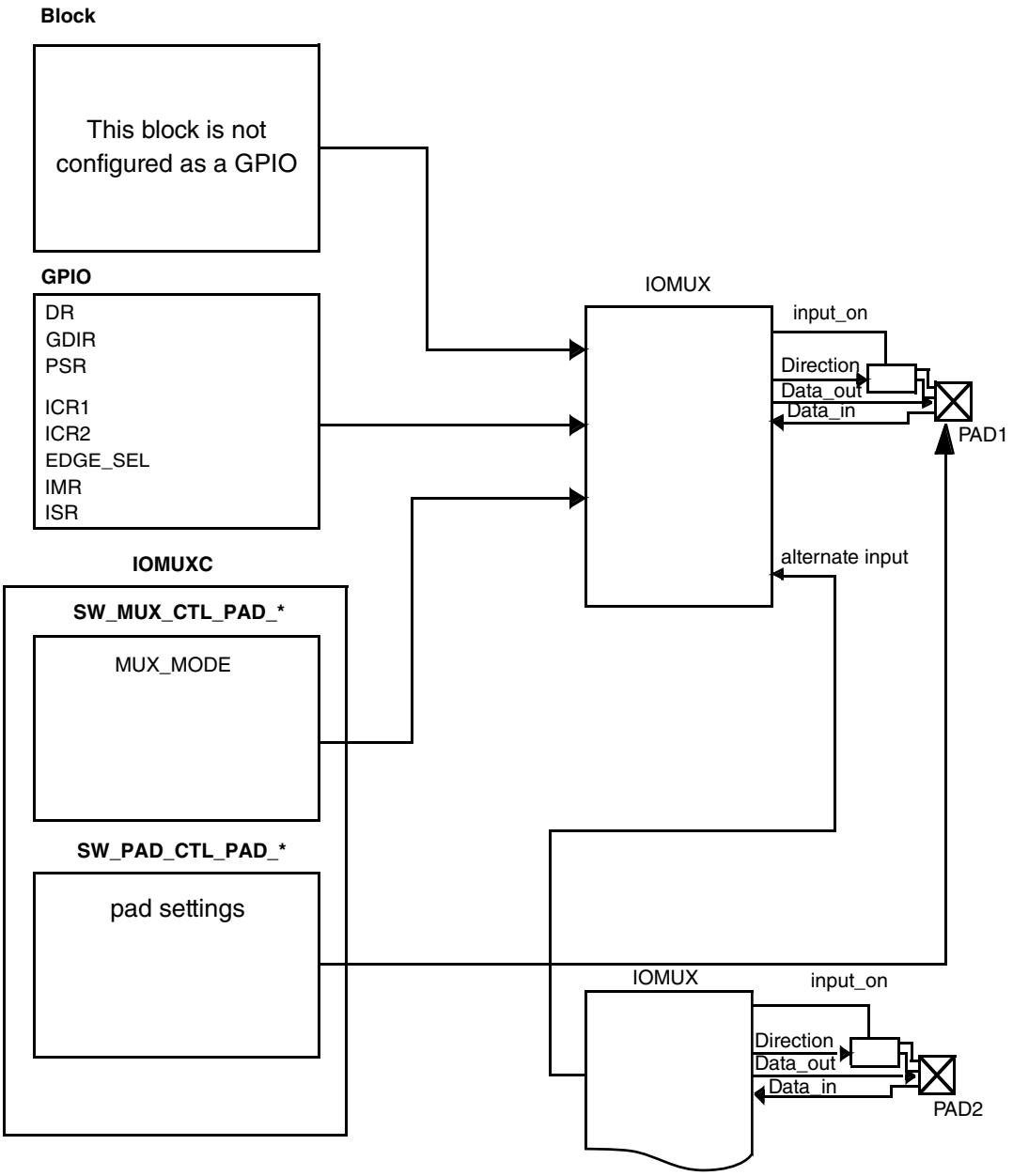


Figure 13-2. Chip IOMUX Scheme

A GPIO signal can operate as a general-purpose input/output when the IOMUX is set to GPIO mode. Each GPIO signal may be independently configured as either an input or an output using the GPIO direction register (GDIR).

When configured as an output (GDIR[GDIR] = 1), the value in the data bit in the GPIO data register (DR) is driven on the corresponding GPIO line. When a signal is configured as an input (GDIR[GDIR] = 0), the state of the input can be read from the corresponding PSR[PSR] bit.

The GPIO functionality is provided through eight registers, an edge-detect circuit, and interrupt generation logic.

The eight registers are:

- Data register (DR)
- GPIO direction register (GDIR)
- Pad sample register (PSR)
- Interrupt control registers (ICR1, ICR2)
- Edge select register (EDGE_SEL)
- Interrupt mask register (IMR)
- Interrupt status register (ISR)

The above registers are described in detail in the Register section.

Each GPIO input has a dedicated edge-detect circuit which can be configured through software to detect rising edges, falling edges, logic low-levels or logic high-levels on the input signals. The outputs of the edge detect circuits are optionally masked by setting the corresponding bit in the interrupt mask register (IMR). These qualified outputs are OR'ed together to generate two one-bit interrupt lines:

- Combined interrupt indication for GPIOx signals 0 - 15
- Combined interrupt indication for GPIOx signals 16 - 31

The GPIO edge detection is described further in Interrupt Control Unit.

13.3.1 GPIO pad structure

The functional diagram for the GPIO pad is provided below.

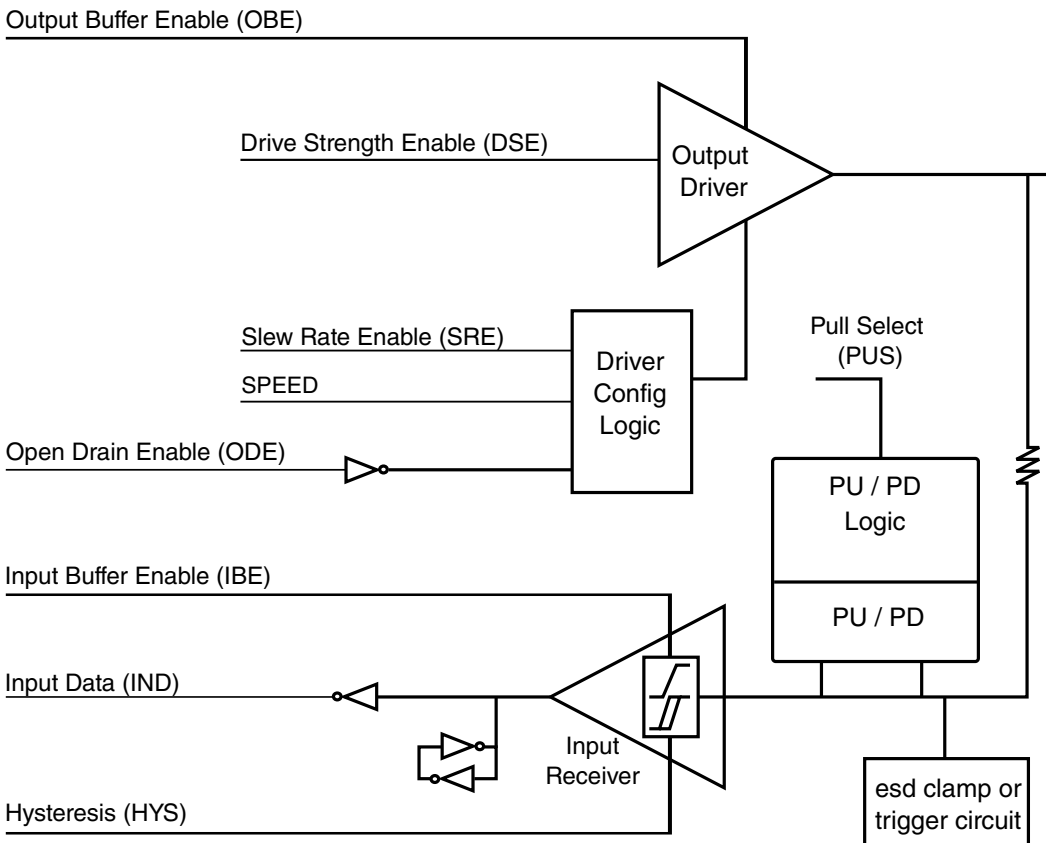


Figure 13-3. GPIO pad functional diagram

13.3.1.1 Input Driver

Input driver characteristics

- Selectable Schmitt trigger or CMOS input mode
- Receiver is tri-stated when I/O supply (OVDD) is powered down.

13.3.1.1.1 Schmitt trigger

The anti-jamming functionality of the Schmitt trigger is illustrated below.

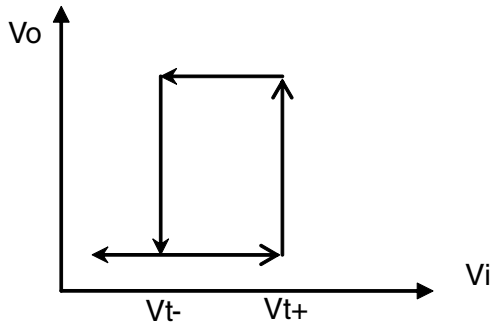


Figure 13-4. Schmitt trigger transfer characteristic

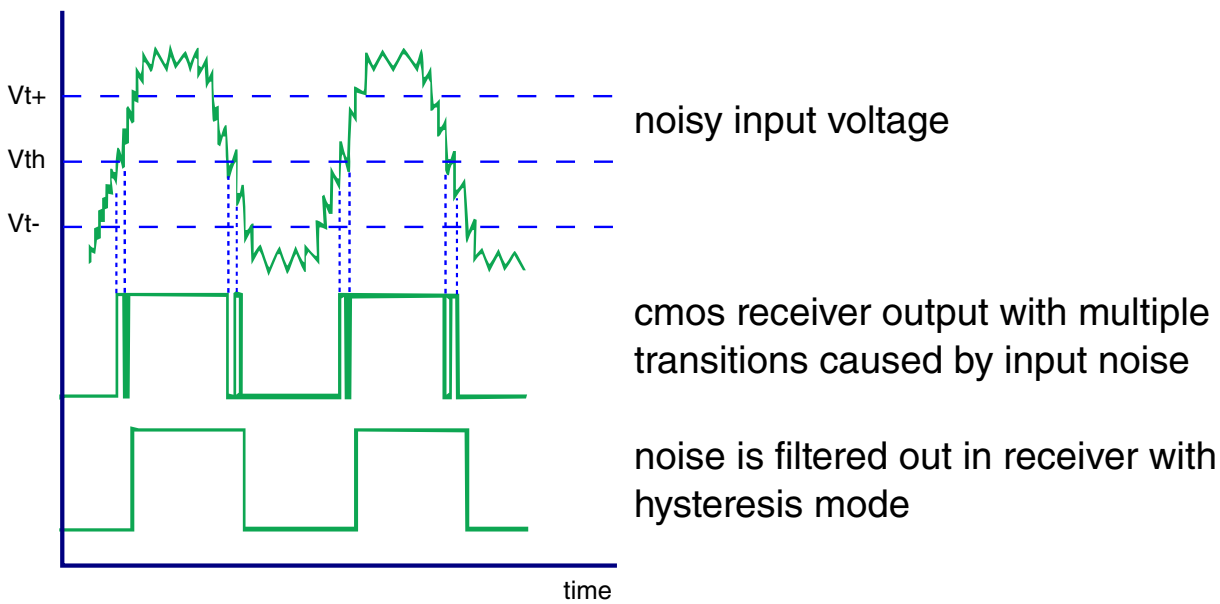


Figure 13-5. Receiver output in CMOS and hysteresis

13.3.1.2 Output Driver

Output driver characteristics

- Selectable CMOS or open-drain output type
- Selectable pull-up resistors of 22K, 47K, 100K and a pull-down resistor of 100KOhm. Unsilicided P+ poly resistor is used to limit resistance variation to within +/- 20%.
- Pull-up and pull-down are disabled in output mode.

- Seven drive strengths in each operating mode
- Additional 2-bit slew rate control to select between 50, 100, and 200 MHz IO cell operation range with reduced switching noise

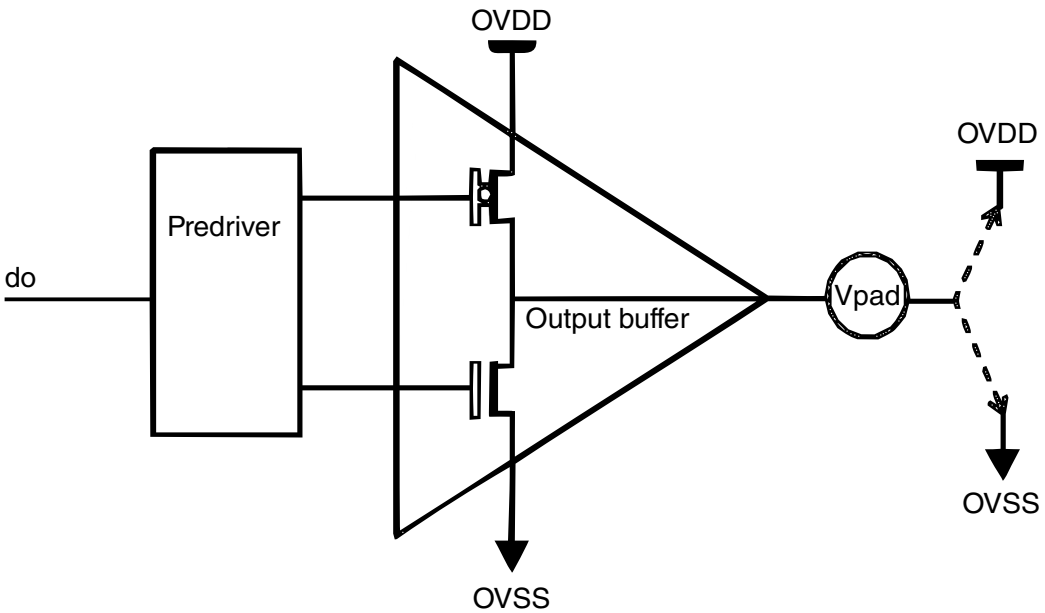


Figure 13-6. Output Driver Functional Diagram

13.3.1.2.1 Drive strength

Drive strength selection can be use to make the impedance matched and get better signal integrity.

13.3.1.2.2 Open drain

Open drain is a circuit technique which allows multiple devices to communication over a single wire bi-directionally. Open drain drivers usually operate with an external or internal pull-up resistor that holds the signal line high until a device sinks enough current to pull the line low, usually used for a bus with multiple devices.

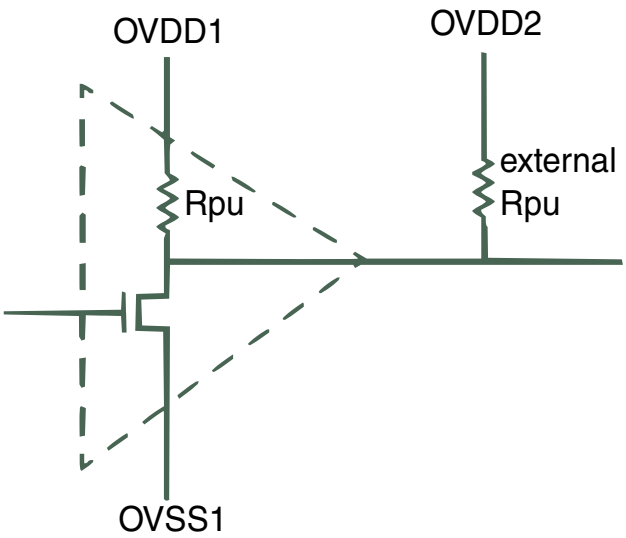


Figure 13-7. Output buffer in open drain mode

13.3.2 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR[GDIR] = 0). The interrupt control registers (ICR1 and ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high). For information about ICR1 and ICR2 settings, see the Register section.

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

13.3.3 Clocks

The following table describes the clock sources for GPIO. Please see the CCM for clock setting, configuration and gating information.

Table 13-2. GPIO Clocks

Clock name	Description
ipg_clk_s	Peripheral access clock

13.4 External Signals

The following table describes the external signals of GPIO.

Table 13-3. GPIO External Signals

Signal	Description
IOn	GPIO Signal

13.5 Application Information

13.5.1 GPIO Read Mode

The programming sequence for reading input signals should be as follows:

1. Configure IOMUX to select GPIO mode (via IOMUX Controller (IOMUXC)).
2. Configure GPIO direction register to input (GDIR[GDIR] = 0b).
3. Read value from data register/pad status register.

A pseudocode description to read [input3:input0] values is as follows:

```
// SET INPUTS TO GPIO MODE.
write sw_mux_ctl_<input0>_<input1>_<input2>_<input3>, 32'h00000000
// SET GDIR TO INPUT.
write GDIR[31:4,input3_bit, input2_bit, input1_bit, input0_bit,] 32'hxxxxxxx0
// READ INPUT VALUE FROM DR.
read DR
// READ INPUT VALUE FROM PSR.
read PSR
```

NOTE

While the GPIO direction is set to input (GDIR[GDIR] = 0), a read access to DR does not return DR data. Instead, it returns the PSR data, which is the corresponding input signal value.

13.5.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (via IOMUXC). If needed, enable SION to read the loopback pad value through PSR.

2. Configure GPIO direction register to output (GDIR[GDIR] = 1b).
3. Write value to data register (DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
// SET PADS TO GPIO MODE VIA IOMUX.
write sw_mux_ctl_pad_<output[0-3]>.mux_mode, <GPIO_MUX_MODE>
// Enable loopback to capture pad value into PSR in output mode
write sw_mux_ctl_pad_<output[0-3]>.sion, 1
// SET GDIR=1 TO OUTPUT BITS.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxF
// WRITE OUTPUT VALUE=4'b0101 TO DR.
write DR, 32'hxxxxxxx5
// READ OUTPUT VALUE FROM PSR ONLY.
read_cmp PSR, 32'hxxxxxxx5
```

13.6 Memory Map/Register Definition

This section includes the memory map and descriptions of all registers.

13.6.1 GPIO register descriptions

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

13.6.1.1 GPIO memory map

CM7_GPIO2 base address: 4200_8000h

CM7_GPIO3 base address: 4200_C000h

GPIO1 base address: 4012_C000h

GPIO2 base address: 4013_0000h

GPIO3 base address: 4013_4000h

GPIO4 base address: 4013_8000h

GPIO5 base address: 4013_C000h

GPIO6 base address: 4014_0000h

GPIO7 base address: 40C5_C000h

GPIO8 base address: 40C6_0000h

GPIO9 base address: 40C6_4000h

GPIO10 base address: 40C6_8000h

GPIO11 base address: 40C6_C000h

GPIO12 base address: 40C7_0000h

GPIO13 base address: 40CA_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPIO data register (DR)	32	RW	0000_0000h
4h	GPIO direction register (GDIR)	32	RW	0000_0000h
8h	GPIO pad status register (PSR)	32	R	0000_0000h
Ch	GPIO interrupt configuration register1 (ICR1)	32	RW	0000_0000h
10h	GPIO interrupt configuration register2 (ICR2)	32	RW	0000_0000h
14h	GPIO interrupt mask register (IMR)	32	RW	0000_0000h
18h	GPIO interrupt status register (ISR)	32	RW	0000_0000h
1Ch	GPIO edge select register (EDGE_SEL)	32	RW	0000_0000h
84h	GPIO data register SET (DR_SET)	32	W	0000_0000h
88h	GPIO data register CLEAR (DR_CLEAR)	32	W	0000_0000h
8Ch	GPIO data register TOGGLE (DR_TOGGLE)	32	W	0000_0000h

13.6.1.2 GPIO data register (DR)

13.6.1.2.1 Offset

Register	Offset
DR	0h

13.6.1.2.2 Function

The 32-bit DR register stores data that is ready to be driven to the output lines. If the IOMUXC is in GPIO mode and a given GPIO direction bit is set, then the corresponding DR bit is driven to the output. If a given GPIO direction bit is cleared, then a read of DR register reflects the value of the corresponding signal. Two wait states are required in read access for synchronization.

The results of a read of a DR bit depends on the IOMUXC input mode settings and the corresponding GDIR bit as follows:

- If GDIR[*n*] is set and IOMUXC input mode is GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is GPIO, then reading DR[*n*] returns the corresponding input signal's value.
- If GDIR[*n*] is set and IOMUXC input mode is not GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is not GPIO, then reading DR[*n*] always returns zero.

13.6.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.2.4 Fields

Field	Function
31-0 DR	DR data bits This register defines the value of the GPIO output when the signal is configured as an output (GDIR[n]=1). Writes to this register are stored in a register. Reading DR register returns the value stored in the register if the signal is configured as an output (GDIR[n]=1), or the input signal's value if configured as an input (GDIR[n]=0). **NOTE:** The IOMUXC must be configured to GPIO mode for the DR register value to connect with the signal. Reading the data register with the input path disabled always returns a zero value.

13.6.1.3 GPIO direction register (GDIR)

13.6.1.3.1 Offset

Register	Offset
GDIR	4h

13.6.1.3.2 Function

The GDIR register functions as direction control when the IOMUXC is in GPIO mode. Each bit specifies the direction of a one-bit signal. The mapping of each DIR bit to a corresponding SoC signal is determined by the SoC's pin assignment and the IOMUX table. For more details consult the IOMUXC chapter.

13.6.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GDIR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDIR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.3.4 Fields

Field	Function
31-0 GDIR	GPIO direction bits Bit n of this register defines the direction of the GPIO[n] signal. **NOTE:** The GDIR register affects only the direction of the I/O signal when the corresponding bit in the IOMUXC is configured for GPIO. 0b - GPIO is configured as input. 1b - GPIO is configured as output.

13.6.1.4 GPIO pad status register (PSR)

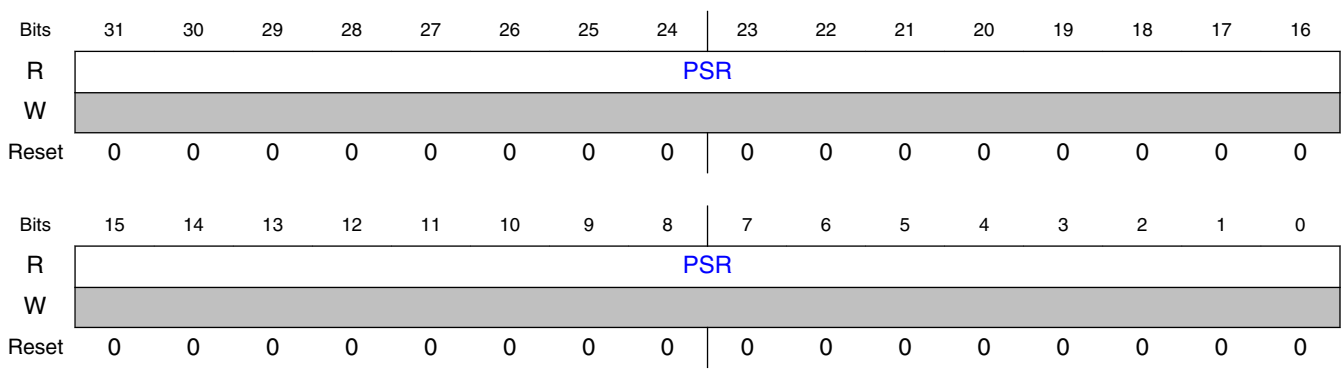
13.6.1.4.1 Offset

Register	Offset
PSR	8h

13.6.1.4.2 Function

The PSR register is a read-only register. Each bit stores the value of the corresponding input signal (as configured in the IOMUX). This register is clocked with the ipg_clk_s clock, meaning that the input signal is sampled only when accessing this location. Two wait states are required any time this register is accessed for synchronization.

13.6.1.4.3 Diagram



13.6.1.4.4 Fields

Field	Function
31-0 PSR	GPIO pad status bits Reading the PSR register returns the state of the corresponding input signal. **NOTE:** The IOMUXC must be configured to GPIO mode for the PSR register to reflect the state of the corresponding signal.

13.6.1.5 GPIO interrupt configuration register1 (ICR1)

13.6.1.5.1 Offset

Register	Offset
ICR1	Ch

13.6.1.5.2 Function

The ICR1 register contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

13.6.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR15		ICR14		ICR13		ICR12		ICR11		ICR10		ICR9		ICR8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR7		ICR6		ICR5		ICR4		ICR3		ICR2		ICR1		ICR0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.5.4 Fields

Field	Function
31-30 ICR15	Interrupt configuration field for GPIO interrupt 15 This bit field controls the active condition of the interrupt function for GPIO interrupt 15. 00b - Interrupt 15 is low-level sensitive. 01b - Interrupt 15 is high-level sensitive. 10b - Interrupt 15 is rising-edge sensitive. 11b - Interrupt 15 is falling-edge sensitive.
29-28 ICR14	Interrupt configuration field for GPIO interrupt 14 This bit field controls the active condition of the interrupt function for GPIO interrupt 14. 00b - Interrupt 14 is low-level sensitive. 01b - Interrupt 14 is high-level sensitive. 10b - Interrupt 14 is rising-edge sensitive. 11b - Interrupt 14 is falling-edge sensitive.
27-26	Interrupt configuration field for GPIO interrupt 13

Table continues on the next page...

Field	Function
ICR13	This bit field controls the active condition of the interrupt function for GPIO interrupt 13. 00b - Interrupt 13 is low-level sensitive. 01b - Interrupt 13 is high-level sensitive. 10b - Interrupt 13 is rising-edge sensitive. 11b - Interrupt 13 is falling-edge sensitive.
25-24	Interrupt configuration field for GPIO interrupt 12
ICR12	This bit field controls the active condition of the interrupt function for GPIO interrupt 12. 00b - Interrupt 12 is low-level sensitive. 01b - Interrupt 12 is high-level sensitive. 10b - Interrupt 12 is rising-edge sensitive. 11b - Interrupt 12 is falling-edge sensitive.
23-22	Interrupt configuration field for GPIO interrupt 11
ICR11	This bit field controls the active condition of the interrupt function for GPIO interrupt 11. 00b - Interrupt 11 is low-level sensitive. 01b - Interrupt 11 is high-level sensitive. 10b - Interrupt 11 is rising-edge sensitive. 11b - Interrupt 11 is falling-edge sensitive.
21-20	Interrupt configuration field for GPIO interrupt 10
ICR10	This bit field controls the active condition of the interrupt function for GPIO interrupt 10. 00b - Interrupt 10 is low-level sensitive. 01b - Interrupt 10 is high-level sensitive. 10b - Interrupt 10 is rising-edge sensitive. 11b - Interrupt 10 is falling-edge sensitive.
19-18	Interrupt configuration field for GPIO interrupt 9
ICR9	This bit field controls the active condition of the interrupt function for GPIO interrupt 9. 00b - Interrupt 9 is low-level sensitive. 01b - Interrupt 9 is high-level sensitive. 10b - Interrupt 9 is rising-edge sensitive. 11b - Interrupt 9 is falling-edge sensitive.
17-16	Interrupt configuration field for GPIO interrupt 8
ICR8	This bit field controls the active condition of the interrupt function for GPIO interrupt 8. 00b - Interrupt 8 is low-level sensitive. 01b - Interrupt 8 is high-level sensitive. 10b - Interrupt 8 is rising-edge sensitive. 11b - Interrupt 8 is falling-edge sensitive.
15-14	Interrupt configuration field for GPIO interrupt 7
ICR7	This bit field controls the active condition of the interrupt function for GPIO interrupt 7. 00b - Interrupt 7 is low-level sensitive. 01b - Interrupt 7 is high-level sensitive. 10b - Interrupt 7 is rising-edge sensitive. 11b - Interrupt 7 is falling-edge sensitive.
13-12	Interrupt configuration field for GPIO interrupt 6
ICR6	This bit field controls the active condition of the interrupt function for GPIO interrupt 6. 00b - Interrupt 6 is low-level sensitive. 01b - Interrupt 6 is high-level sensitive. 10b - Interrupt 6 is rising-edge sensitive. 11b - Interrupt 6 is falling-edge sensitive.

Table continues on the next page...

Field	Function
11-10 ICR5	Interrupt configuration field for GPIO interrupt 5 This bit field controls the active condition of the interrupt function for GPIO interrupt 5. 00b - Interrupt 5 is low-level sensitive. 01b - Interrupt 5 is high-level sensitive. 10b - Interrupt 5 is rising-edge sensitive. 11b - Interrupt 5 is falling-edge sensitive.
9-8 ICR4	Interrupt configuration field for GPIO interrupt 4 This bit field controls the active condition of the interrupt function for GPIO interrupt 4. 00b - Interrupt 4 is low-level sensitive. 01b - Interrupt 4 is high-level sensitive. 10b - Interrupt 4 is rising-edge sensitive. 11b - Interrupt 4 is falling-edge sensitive.
7-6 ICR3	Interrupt configuration field for GPIO interrupt 3 This bit field controls the active condition of the interrupt function for GPIO interrupt 3. 00b - Interrupt 3 is low-level sensitive. 01b - Interrupt 3 is high-level sensitive. 10b - Interrupt 3 is rising-edge sensitive. 11b - Interrupt 3 is falling-edge sensitive.
5-4 ICR2	Interrupt configuration field for GPIO interrupt 2 This bit field controls the active condition of the interrupt function for GPIO interrupt 2. 00b - Interrupt 2 is low-level sensitive. 01b - Interrupt 2 is high-level sensitive. 10b - Interrupt 2 is rising-edge sensitive. 11b - Interrupt 2 is falling-edge sensitive.
3-2 ICR1	Interrupt configuration field for GPIO interrupt 1 This bit field controls the active condition of the interrupt function for GPIO interrupt 1. 00b - Interrupt 1 is low-level sensitive. 01b - Interrupt 1 is high-level sensitive. 10b - Interrupt 1 is rising-edge sensitive. 11b - Interrupt 1 is falling-edge sensitive.
1-0 ICR0	Interrupt configuration field for GPIO interrupt 0 This bit field controls the active condition of the interrupt function for GPIO interrupt 0. 00b - Interrupt 0 is low-level sensitive. 01b - Interrupt 0 is high-level sensitive. 10b - Interrupt 0 is rising-edge sensitive. 11b - Interrupt 0 is falling-edge sensitive.

13.6.1.6 GPIO interrupt configuration register2 (ICR2)

13.6.1.6.1 Offset

Register	Offset
ICR2	10h

13.6.1.6.2 Function

The ICR2 register contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

13.6.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR31		ICR30		ICR29		ICR28		ICR27		ICR26		ICR25		ICR24	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR23		ICR22		ICR21		ICR20		ICR19		ICR18		ICR17		ICR16	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.6.4 Fields

Field	Function
31-30 ICR31	Interrupt configuration field for GPIO interrupt 31 This bit field controls the active condition of the interrupt function for GPIO interrupt 31. 00b - Interrupt 31 is low-level sensitive. 01b - Interrupt 31 is high-level sensitive. 10b - Interrupt 31 is rising-edge sensitive. 11b - Interrupt 31 is falling-edge sensitive.
29-28 ICR30	Interrupt configuration field for GPIO interrupt 30 This bit field controls the active condition of the interrupt function for GPIO interrupt 30. 00b - Interrupt 30 is low-level sensitive. 01b - Interrupt 30 is high-level sensitive. 10b - Interrupt 30 is rising-edge sensitive. 11b - Interrupt 30 is falling-edge sensitive.
27-26 ICR29	Interrupt configuration field for GPIO interrupt 29 This bit field controls the active condition of the interrupt function for GPIO interrupt 29. 00b - Interrupt 29 is low-level sensitive. 01b - Interrupt 29 is high-level sensitive. 10b - Interrupt 29 is rising-edge sensitive. 11b - Interrupt 29 is falling-edge sensitive.
25-24 ICR28	Interrupt configuration field for GPIO interrupt 28 This bit field controls the active condition of the interrupt function for GPIO interrupt 28. 00b - Interrupt 28 is low-level sensitive. 01b - Interrupt 28 is high-level sensitive.

Table continues on the next page...

Field	Function
	10b - Interrupt 28 is rising-edge sensitive. 11b - Interrupt 28 is falling-edge sensitive.
23-22 ICR27	Interrupt configuration field for GPIO interrupt 27 This bit field controls the active condition of the interrupt function for GPIO interrupt 27. 00b - Interrupt 27 is low-level sensitive. 01b - Interrupt 27 is high-level sensitive. 10b - Interrupt 27 is rising-edge sensitive. 11b - Interrupt 27 is falling-edge sensitive.
21-20 ICR26	Interrupt configuration field for GPIO interrupt 26 This bit field controls the active condition of the interrupt function for GPIO interrupt 26. 00b - Interrupt 26 is low-level sensitive. 01b - Interrupt 26 is high-level sensitive. 10b - Interrupt 26 is rising-edge sensitive. 11b - Interrupt 26 is falling-edge sensitive.
19-18 ICR25	Interrupt configuration field for GPIO interrupt 25 This bit field controls the active condition of the interrupt function for GPIO interrupt 25. 00b - Interrupt 25 is low-level sensitive. 01b - Interrupt 25 is high-level sensitive. 10b - Interrupt 25 is rising-edge sensitive. 11b - Interrupt 25 is falling-edge sensitive.
17-16 ICR24	Interrupt configuration field for GPIO interrupt 24 This bit field controls the active condition of the interrupt function for GPIO interrupt 24. 00b - Interrupt 24 is low-level sensitive. 01b - Interrupt 24 is high-level sensitive. 10b - Interrupt 24 is rising-edge sensitive. 11b - Interrupt 24 is falling-edge sensitive.
15-14 ICR23	Interrupt configuration field for GPIO interrupt 23 This bit field controls the active condition of the interrupt function for GPIO interrupt 23. 00b - Interrupt 23 is low-level sensitive. 01b - Interrupt 23 is high-level sensitive. 10b - Interrupt 23 is rising-edge sensitive. 11b - Interrupt 23 is falling-edge sensitive.
13-12 ICR22	Interrupt configuration field for GPIO interrupt 22 This bit field controls the active condition of the interrupt function for GPIO interrupt 22. 00b - Interrupt 22 is low-level sensitive. 01b - Interrupt 22 is high-level sensitive. 10b - Interrupt 22 is rising-edge sensitive. 11b - Interrupt 22 is falling-edge sensitive.
11-10 ICR21	Interrupt configuration field for GPIO interrupt 21 This bit field controls the active condition of the interrupt function for GPIO interrupt 21. 00b - Interrupt 21 is low-level sensitive. 01b - Interrupt 21 is high-level sensitive. 10b - Interrupt 21 is rising-edge sensitive. 11b - Interrupt 21 is falling-edge sensitive.
9-8 ICR20	Interrupt configuration field for GPIO interrupt 20 This bit field controls the active condition of the interrupt function for GPIO interrupt 20.

Table continues on the next page...

Field	Function
	00b - Interrupt 20 is low-level sensitive. 01b - Interrupt 20 is high-level sensitive. 10b - Interrupt 20 is rising-edge sensitive. 11b - Interrupt 20 is falling-edge sensitive.
7-6 ICR19	Interrupt configuration field for GPIO interrupt 19 This bit field controls the active condition of the interrupt function for GPIO interrupt 19. 00b - Interrupt 19 is low-level sensitive. 01b - Interrupt 19 is high-level sensitive. 10b - Interrupt 19 is rising-edge sensitive. 11b - Interrupt 19 is falling-edge sensitive.
5-4 ICR18	Interrupt configuration field for GPIO interrupt 18 This bit field controls the active condition of the interrupt function for GPIO interrupt 18. 00b - Interrupt 18 is low-level sensitive. 01b - Interrupt 18 is high-level sensitive. 10b - Interrupt 18 is rising-edge sensitive. 11b - Interrupt 18 is falling-edge sensitive.
3-2 ICR17	Interrupt configuration field for GPIO interrupt 17 This bit field controls the active condition of the interrupt function for GPIO interrupt 17. 00b - Interrupt 17 is low-level sensitive. 01b - Interrupt 17 is high-level sensitive. 10b - Interrupt 17 is rising-edge sensitive. 11b - Interrupt 17 is falling-edge sensitive.
1-0 ICR16	Interrupt configuration field for GPIO interrupt 16 This bit field controls the active condition of the interrupt function for GPIO interrupt 16. 00b - Interrupt 16 is low-level sensitive. 01b - Interrupt 16 is high-level sensitive. 10b - Interrupt 16 is rising-edge sensitive. 11b - Interrupt 16 is falling-edge sensitive.

13.6.1.7 GPIO interrupt mask register (IMR)

13.6.1.7.1 Offset

Register	Offset
IMR	14h

13.6.1.7.2 Function

The IMR register contains masking bits for each interrupt line.

13.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.7.4 Fields

Field	Function
31-0 IMR	Interrupt Mask bits This register is used to enable or disable the interrupt function on each of the 32 GPIO signals. Bit IMR[n] (n=0...31) controls interrupt n as follows: 0b - Interrupt n is disabled. 1b - Interrupt n is enabled.

13.6.1.8 GPIO interrupt status register (ISR)

13.6.1.8.1 Offset

Register	Offset
ISR	18h

13.6.1.8.2 Function

The ISR register functions as an interrupt status indicator. Each bit indicates whether an interrupt condition has been met for the corresponding input signal. When an interrupt condition is met (as determined by the corresponding interrupt condition register field), the corresponding bit in this register is set. Two wait states are required in read access for synchronization. One wait state is required for reset.

13.6.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISR															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.8.4 Fields

Field	Function
31-0 ISR	Interrupt status bits Bit n of this register is asserted (active high) when the active condition (as determined by the corresponding ICR bit) is detected on the GPIO input and is waiting for service. The value of this register is independent of the value in IMR register. When the active condition has been detected, the corresponding bit remains set until cleared by software. Status flags are cleared by writing a 1 to the corresponding bit position.

13.6.1.9 GPIO edge select register (EDGE_SEL)

13.6.1.9.1 Offset

Register	Offset
EDGE_SEL	1Ch

13.6.1.9.2 Function

The EDGE_SEL register may be used to override the ICR registers' configuration. If the GPIO_EDGE_SEL bit is set, then a rising edge or falling edge in the corresponding signal generates an interrupt. This register provides backward compatibility. On reset all bits are cleared (ICR is not overridden).

13.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPIO_EDGE_SEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO_EDGE_SEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.9.4 Fields

Field	Function
31-0 GPIO_EDGE_SEL	Edge select When EDGE_SEL[*n*] is set, the GPIO disregards the ICR[*n*] setting, and detects any edge on the corresponding input signal.

13.6.1.10 GPIO data register SET (DR_SET)

13.6.1.10.1 Offset

Register	Offset
DR_SET	84h

13.6.1.10.2 Function

The SET register of DR.

13.6.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DR_SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DR_SET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.10.4 Fields

Field	Function
31-0 DR_SET	Set Writing a 1 to a bit in this register sets the corresponding bit in DR

13.6.1.11 GPIO data register CLEAR (DR_CLEAR)

13.6.1.11.1 Offset

Register	Offset
DR_CLEAR	88h

13.6.1.11.2 Function

The CLEAR register of DR.

13.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DR_CLEAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DR_CLEAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.11.4 Fields

Field	Function
31-0	Clear
DR_CLEAR	Writing a 1 to a bit in this register clears the corresponding bit in DR

13.6.1.12 GPIO data register TOGGLE (DR_TOGGLE)

13.6.1.12.1 Offset

Register	Offset
DR_TOGGLE	8Ch

13.6.1.12.2 Function

The TOGGLE register of DR.

13.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DR_TOGGLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DR_TOGGLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.6.1.12.4 Fields

Field	Function
31-0 DR_TOGGLE	Toggle Writing a 1 to a bit in this register toggles the corresponding bit in DR

Chapter 14
Clock and Power Management

14.1 Introduction

This chip targets many applications with power requirements that include low power consumption, long battery life, always-on capability, instant-on capability, and doesn't require active cooling. To meet these requirements, the chip design focuses on reducing current consumption, while simultaneously enabling the maximum level of peak performance, and also a balanced level of sustained performance. The chip architecture uses a wide range of power-management techniques, used in combination, for system design flexibility.

This chapter describes these architecture details as a high-level clock and power management overview of the chip. This overview chapter covers:

- Structural components of the power and clock management systems
- Clock generation and distribution system
- Power generation and distribution system
- Power mode definition
- Power and clock and management techniques supported.

NOTE

All numerical values in this chapter are either typical values or used as an example. Please see the datasheet for accurate values.

14.2 Components of Clock and Power Management

The centralized clock generation, power generation and distribution are implemented by the following blocks:

Table 14-1. Clock and Power Components

IP Module	Description
Crystal OSC (XTALOSC)	The XTALOSC includes a 24 MHz and 32 kHz oscillators. The 24MHz crystal oscillator is the primary clock source for all the PLLs, and clock generation for the CPU and high-speed interfaces. The 32KHz crystal oscillator is the primary clock source for the RTC as well as the low-speed clock source for CCM/SRC/GPC. See the Crystal Oscillator (XTALOSC) chapter for details of the XTALOSC block.
PLLs and PFDs	The PLLs and their associated PFDs generate the clocks with various frequencies required to feed the CCM clock generator that supplies the different functional blocks. See the PLL and PFD section in Clock Controller Module (CCM) for information on the PLL and PFD architecture, functional description and programming model.
Clock Control Module (CCM)	The CCM module provides control for clock generation, division, distribution, synchronization, and coarse-level gating. See Clock Controller Module (CCM) for information on the CCM architecture, functional description and programming model.
Low Power Clock Gating (LPCG)	The LPCG distributes the clocks to all blocks in the SoC and handles automated clock gating and block level software-controllable clock gating. See Clock Controller Module (CCM) for information on the LPCG architecture and functional description.
General Power Controller (GPC)	This module controls the power state of the SoC. The GPC handles the power gating under low power modes and also manages the power up / power down sequences. See the General Power Controller (GPC) chapter for information on the GPC architecture, functional description and programming model.
Power Management Unit (PMU)	This module generates the internal power supplies distributed to the entire SoC. See the Power Management Unit (PMU) chapter for information on the PMU architecture, functional description, and programming model.
System Reset Controller (SRC)	This module generates the reset signals to all the modules in the SoC. The SRC will appropriately assert reset signals for power transitions, entry, and exit. See the System Reset Controller (SRC) chapter for information on the SRC architecture, functional description and programming model.
Power Gating and Memory Controller (PGMC)	This module is a power management component, which controls the power gating of power domains and memory low power mode. See Power Gating and Memory Controller (PGMC) chapter for information on PGMC architecture, functional description, and programming model.
DCDC Converter (DCDC)	This module generates the power supply for chip's core logic. See DCDC converter (DCDC) chapter for information on functional description, and programming model.

Together, the modules listed above provide enhanced power-management features with the centralized control for the clock, reset, and power-management signals on the SoC.

14.3 Power Management

The power management design on the chip satisfies many use cases that depend on a key balance of performance and low-power consumption. In order to meet these criteria, the system power management needs to be dynamic and flexible. This is accomplished on a multi-core system by utilizing a power scheme that considers each CPU domain, and arbitrates the appropriate power mode configuration to satisfy the needs of both CPUs, and the system as a whole. The following are additional features supported by the power management architecture:

- Programmable power mode transition rules
- Independent operation mode for individual CPU cores
- Intelligent clock and power management for multi-core architecture
- Unified power management interface (UPI) for all on-chip resources, including clock, power, reset, and peripherals

A high-level view of the power management controllers and connections is shown below:

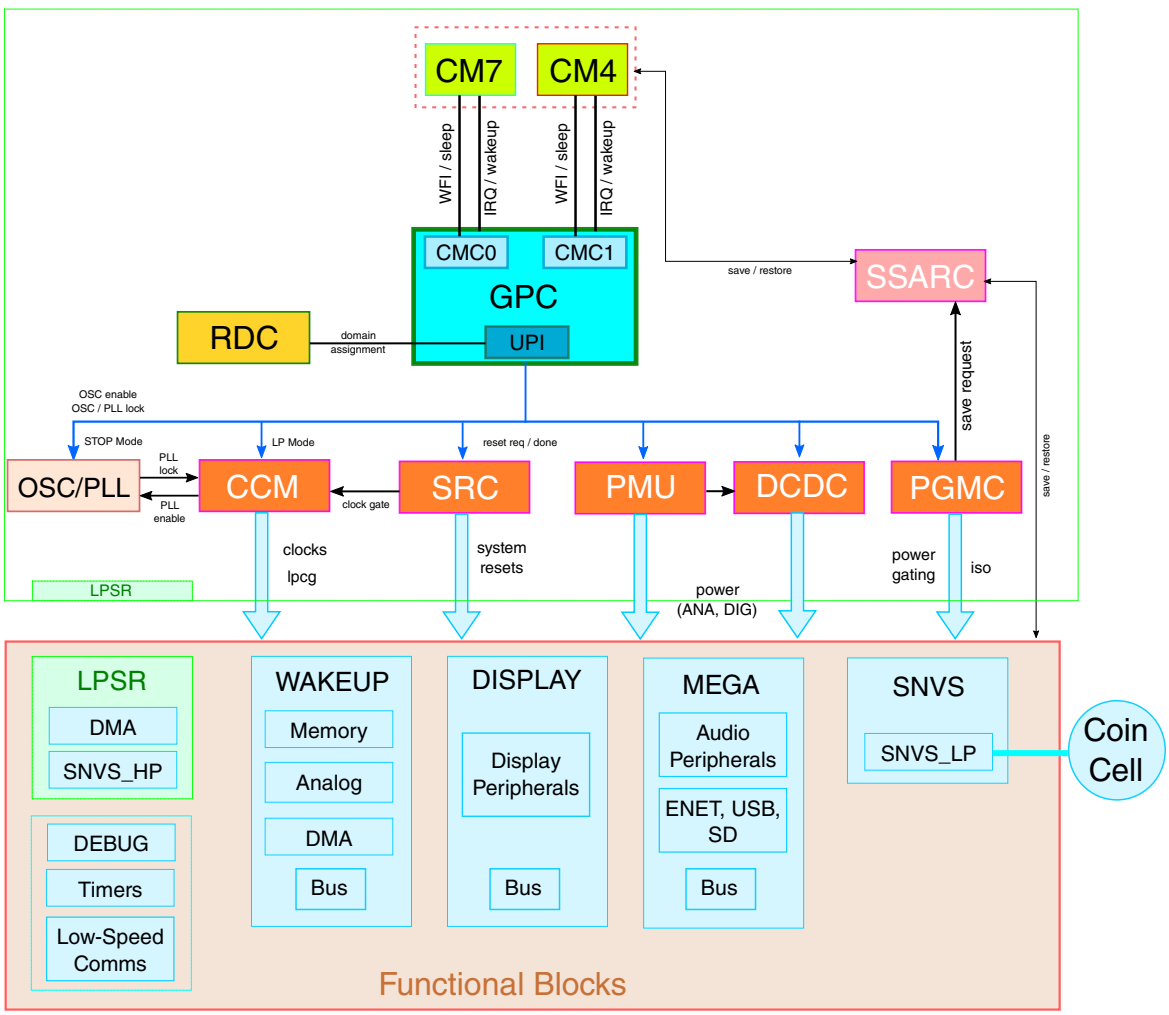


Figure 14-1. Power Management Block Diagram

14.3.1 Resources and Chip Domains

The chip is divided into resources and domains. Resources are controlled by resource controllers. Domains are grouped by central bus controllers and crossbars. Some domains reside within larger domains, which may have power dependencies. For illustrative representation of the power domains and bus configurations, please see the Power Management Block Diagram and the System Bus Diagram. The Resource types and Chip Domains are listed in the tables below:

Table 14-2. Resources

Resource	Description
CPU Platforms	The processor and associated on-platform peripherals that support the complex. The CPU Platforms supported on this chip are the CM7 and the CM4 platforms.
System Controllers	The System Controllers are the associated peripherals that control or define the power management resources for the chip. These modules include GPC, PMU, CCM, SRC, SSARC, PGMC, and RDC.
Functional blocks	The functional blocks are comprised of all the other peripherals, other than System Controllers. These modules include external memory controllers, timers, high-speed and low-speed peripherals, audio and video peripherals, connectivity, and others.

Table 14-3. Chip Domains

Domain	Description
CM7	The Cortex-M7 core and platform, which includes FlexRAM (TCM) and GPIO2-3.
CM4	The Cortex-M4 core and platform, which includes LMEM (TCM).
WAKEUP	The WAKEUP Domain contains the memory controllers and analog on the chip. The CM7 platform (incl JTAGC and debug components) also reside within WAKEUP, but the CM7 functions as it's own domain. The specific modules in WAKEUP include: EWM, WDOG1-3, eDMA, DMAMUX0 (WAKEUP), ACMP1-4, LPADC1-2, ADC_ETC, DAC, XBAR1-3, KPP, IOMUXC, IOMUXC_GPR, GPIO1-6, LPUART1-10, MTR, FLEXCAN1-2 (CANFD), QDC1-4, PIT1, AOI1-2, GPT1-6, TMR1-4, FlexPWM1-4, LPI2C1-4, LPSPI1-4, FlexIO1-2, EMVSIM1-2, FlexSP1-2, SEMC, MECC1-2, XECC (XECC_FLEXSPI1-2, XECC_SEMC), IEE.
MEGA	Contains the connectivity peripherals and audio peripherals. The specific modules include: ENET1G, ENET, ENET QoS, uSDHC, USB, CAAM, ASRC, MQS, SAI1-3, SPDIF.
DISPLAY	The DISPLAY Domain contains the display and graphics peripherals on the chip. The specific modules include: GPU2D, PXP, eLCDIF, LCDIFv2, CSI, MIPI_CSI2, MIPI_DSI, VIDEO_MUX (incl DCIC1,2)
LPSR	The LPSR Domain contains the System Control blocks and associated peripherals. The CM4 platform and portions of SNVS (SNVS_LP) also reside within LPSR, but function as their own domains. The specific modules of LPSR include: GPC, PGMC, DCDC, SRC, SSARC, PMU (ANADIG), TEMPSENSE, XTALOSC, CCM, OCOTP, SNVS_HP, KEYMGR, WDOG (CM4), IOMUXC_SNVS, IOMUXC_LPSR, MU, XRDC2_D0, XRDC2_D1 (CM4, CM7), RDC, SEMA4, SEMA42, eDMA, DMAMUX1 (LPSR), GPIO7-12, PDM, SAI4, PIT2, LPUART11-12, LPSPI5-6, LPI2C5-6, FLEXCAN3 (CANFD)

Table continues on the next page...

Table 14-3. Chip Domains (continued)

Domain	Description
SNVS	The SNVS Domain contains SNVS_LP, which connects to a coin cell or non-interruptible power source. This domain also contains IOMUXC_SNVS, SRAM, and GPIO13.

14.3.2 Power Management Unit (PMU)

The PMU has the following components integrated for power management:

- DCDC
  - Generates the core power supply
  - Supports dynamic voltage scaling
- LDOs
  - Internal logic power supply
- Power Switches for sophisticated power mode management

The typical power architecture of the chip is shown below. The use case illustrated shows the whole system working with a single 3.3V power supply and a coin cell. This is the power system configuration of traditional MCU. Depending on the application, other power supply schemes may be used, including bypassing the DCDC converter and using an external voltage regulator or PMIC.

NOTE

The on-chip DCDC converter is suitable for consumer and industrial applications up to the rated temperature specified in their respective market datasheets. For automotive applications, please check the automotive datasheet or consult with your NXP representative for appropriate power supply requirements.

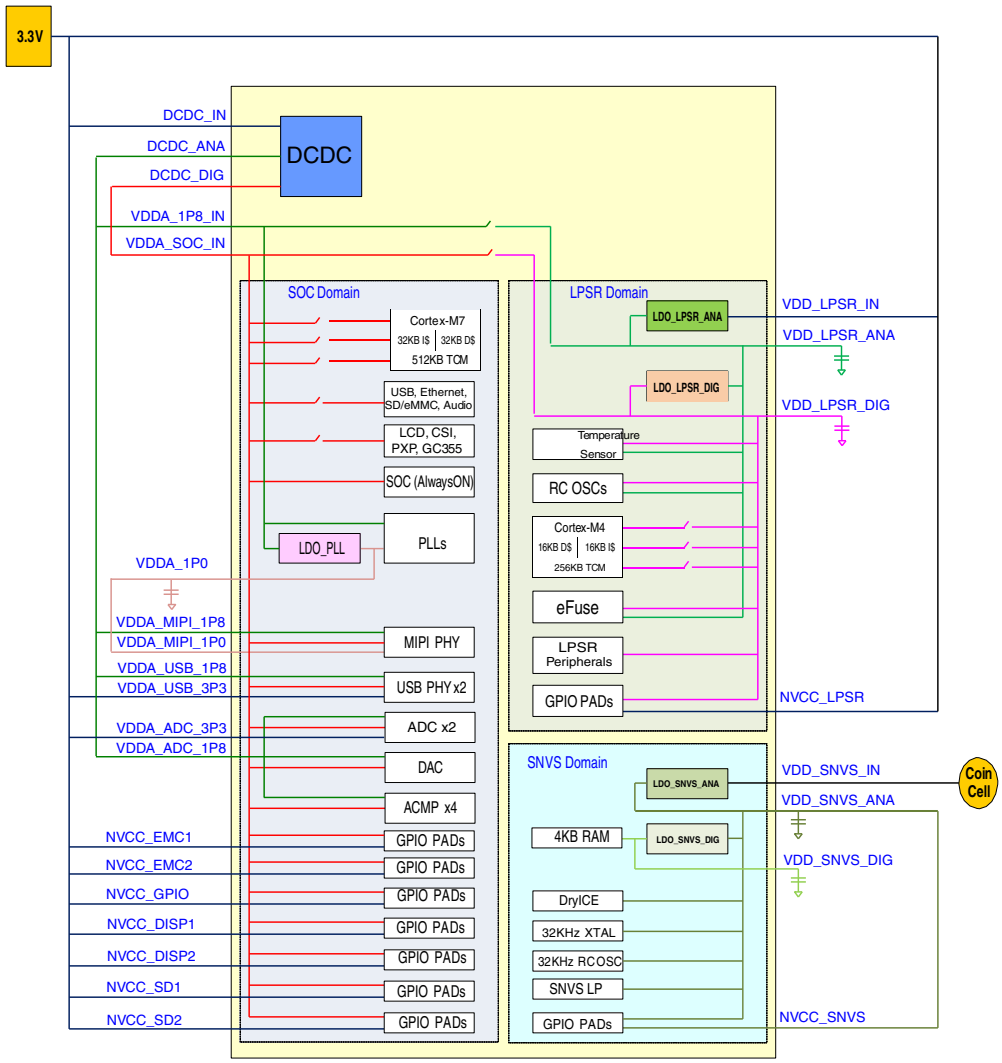


Figure 14-2. Power Architecture

14.3.3 Body Biasing (BB)

Forward Body Biasing (FBB) and Reverse Body Biasing (RBB) are implemented for performance boost and power reduction. The list and figure below define the power domains that support BB.

- FBB is implemented in the CM7 platform in order to achieve higher performance.
- RBB is implemented in the LPSR domain to reduce power consumption in LPSR mode.
- RBB is implemented in the SoC to reduce power consumption in low-power modes.

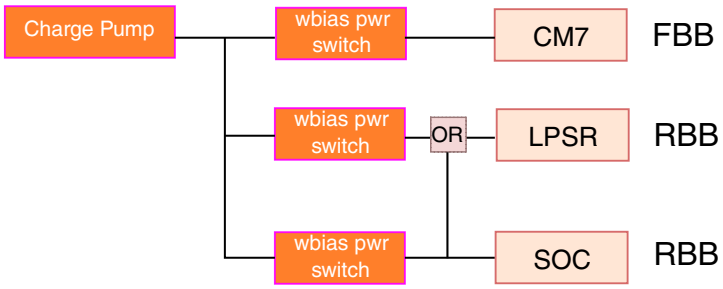


Figure 14-3. Supported Body Biasing

NOTE

Software must ensure that FBB and RBB are not enabled simultaneously. Only one of them can be used at a time.

14.3.4 Power Modes

The chip resources abide by CPU or system-level power modes, depending on their resource assignment. Resources are assigned to one of four categories, which are defined below.

Table 14-4. Resources Categories

Category	Description
Private	A resource that are fully owned and controlled by one CPU platform. These resources can be controlled by hardware and software.
Shared	A resource that is shared by more than one CPU platform. These resources are limited by the other CPU power state. Software must take into regard, shared resources, since they are limited by other CPU power states. If a shared resource supports Setpoint control, then it is recommended that the resource is under Setpoint control, rather than CPU platform control.
Public	A resource that is managed at the system-level, which is not owned and controlled by any of the CPU platforms. These resources can only be controlled by hardware.
Unassigned	A resource that is not assigned as private, shared, or public. This is the default state of a resource after reset.

Resources can be assigned to different CPU domains by the RDC. If one CPU wants to put the chip into a power mode, it must arbitrate with the other CPU to ensure the appropriate power state is chosen for shared resources. Depending on the resource type, varying power control mechanisms are followed. Private and Shared resources follow the power modes defined by the CPU (CPU mode). Public resources are configured by system-level settings (Setpoint mode).

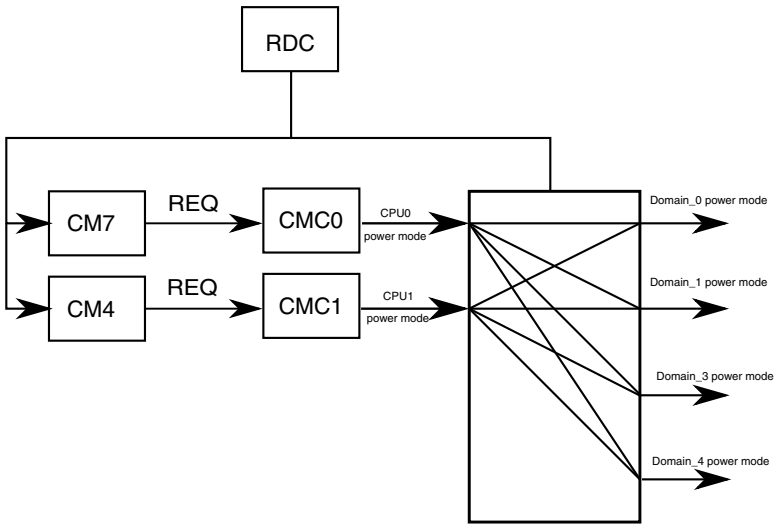


Figure 14-4. Domain Assignment

14.3.4.1 CPU Mode

The CPU platforms both support the same power states that are defined as CPU modes. The CPU Modes are detailed below.

Table 14-5. CPU Mode

CPU Mode	Description
RUN	The CPU is active and running.
WAIT	The CPU is in the WFI state. The CPU core and L1 cache are clock gated, TCM and peripheral can still be alive. The chip can wakeup from this mode from IRQ with a very short latency.
STOP	The CPU is in the WFI state. The CPU core, L1 cache, TCM, and it's private peripherals are all clock gated. The chip requires a longer time to wakeup from this mode.
SUSPEND	The CPU core and L1 cache are power gated (except the M4 cache). The TCM is in retention mode. SUSPEND mode consumes the least amount of power, while keeping parts of the the system alive. But the chip requires a much longer time to wakeup from this mode.

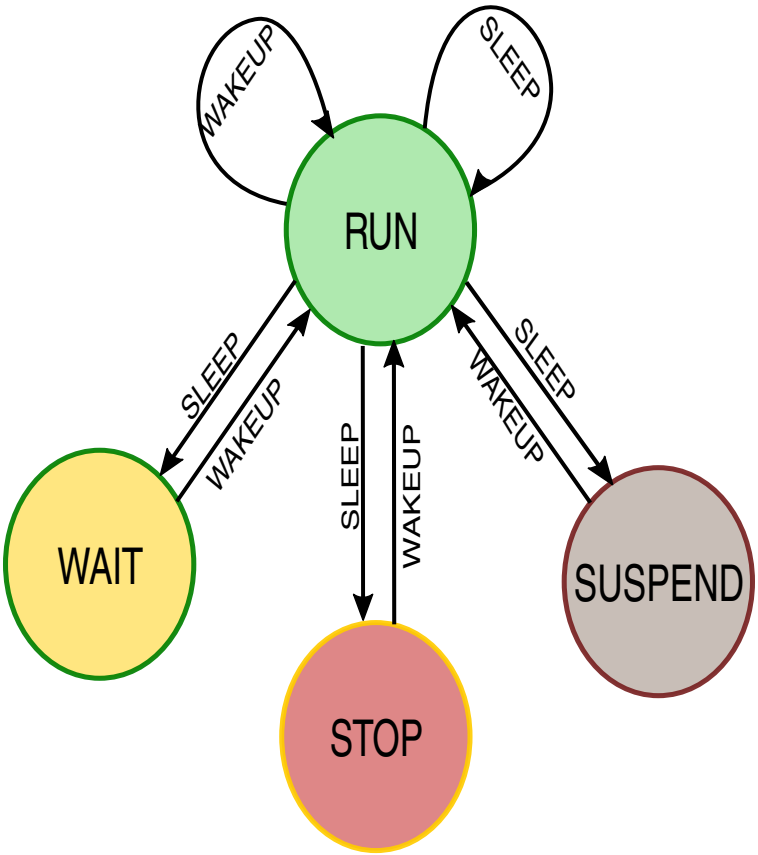


Figure 14-5. CPU Modes

For the modules / resources outside (public) of the CPU platform domain, the power is managed by Setpoint mode. There are up to 16 Setpoints supported. In each Setpoint, the state of the modules is configurable. The power mode of the chip is a combination of the Setpoint and CPU mode. Please refer to the specific System Controller blocks / modules for more information on their specific configuration options for Setpoint mode.

14.3.4.1.1 Memory Power Connection

A few of the power connections that exist in this chip are shown below:

- M7 Cache and TCM memory:
  - M7 core and memory peripheral logics are on the same power domain, which are on/off together
  - When operating, memory array is supplied from VDD_SOC_IN
  - Supported modes
    - M7 on, cache and TCM on
    - M7 off, cache off, TCM off
  - No access to TCM memory when M7 core is power gated
- M4 Cache & TCM memory:
  - Supported modes

- M4 on, cache and TCM on
- M4 off, cache and TCM on
- M4 off, cache on, TCM off

- OCRAM memories:
  - Controller and memory peripheral logics are on the same power domain, which are always on when VDD_SOC_IN is there
  - It will only be off when then entire SOC domain is off

14.3.4.2 Setpoint Mode

Setpoints are preconfigured power states (up to 16) that are defined within the System Controller resources. Resources that aren't owned by either CPU, follow Setpoint mode to define the system-level settings for clocking, power, BB, and reset. Please see the respective System Controller chapters for more details on the specifc controls supported by Setpoint mode.

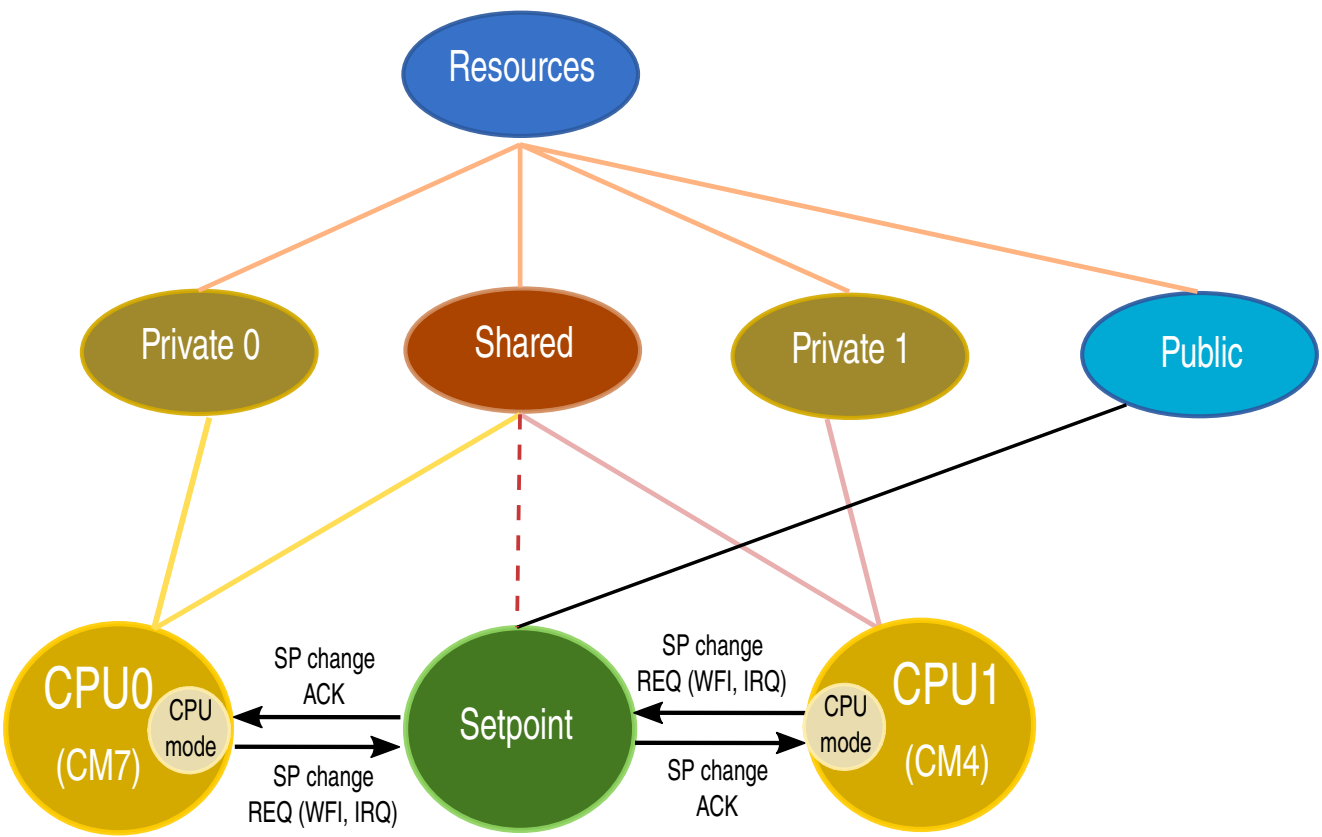


Figure 14-6. Setpoint Mode

14.3.4.2.1 Setpoint Arbitration

The power mode of the CPU platforms is independent of the other. But the shared resources must have a target Setpoint and a lookup table of allowed Setpoints for each CPU mode transition. The GPC will arbitrate between the target Setpoint, and allowed Setpoints, in order to determine the appropriate and accepted Setpoint setting. The lookup tables in GPC are a 16x16 bit table, that work for one domain. If more than one Setpoint is allowed, control logic first checks if any Setpoints match the request from CM7 or CM4. If both match, the selection is done, and based on priority of CPUs. If none match, logic will compare priority of Setpoints, and select the high-priority Setpoint. If the priority is the same, Setpoint in the biggest value will be selected.

14.3.4.3 Standby Mode

Standby mode is the third low-power mode, besides CPU mode and Setpoint mode. Standby mode is related to the state of both CPU platforms, and has a much shorter transition time than Setpoint. The GPC SBC state machine starts when CMC sends a standby request. It looks at all CMC standby statuses, and if any CPU platform is not in sleep mode, this standby request will be refused. CPU mode and Setpoint transition can't be broken by a wakeup event once it starts. A key benefit of Standby mode, is the Standby transition can be broken after any step completes, due to a wakeup requirement.

14.3.5 Power Sequence

The software flow for low-power entry and exit are listed below.

NOTE

The power flow assumes all resource assignments are complete after reset

1. CPU decides to enter any low power mode according its current loading
2. Software handshaking: set ipg_stop/doze in IOMUXC_GPR register; polling corresponding ips_stop_ack
   - Note: DMA/ENET (bus masters) should complete the handshake first, then flash/memories (bus slaves)
3. Set CPU target mode and system target/allowed setpoint in GPC registers
4. WFI/WFE
   - CPU mode transition stage: SSARC restore database; clock/power of CPU private resources are shut off
   - Setpoint transition stage: clock/power of public resource are shut off
   - System enter target low power state and waiting for wakeup event

5. Wakeup event comes
    - Setpoint transition stage: power recover, then clock recover for public resources
    - CPU transition stage: power recover, then clock recover for CPU private resources; SSARC recover database
    - CPU wakeup when core clock is recovered
6. CPU enter wakeup event ISR and recover the field before entering low power state

Please see GPC for a detailed sequence.

14.4 Clock Generation

The clock system is designed to provide a scalable, powerful, and easy-to-use clock solution.

14.4.1 Overview

The overall clock system for the chip is shown in the diagram below.

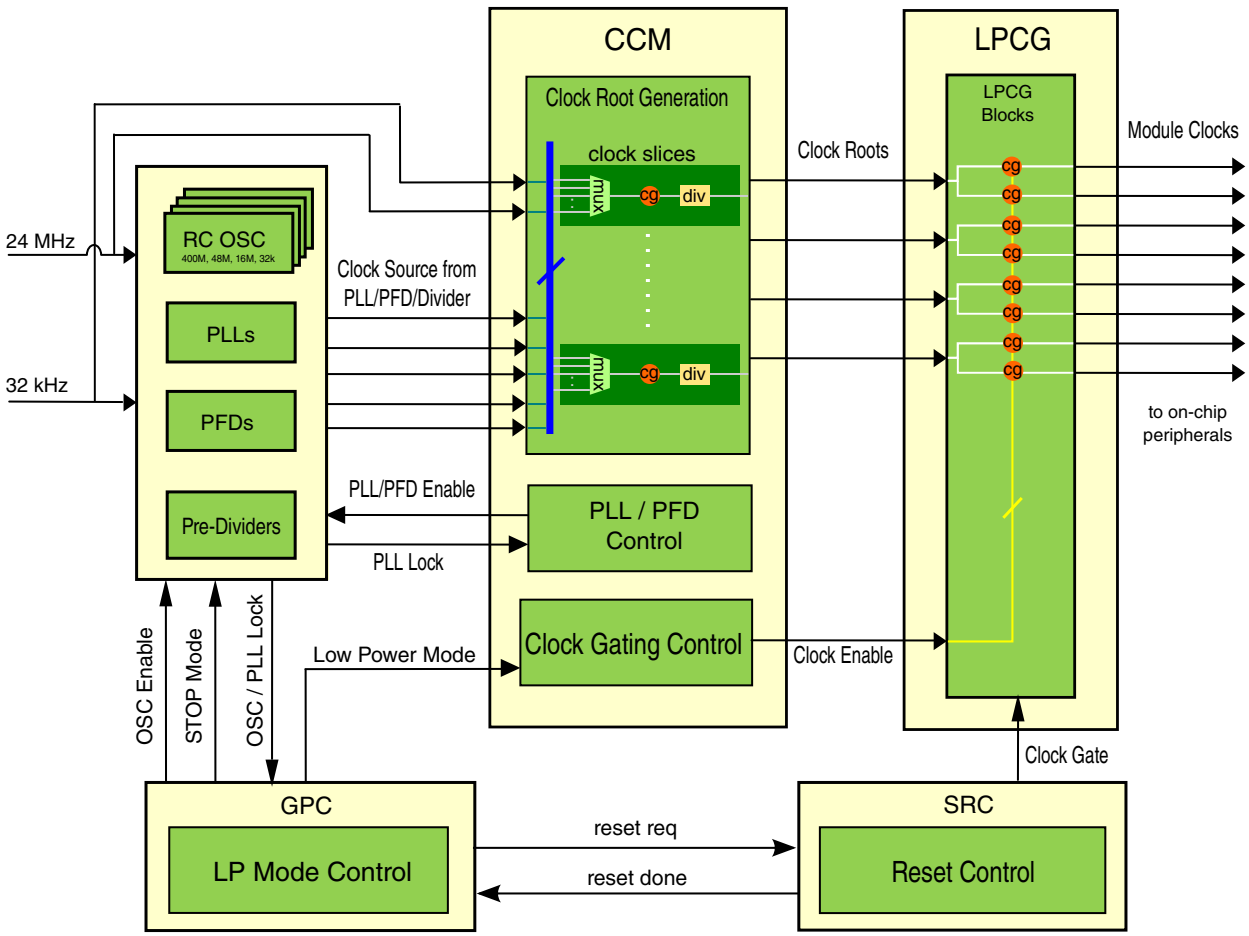


Figure 14-7. Clock System

The OSCs, PLLs, PFDs and Pre-Dividers generate the clock sources with fixed or variable frequency. The clock sources feed the clock root generation logic inside CCM, which generates numerous clock roots required for core, bus, and peripheral blocks. These clock roots are branched to each module through the LPCG logic, which contains the clock gating cells for each clock. Control signals for the LPCGs are in CCM (clock enable signal). Some of the clocks need to be gated off during reset, because of this, the SRC module will also send clock gate signals to the LPCG.

In low power mode, GPC will drive the low power mode state signal to CCM, then CCM will enable clock gating based on the configuration. At the same time, CCM may also deassert enable signals for PLLs and PFDs, so these modules can power off during low power modes. When the chip enters STOP mode, CCM may deassert the OSC enable signal to shut off the OSC, it may also assert the STOP mode signal to set OSC to low power mode.

The detailed functional description for each block will be described in the following sections.

14.4.2 Oscillator

This chip has two crystal oscillators and four RC oscillators:

Table 14-6. Oscillators

Oscillator	Description
24 MHz Crystal Oscillator (24M XTALOSC)	The primary clock source for all PLLs to generate the clocks for CPU, bus, and high-speed interfaces
32 kHz Crystal Oscillator (32K XTALOSC)	The primary clock source for RTC as well as low-power clock source for CCM, SRC, and GPC. The typical frequency is 32.768KHz.
48 MHz RC Oscillator (48M RCOSC)	Generates the 24 MHz clock source for the chip during startup, before the 24MHz crystal oscillator is ready. The 48M RCOSC can also be used as an alternative 24 MHz clock source in low power mode, when the crystal oscillator is turned off, or in a system that doesn't have a crystal oscillator.
32 kHz RC Oscillator (32K RCOSC)	The 32 kHz clock source for the chip during startup when 32 kHz crystal is not ready. It can also be used as alternative 32 kHz clock source when the external 32 kHz crystal oscillator is not stable, or in system which does not have any crystal oscillator. The switch from external 32 kHz to internal 32 kHz RC is controlled by hardware, and it happens automatically when the system detects a loss of the 32 kHz crystal clock.
16 MHz RC Oscillator (16M RCOSC)	The clock source in low power mode.
400 MHz RC Oscillator (400M RCOSC)	The clock source during chip boot up, before PLLs are enabled, or in low-power mode when the PLLs are turned off.

See Crystal Oscillator (XTALOSC) chapter for functional description details and the programming model on these blocks.

14.4.3 PLL and PFD

The PLLs are used in the chip to generate the high speed clocks required by the modules in SoC. A few PLLs are equipped with Phase Fractional Dividers (PFDs) in order to generate additional frequencies. The following PLLs are used in this chip:

Table 14-7. PLLs

PLL	Description	Spread Spectrum	Configurable
ARM_PLL	The ARM_PLL is used to generate the clock for the Arm Cortex-M7 platform. It is a programmable integer-frequency multiplier PLL, capable of an output frequency over 1 GHz.	-	Yes
SYS_PLL1	An AVPLL that is used primarily by the ENET module. It has a dedicated frequency of 1GHz.	Yes	1 GHz

Table continues on the next page...

Table 14-7. PLLs (continued)

PLL	Description	Spread Spectrum	Configurable
SYS_PLL2	The SYS_PLL2 main output clock has a dedicated frequency of 528MHz, but has several configurable PFDs.	Yes	528 MHz
SYS_PLL3	The SYS_PLL3 main output clock has a dedicated frequency of 480MHz, but has several configurable PFDs.	-	480 MHz
AUDIO_PLL	An AVPLL used to generate reference clocks, mainly for serial audio interfaces, and external audio codecs. This fractional multiplier PLL provides a low-jitter and high precision audio clock with standardized audio frequencies. The PLL's oscillator frequency range is from 650MHz to 1300MHz, and the frequency resolution is better than 1Hz.	Yes	Yes
VIDEO_PLL	An AVPLL used to generate reference clocks, mainly for display and video interfaces. This fractional multiplier PLL provides a low-jitter and high-precision video clock at standardized video frequencies. The PLL's oscillator frequency range is from 650MHz to 1300MHz, and the frequency resolution is better than 1Hz.	Yes	Yes

NOTE

Please see the datasheet for all supported maximum PLL frequencies

The following PLLs also feature the following dedicated dividers, which provides fixed frequencies for system usage:

- SYS_PLL1_DIV2: dedicated 500MHz clock
- SYS_PLL1_DIV5: dedicated 200MHz clock
- SYS_PLL3_DIV2: dedicated 240MHz clock

Each PLL can use one of the following clock sources as it's reference clock:

- Primary Option: 24 MHz clock from 24 MHz oscillator (XTAL)
- Secondary Option: 24 MHz clock from 48 MHz RCOSC (divided by 2)

Only the AUDIO_PLL and VIDEO_PLL have the capability to spread spectrum the generated signal. They do not require exact/constant frequency, and can be changed as a part of dynamic frequency scaling procedure and/or can be spread-spectrum modulated.

The main output clocks for the System PLLs (SYS_PLLx_CLK) are not configurable, but their PFDs are. Each PFD works independently by interpolating the VCO of the PLL to which it is connected. It effectively takes the PLL VCO frequency and produces 18/N x Fvco at it's output where N ranges from 12 to 35. PFD is a completely digital design with no analog components or feedback loops. The frequency switch time is much faster than a PLL because keeping the base PLL locked and changing the integer N only

changes the logical combination of the interpolated outputs of the VCO. Note that the PFD not only enables faster frequency changes than a PLL, but also allows the configuration to be safely changed "on-the-fly" without going through the output clock disabling/enabling process.

The basic logical connections for the PLLs, PFDs and Dividers is shown in the diagram below:

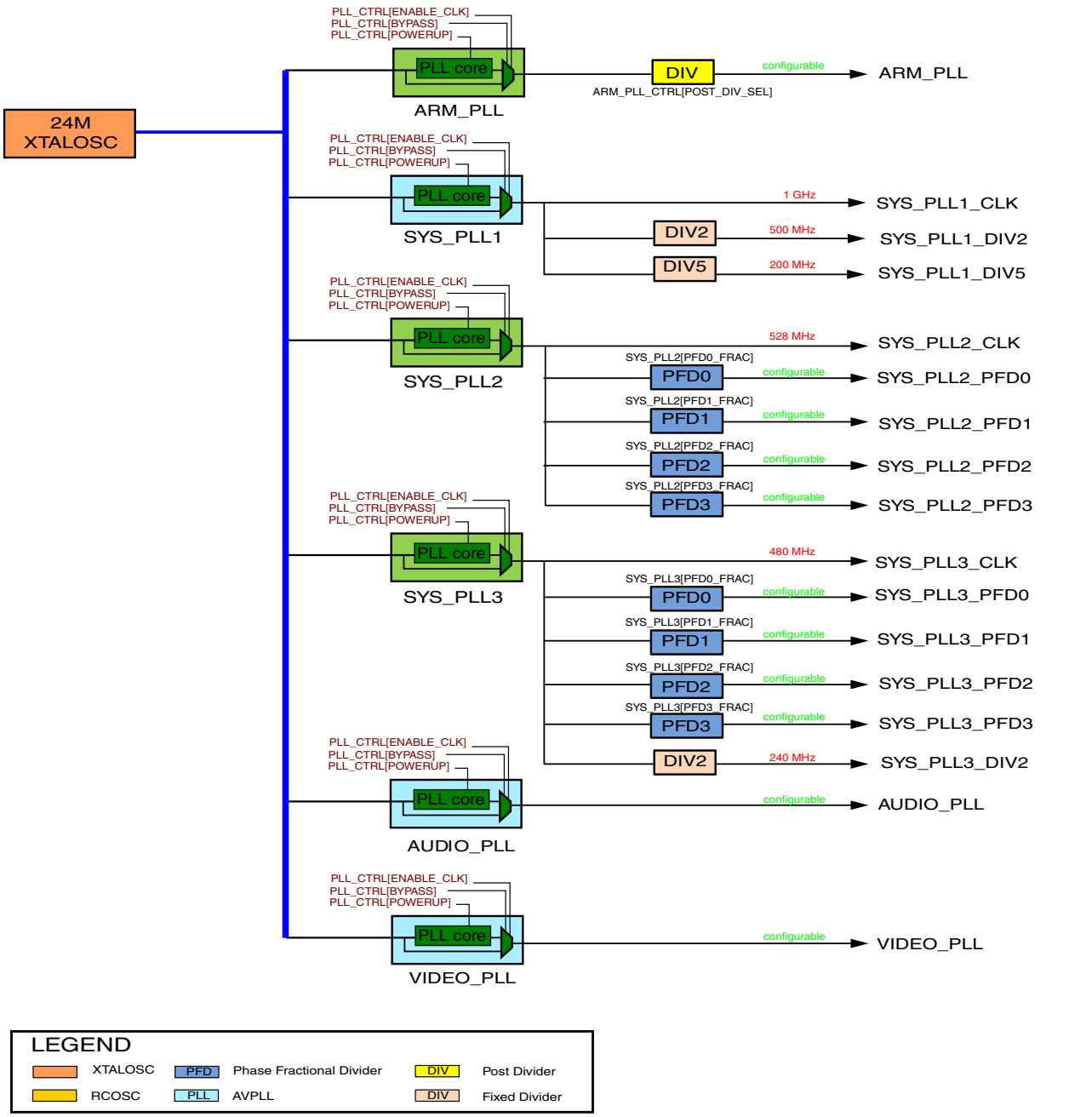


Figure 14-8. PLL and PFD

PLL configuration and control functions are accessible via individual PLL and PFD configuration and status registers. Each PLL has a dedicated configuration register. Software can directly enable / disable the PLL, or change it's clock frequency. In addition to the register interface, each PLL also has a dedicated ENABLE input, and a LOCK

output, to support the PLL disable / enable control from CCM. The CCM block contains the detailed descriptions of the memory mapped registers, and control functions of the clock generation sub-module. Please see CCM for more information.

14.4.4 Clock Root Generation

Clock generation in the CCM creates clock roots for on-chip peripherals. Each clock root will have a dedicated clock slice. It takes the clock source from Oscillators, PLL/PFD or Pre-Dividers, and generate the clock root with required frequency.

The CCM can manage on-chip peripheral clocks by controlling the low power clock gating cells (LPCGs) before the clock roots enter the peripherals.

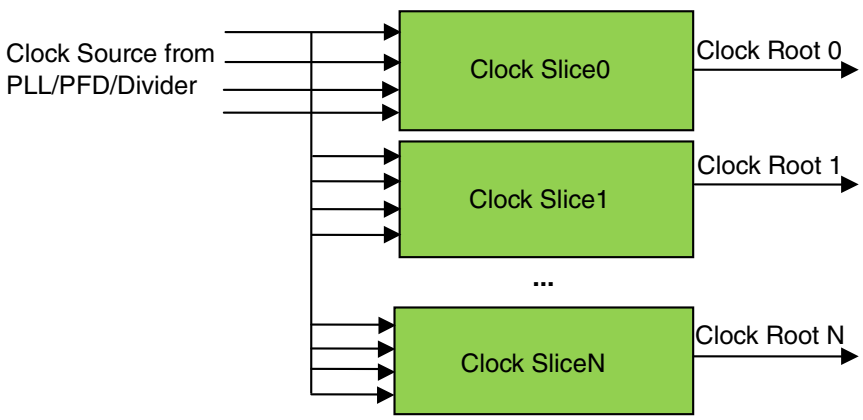


Figure 14-9. Clock Root Generation from Clock Slice

The LPCGs are implemented to control peripheral functions, and there could be several LPCGs for each functions in a peripheral.

For details on the programming model, please refer to CCM Memory Map/Register Definition.

14.4.5 Low Power Clock Gating (LPCG)

The clock roots generated inside CCM are distributed to the entire SoC through LPCG. The LPCG block receives the root clocks from CCM and splits them to clock branches for each functional block. The clock branches are individually gated clocks.

The LPCG can be programmed directly or controlled by CPU Low-Power Mode (CPULPM) according to the low-power signal from GPC. The settings for LPCG in CPULPM are defined by a 3-bit clock dependent level setting in the LPCGx_DOMAIN register. The configuration of the LPCGx_DOMAIN[LEVELn] settings are as follows:

- 3b’000: The clock source is not needed in any mode, and can be turned off
- 3’b001: The clock source is needed in RUN mode, but not needed in WAIT or STOP mode
- 3’b010: The clock source is needed in RUN and WAIT mode, but not needed in STOP mode
- 3’b011: The clock source is needed in RUN, WAIT, and STOP mode
- 3’b100: The clock source is always on in any mode (including SUSPEND)
- Others: Reserved

Chapter 15
Clock Controller Module (CCM)

15.1 Chip-specific CCM information

Table 15-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

- SYS_PLL1 has a fixed frequency of 1GHz and is primarily used by the ENET modules
- SYS_PLL2 has a fixed frequency of 528MHz
- SYS_PLL3 has a fixed frequency of 480MHz

15.2 Overview

The Clock Control Module (CCM) manages the on-chip module clocks. The oscillators, PLLs, and Phase Fractional Dividers (PFD) will generate clock sources with fixed or variable frequencies.

15.2.1 Block diagram

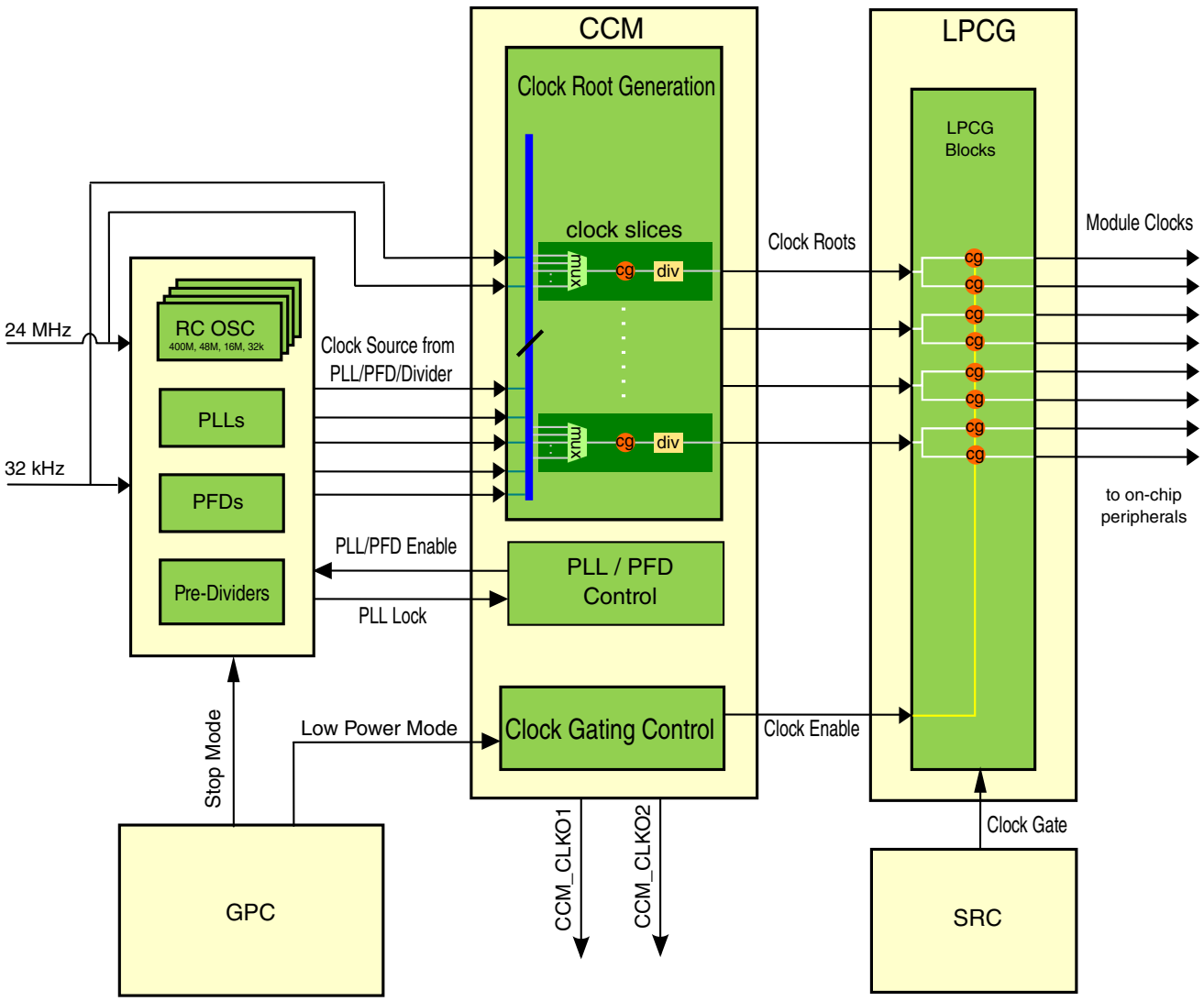


Figure 15-1. CCM block diagram

The clock root generation logic inside CCM will generate various clock roots required for core, bus, and peripheral blocks. These clock roots will be delivered to each module through LPCG, which contains the clock gating logic for each clock.

The control signal for LPCGs in CCM will be the source for clock enable signals. Since some of the clocks need to be gated off during the reset, the System Reset Controller (SRC) will also send clock gate signals to the LPCG.

In low power mode, the General Power Controller (GPC) will drive the low power mode state signal to CCM, and CCM will enable clock gating based on the configuration. At the same time, CCM may also de-assert enable signals for PLLs, so these modules can be powered off during the low power mode.

15.2.2 Features

The CCM includes the following features:

- Clock Root
  - Supports 8 clock sources for each clock root
  - 8-bit divider supports up to divide by 256 in each clock root
  - Each clock root can be shutdown by software
  - 4-bit fractional divide (numerator and denominator)
  - 16 preset clock root Setpoints transitions on power mode change
- Clock Group
  - Synchronous clock group where each clock can change on-the-fly
  - Clock group can be shutdown by software
  - 16 preset clock group Setpoint transitions on power mode change
- Peripheral clocks can be auto gated according to CPU platform, software, or system power mode
- PLLs and oscillators can be auto shutdown according to CPU platform, software, or system power mode
- Two access control schemes: Trustzone, and Domain
  - Each component can be protected independently
  - Independent setting for user, privileged, secure, non-secure
  - Whitelist that can control Domain Mode access
  - Access control setting can be locked independently

15.3 System Clocks

The table found here shows the CCM output clocks' system-level connectivity.

Table 15-2. System Clocks Table

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
ACMPn	acmp1_fast_clk	ACMP_CLK_ROOT	clk_enable_acmp1 (LPCG58)
	acmp1_ipg_clk	BUS_CLK_ROOT	clk_enable_acmp1 (LPCG58)
	acmp1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_acmp1 (LPCG58)
	acmp1_slow_clk	OSC_32K	clk_enable_acmp1 (LPCG58)
	acmp2_fast_clk	ACMP_CLK_ROOT	clk_enable_acmp2 (LPCG59)
	acmp2_ipg_clk	BUS_CLK_ROOT	clk_enable_acmp2 (LPCG59)
	acmp2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_acmp2 (LPCG59)
	acmp2_slow_clk	BUS_CLK_ROOT	clk_enable_acmp2 (LPCG59)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	acmp3_fast_clk	ACMP_CLK_ROOT	clk_enable_acmp3 (LPCG60)
	acmp3_ipg_clk	BUS_CLK_ROOT	clk_enable_acmp3 (LPCG60)
	acmp3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_acmp3 (LPCG60)
	acmp3_slow_clk	BUS_CLK_ROOT	clk_enable_acmp3 (LPCG60)
	acmp4_fast_clk	ACMP_CLK_ROOT	clk_enable_acmp4 (LPCG61)
	acmp4_ipg_clk	BUS_CLK_ROOT	clk_enable_acmp4 (LPCG61)
	acmp4_ipg_clk_s	BUS_CLK_ROOT	clk_enable_acmp4 (LPCG61)
	acmp4_clk	BUS_CLK_ROOT	clk_enable_acmp4 (LPCG61)
ADC_ETC	adc_etc_ipg_clk	BUS_CLK_ROOT	clk_enable_adc_etc (LPCG48)
AIPS	aips_tz1_hclk	BUS_CLK_ROOT	clk_enable_sim_per (LPCG5)
	aips_tz1_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_per (LPCG5)
	aips_tz2_hclk	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	aips_tz2_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	aips_tz3_hclk	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	aips_tz3_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	aips_tz4_hclk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	aips_tz4_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
AOIn	aoi1_ipg_clk	BUS_CLK_ROOT	clk_enable_aoi1 (LPCG46)
	aoi2_ipg_clk	BUS_CLK_ROOT	clk_enable_aoi2 (LPCG47)
ARM CM4	cm4_imxrt_FCLK	M4_CLK_ROOT	clk_enable_cm4 (LPCG1)
	cm4_imxrt_HCLK	M4_CLK_ROOT	clk_enable_cm4 (LPCG1)
	cm4_imxrt_mmcau_hclk	M4_CLK_ROOT	clk_enable_cm4 (LPCG1)
	cm4_imxrt_tcmc_hclk	M4_CLK_ROOT	clk_enable_lpsrmem (LPCG27)
	cm4_lp_gasket_clk_free_running	M4_CLK_ROOT	-
	nmi_glue_cm4_fclk	M4_CLK_ROOT	clk_enable_cm4 (LPCG1)
ARM CM7	cm7_imxrt_ccm_cm7_clk_root	M7_CLK_ROOT	clk_enable_cm7 (LPCG0)
	cm7_imxrt_cm7_core_stclk	M7_SYSTICK_CLK_ROOT	clk_enable_cm7 (LPCG0)
ASRC	asrc_asrck_clock_a	SAI2_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_asrck_clock_b	SAI3_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_asrck_clock_c	SAI4_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_asrck_clock_d	MIC_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_asrck_clock_e	MQS_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_ipg_clk	ASRC_CLK_ROOT	clk_enable_asrc (LPCG119)
	asrc_mem_ipg_clk	ASRC_CLK_ROOT	clk_enable_asrc (LPCG119)
CAAM	caam_aclk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_exsc_aclk_exsc	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	caam_exsc_ipg_clk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_ipg_clk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_ipg_clk_s	BUS_CLK_ROOT	clk_enable_caam (LPCG40)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	caam_kma_caam_ips_clk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_mem_caam_clk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_rng_async_bridge_clock	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
	caam_snvs_ipg_clk	BUS_CLK_ROOT	clk_enable_caam (LPCG40)
CCM	ccm_ipg_clk	RCOSC_16M	clk_enable_ccm (LPCG10)
	ccm_ipg_clk_s	RCOSC_16M	clk_enable_ccm (LPCG10)
CANn	can1_ipg_clk	BUS_CLK_ROOT	clk_enable_can1 (LPCG83)
	can1_ipg_clk_chi	BUS_CLK_ROOT	clk_enable_can1 (LPCG83)
	can1_ipg_clk_pe	CAN1_CLK_ROOT or BUS_CLK_ROOT	clk_enable_can1 (LPCG83)
	can1_ipg_clk_pe_nogate	CAN1_CLK_ROOT or BUS_CLK_ROOT	clk_enable_can1 (LPCG83)
	can1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_can1 (LPCG83)
	can1_osc_clk	OSC_24M	clk_enable_can1 (LPCG83)
	can1_ipt_clk	OSC_24M	clk_enable_can1 (LPCG83)
	can2_ipg_clk	BUS_CLK_ROOT	clk_enable_can2 (LPCG84)
	can2_ipg_clk_chi	BUS_CLK_ROOT	clk_enable_can2 (LPCG84)
	can2_ipg_clk_pe	CAN2_CLK_ROOT or BUS_CLK_ROOT	clk_enable_can2 (LPCG84)
	can2_ipg_clk_pe_nogate	CAN2_CLK_ROOT or BUS_CLK_ROOT	clk_enable_can2 (LPCG84)
	can2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_can2 (LPCG84)
	can2_osc_clk	OSC_24M	clk_enable_can2 (LPCG84)
	can2_ipt_clk	OSC_24M	clk_enable_can2 (LPCG84)
	can3_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_can3 (LPCG85)
	can3_ipg_clk_chi	BUS_LPSR_CLK_ROOT	clk_enable_can3 (LPCG85)
	can3_ipg_clk_pe	CAN3_CLK_ROOT or BUS_LPSR_CLK_ROOT	clk_enable_can3 (LPCG85)
	can3_ipg_clk_pe_nogate	CAN3_CLK_ROOT or BUS_LPSR_CLK_ROOT	clk_enable_can3 (LPCG85)
	can3_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_can3 (LPCG85)
	can3_osc_clk	OSC_24M	clk_enable_can3 (LPCG85)
	can3_ipt_clk	OSC_24M	clk_enable_can3 (LPCG85)
CDOG	cdog_ahb_clk	BUS_LPSR_CLK_ROOT	clk_enable_cdog (LPCG116)
CSI	csi_csi_hclk	BUS_CLK_ROOT	clk_enable_csi (LPCG133)
	csi_ipg_clk	BUS_CLK_ROOT	clk_enable_csi (LPCG133)
	csi_ipg_clk_s	BUS_CLK_ROOT	clk_enable_csi (LPCG133)
	csi_ipg_clk_s_raw	BUS_CLK_ROOT	clk_enable_csi (LPCG133)
DAC	dac_ipg_clk	BUS_CLK_ROOT	clk_enable_dac (LPCG57)
	dac_ipg_clk_s	BUS_CLK_ROOT	clk_enable_dac (LPCG57)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
DMA_CH_MUXn	dma_ch_mux_ipg_clk	BUS_CLK_ROOT	clk_enable_edma (LPCG22)
	dma_ch_mux_ipg_clk_s	BUS_CLK_ROOT	clk_enable_edma (LPCG22)
	dma_ch_mux_lpsr_ipg_clk	M4_CLK_ROOT	clk_enable_edma_lpsr (LPCG23)
	dma_ch_mux_lpsr_ipg_clk_s	M4_CLK_ROOT	clk_enable_edma_lpsr (LPCG23)
DCDC	dcdc_clk_24M	RCOSC_16M	clk_enable_dcdc (LPCG8)
	dcdc_ipg_clk	RCOSC_16M	clk_enable_dcdc (LPCG8)
eDMAn	edma_hclk	BUS_CLK_ROOT	clk_enable_edma (LPCG22)
	edma_ipg_clk	BUS_CLK_ROOT	clk_enable_edma (LPCG22)
	edma_lpsr_hclk	M4_CLK_ROOT	clk_enable_edma_lpsr (LPCG23)
	edma_lpsr_ipg_clk	M4_CLK_ROOT	clk_enable_edma_lpsr (LPCG23)
ENET1G	enet_1g_ipg_clk	BUS_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_ipg_clk_mac0	BUS_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_ipg_clk_mac0_s	BUS_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_ipg_clk_rmii	ENET2_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_ipg_clk_s	BUS_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_ipg_clk_time	ENET_TIMER2_CLK_ROOT	clk_enable_enet_1g (LPCG113)
	enet_1g_txc_sampling_clk	ENET2_CLK_ROOT	clk_enable_enet_1g (LPCG113)
ENET	enet_ipg_clk	BUS_CLK_ROOT	clk_enable_enet (LPCG112)
	enet_ipg_clk_mac0	BUS_CLK_ROOT	clk_enable_enet (LPCG112)
	enet_ipg_clk_mac0_s	BUS_CLK_ROOT	clk_enable_enet (LPCG112)
	enet_ipg_clk_rmii	ENET1_CLK_ROOT	clk_enable_enet (LPCG112)
	enet_ipg_clk_s	BUS_CLK_ROOT	clk_enable_enet (LPCG112)
	enet_ipg_clk_time	ENET_TIMER1_CLK_ROOT	clk_enable_enet (LPCG112)
ENET QOS	enet_qos_aclk_i	BUS_CLK_ROOT	clk_enable_enet_qos (LPCG114)
	enet_qos_clk_csr_i	BUS_CLK_ROOT	clk_enable_enet_qos (LPCG114)
	enet_qos_clk_ptp_ref_i	ENET_TIMER3_CLK_ROOT	clk_enable_enet_qos (LPCG114)
EWM	ewm_ipg_clk	BUS_CLK_ROOT	clk_enable_ewm (LPCG18)
	ewm_ipg_clk_s	BUS_CLK_ROOT	clk_enable_ewm (LPCG18)
	ewm_lpo_clk_0	RCOSC_16M	clk_enable_ewm (LPCG18)
	ewm_lpo_clk_1	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_ewm (LPCG18)
	ewm_lpo_clk_2	BUS_CLK_ROOT	clk_enable_ewm (LPCG18)
	ewm_lpo_clk_3	BUS_CLK_ROOT	clk_enable_ewm (LPCG18)
FlexIOn	flexio1_flexio_clk	FLEXIO1_CLK_ROOT	clk_enable_flexio1 (LPCG53)
	flexio1_ipt_clk	FLEXIO1_CLK_ROOT	clk_enable_flexio1 (LPCG53)
	flexio1_pclk	BUS_CLK_ROOT	clk_enable_flexio1 (LPCG53)
	flexio2_flexio_clk	FLEXIO2_CLK_ROOT	clk_enable_flexio2 (LPCG54)
	flexio2_ipt_clk	FLEXIO2_CLK_ROOT	clk_enable_flexio2 (LPCG54)
	flexio2_pclk	BUS_CLK_ROOT	clk_enable_flexio2 (LPCG54)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
FlexPWMn	flexpwm1_ipg_clk_0	BUS_CLK_ROOT	clk_enable_flexpwm1 (LPCG79)
	flexpwm1_ipg_clk_1	BUS_CLK_ROOT	clk_enable_flexpwm1 (LPCG79)
	flexpwm1_ipg_clk_2	BUS_CLK_ROOT	clk_enable_flexpwm1 (LPCG79)
	flexpwm1_ipg_clk_3	BUS_CLK_ROOT	clk_enable_flexpwm1 (LPCG79)
	flexpwm2_ipg_clk_0	BUS_CLK_ROOT	clk_enable_flexpwm2 (LPCG80)
	flexpwm2_ipg_clk_1	BUS_CLK_ROOT	clk_enable_flexpwm2 (LPCG80)
	flexpwm2_ipg_clk_2	BUS_CLK_ROOT	clk_enable_flexpwm2 (LPCG80)
	flexpwm2_ipg_clk_3	BUS_CLK_ROOT	clk_enable_flexpwm2 (LPCG80)
	flexpwm3_ipg_clk_0	BUS_CLK_ROOT	clk_enable_flexpwm3 (LPCG81)
	flexpwm3_ipg_clk_1	BUS_CLK_ROOT	clk_enable_flexpwm3 (LPCG81)
	flexpwm3_ipg_clk_2	BUS_CLK_ROOT	clk_enable_flexpwm3 (LPCG81)
	flexpwm3_ipg_clk_3	BUS_CLK_ROOT	clk_enable_flexpwm3 (LPCG81)
	flexpwm4_ipg_clk_0	BUS_CLK_ROOT	clk_enable_flexpwm4 (LPCG82)
	flexpwm4_ipg_clk_1	BUS_CLK_ROOT	clk_enable_flexpwm4 (LPCG82)
	flexpwm4_ipg_clk_2	BUS_CLK_ROOT	clk_enable_flexpwm4 (LPCG82)
	flexpwm4_ipg_clk_3	BUS_CLK_ROOT	clk_enable_flexpwm4 (LPCG82)
FlexSPIn	flexspi1_hclk	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	flexspi1_ipg_clk	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	flexspi1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	flexspi1_ipg_clk_sfck	FLEXSPI1_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	flexspi2_hclk	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	flexspi2_ipg_clk	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	flexspi2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	flexspi2_ipg_clk_sfck	FLEXSPI2_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
GPC	gpc_ipg_clk	RCOSC_16M	clk_enable_gpc (LPCG11)
	gpc_clk_32k	RCOSC_16M	clk_enable_gpc (LPCG11)
GPIOn	gpio1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio4_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio5_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio6_ipg_clk_s	BUS_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio7_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio8_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio9_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio10_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio11_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio12_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_gpio (LPCG51)
	gpio13_ipg_clk_s	RCOSC_16M	clk_enable_gpio (LPCG51)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
GPTn	gpt1_ipg_clk	GPT1_CLK_ROOT	clk_enable_gpt1 (LPCG64)
	gpt1_ipg_clk_s	GPT1_CLK_ROOT	clk_enable_gpt1 (LPCG64)
	gpt1_ipg_clk_32k	OSC32K	clk_enable_gpt1 (LPCG64)
	gpt1_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt1 (LPCG64)
	gpt1_ipg_clk_24m	RCOSC_16M	clk_enable_gpt1 (LPCG64)
	gpt2_ipg_clk	GPT2_CLK_ROOT	clk_enable_gpt2 (LPCG65)
	gpt2_ipg_clk_s	GPT2_CLK_ROOT	clk_enable_gpt2 (LPCG65)
	gpt2_ipg_clk_32k	OSC32K	clk_enable_gpt2 (LPCG65)
	gpt2_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt2 (LPCG65)
	gpt2_ipg_clk_24m	RCOSC_16M	clk_enable_gpt2 (LPCG65)
	gpt3_ipg_clk	GPT3_CLK_ROOT	clk_enable_gpt3 (LPCG66)
	gpt3_ipg_clk_s	GPT3_CLK_ROOT	clk_enable_gpt3 (LPCG66)
	gpt3_ipg_clk_32k	OSC32K	clk_enable_gpt3 (LPCG66)
	gpt3_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt3 (LPCG66)
	gpt3_ipg_clk_24m	RCOSC_16M	clk_enable_gpt3 (LPCG66)
	gpt4_ipg_clk	GPT4_CLK_ROOT	clk_enable_gpt4 (LPCG67)
	gpt4_ipg_clk_s	GPT4_CLK_ROOT	clk_enable_gpt4 (LPCG67)
	gpt4_ipg_clk_32k	OSC32K	clk_enable_gpt4 (LPCG67)
	gpt4_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt4 (LPCG67)
	gpt4_ipg_clk_24m	RCOSC_16M	clk_enable_gpt4 (LPCG67)
	gpt5_ipg_clk	GPT5_CLK_ROOT	clk_enable_gpt5 (LPCG68)
	gpt5_ipg_clk_s	GPT5_CLK_ROOT	clk_enable_gpt5 (LPCG68)
	gpt5_ipg_clk_32k	OSC32K	clk_enable_gpt5 (LPCG68)
	gpt5_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt5 (LPCG68)
	gpt5_ipg_clk_24m	RCOSC_16M	clk_enable_gpt5 (LPCG68)
	gpt6_ipg_clk	GPT6_CLK_ROOT	clk_enable_gpt6 (LPCG69)
	gpt6_ipg_clk_s	GPT6_CLK_ROOT	clk_enable_gpt6 (LPCG69)
	gpt6_ipg_clk_32k	OSC32K	clk_enable_gpt6 (LPCG69)
	gpt6_ipg_clk_highfreq	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_gpt6 (LPCG69)
	gpt6_ipg_clk_24m	RCOSC_16M	clk_enable_gpt6 (LPCG69)
GPU2D	gpu2d_aclk	BUS_CLK_ROOT	clk_enable_gpu2d (LPCG128)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	gpu2d_clk2x	GPU2D_CLK_ROOT	clk_enable_gpu2d (LPCG128)
	gpu2d_hclk	BUS_CLK_ROOT	clk_enable_gpu2d (LPCG128)
IEE	iee_iee_clk	BUS_CLK_ROOT	clk_enable_iee (LPCG35)
	iee_pclk	BUS_CLK_ROOT	clk_enable_iee (LPCG35)
IOMUXC	iomux_ipt_clk_io	BUS_CLK_ROOT	clk_enable_iomuxc (LPCG49)
	iomuxc_ipg_clk_s	BUS_CLK_ROOT	clk_enable_iomuxc (LPCG49)
	iomux_lpsr_ipt_clk_io	BUS_LPSR_CLK_ROOT	clk_enable_iomuxc_lpsr (LPCG50)
	iomuxc_gpr_ipg_clk_s	BUS_CLK_ROOT	clk_enable_iomuxc (LPCG49)
	iomuxc_ipg_clk_s	BUS_CLK_ROOT	clk_enable_iomuxc (LPCG49)
	iomuxc_lpsr_gpr_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_iomuxc_lpsr (LPCG50)
	iomuxc_lpsr_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_iomuxc_lpsr (LPCG50)
JTAG	jtag_mux_ipg_clk	M4_CLK_ROOT	clk_enable_jtag_mux (LPCG41)
	jtag_mux_tck	-	-
KPP	kpp_ipg_clk_s	BUS_CLK_ROOT	clk_enable_kpp (LPCG52)
	kpp_ipg_clk_32k	BUS_CLK_ROOT	clk_enable_kpp (LPCG52)
eLCDIF	elcdif_apb_clk	BUS_CLK_ROOT	clk_enable_elcdif (LPCG129)
	elcdif_pix_clk	ELCDIF_CLK_ROOT	clk_enable_elcdif (LPCG129)
LCDIFv2	lcdifv2_apb_clk	BUS_CLK_ROOT	clk_enable_lcdifv2 (LPCG130)
	lcdifv2_bus_clk	BUS_CLK_ROOT	clk_enable_lcdifv2 (LPCG130)
	lcdifv2_pix_clk	LCDIFV2_CLK_ROOT	clk_enable_lcdifv2 (LPCG130)
LPADCn	adc1_adc_clk	ADC1_CLK_ROOT	clk_enable_adc1 (LPCG55)
	adc1_ipg_clk	BUS_CLK_ROOT	clk_enable_adc1 (LPCG55)
	adc1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_adc1 (LPCG55)
	adc2_adc_clk	ADC2_CLK_ROOT	clk_enable_adc2 (LPCG56)
	adc2_ipg_clk	BUS_CLK_ROOT	clk_enable_adc2 (LPCG56)
	adc2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_adc2 (LPCG56)
LPI2Cn	lpi2c1_ipg_clk	BUS_CLK_ROOT	clk_enable_lpi2c1 (LPCG98)
	lpi2c1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpi2c1 (LPCG98)
	lpi2c1_ipt_clk	LPI2C1_CLK_ROOT	clk_enable_lpi2c1 (LPCG98)
	lpi2c1_lpi2c_clk	LPI2C1_CLK_ROOT	clk_enable_lpi2c1 (LPCG98)
	lpi2c1_lpi2c_div_clk	LPI2C1_CLK_ROOT	clk_enable_lpi2c1 (LPCG98)
	lpi2c2_ipg_clk	BUS_CLK_ROOT	clk_enable_lpi2c2 (LPCG99)
	lpi2c2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpi2c2 (LPCG99)
	lpi2c2_ipt_clk	LPI2C2_CLK_ROOT	clk_enable_lpi2c2 (LPCG99)
	lpi2c2_lpi2c_clk	LPI2C2_CLK_ROOT	clk_enable_lpi2c2 (LPCG99)
	lpi2c2_lpi2c_div_clk	LPI2C2_CLK_ROOT	clk_enable_lpi2c2 (LPCG99)
	lpi2c3_ipg_clk	BUS_CLK_ROOT	clk_enable_lpi2c3 (LPCG100)
	lpi2c3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpi2c3 (LPCG100)
	lpi2c3_ipt_clk	LPI2C3_CLK_ROOT	clk_enable_lpi2c3 (LPCG100)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	lpi2c3_lpi2c_clk	LPI2C3_CLK_ROOT	clk_enable_lpi2c3 (LPCG100)
	lpi2c3_lpi2c_div_clk	LPI2C3_CLK_ROOT	clk_enable_lpi2c3 (LPCG100)
	lpi2c4_ipg_clk	BUS_CLK_ROOT	clk_enable_lpi2c4 (LPCG101)
	lpi2c4_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpi2c4 (LPCG101)
	lpi2c4_ipt_clk	LPI2C4_CLK_ROOT	clk_enable_lpi2c4 (LPCG101)
	lpi2c4_lpi2c_clk	LPI2C4_CLK_ROOT	clk_enable_lpi2c4 (LPCG101)
	lpi2c4_lpi2c_div_clk	LPI2C4_CLK_ROOT	clk_enable_lpi2c4 (LPCG101)
	lpi2c5_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpi2c5 (LPCG102)
	lpi2c5_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpi2c5 (LPCG102)
	lpi2c5_ipt_clk	LPI2C5_CLK_ROOT	clk_enable_lpi2c5 (LPCG102)
	lpi2c5_lpi2c_clk	LPI2C5_CLK_ROOT	clk_enable_lpi2c5 (LPCG102)
	lpi2c5_lpi2c_div_clk	LPI2C5_CLK_ROOT	clk_enable_lpi2c5 (LPCG102)
	lpi2c6_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpi2c6 (LPCG103)
	lpi2c6_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpi2c6 (LPCG103)
	lpi2c6_ipt_clk	LPI2C6_CLK_ROOT	clk_enable_lpi2c6 (LPCG103)
	lpi2c6_lpi2c_clk	LPI2C6_CLK_ROOT	clk_enable_lpi2c6 (LPCG103)
	lpi2c6_lpi2c_div_clk	LPI2C6_CLK_ROOT	clk_enable_lpi2c6 (LPCG103)
LPSPIn	lpspi1_ipg_clk	BUS_CLK_ROOT	clk_enable_lpspi1 (LPCG104)
	lpspi1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpspi1 (LPCG104)
	lpspi1_ipt_clk_lpspi	LPSPI1_CLK_ROOT	clk_enable_lpspi1 (LPCG104)
	lpspi1_lpspi_clk	LPSPI1_CLK_ROOT	clk_enable_lpspi1 (LPCG104)
	lpspi1_lpspi_div_clk	LPSPI1_CLK_ROOT	clk_enable_lpspi1 (LPCG104)
	lpspi2_ipg_clk	BUS_CLK_ROOT	clk_enable_lpspi2 (LPCG105)
	lpspi2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpspi2 (LPCG105)
	lpspi2_ipt_clk_lpspi	LPSPI2_CLK_ROOT	clk_enable_lpspi2 (LPCG105)
	lpspi2_lpspi_clk	LPSPI2_CLK_ROOT	clk_enable_lpspi2 (LPCG105)
	lpspi2_lpspi_div_clk	LPSPI2_CLK_ROOT	clk_enable_lpspi2 (LPCG105)
	lpspi3_ipg_clk	BUS_CLK_ROOT	clk_enable_lpspi3 (LPCG106)
	lpspi3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpspi3 (LPCG106)
	lpspi3_ipt_clk_lpspi	LPSPI3_CLK_ROOT	clk_enable_lpspi3 (LPCG106)
	lpspi3_lpspi_clk	LPSPI3_CLK_ROOT	clk_enable_lpspi3 (LPCG106)
	lpspi3_lpspi_div_clk	LPSPI3_CLK_ROOT	clk_enable_lpspi3 (LPCG106)
	lpspi4_ipg_clk	BUS_CLK_ROOT	clk_enable_lpspi4 (LPCG107)
	lpspi4_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpspi4 (LPCG107)
	lpspi4_ipt_clk_lpspi	LPSPI4_CLK_ROOT	clk_enable_lpspi4 (LPCG107)
	lpspi4_lpspi_clk	LPSPI4_CLK_ROOT	clk_enable_lpspi4 (LPCG107)
	lpspi4_lpspi_div_clk	LPSPI4_CLK_ROOT	clk_enable_lpspi4 (LPCG107)
	lpspi5_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpspi5 (LPCG108)
	lpspi5_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpspi5 (LPCG108)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	lpspi5_ipt_clk_lpspi	LPSPI5_CLK_ROOT	clk_enable_lpspi5 (LPCG108)
	lpspi5_lpspi_clk	LPSPI5_CLK_ROOT	clk_enable_lpspi5 (LPCG108)
	lpspi5_lpspi_div_clk	LPSPI5_CLK_ROOT	clk_enable_lpspi5 (LPCG108)
	lpspi6_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpspi6 (LPCG109)
	lpspi6_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpspi6 (LPCG109)
	lpspi6_ipt_clk_lpspi	LPSPI6_CLK_ROOT	clk_enable_lpspi6 (LPCG109)
	lpspi6_lpspi_clk	LPSPI6_CLK_ROOT	clk_enable_lpspi6 (LPCG109)
	lpspi6_lpspi_div_clk	LPSPI6_CLK_ROOT	clk_enable_lpspi6 (LPCG109)
LPUARTn	lpuart1_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart1 (LPCG86)
	lpuart1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart1 (LPCG86)
	lpuart1_lpuart_baud_clk	LPUART1_CLK_ROOT	clk_enable_lpuart1 (LPCG86)
	lpuart1_lpuart_baud_gated_clk	LPUART1_CLK_ROOT	clk_enable_lpuart1 (LPCG86)
	lpuart1_lpuart_baud_gated_clk_b	LPUART1_CLK_ROOT	clk_enable_lpuart1 (LPCG86)
	lpuart2_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart2 (LPCG87)
	lpuart2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart2 (LPCG87)
	lpuart2_lpuart_baud_clk	LPUART2_CLK_ROOT	clk_enable_lpuart2 (LPCG87)
	lpuart2_lpuart_baud_gated_clk	LPUART2_CLK_ROOT	clk_enable_lpuart2 (LPCG87)
	lpuart2_lpuart_baud_gated_clk_b	LPUART2_CLK_ROOT	clk_enable_lpuart2 (LPCG87)
	lpuart3_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart3 (LPCG88)
	lpuart3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart3 (LPCG88)
	lpuart3_lpuart_baud_clk	LPUART3_CLK_ROOT	clk_enable_lpuart3 (LPCG88)
	lpuart3_lpuart_baud_gated_clk	LPUART3_CLK_ROOT	clk_enable_lpuart3 (LPCG88)
	lpuart3_lpuart_baud_gated_clk_b	LPUART3_CLK_ROOT	clk_enable_lpuart3 (LPCG88)
	lpuart4_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart4 (LPCG89)
	lpuart4_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart4 (LPCG89)
	lpuart4_lpuart_baud_clk	LPUART4_CLK_ROOT	clk_enable_lpuart4 (LPCG89)
	lpuart4_lpuart_baud_gated_clk	LPUART4_CLK_ROOT	clk_enable_lpuart4 (LPCG89)
	lpuart4_lpuart_baud_gated_clk_b	LPUART4_CLK_ROOT	clk_enable_lpuart4 (LPCG89)
	lpuart5_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart5 (LPCG90)
	lpuart5_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart5 (LPCG90)
	lpuart5_lpuart_baud_clk	LPUART5_CLK_ROOT	clk_enable_lpuart5 (LPCG90)
	lpuart5_lpuart_baud_gated_clk	LPUART5_CLK_ROOT	clk_enable_lpuart5 (LPCG90)
	lpuart5_lpuart_baud_gated_clk_b	LPUART5_CLK_ROOT	clk_enable_lpuart5 (LPCG90)
	lpuart6_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart6 (LPCG91)
	lpuart6_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart6 (LPCG91)
	lpuart6_lpuart_baud_clk	LPUART6_CLK_ROOT	clk_enable_lpuart6 (LPCG91)
	lpuart6_lpuart_baud_gated_clk	LPUART6_CLK_ROOT	clk_enable_lpuart6 (LPCG91)
	lpuart6_lpuart_baud_gated_clk_b	LPUART6_CLK_ROOT	clk_enable_lpuart6 (LPCG91)
	lpuart7_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart7 (LPCG92)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	lpuart7_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart7 (LPCG92)
	lpuart7_lpuart_baud_clk	LPUART7_CLK_ROOT	clk_enable_lpuart7 (LPCG92)
	lpuart7_lpuart_baud_gated_clk	LPUART7_CLK_ROOT	clk_enable_lpuart7 (LPCG92)
	lpuart7_lpuart_baud_gated_clk_b	LPUART7_CLK_ROOT	clk_enable_lpuart7 (LPCG92)
	lpuart8_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart8 (LPCG93)
	lpuart8_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart8 (LPCG93)
	lpuart8_lpuart_baud_clk	LPUART8_CLK_ROOT	clk_enable_lpuart8 (LPCG93)
	lpuart8_lpuart_baud_gated_clk	LPUART8_CLK_ROOT	clk_enable_lpuart8 (LPCG93)
	lpuart8_lpuart_baud_gated_clk_b	LPUART8_CLK_ROOT	clk_enable_lpuart8 (LPCG93)
	lpuart9_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart9 (LPCG94)
	lpuart9_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart9 (LPCG94)
	lpuart9_lpuart_baud_clk	LPUART9_CLK_ROOT	clk_enable_lpuart9 (LPCG94)
	lpuart9_lpuart_baud_gated_clk	LPUART9_CLK_ROOT	clk_enable_lpuart9 (LPCG94)
	lpuart9_lpuart_baud_gated_clk_b	LPUART9_CLK_ROOT	clk_enable_lpuart9 (LPCG94)
	lpuart10_ipg_clk	BUS_CLK_ROOT	clk_enable_lpuart10 (LPCG95)
	lpuart10_ipg_clk_s	BUS_CLK_ROOT	clk_enable_lpuart10 (LPCG95)
	lpuart10_lpuart_baud_clk	LPUART10_CLK_ROOT	clk_enable_lpuart10 (LPCG95)
	lpuart10_lpuart_baud_gated_clk	LPUART10_CLK_ROOT	clk_enable_lpuart10 (LPCG95)
	lpuart10_lpuart_baud_gated_clk_b	LPUART10_CLK_ROOT	clk_enable_lpuart10 (LPCG95)
	lpuart11_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpuart11 (LPCG96)
	lpuart11_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpuart11 (LPCG96)
	lpuart11_lpuart_baud_clk	LPUART11_CLK_ROOT	clk_enable_lpuart11 (LPCG96)
	lpuart11_lpuart_baud_gated_clk	LPUART11_CLK_ROOT	clk_enable_lpuart11 (LPCG96)
	lpuart11_lpuart_baud_gated_clk_b	LPUART11_CLK_ROOT	clk_enable_lpuart11 (LPCG96)
	lpuart12_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_lpuart12 (LPCG97)
	lpuart12_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_lpuart12 (LPCG97)
	lpuart12_lpuart_baud_clk	LPUART12_CLK_ROOT	clk_enable_lpuart12 (LPCG97)
	lpuart12_lpuart_baud_gated_clk	LPUART12_CLK_ROOT	clk_enable_lpuart12 (LPCG97)
	lpuart12_lpuart_baud_gated_clk_b	LPUART12_CLK_ROOT	clk_enable_lpuart12 (LPCG97)
MECCn	mecc1_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	mecc1_ipg_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	mecc1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	mecc2_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	mecc2_ipg_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	mecc2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
MIPI CSI	mipi_csi_clk	CSI2_CLK_ROOT	clk_enable_mipi_csi (LPCG132)
	mipi_csi_clk_esc	CSI2_ESC_CLK_ROOT	clk_enable_mipi_csi (LPCG132)
	mipi_csi_clk_ui	CSI2_UI_CLK_ROOT	clk_enable_mipi_csi (LPCG132)
	mipi_csi_pclk	BUS_CLK_ROOT	clk_enable_mipi_csi (LPCG132)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
MIPI DSI	mipi_dsi_CLKREF	MIPI_REF_CLK_ROOT	clk_enable_mipi_dsi (LPCG131)
	mipi_dsi_dpi_pclk	BUS_CLK_ROOT	clk_enable_mipi_dsi (LPCG131)
	mipi_dsi_pclk	BUS_CLK_ROOT	clk_enable_mipi_dsi (LPCG131)
	mipi_dsi_RxClkEsc	MIPI_ESC_CLK_ROOT	clk_enable_mipi_dsi (LPCG131)
	mipi_dsi_TxClkEsc	MIPI_ESC_CLK_ROOT	clk_enable_mipi_dsi (LPCG131)
MQS	mqs_bclk	MQS_CLK_ROOT	clk_enable_mqs (LPCG120)
	mqs_hmclk	SAI3_CLK_ROOT	clk_enable_mqs (LPCG120)
MU	mu_ipg_clk_dsp	BUS_LPSR_CLK_ROOT	clk_enable_mu_b (LPCG21)
	mu_ipg_clk_mcu	BUS_LPSR_CLK_ROOT	clk_enable_mu_a (LPCG20)
	mu_ipg_clk_s_dsp	BUS_LPSR_CLK_ROOT	clk_enable_mu_b (LPCG21)
	mu_ipg_clk_s_mcu	BUS_LPSR_CLK_ROOT	clk_enable_mu_a (LPCG20)
OCRAMn	ocram1_mem_mecc_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
	ocram2_mem_mecc_clk	BUS_CLK_ROOT	clk_enable_ocram (LPCG25)
PDM	mic_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_app	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_app_fifo	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_app_nonstop	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dc	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec1	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec2	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec3	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec4	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec5	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec6	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec7	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec160	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec161	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec162	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec163	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec164	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec165	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec166	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_dec167	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_fir	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_hb	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_inp	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_vad0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	mic_ipg_clk_vad_dec0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_vad_dec160	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_vad_ndec0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_vad_noise0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
	mic_ipg_clk_vad_zcd0	MIC_CLK_ROOT	clk_enable_mic (LPCG121)
PITn	pit1_ipg_clk	BUS_CLK_ROOT	clk_enable_pit1 (LPCG62)
	pit1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_pit1 (LPCG62)
	pit1_ipg_clk_sync	BUS_CLK_ROOT	clk_enable_pit1 (LPCG62)
	pit2_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_pit2 (LPCG63)
	pit2_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_pit2 (LPCG63)
	pit2_ipg_clk_sync	BUS_LPSR_CLK_ROOT	clk_enable_pit2 (LPCG63)
PXP	pxp_clk	BUS_CLK_ROOT	clk_enable_pxp (LPCG127)
QDCn	enc1_ipg_clk	BUS_CLK_ROOT	clk_enable_enc1 (LPCG74)
	enc2_ipg_clk	BUS_CLK_ROOT	clk_enable_enc2 (LPCG75)
	enc3_ipg_clk	BUS_CLK_ROOT	clk_enable_enc3 (LPCG76)
	enc4_ipg_clk	BUS_CLK_ROOT	clk_enable_enc4 (LPCG77)
QTIMERn	qtimer1_clk0	BUS_CLK_ROOT	clk_enable_qtimer1 (LPCG70)
	qtimer1_clk1	BUS_CLK_ROOT	clk_enable_qtimer1 (LPCG70)
	qtimer1_clk2	BUS_CLK_ROOT	clk_enable_qtimer1 (LPCG70)
	qtimer1_clk3	BUS_CLK_ROOT	clk_enable_qtimer1 (LPCG70)
	qtimer2_clk0	BUS_CLK_ROOT	clk_enable_qtimer2 (LPCG71)
	qtimer2_clk1	BUS_CLK_ROOT	clk_enable_qtimer2 (LPCG71)
	qtimer2_clk2	BUS_CLK_ROOT	clk_enable_qtimer2 (LPCG71)
	qtimer2_clk3	BUS_CLK_ROOT	clk_enable_qtimer2 (LPCG71)
	qtimer3_clk0	BUS_CLK_ROOT	clk_enable_qtimer3 (LPCG72)
	qtimer3_clk1	BUS_CLK_ROOT	clk_enable_qtimer3 (LPCG72)
	qtimer3_clk2	BUS_CLK_ROOT	clk_enable_qtimer3 (LPCG72)
	qtimer3_clk3	BUS_CLK_ROOT	clk_enable_qtimer3 (LPCG72)
	qtimer4_clk0	BUS_CLK_ROOT	clk_enable_qtimer4 (LPCG73)
	qtimer4_clk1	BUS_CLK_ROOT	clk_enable_qtimer4 (LPCG73)
	qtimer4_clk2	BUS_CLK_ROOT	clk_enable_qtimer4 (LPCG73)
	qtimer4_clk3	BUS_CLK_ROOT	clk_enable_qtimer4 (LPCG73)
RDC	rdc_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_rdc (LPCG30)
	rdc_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_rdc (LPCG30)
SAIn	sai1_ipg_clk	BUS_CLK_ROOT	clk_enable_sai1 (LPCG123)
	sai1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_sai1 (LPCG123)
	sai1_ipg_clk_sai_mclk	IOMUXC_GPR_GPR0[SAI_MCLK1_SEL]	clk_enable_sai1 (LPCG123)
	sai1_ipt_clk_sai_bclk	SAI1_CLK_ROOT	clk_enable_sai1 (LPCG123)
	sai1_ipt_clk_sai_bclk_b	SAI1_CLK_ROOT	clk_enable_sai1 (LPCG123)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	sai1_mclk_in2	IOMUXC_GPR_GPR0[SAI_MCLK2_SEL]	clk_enable_sai1 (LPCG123)
	sai1_mclk_in3	IOMUXC_GPR_GPR0[SAI_MCLK3_SEL]	clk_enable_sai1 (LPCG123)
	sai2_ipg_clk	BUS_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_ipg_clk_sai_mclk	SAI2_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_ipt_clk_sai_bclk	SAI2_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_ipt_clk_sai_bclk_b	SAI2_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_mclk_in2	SAI2_CLK_ROOT	clk_enable_sai2 (LPCG124)
	sai2_mclk_in3	IOMUXC_GPR_GPR1[SA2_MCLK3_SEL]	clk_enable_sai2 (LPCG124)
	sai3_ipg_clk	BUS_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_ipg_clk_sai_mclk	SAI3_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_ipt_clk_sai_bclk	SAI3_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_ipt_clk_sai_bclk_b	SAI3_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_mclk_in2	SAI3_CLK_ROOT	clk_enable_sai3 (LPCG125)
	sai3_mclk_in3	IOMUXC_GPR_GPR2[SA3_MCLK3_SEL]	clk_enable_sai3 (LPCG125)
	sai4_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_ipg_clk_sai_mclk	SAI4_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_ipt_clk_sai_bclk	SAI4_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_ipt_clk_sai_bclk_b	SAI4_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_mclk_in2	SAI4_CLK_ROOT	clk_enable_sai4 (LPCG126)
	sai4_mclk_in3	SAI4_CLK_ROOT	clk_enable_sai4 (LPCG126)
SEMAn	sema_hs_clk	BUS_LPSR_CLK_ROOT	clk_enable_sema (LPCG19)
	sema1_clk	BUS_LPSR_CLK_ROOT	clk_enable_sema (LPCG19)
	sema2_clk	BUS_LPSR_CLK_ROOT	clk_enable_sema (LPCG19)
SEMC	semc_ipg_clk	SEMC_CLK_ROOT	clk_enable_semc (LPCG33)
EMVSIMn	sim1_ipg_clk	BUS_CLK_ROOT	clk_enable_sim1 (LPCG110)
	sim1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_sim1 (LPCG110)
	sim1_ipg_sim_clk	EMV1_CLK_ROOT	clk_enable_sim1 (LPCG110)
	sim1_ipg_ungated_sim_clk	EMV1_CLK_ROOT	clk_enable_sim1 (LPCG110)
	sim2_ipg_clk	BUS_CLK_ROOT	clk_enable_sim2 (LPCG111)
	sim2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_sim2 (LPCG111)
	sim2_ipg_sim_clk	EMV2_CLK_ROOT	clk_enable_sim2 (LPCG111)
	sim2_ipg_ungated_sim_clk	EMV2_CLK_ROOT	clk_enable_sim2 (LPCG111)
SNVS	snvs_hp_wrapper_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_snvs_hp (LPCG38)
	snvs_hp_wrapper_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_snvs_hp (LPCG38)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
SPDIF	spdif_gclkw_t0	BUS_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_extal_clk	OSC_24M	clk_enable_spdif (LPCG122)
	spdif_ipg_clk_s	BUS_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk	SPDIF_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk1	SAI1_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk2	SAI2_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk3	SAI3_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk4	SAI4_CLK_ROOT	clk_enable_spdif (LPCG122)
	spdif_tx_clk5	MIC_CLK_ROOT	clk_enable_spdif (LPCG122)
SSARC	ssarc_hp_ahb_clk	M4_CLK_ROOT	clk_enable_ssarc (LPCG12)
	ssarc_lp_ipg_clk	RCOSC_16M	clk_enable_ssarc (LPCG12)
	ssarc_lp_ahb_clk	M4_CLK_ROOT	clk_enable_ssarc (LPCG12)
USB	usb_ipg_ahb_clk	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
	usb_ipg_clk_s	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
	usb_ipg_clk_s_pl301	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
USBPHYn	usbphy1_ipg_clk	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
	usbphy1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
	usbphy1_clk_xtal	OSC_24M	clk_enable_usb (LPCG115)
	usbphy2_ipg_clk	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
	usbphy2_clk_xtal	OSC_24M	clk_enable_usb (LPCG115)
	usbphy2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_usb (LPCG115)
USDHCn	usdhc1_hclk	BUS_CLK_ROOT	clk_enable_usdhc1 (LPCG117)
	usdhc1_ipg_clk	BUS_CLK_ROOT	clk_enable_usdhc1 (LPCG117)
	usdhc1_ipg_clk_perclk	USDHC1_CLK_ROOT	clk_enable_usdhc1 (LPCG117)
	usdhc1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_usdhc1 (LPCG117)
	usdhc2_hclk	BUS_CLK_ROOT	clk_enable_usdhc2 (LPCG118)
	usdhc2_ipg_clk	BUS_CLK_ROOT	clk_enable_usdhc2 (LPCG118)
	usdhc2_ipg_clk_perclk	USDHC2_CLK_ROOT	clk_enable_usdhc2 (LPCG118)
	usdhc2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_usdhc2 (LPCG118)
VIDMUX	video_mux_dcic1_hsp_clk	BUS_CLK_ROOT	clk_enable_dcic_mipi (LPCG134)
	video_mux_dcic2_hsp_clk	BUS_CLK_ROOT	clk_enable_dcic_lcd (LPCG135)
	video_mux_ipg_clk_s	BUS_CLK_ROOT	clk_enable_video_mux (LPCG136)
WDOGn	wdog1_ipg_clk	BUS_CLK_ROOT	clk_enable_wdog1 (LPCG14)
	wdog1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_wdog1 (LPCG14)
	wdog1_ipg_clk_32k	BUS_CLK_ROOT	clk_enable_wdog1 (LPCG14)
	wdog2_ipg_clk	BUS_CLK_ROOT	clk_enable_wdog2 (LPCG15)
	wdog2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_wdog2 (LPCG15)
	wdog2_ipg_clk_32k	BUS_CLK_ROOT	clk_enable_wdog2 (LPCG15)
	wdog3_ipg_clk_gated	BUS_CLK_ROOT	clk_enable_wdog3 (LPCG16)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	wdog3_ipg_clk_s	BUS_CLK_ROOT	clk_enable_wdog3 (LPCG16)
	wdog3_ipg_clk_ungated	BUS_CLK_ROOT	clk_enable_wdog3 (LPCG16)
	wdog3_ipt_clk	BUS_CLK_ROOT	clk_enable_wdog3 (LPCG16)
	wdog3_ext_clk	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_wdog3 (LPCG16)
	wdog4_ipg_clk_gated	BUS_LPSR_CLK_ROOT	clk_enable_wdog4 (LPCG17)
	wdog4_ipg_clk_s	BUS_LPSR_CLK_ROOT	clk_enable_wdog4 (LPCG17)
	wdog4_ipg_clk_ungated	BUS_LPSR_CLK_ROOT	clk_enable_wdog4 (LPCG17)
	wdog4_ipt_clk	BUS_LPSR_CLK_ROOT	clk_enable_wdog4 (LPCG17)
	wdog4_ext_clk	ref_1m_clk (1MHz clock generated from the 400MHz RC oscillator)	clk_enable_wdog4 (LPCG17)
XBARn	xbar1_ipb_clk	BUS_CLK_ROOT	clk_enable_xbar1 (LPCG43)
	xbar2_ipb_clk	BUS_CLK_ROOT	clk_enable_xbar2 (LPCG44)
	xbar3_ipb_clk	BUS_CLK_ROOT	clk_enable_xbar3 (LPCG45)
XECC	xecc_flexspi1_clk	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	xecc_flexspi1_ipg_clk	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	xecc_flexspi1_ipg_clk_s	BUS_CLK_ROOT	clk_enable_flexspi1 (LPCG28)
	xecc_flexspi2_clk	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	xecc_flexspi2_ipg_clk	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	xecc_flexspi2_ipg_clk_s	BUS_CLK_ROOT	clk_enable_flexspi2 (LPCG29)
	xecc_semc_clk	SEMC_CLK_ROOT	clk_enable_semc (LPCG33)
	xecc_semc_ipg_clk	SEMC_CLK_ROOT	clk_enable_semc (LPCG33)
	xecc_semc_ipg_clk_s	SEMC_CLK_ROOT	clk_enable_semc (LPCG33)
XRDC2	xrdc2_mdac_ahb_m7_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mdac_ahbc_m4_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mdac_ahbs_m4_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mdac_axi_m7_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mdac_caam_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_caam_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_csi_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_csi_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_edma_m4_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mdac_edma_m7_ipg_clk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mdac_enet_1g_rx_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_1g_rx_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_1g_tx_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)

Table continues on the next page...

Table 15-2. System Clocks Table (continued)

Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	xrdc2_mdac_enet_1g_tx_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_qos_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_enet_qos_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_gpu_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_gpu_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_elcdif_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_elcdif_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_lcdifv2_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_lcdifv2_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_pxp_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_pxp_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_mdac_ssarc_ipg_clk_d0	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mdac_ssarc_ipg_clk_d1	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mdac_usb_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_usb_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_usdhc1_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_usdhc1_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_usdhc2_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mdac_usdhc2_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m (LPCG3)
	xrdc2_mgr_d0_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_m7_xrdc (LPCG31)
	xrdc2_mgr_d1_ipg_clk	BUS_LPSR_CLK_ROOT	clk_enable_m4_xrdc (LPCG32)
	xrdc2_mrc_caam_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_caam_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_caam_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi1_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi1_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi1_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi2_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi2_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_flexspi2_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_m4lmem_hclk	M4_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mrc_m4lmem_ipg_clk_d0	M4_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mrc_m4lmem_ipg_clk_d1	M4_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_mrc_m7_ocram_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_m7_ocram_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_m7_ocram_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_mecc1_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)

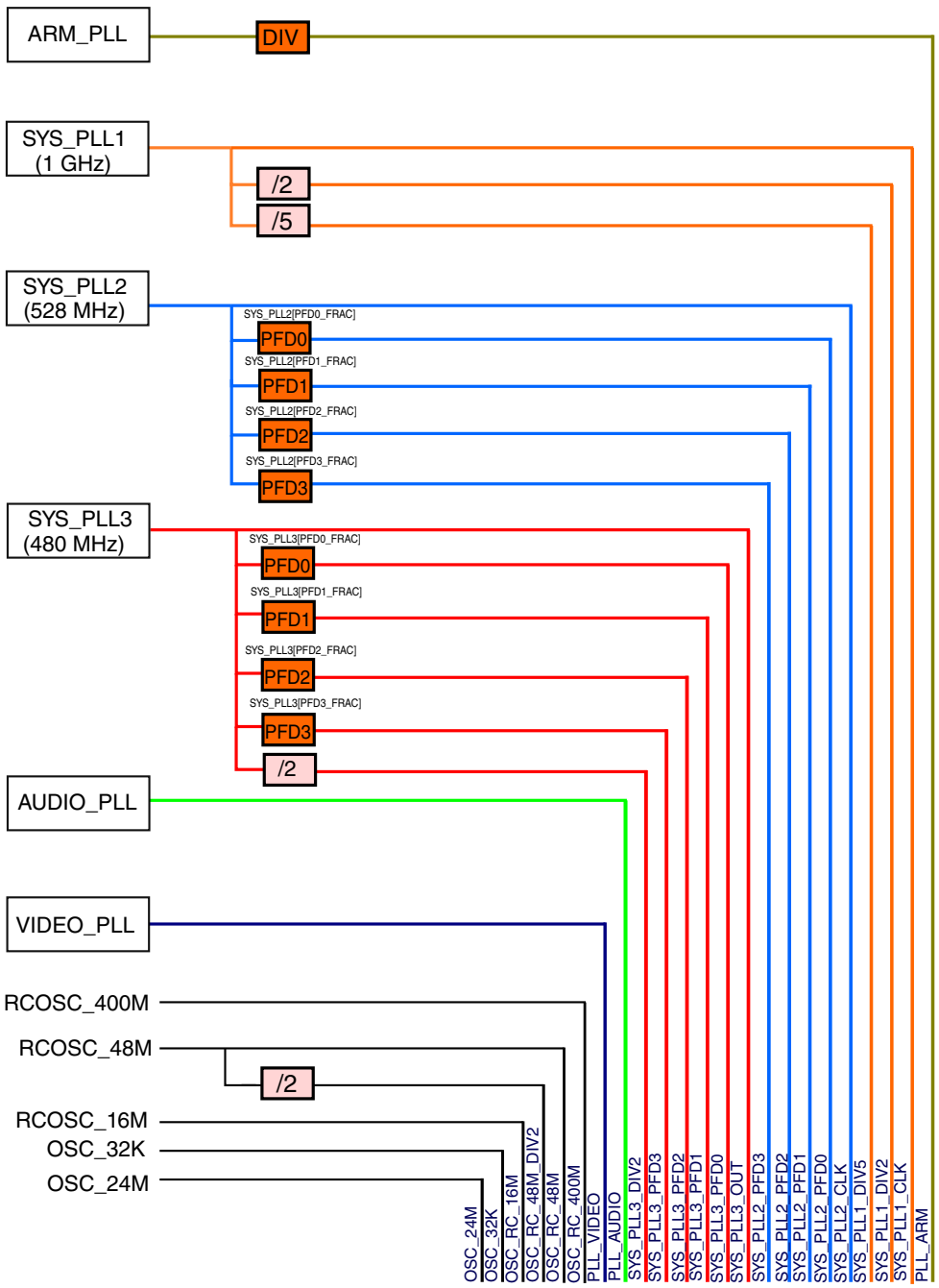
Table continues on the next page...

Table 15-2. System Clocks Table (continued)

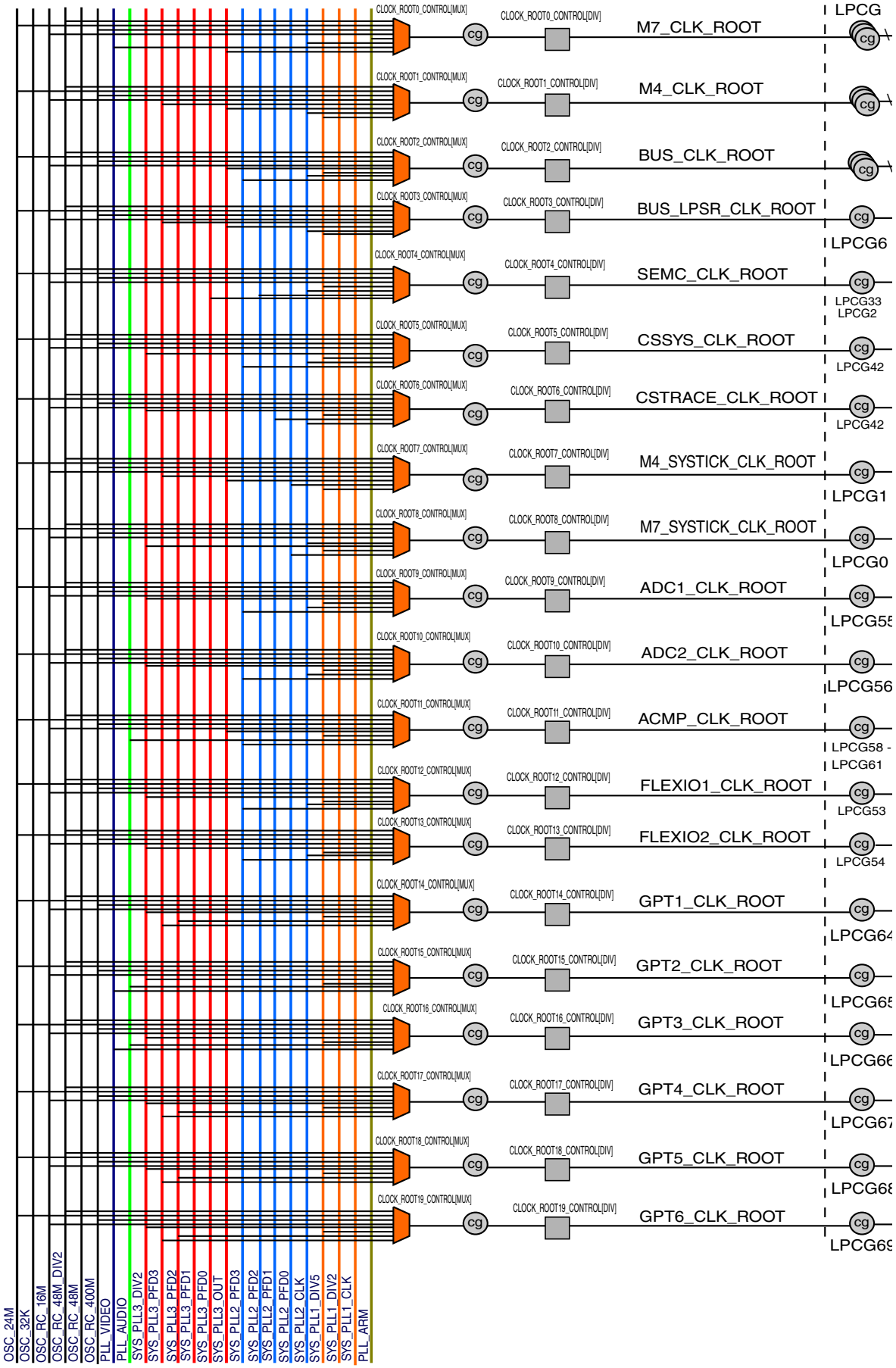
Module	Module Clock	Clock Root	LPCG Enable (LPCGx_DIRECT[ON])
	xrdc2_mrc_mecc1_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_mecc1_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_mecc2_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_mecc2_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_mecc2_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_semc_hclk	SEMC_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_semc_ipg_clk_d0	SEMC_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_mrc_semc_ipg_clk_d1	SEMC_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv0_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv0_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv0_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv0_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv0_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv1_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv1_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv1_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv4_hclk	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv4_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_gpv4_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_msc_romcp_hclk	M4_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_msc_romcp_ipg_clk_d0	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_msc_romcp_ipg_clk_d1	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_pac_aips1_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips1_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips2_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips2_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips3_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips3_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_m7 (LPCG2)
	xrdc2_pac_aips4_ipg_clk_d0	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_pac_aips4_ipg_clk_d1	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_pac_gpu_hclk	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_pac_gpu_ipg_clk_d0	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_pac_gpu_ipg_clk_d1	BUS_CLK_ROOT	clk_enable_sim_disp (LPCG4)
	xrdc2_pac_scdog_hclk	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_pac_scdog_ipg_clk_d0	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)
	xrdc2_pac_scdog_ipg_clk_d1	BUS_LPSR_CLK_ROOT	clk_enable_sim_lpsr (LPCG6)

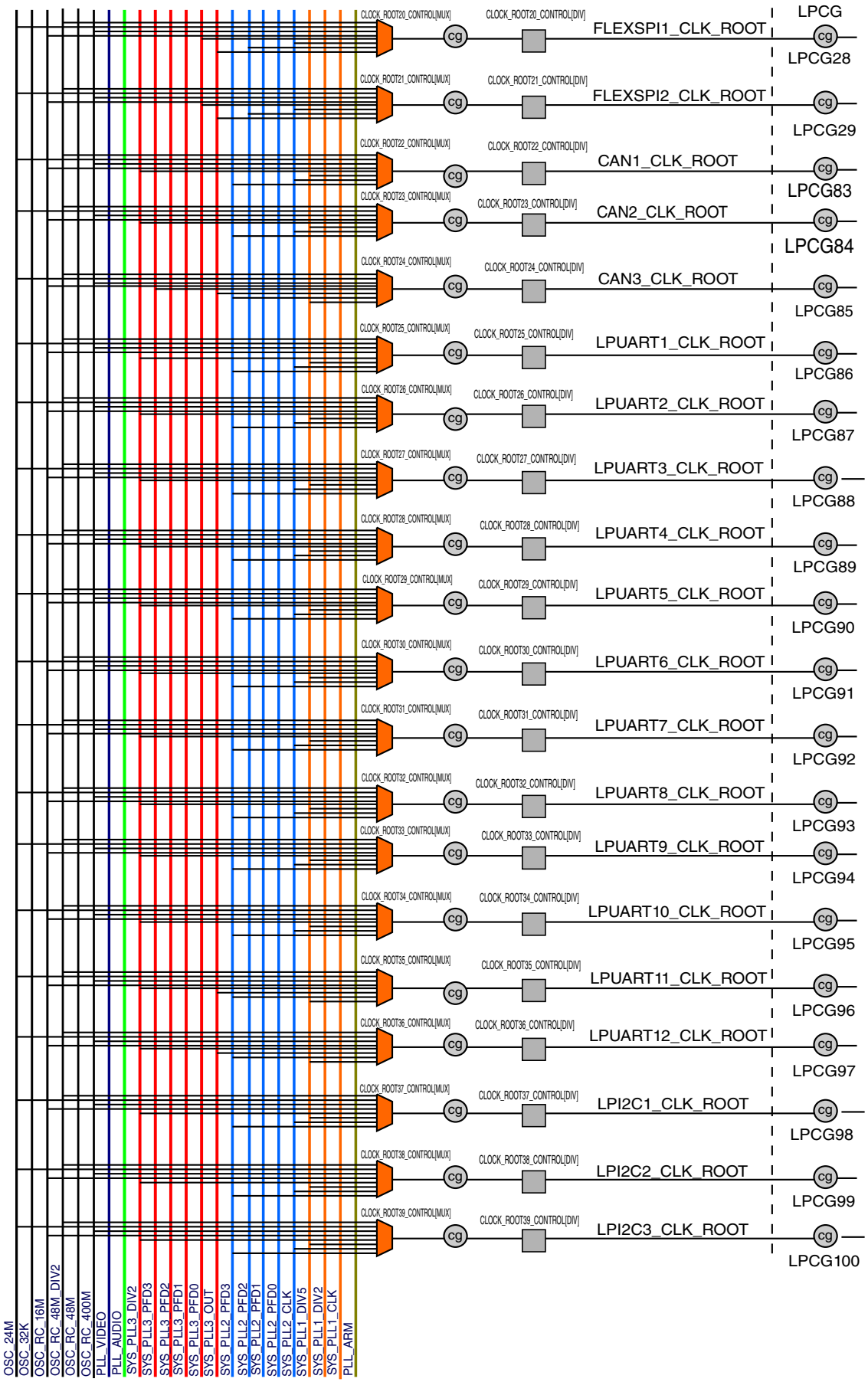
15.4 Clock Tree

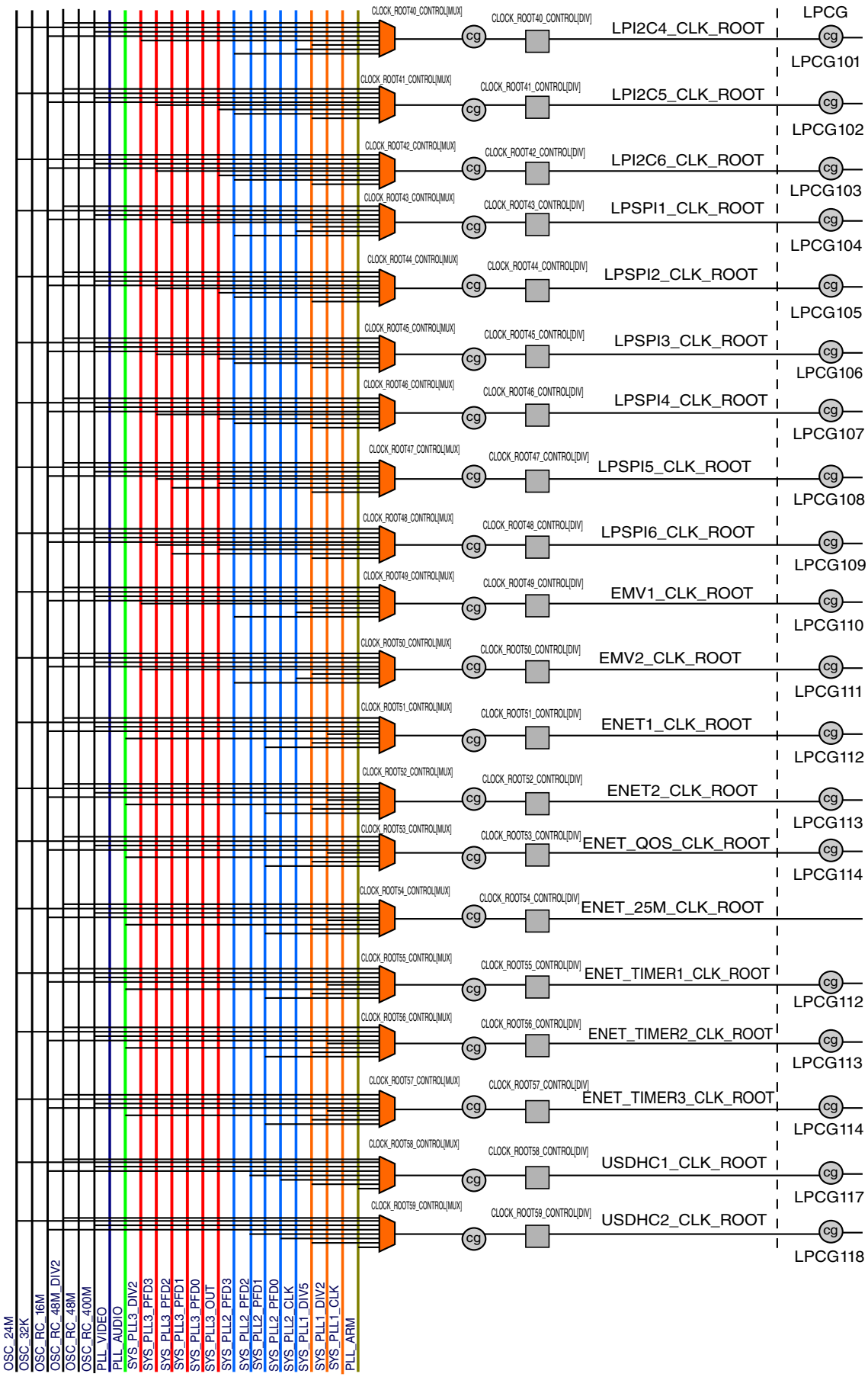
The figure below illustrates the clock sources from the PLLs.

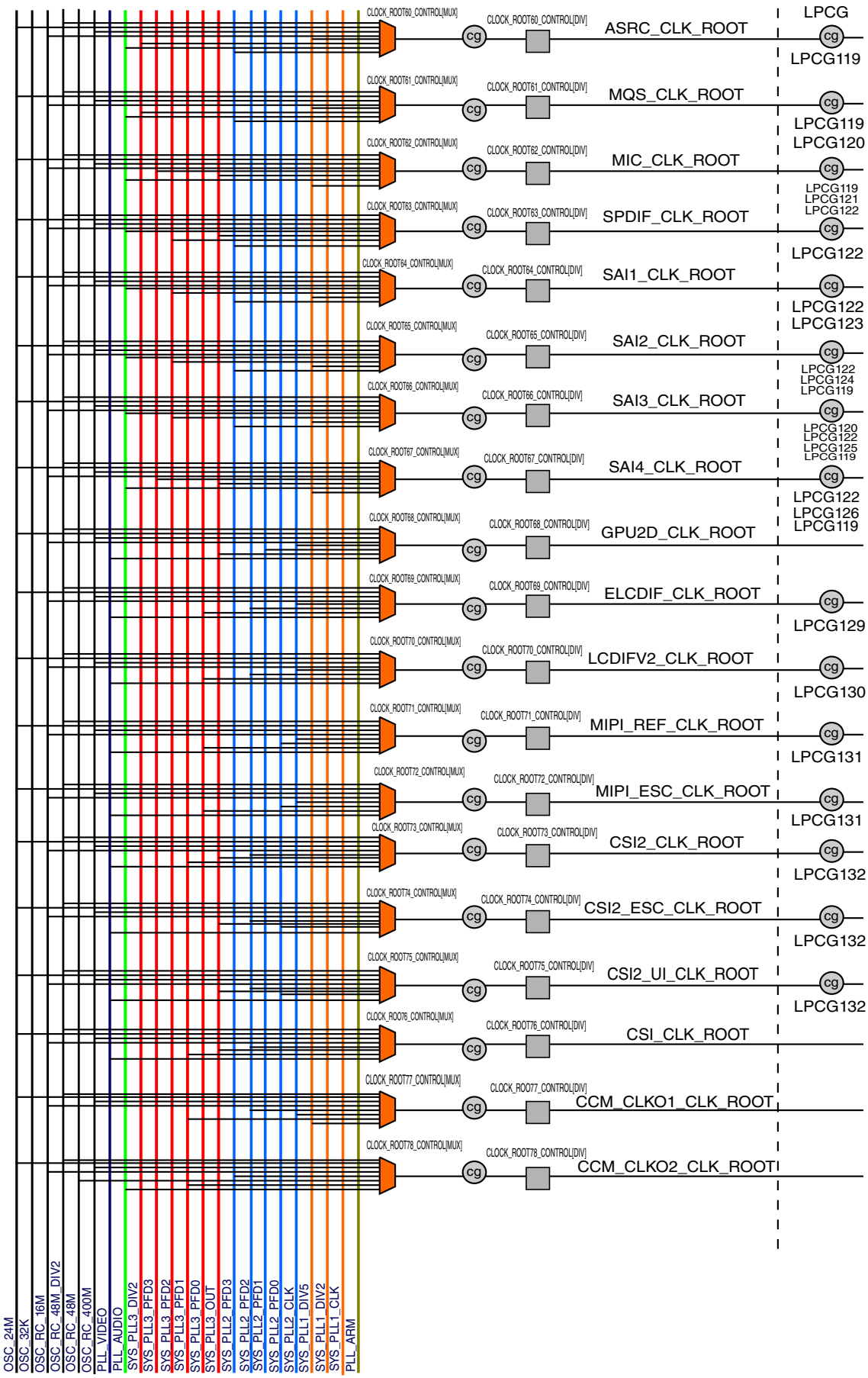


The figure below illustrates the clock roots of CCM and clock root generation.


CLOCK_ROOT0_CONTROL[MUX]
CLOCK_ROOT0_CONTROL[DIV]
M7_CLK_ROOT
CLOCK_ROOT1_CONTROL[MUX]
CLOCK_ROOT1_CONTROL[DIV]
M4_CLK_ROOT
CLOCK_ROOT2_CONTROL[MUX]
CLOCK_ROOT2_CONTROL[DIV]
BUS_CLK_ROOT
CLOCK_ROOT3_CONTROL[MUX]
CLOCK_ROOT3_CONTROL[DIV]
BUS_LPSR_CLK_ROOT
CLOCK_ROOT4_CONTROL[MUX]
CLOCK_ROOT4_CONTROL[DIV]
SEMC_CLK_ROOT
CLOCK_ROOT5_CONTROL[MUX]
CLOCK_ROOT5_CONTROL[DIV]
CSSYS_CLK_ROOT
CLOCK_ROOT6_CONTROL[MUX]
CLOCK_ROOT6_CONTROL[DIV]
CSTRACE_CLK_ROOT
CLOCK_ROOT7_CONTROL[MUX]
CLOCK_ROOT7_CONTROL[DIV]
M4_SYSTICK_CLK_ROOT
CLOCK_ROOT8_CONTROL[MUX]
CLOCK_ROOT8_CONTROL[DIV]
M7_SYSTICK_CLK_ROOT
CLOCK_ROOT9_CONTROL[MUX]
CLOCK_ROOT9_CONTROL[DIV]
ADC1_CLK_ROOT
CLOCK_ROOT10_CONTROL[MUX]
CLOCK_ROOT10_CONTROL[DIV]
ADC2_CLK_ROOT
CLOCK_ROOT11_CONTROL[MUX]
CLOCK_ROOT11_CONTROL[DIV]
ACMP_CLK_ROOT
CLOCK_ROOT12_CONTROL[MUX]
CLOCK_ROOT12_CONTROL[DIV]
FLEXIO1_CLK_ROOT
CLOCK_ROOT13_CONTROL[MUX]
CLOCK_ROOT13_CONTROL[DIV]
FLEXIO2_CLK_ROOT
CLOCK_ROOT14_CONTROL[MUX]
CLOCK_ROOT14_CONTROL[DIV]
GPT1_CLK_ROOT
CLOCK_ROOT15_CONTROL[MUX]
CLOCK_ROOT15_CONTROL[DIV]
GPT2_CLK_ROOT
CLOCK_ROOT16_CONTROL[MUX]
CLOCK_ROOT16_CONTROL[DIV]
GPT3_CLK_ROOT
CLOCK_ROOT17_CONTROL[MUX]
CLOCK_ROOT17_CONTROL[DIV]
GPT4_CLK_ROOT
CLOCK_ROOT18_CONTROL[MUX]
CLOCK_ROOT18_CONTROL[DIV]
GPT5_CLK_ROOT
CLOCK_ROOT19_CONTROL[MUX]
CLOCK_ROOT19_CONTROL[DIV]
GPT6_CLK_ROOT
cg
LPCG
LPCG6
LPCG33
LPCG2
LPCG42
LPCG42
LPCG1
LPCG0
LPCG55
LPCG56
LPCG58 -
LPCG61
LPCG53
LPCG54
LPCG64
LPCG65
LPCG66
LPCG67
LPCG68
LPCG69
OSC_24M
OSC_32K
OSC_RC_16M
OSC_RC_48M_DIV2
OSC_RC_48M
OSC_RC_400M
PLL_VIDEO
PLL_AUDIO
SYS_PLL3_DIV2
SYS_PLL3_PFD3
SYS_PLL3_PFD2
SYS_PLL3_PFD1
SYS_PLL3_PFD0
SYS_PLL3_OUT
SYS_PLL2_PFD3
SYS_PLL2_PFD2
SYS_PLL2_PFD1
SYS_PLL2_PFD0
SYS_PLL2_CLK
SYS_PLL1_DIV5
SYS_PLL1_DIV2
SYS_PLL1_CLK
PLL_ARM


LPCG
FLEXSPI1_CLK_ROOT
LPCG28
FLEXSPI2_CLK_ROOT
LPCG29
CAN1_CLK_ROOT
LPCG83
CAN2_CLK_ROOT
LPCG84
CAN3_CLK_ROOT
LPCG85
LPUART1_CLK_ROOT
LPCG86
LPUART2_CLK_ROOT
LPCG87
LPUART3_CLK_ROOT
LPCG88
LPUART4_CLK_ROOT
LPCG89
LPUART5_CLK_ROOT
LPCG90
LPUART6_CLK_ROOT
LPCG91
LPUART7_CLK_ROOT
LPCG92
LPUART8_CLK_ROOT
LPCG93
LPUART9_CLK_ROOT
LPCG94
LPUART10_CLK_ROOT
LPCG95
LPUART11_CLK_ROOT
LPCG96
LPUART12_CLK_ROOT
LPCG97
LPI2C1_CLK_ROOT
LPCG98
LPI2C2_CLK_ROOT
LPCG99
LPI2C3_CLK_ROOT
LPCG100
cg
OSC_24M
OSC_32K
OSC_RC_16M
OSC_RC_48M_DIV2
OSC_RC_48M
OSC_RC_400M
PLL_VIDEO
PLL_AUDIO
SYS_PLL3_DIV2
SYS_PLL3_PFD3
SYS_PLL3_PFD2
SYS_PLL3_PFD1
SYS_PLL3_PFD0
SYS_PLL3_OUT
SYS_PLL2_PFD3
SYS_PLL2_PFD2
SYS_PLL2_PFD1
SYS_PLL2_PFD0
SYS_PLL2_CLK
SYS_PLL1_DIV5
SYS_PLL1_DIV2
SYS_PLL1_CLK
PLL_ARM


CLOCK_ROOT40_CONTROL[MUX]
CLOCK_ROOT40_CONTROL[DIV]
LPI2C4_CLK_ROOT
LPCG
LPCG101
CLOCK_ROOT41_CONTROL[MUX]
CLOCK_ROOT41_CONTROL[DIV]
LPI2C5_CLK_ROOT
LPCG102
CLOCK_ROOT42_CONTROL[MUX]
CLOCK_ROOT42_CONTROL[DIV]
LPI2C6_CLK_ROOT
LPCG103
CLOCK_ROOT43_CONTROL[MUX]
CLOCK_ROOT43_CONTROL[DIV]
LPSPI1_CLK_ROOT
LPCG104
CLOCK_ROOT44_CONTROL[MUX]
CLOCK_ROOT44_CONTROL[DIV]
LPSPI2_CLK_ROOT
LPCG105
CLOCK_ROOT45_CONTROL[MUX]
CLOCK_ROOT45_CONTROL[DIV]
LPSPI3_CLK_ROOT
LPCG106
CLOCK_ROOT46_CONTROL[MUX]
CLOCK_ROOT46_CONTROL[DIV]
LPSPI4_CLK_ROOT
LPCG107
CLOCK_ROOT47_CONTROL[MUX]
CLOCK_ROOT47_CONTROL[DIV]
LPSPI5_CLK_ROOT
LPCG108
CLOCK_ROOT48_CONTROL[MUX]
CLOCK_ROOT48_CONTROL[DIV]
LPSPI6_CLK_ROOT
LPCG109
CLOCK_ROOT49_CONTROL[MUX]
CLOCK_ROOT49_CONTROL[DIV]
EMV1_CLK_ROOT
LPCG110
CLOCK_ROOT50_CONTROL[MUX]
CLOCK_ROOT50_CONTROL[DIV]
EMV2_CLK_ROOT
LPCG111
CLOCK_ROOT51_CONTROL[MUX]
CLOCK_ROOT51_CONTROL[DIV]
ENET1_CLK_ROOT
LPCG112
CLOCK_ROOT52_CONTROL[MUX]
CLOCK_ROOT52_CONTROL[DIV]
ENET2_CLK_ROOT
LPCG113
CLOCK_ROOT53_CONTROL[MUX]
CLOCK_ROOT53_CONTROL[DIV]
ENET_QOS_CLK_ROOT
LPCG114
CLOCK_ROOT54_CONTROL[MUX]
CLOCK_ROOT54_CONTROL[DIV]
ENET_25M_CLK_ROOT
CLOCK_ROOT55_CONTROL[MUX]
CLOCK_ROOT55_CONTROL[DIV]
ENET_TIMER1_CLK_ROOT
LPCG112
CLOCK_ROOT56_CONTROL[MUX]
CLOCK_ROOT56_CONTROL[DIV]
ENET_TIMER2_CLK_ROOT
LPCG113
CLOCK_ROOT57_CONTROL[MUX]
CLOCK_ROOT57_CONTROL[DIV]
ENET_TIMER3_CLK_ROOT
LPCG114
CLOCK_ROOT58_CONTROL[MUX]
CLOCK_ROOT58_CONTROL[DIV]
USDHC1_CLK_ROOT
LPCG117
CLOCK_ROOT59_CONTROL[MUX]
CLOCK_ROOT59_CONTROL[DIV]
USDHC2_CLK_ROOT
LPCG118
cg
OSC_24M
OSC_32K
OSC_RC_16M
OSC_RC_48M_DIV2
OSC_RC_48M
OSC_RC_400M
PLL_VIDEO
PLL_AUDIO
SYS_PLL3_DIV2
SYS_PLL3_PFD3
SYS_PLL3_PFD2
SYS_PLL3_PFD1
SYS_PLL3_PFD0
SYS_PLL3_OUT
SYS_PLL2_PFD3
SYS_PLL2_PFD2
SYS_PLL2_PFD1
SYS_PLL2_PFD0
SYS_PLL2_CLK
SYS_PLL1_DIV5
SYS_PLL1_DIV2
SYS_PLL1_CLK
PLL_ARM


OSC_24M
OSC_32K
OSC_RC_16M
OSC_RC_48M_DIV2
OSC_RC_48M
OSC_RC_400M
PLL_VIDEO
PLL_AUDIO
SYS_PLL3_DIV2
SYS_PLL3_PFD3
SYS_PLL3_PFD2
SYS_PLL3_PFD1
SYS_PLL3_PFD0
SYS_PLL3_OUT
SYS_PLL2_PFD3
SYS_PLL2_PFD2
SYS_PLL2_PFD1
SYS_PLL2_PFD0
SYS_PLL2_CLK
SYS_PLL1_DIV5
SYS_PLL1_DIV2
SYS_PLL1_CLK
PLL_ARM
CLOCK_ROOT60_CONTROL[MUX]
CLOCK_ROOT60_CONTROL[DIV]
ASRC_CLK_ROOT
CLOCK_ROOT61_CONTROL[MUX]
CLOCK_ROOT61_CONTROL[DIV]
MQS_CLK_ROOT
CLOCK_ROOT62_CONTROL[MUX]
CLOCK_ROOT62_CONTROL[DIV]
MIC_CLK_ROOT
CLOCK_ROOT63_CONTROL[MUX]
CLOCK_ROOT63_CONTROL[DIV]
SPDIF_CLK_ROOT
CLOCK_ROOT64_CONTROL[MUX]
CLOCK_ROOT64_CONTROL[DIV]
SAI1_CLK_ROOT
CLOCK_ROOT65_CONTROL[MUX]
CLOCK_ROOT65_CONTROL[DIV]
SAI2_CLK_ROOT
CLOCK_ROOT66_CONTROL[MUX]
CLOCK_ROOT66_CONTROL[DIV]
SAI3_CLK_ROOT
CLOCK_ROOT67_CONTROL[MUX]
CLOCK_ROOT67_CONTROL[DIV]
SAI4_CLK_ROOT
CLOCK_ROOT68_CONTROL[MUX]
CLOCK_ROOT68_CONTROL[DIV]
GPU2D_CLK_ROOT
CLOCK_ROOT69_CONTROL[MUX]
CLOCK_ROOT69_CONTROL[DIV]
ELCDIF_CLK_ROOT
CLOCK_ROOT70_CONTROL[MUX]
CLOCK_ROOT70_CONTROL[DIV]
LCDIFV2_CLK_ROOT
CLOCK_ROOT71_CONTROL[MUX]
CLOCK_ROOT71_CONTROL[DIV]
MIPI_REF_CLK_ROOT
CLOCK_ROOT72_CONTROL[MUX]
CLOCK_ROOT72_CONTROL[DIV]
MIPI_ESC_CLK_ROOT
CLOCK_ROOT73_CONTROL[MUX]
CLOCK_ROOT73_CONTROL[DIV]
CSI2_CLK_ROOT
CLOCK_ROOT74_CONTROL[MUX]
CLOCK_ROOT74_CONTROL[DIV]
CSI2_ESC_CLK_ROOT
CLOCK_ROOT75_CONTROL[MUX]
CLOCK_ROOT75_CONTROL[DIV]
CSI2_UI_CLK_ROOT
CLOCK_ROO76_CONTROL[MUX]
CLOCK_ROOT76_CONTROL[DIV]
CSI_CLK_ROOT
CLOCK_ROOT77_CONTROL[MUX]
CLOCK_ROO77_CONTROL[DIV]
CCM_CLKO1_CLK_ROOT
CLOCK_ROOT78_CONTROL[MUX]
CLOCK_ROOT78_CONTROL[DIV]
CCM_CLKO2_CLK_ROOT
cg
LPCG
LPCG119
LPCG119
LPCG120
LPCG119
LPCG121
LPCG122
LPCG122
LPCG122
LPCG123
LPCG122
LPCG124
LPCG119
LPCG120
LPCG122
LPCG125
LPCG119
LPCG122
LPCG126
LPCG119
LPCG129
LPCG130
LPCG131
LPCG131
LPCG132
LPCG132
LPCG132

15.5 Functional description

The following sections describe the functionality of CCM:

- Clock Source
- Clock Root
- Clock Gate
- Clock Group
- General Purpose Registers (GPR)

15.5.1 CCM Modes

CCM functional blocks support the following modes:

- Unassigned Mode
- Domain Mode
- CPU Low Power Mode (CPULPM)
- Setpoint Mode

NOTE

In any of the modes, every register in any Domain is readable. But write access will be restricted for protection.

15.5.1.1 Unassigned Mode

If a clock is not assigned to Domain or Setpoint Mode or CPU Low Power Mode, it automatically assumes Unassigned Mode. After reset, all clock sources, clock roots, clock gates, and clock groups remain in Unassigned Mode.

In Unassigned Mode, Domain access control scheme is disabled, but Trustzone access control is still active.

Clock Source

In Unassigned Mode, clock source setting comes from OSCPLLn_DIRECT registers (PLL/OSC is required to function in GPC mode). These register can only be accessed in Unassigned Mode if Trustzone authentication is passed. They have no effect in CPULPM or Setpoint Mode and are not writable in these modes.

Clock Root

In this mode, the clock root setting comes from the CLOCK_ROOTn_CONTROL registers. These registers can only be accessed in Unassigned Mode if Trustzone authentication is passed.

Clock Group

In Unassigned Mode, the clock group setting comes from CLOCK_GROUPn_CONTROLregisters. These registers can be modified from any domain in Unassigned Mode after passing Trustzone authentication.

Clock Gate (LPCG)

For this mode, the clock gate setting comes from the LPCGn_DIRECT registers. These register can only be accessed in Unassigned Mode if Trustzone authentication is passed. They cannot be modified or will not have any effect in CPULPM or Setpoint Mode.

15.5.1.2 Domain Mode

In Domain Mode, domain access control is enabled.

Clock Source

In this mode, clock source setting comes from OSCPLLn_DIRECT register. This register can only be accessed if Trustzone authentication is passed and access comes from domain in Whitelist. This register have no effect in CPU Low Power Mode and Setpoint Mode and is not writable.

Clock Root

In this mode, the Domain based authentication is activated and only domains on the Whitelist can change the value of the CLOCK_ROOTn_CONTROL register.

Clock Group

In this mode, the clock group setting comes from CLOCK_GROUPn_CONTROL. Since Domain based authentication is active in this mode, only the domains available on the Whitelist can modify these registers.

Clock Gate (LPCG)

For this mode, the clock gate setting come from the LPCGn_DIRECT register. This register can only be accessed in if Trustzone authentication is passed and access comes from domain in Whitelist. This register cannot be modified or will not have any effect in CPU Low Power Mode or Setpoint Mode.

NOTE

The Domain setting register can be written from any domain in Unassigned Mode, but need to pass Trustzone authentication.

15.5.1.3 CPU Low Power Mode (CPULPM)

The CPU Low power Mode will control the OSC/PLL/LPCG according to the low power signal from the GPC.

Clock Source

In this mode, there are 3-bit fields in OSCPLLn_DOMAIN register, which helps select the clock source dependency level for each CPU platform. The valid values are as follows:

- 0: This clock source is not needed in any mode
- 1: This clock source is needed in RUN mode, but not needed in WAIT, STOP mode
- 2: This clock source is needed in RUN and WAIT mode, but not needed in STOP mode
- 3: This clock source is needed in RUN, WAIT and STOP mode
- 4: this clock source is needed in RUN, WAIT, STOP and SUSPEND mode

Clock Root

Clock Root does not support this mode.

Clock Group

Clock Group does not support this mode.

Clock Gate (LPCG)

There are 3-bit clock dependent levels setting for each CPU platform in the register LPCGn_DOMAIN. The valid values are the same as the values shown for clock source.

The clock gate calculates whether the clock is dependent for each CPU platform and if not required, it will be shut down.

15.5.1.4 Setpoint Mode

Clock Source

The Setpoint registers define 16 work points of clock source using 16 bits for system work modes through OSCPLLn_SETPOINT registers.

As the system work mode transitions from one Setpoint to another, the clock source will look up the corresponding value and source the clock according to the response on the request from General Power Controller (GPC).

Clock Root

For clock roots running in this mode, the CLOCK_ROOTn_CONTROL register does not affect the clock root, and all the write access will be blocked.

The Setpoint registers define 16 work points of clock roots using 16 bits for system work modes. As the system work mode transition from one Setpoint to another, the clock root will look up corresponding value and changes the clock root setting to the value requested by GPC.

Please refer to CLOCK_ROOTn_SETPOINTm register to see which clock roots support Setpoints.

Speed Grade:

In Setpoints, speed grade information needs to be provided for clock root channel through CLOCK_ROOTn_SETPOINTm. The clock channel uses this information to determine whether to change the clock setting before or after a change in system supply voltage. A smaller grade value refers to faster clock speed.

Speed grade is being used by Setpoint logic to determine whether the clock is transitioning up or down between a Setpoint transition. Setpoint logic handles relationship between other components, such as power control. Each Setpoint setting is associated with a speed grade setting. A smaller speed grade value indicates higher speed. For example, the value of 0 refers to the fastest clock and the value of 15 refers to the slowest clock. For Setpoint setting with same speed, speed grade relation can be any or higher, lower or equal.

Clock Group

In this mode, the CLOCK_GROUPn_CONTROL registers do not affect the clock group and all write access is blocked.

The Setpoint registers define 16 work points of clock channel for system work modes. As the system work mode transition from one Setpoint to another, clock group will look up the corresponding value and change the clock group setting to the corresponding value requested from GPC.

In Setpoints, speed grade information needs to be provided for clock root channel through CLOCK_ROOTn_SETPOINTm.

Clock Gate (LPCG)

The Setpoint registers define 16 work points of clock gate for system work modes. As the system work mode transition from one Setpoint to another, clock gate will look up corresponding value and change the clock gate setting to the corresponding value requested from GPC.

In Setpoint Mode, the Domain setting register will not affect clock source, and all write access is blocked.

NOTE

These registers can only be changed in Unassigned Mode after passing Trustzone authentication.

After the clock channel, source, root, or gate is assigned to Setpoint Mode, the registers cannot be changed by application. If application needs to change the Setpoint value, the clock channel should be switched to Unassigned Mode.

15.5.2 Clock Sources

Clock roots are generated from various clock sources, which can be either oscillators or PLLs. Clock sources are implemented to control PLLs and oscillators by automatically turning them OFF or ON on system low power actions.

The following table shows the clock sources and the associated control registers.

Table 15-3. Clock Sources

Control Register	Clock Source	Typical PFD value	Description
OSCPLL0	OSC_RC_16M	-	16MHz RC OSC Output
OSCPLL1	OSC_RC_48M	-	48MHz RC OSC Output
OSCPLL2	OSC_RC_48M_DIV2	-	48MHz divided by 2 clock output
OSCPLL3	OSC_RC_400M	-	400MHz RC OSC output
OSCPLL4	OSC_24M	-	VCO (Not connected to the clock tree)
OSCPLL5	OSC_24M_CLK	-	24MHz main output clock
OSCPLL6	ARM_PLL	-	VCO (Not connected to the clock tree)
OSCPLL7	ARM_PLL_CLK	4	ARM PLL main output clock
OSCPLL8	SYS_PLL2	-	VCO (Not connected to the clock tree)
OSCPLL9	SYS_PLL2_CLK	1	System PLL2 main output clock
OSCPLL10	SYS_PLL2_PFD0	27	System PLL2 PFD0 clock
OSCPLL11	SYS_PLL2_PFD1	16	System PLL2 PFD1 clock

Table continues on the next page...

Table 15-3. Clock Sources (continued)

Control Register	Clock Source	Typical PFD value	Description
OSCPLL12	SYS_PLL2_PFD2	24	System PLL2 PFD2 clock
OSCPLL13	SYS_PLL2_PFD3	32	System PLL2 PFD3 clock
OSCPLL14	SYS_PLL3	-	VCO (Not connected to the clock tree)
OSCPLL15	SYS_PLL3_CLK	1	System PLL3 main output
OSCPLL16	SYS_PLL3_DIV2	2	System PLL3 divided by 2 clock
OSCPLL17	SYS_PLL3_PFD0	13	System PLL3 PFD0 clock
OSCPLL18	SYS_PLL3_PFD1	17	System PLL3 PFD1 clock
OSCPLL19	SYS_PLL3_PFD2	32	System PLL3 PFD2 clock
OSCPLL20	SYS_PLL3_PFD3	24	System PLL3 PFD3 clock
OSCPLL21	SYS_PLL1	-	VCO (Not connected to the clock tree)
OSCPLL22	SYS_PLL1_CLK	1	System PLL1 main clock output
OSCPLL23	SYS_PLL1_DIV2	2	System PLL1 divided by 2 clock
OSCPLL24	SYS_PLL1_DIV5	5	System PLL1 divided by 5 clock
OSCPLL25	AUDIO_PLL	-	VCO (Not connected to the clock tree)
OSCPLL26	AUDIO_PLL_CLK	-	Audio PLL main clock
OSCPLL27	VIDEO_PLL	-	VCO (Not connected to the clock tree)
OSCPLL28	VIDEO_PLL_CLK	-	Video PLL main clock output

NOTE

The clock source calculates whether the clock is required for each CPU domain. If clock source is not needed for any domain, clock will be shutdown.

The typical PDF value for AUDIO_PLL_CLK and VIDEO_PLL_CLK depends on the user application.

15.5.3 Clock Root

CCM clock root generation contains multiple clock root channels. All clock roots are asynchronous, even when all the clock root setting are the same.

All clock channel contains an 8-to-1 MUX, and an 8-bit divider. The clock MUX selects 1 clock out of 8 clock inputs. The 8-bit divider can divide selected clock by up to 256. The clock output of the clock channel can be gated off.

The clock root channel setting can be changed from any value to any value, and at any time. It is also possible to change the clock setting when the clock root channel is gating off. If more than one setting is changed at the same time, internal logic will use a

procedure to change the setting. The procedure will make sure the clock output is not faster than the current clock root setting or target clock root setting. If the application does care about the clock during transition, software will need to change the setting one field at a time. This typically happens when a clock must be faster than the given frequency. In this case, the application needs to change each field one at a time and wait for the field to take effect before changing the next field.

Although a clock root channel is designed to be able to switch clocks regardless of clock input status. When the current selected clock input is off, the application needs to avoid changing the clock MUX after the clock input is turned on. This will lead to unstable behavior for the clock root and loading peripherals (peripherals driven by a clock). To avoid this scenario, the application can either switch the clock MUX before turning on the clock input, or wait until the clock input is stable before changing the MUX option.

The clock root channels have 3 working modes, Unassigned Mode, Domain Mode, and Setpoint Mode. Unassigned Mode is an implicit mode, if a clock channel is not assigned to Domain or Setpoint Mode, it defaults to Unassigned Mode. After reset, all clock root channels are in Unassigned Mode.

All registers are readable for any Domain in any mode, but write access are restricted. In Unassigned Mode, Domain access control scheme is disabled, but Trustzone access control is still active. In Unassigned Mode and Domain Mode, the clock root setting comes from the CLOCK_ROOTn_CONTROL register. This register can be written from any Domain in Unassigned Mode, but needs to pass Trustzone authentication.

In Domain control Mode, Domain based authentication is activate and only the Domain on the Whitelist can change the value of CLOCK_ROOTn_CONTROL register. In Setpoint Mode, this register does not affect the clock root, and all write access will be blocked. CLOCK_ROOTn_SETPOINT registers define 16 work points of the clock channel for system work modes. As system work mode transitions from one Setpoint to another, the clock root will look up the corresponding value and will change the clock root setting to the value requested from GPC. These registers can only be changed in Unassigned Mode, after passing Trustzone authentication.

After the clock channel is assigned to Setpoint Mode, CLOCK_ROOTn_SETPOINT registers cannot be changed by the application. If the application needs to change the Setpoint value, Unassigned Mode needs to be switched to, before doing any modification. In Setpoints, speed grade information needs to be provided for the clock root channel.

The table below shows the Clock roots and the associated control registers.

UD, NM and OD represent Under-drive, Nominal, and Over-drive frequencies.

If 'SP' is blank, Setpoint is present, and if 'SP' is N, Setpoint is not present.

If 'Frac' is blank, Fraction is present, and if 'Frac' is N, Fraction is not present.

Table 15-4. Clock Roots

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
CLOCK_ROOT0	M7_CLK_ROOT	240	700 600 +ECC	1000 800 +ECC			000 - OSC_RC_48M_DIV2 001 - OSC_24M 010 - OSC_RC_400M 011 - OSC_RC_16M 100 - PLL_ARM_CLK 101 - SYS_PLL1_CLK 110 - SYS_PLL3_CLK 111 - VIDEO_PLL_CLK
CLOCK_ROOT1	M4_CLK_ROOT	120	240	400			000 - OSC_RC_48M_DIV2 001 - OSC_24M 010 - OSC_RC_400M 011 - OSC_RC_16M 100 - SYS_PLL3_PFD3 101 - SYS_PLL3_CLK 110 - SYS_PLL2_CLK 111 - SYS_PLL1_DIV5
CLOCK_ROOT2	BUS_CLK_ROOT	100	200	240			000 - OSC_RC_48M_DIV2 001 - OSC_24M 010 - OSC_RC_400M 011 - OSC_RC_16M 100 - SYS_PLL3_CLK 101 - SYS_PLL1_DIV5 110 - SYS_PLL2_CLK 111 - SYS_PLL2_PFD3
CLOCK_ROOT3	BUS_LPSR_CLK_ROOT	60	120	160			000 - OSC_RC_48M_DIV2 001 - OSC_24M 010 - OSC_RC_400M 011 - OSC_RC_16M 100 - SYS_PLL3_PFD3 101 - SYS_PLL3_CLK 110 - SYS_PLL2_CLK 111 - SYS_PLL1_DIV5
CLOCK_ROOT4	SEMC_CLK_ROOT	132	200	200			000 - OSC_RC_48M_DIV2 001 - OSC_24M 010 - OSC_RC_400M 011 - OSC_RC_16M 100 - SYS_PLL1_DIV5 101 - SYS_PLL2_CLK

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							110 - SYS_PLL2_PFD1
							111 - SYS_PLL3_PFD0
CLOCK_ROOT5	CSSYS_CLK_ROOT	66	132	132	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT6	CSTRACE_CLK_ROOT	88	132	132	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_PFD1
							111 - SYS_PLL2_CLK
CLOCK_ROOT7	M4_SYSTICK_CLK_ROOT	50	50	50	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD0
							111 - SYS_PLL1_DIV5
CLOCK_ROOT8	M7_SYSTICK_CLK_ROOT	50	50	50	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL3_DIV2
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD0
CLOCK_ROOT9	ADC1_CLK_ROOT	66	88	88	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT10	ADC2_CLK_ROOT	66	88	88	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT11	ACMP_CLK_ROOT	240	240	240	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_CLK
							101 - SYS_PLL1_DIV5
							110 - AUDIO_PLL_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT12	FLEXIO1_CLK_ROOT	80	120	120	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT13	FLEXIO2_CLK_ROOT	80	120	120	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT14	GPT1_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL3_PFD3
CLOCK_ROOT15	GPT2_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - AUDIO_PLL_CLK
							111 - VIDEO_PLL_CLK
CLOCK_ROOT16	GPT3_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - AUDIO_PLL_CLK
							111 - VIDEO_PLL_CLK
CLOCK_ROOT17	GPT4_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL3_PFD3
CLOCK_ROOT18	GPT5_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL3_PFD3

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
CLOCK_ROOT19	GPT6_CLK_ROOT	80	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL3_PFD3
CLOCK_ROOT20	FLEXSPI1_CLK_ROOT	111	332	332		N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD0
							101 - SYS_PLL2_CLK
							110 - SYS_PLL2_PFD2
							111 - SYS_PLL3_CLK
CLOCK_ROOT21	FLEXSPI2_CLK_ROOT	111	332	332		N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD0
							101 - SYS_PLL2_CLK
							110 - SYS_PLL2_PFD2
							111 - SYS_PLL3_CLK
CLOCK_ROOT22	CAN1_CLK_ROOT	80	80	80	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT23	CAN2_CLK_ROOT	80	80	80	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT24	CAN3_CLK_ROOT	80	80	80	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD3
							111 - SYS_PLL1_DIV5
CLOCK_ROOT25	LPUART1_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT26	LPUART2_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT27	LPUART3_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT28	LPUART4_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT29	LPUART5_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT30	LPUART6_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT31	LPUART7_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT32	LPUART8_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT33	LPUART9_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT34	LPUART10_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT35	LPUART11_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD3
							111 - SYS_PLL1_DIV5
CLOCK_ROOT36	LPUART12_CLK_ROOT	80	80	80	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD3
							111 - SYS_PLL1_DIV5
CLOCK_ROOT37	LPI2C1_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
CLOCK_ROOT38	LPI2C2_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT39	LPI2C3_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT40	LPI2C4_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT41	LPI2C5_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD3
							111 - SYS_PLL1_DIV5
CLOCK_ROOT42	LPI2C6_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							110 - SYS_PLL2_PFD3
							111 - SYS_PLL1_DIV5
CLOCK_ROOT43	LPSPI1_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT44	LPSPI2_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT45	LPSPI3_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT46	LPSPI4_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT47	LPSPI5_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL1_DIV5
CLOCK_ROOT48	LPSPI6_CLK_ROOT	90	135	135	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL1_DIV5
CLOCK_ROOT49	EMV1_CLK_ROOT	66	66	66	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT50	EMV2_CLK_ROOT	66	66	66	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_DIV2
							101 - SYS_PLL1_DIV5
							110 - SYS_PLL2_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT51	ENET1_CLK_ROOT	50	50	50	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT52	ENET2_CLK_ROOT	50	125	125	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT53	ENET_QOS_CLK_ROOT	50	125	125	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT54	ENET_25M_CLK_ROOT	50	50	50	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT55	ENET_TIMER1_CLK_ROOT	100	200	200	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT56	ENET_TIMER2_CLK_ROOT	100	200	200	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
CLOCK_ROOT57	ENET_TIMER3_CLK_ROOT	100	200	200	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV2
							101 - AUDIO_PLL_CLK
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD1
CLOCK_ROOT58	USDHC1_CLK_ROOT	100	200	200	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL2_PFD0
							110 - SYS_PLL1_DIV5
							111 - PLL_ARM_CLK
CLOCK_ROOT59	USDHC2_CLK_ROOT	100	400	400	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL2_PFD0
							110 - SYS_PLL1_DIV5
							111 - PLL_ARM_CLK
CLOCK_ROOT60	ASRC_CLK_ROOT	60	240	240	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV5
							101 - SYS_PLL3_DIV2
							110 - AUDIO_PLL_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT61	MQS_CLK_ROOT	66	66	66	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL1_DIV5
							101 - SYS_PLL3_DIV2

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							110 - AUDIO_PLL_CLK
							111 - SYS_PLL2_PFD3
CLOCK_ROOT62	MIC_CLK_ROOT	60	80	80	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - AUDIO_PLL_CLK
							111 - SYS_PLL1_DIV5
CLOCK_ROOT63	SPDIF_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - AUDIO_PLL_CLK
							101 - SYS_PLL3_CLK
							110 - SYS_PLL3_PFD2
							111 - SYS_PLL2_PFD3
CLOCK_ROOT64	SAI1_CLK_ROOT	33	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - AUDIO_PLL_CLK
							101 - SYS_PLL3_PFD2
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD3
CLOCK_ROOT65	SAI2_CLK_ROOT	33	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - AUDIO_PLL_CLK
							101 - SYS_PLL3_PFD2
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD3
CLOCK_ROOT66	SAI3_CLK_ROOT	33	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							100 - AUDIO_PLL_CLK
							101 - SYS_PLL3_PFD2
							110 - SYS_PLL1_DIV5
							111 - SYS_PLL2_PFD3
CLOCK_ROOT67	SAI4_CLK_ROOT	33	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL3_PFD3
							101 - SYS_PLL3_CLK
							110 - AUDIO_PLL_CLK
							111 - SYS_PLL1_DIV5
CLOCK_ROOT68	GPU2D_CLK_ROOT	176	500	500	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL2_PFD1
							110 - SYS_PLL3_CLK
							111 - VIDEO_PLL_CLK
CLOCK_ROOT69	ELCDIF_CLK_ROOT	82	150	150	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL2_PFD2
							110 - SYS_PLL3_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT70	LCDIFV2_CLK_ROOT	82	150	150	N		000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL2_PFD2
							110 - SYS_PLL3_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT71	MIPI_REF_CLK_ROOT	133	133	133	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL2_PFD0
							110 - SYS_PLL3_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT72	MIPI_ESC_CLK_ROOT	66	133	133	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_CLK
							101 - SYS_PLL2_PFD0
							110 - SYS_PLL3_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT73	CSI2_CLK_ROOT	133	200	200	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT74	CSI2_ESC_CLK_ROOT	133	133	133	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD0
							111 - VIDEO_PLL_CLK
CLOCK_ROOT75	CSI2_UI_CLK_ROOT	200	200	200	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL3_CLK
							110 - SYS_PLL2_PFD0
							111 - VIDEO_PLL_CLK

Table continues on the next page...

Table 15-4. Clock Roots (continued)

Control Register	Clock Roots	Max Frequency (MHz)			SP	Frac	Source Select
		UD	NM	OD (FBB)			(CLOCK_ROOTn_CONTROL[MUX])
CLOCK_ROOT76	CSI_CLK_ROOT	66	66	66	N	N	000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL3_CLK
							110 - SYS_PLL3_PFD1
							111 - VIDEO_PLL_CLK
CLOCK_ROOT77	CCM_CLKO1_CLK_ROOT	120	120	120			000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD2
							101 - SYS_PLL2_CLK
							110 - SYS_PLL3_PFD1
							111 - SYS_PLL1_DIV5
CLOCK_ROOT78	CCM_CLKO2_CLK_ROOT	200	200	200			000 - OSC_RC_48M_DIV2
							001 - OSC_24M
							010 - OSC_RC_400M
							011 - OSC_RC_16M
							100 - SYS_PLL2_PFD3
							101 - OSC_RC_48M
							110 - SYS_PLL3_PFD1
							111 - AUDIO_PLL_CLK

NOTE

The frequency values of the cores are dependent on the part configuration. Please see the datasheet for supported core frequencies.

15.5.4 Clock Gate

Before a clock root goes to on-chip peripherals, the clock roots go through the Low Power Clock Gates (LPCG). These LPCGs are implemented to perform automatic clock gating when a domain enters or leaves low-power states. LPCGs are implemented to control peripheral functions. Different LPCGs could interface different clocks of the same peripheral.

Clock gates implements four modes: Unassigned Mode, Domain Mode, CPU Low Power Mode, and Setpoint Mode. Unassigned Mode is an implicit mode. If a clock gate is not assigned to Domain, Setpoint Mode or CPU Low Power Mode, the default mode is Unassigned Mode. After reset, all clock gates work in Unassigned Mode. All registers are readable for any domain in any mode, but write access is restricted for protection. In Unassigned Mode, Domain access control scheme is disabled, but Trustzone access control is still active.

In Unassigned Mode, clock gate settings come from LPCGn_DIRECT registers. Application can only change clock gate status by accessing LPCGn_DIRECT registers. These can only be accessed in Unassigned Mode if Trustzone authentication was passed. LPCGn_DIRECT registers have no effect and are not writable in CPULPM or Setpoint Mode.

In Domain Mode, the clock gate setting come from the LPCGn_DIRECT register. This register can only be accessed in if Trustzone authentication is passed and access comes from domain in Whitelist.

In CPULPM, there are 3-bit clock dependent level settings for each CPU platform in the LPCGx_DOMAIN registers. The bigger the 3-bit value is, the stronger the clock dependency. Valid values are as follows:

- 0: This clock is not needed in any mode
- 1: This clock is needed in RUN mode, but not needed in WAIT, STOP mode
- 2: This clock is needed in RUN and WAIT mode, but not needed in STOP mode
- 3: This clock is needed in RUN, WAIT and STOP mode
- 4: This clock is always on in any mode (including SUSPEND)

NOTE

The clock gates calculate whether the clock is dependent for each CPU Domain. If clock is not needed for any domain, the clock will be shutdown. The Domain setting register can be written from any domain in Unassigned Mode, but needs to pass Trustzone authentication. In Domain control Mode, the Domain setting register is locked and not changeable. If the application needs to change the Domain setting, it should switch the clock gate back into Unassigned Mode. In Setpoint

Mode, the Domain register does not affect the clock gate, and all write accesses are blocked.

The Setpoint registers define 16 work points of the clock gates by 16 register bits for system work modes. As the work modes transition from one Setpoint to another, the clock gate will look up corresponding value and will gate the clock according to the value requested by GPC. These registers can only be changed in Unassigned Mode, after passing Trustzone authentication. After the clock gate is assigned to Setpoint Mode, Setpoint registers cannot be changed by the application. If the application needs to change Setpoint value, the clock gate must be switched to Unassigned Mode as a first step.

Table 15-5. Clock Gate Table

Gating Registers	LPCG Enable
LPCG0	clk_enable_cm7
LPCG1	clk_enable_cm4
LPCG2	clk_enable_sim_m7
LPCG3	clk_enable_sim_m
LPCG4	clk_enable_sim_disp
LPCG5	clk_enable_sim_per
LPCG6	clk_enable_sim_lpsr
LPCG7	clk_enable_anadig
LPCG8	clk_enable_dcdc
LPCG9	clk_enable_src
LPCG10	clk_enable_ccm
LPCG11	clk_enable_gpc
LPCG12	clk_enable_ssarc
LPCG13	clk_enable_sim_r
LPCG14	clk_enable_wdog1
LPCG15	clk_enable_wdog2
LPCG16	clk_enable_wdog3
LPCG17	clk_enable_wdog4
LPCG18	clk_enable_ewm
LPCG19	clk_enable_sema
LPCG20	clk_enable_mu_a
LPCG21	clk_enable_mu_b
LPCG22	clk_enable_edma
LPCG23	clk_enable_edma_lpsr
LPCG24	clk_enable_romcp
LPCG25	clk_enable_ocram
LPCG26	clk_enable_flexram
LPCG27	clk_enable_lpsrmem

Table continues on the next page...

Table 15-5. Clock Gate Table (continued)

Gating Registers	LPCG Enable
LPCG28	clk_enable_flexspi1
LPCG29	clk_enable_flexspi2
LPCG30	clk_enable_rdc
LPCG31	clk_enable_m7_xrdc
LPCG32	clk_enable_m4_xrdc
LPCG33	clk_enable_semc
LPCG34	clk_enable_xecc
LPCG35	clk_enable_iee
LPCG36	clk_enable_key_manager
LPCG37	clk_enable_ocotp
LPCG38	clk_enable_snvs_hp
LPCG39	clk_enable_snvs
LPCG40	clk_enable_caam
LPCG41	clk_enable_jtag_mux
LPCG42	clk_enable_cstrace
LPCG43	clk_enable_xbar1
LPCG44	clk_enable_xbar2
LPCG45	clk_enable_xbar3
LPCG46	clk_enable_aoi1
LPCG47	clk_enable_aoi2
LPCG48	clk_enable_adc_etc
LPCG49	clk_enable_iomuxc
LPCG50	clk_enable_iomuxc_lpsr
LPCG51	clk_enable_gpio
LPCG52	clk_enable_kpp
LPCG53	clk_enable_flexio1
LPCG54	clk_enable_flexio2
LPCG55	clk_enable_adc1
LPCG56	clk_enable_adc2
LPCG57	clk_enable_dac
LPCG58	clk_enable_acmp1
LPCG59	clk_enable_acmp2
LPCG60	clk_enable_acmp3
LPCG61	clk_enable_acmp4
LPCG62	clk_enable_pit1
LPCG63	clk_enable_pit2
LPCG64	clk_enable_gpt1
LPCG65	clk_enable_gpt2
LPCG66	clk_enable_gpt3

Table continues on the next page...

Table 15-5. Clock Gate Table (continued)

Gating Registers	LPCG Enable
LPCG67	clk_enable_gpt4
LPCG68	clk_enable_gpt5
LPCG69	clk_enable_gpt6
LPCG70	clk_enable_qtimer1
LPCG71	clk_enable_qtimer2
LPCG72	clk_enable_qtimer3
LPCG73	clk_enable_qtimer4
LPCG74	clk_enable_enc1
LPCG75	clk_enable_enc2
LPCG76	clk_enable_enc3
LPCG77	clk_enable_enc4
LPCG78	Reserved
LPCG79	clk_enable_flexpwm1
LPCG80	clk_enable_flexpwm2
LPCG81	clk_enable_flexpwm3
LPCG82	clk_enable_flexpwm4
LPCG83	clk_enable_can1
LPCG84	clk_enable_can2
LPCG85	clk_enable_can3
LPCG86	clk_enable_lpuart1
LPCG87	clk_enable_lpuart2
LPCG88	clk_enable_lpuart3
LPCG89	clk_enable_lpuart4
LPCG90	clk_enable_lpuart5
LPCG91	clk_enable_lpuart6
LPCG92	clk_enable_lpuart7
LPCG93	clk_enable_lpuart8
LPCG94	clk_enable_lpuart9
LPCG95	clk_enable_lpuart10
LPCG96	clk_enable_lpuart11
LPCG97	clk_enable_lpuart12
LPCG98	clk_enable_lpi2c1
LPCG99	clk_enable_lpi2c2
LPCG100	clk_enable_lpi2c3
LPCG101	clk_enable_lpi2c4
LPCG102	clk_enable_lpi2c5
LPCG103	clk_enable_lpi2c6
LPCG104	clk_enable_lpspi1
LPCG105	clk_enable_lpspi2

Table continues on the next page...

Table 15-5. Clock Gate Table (continued)

Gating Registers	LPCG Enable
LPCG106	clk_enable_lpspi3
LPCG107	clk_enable_lpspi4
LPCG108	clk_enable_lpspi5
LPCG109	clk_enable_lpspi6
LPCG110	clk_enable_sim1
LPCG111	clk_enable_sim2
LPCG112	clk_enable_enet
LPCG113	clk_enable_enet_1g
LPCG114	clk_enable_enet_qos
LPCG115	clk_enable_usb
LPCG116	clk_enable_cdog
LPCG117	clk_enable_usdhc1
LPCG118	clk_enable_usdhc2
LPCG119	clk_enable_asrc
LPCG120	clk_enable_mqs
LPCG121	clk_enable_mic
LPCG122	clk_enable_spdif
LPCG123	clk_enable_sai1
LPCG124	clk_enable_sai2
LPCG125	clk_enable_sai3
LPCG126	clk_enable_sai4
LPCG127	clk_enable_pxp
LPCG128	clk_enable_gpu2d
LPCG129	clk_enable_elcdif
LPCG130	clk_enable_lcdifv2
LPCG131	clk_enable_mipi_dsi
LPCG132	clk_enable_mipi_csi
LPCG133	clk_enable_csi
LPCG134	clk_enable_dcic_mipi
LPCG135	clk_enable_dcic_lcd
LPCG136	clk_enable_video_mux

15.5.5 Clock Group

Clock group creates a divided clock for a secondary clock input or a peripheral. The generated clock is synchronized to its parent clock. Each group contains a 4-bit clock divider which can divide up to 16. There is an 8-bit reset divider on clock divider. Clock divider will be synchronized on reset divider overflow. Reset divider and clock divider should be set in same write operation.

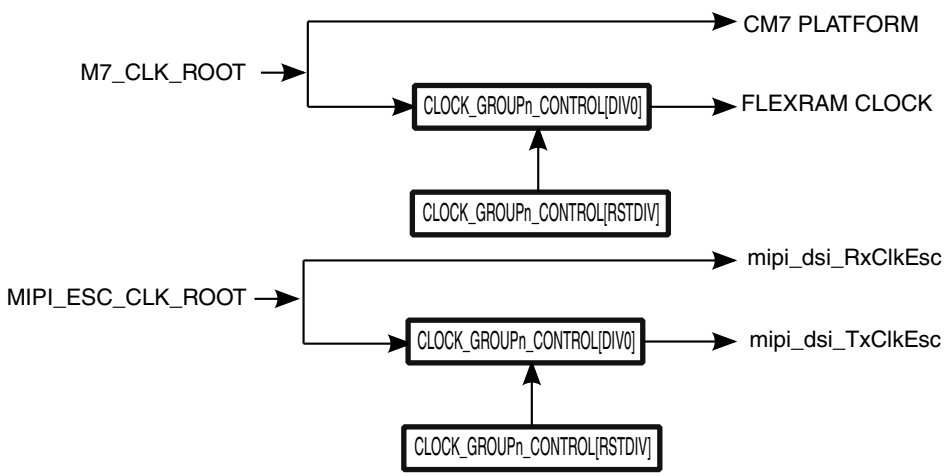
There are 2 clock groups in this chip as shown in the table below. Flexram clock group generates flexram AXI port clock, and the mipi_esc clock group generates mipi esc clock.

Each clock supports a 4-bit divider, which can divide down by 16. In a clock group, a restart cycle needs to be set. This restart cycle value should be a common multiple of all dividers in the clock group.

NOTE

If the wrong restart cycle value is set, the group behaviour will become unpredictable and the clocks cannot be recovered even after writing the correct value.

Each clock group contains a Manage counter. The figure below shows the manage counter and the group controls.



The clock group setting can be changed from any value to another value at any time. It can also be changed while the clock group is gated OFF. The clock group will update the setting on restart counter overflow. However, the clock group will take a few cycles to complete the action due to synchronous signal generation.

The clock groups can work in all modes.

The following table lists the clock groups.

Control Register	Clock Group
CLOCK_GROUP0	flexram_clk_group
CLOCK_GROUP1	mipi_esc_clk_group

15.5.6 Clock Observe

The clock observe slices are used to observe on-chip clocks or signals. Each observe channel contains a 512-to-1 MUX, an inverter, and an 8-bit divider. The clock MUX selects one observe signal out of 512 inputs. The 8-bit divider can divide the selected clock by up to 256. The inverter is selectable by software. If any of the settings change, the logic will be automatically reset. However, if the settings remain unchanged, a reset can be forced by writing "1" to OBSERVEn_CONTROL[RESET]. The table below shows the target signal, selection index, and the slice number. The target signal can be observed at associated slice number when the corresponding index is selected via OBSERVEn_CONTROL[SELECT].

NOTE

The observe functionality is for debug purposes only, and the accuracy is not guaranteed. Use of this field should be limited to room temperature, and frequency limited to less than 400 MHz.

Target	SELECT index	Slice Number
M7_CLK_ROOT	128	4
M4_CLK_ROOT	129	0
BUS_CLK_ROOT	130	2
BUS_LPSR_CLK_ROOT	131	0
SEMC_CLK_ROOT	132	2
CSSYS_CLK_ROOT	133	2
CSTRACE_CLK_ROOT	134	2
M4_SYSTICK_CLK_ROOT	135	0
M7_SYSTICK_CLK_ROOT	136	2
ADC1_CLK_ROOT	137	2
ADC2_CLK_ROOT	138	2
ACMP_CLK_ROOT	139	2
FLEXIO1_CLK_ROOT	140	2
FLEXIO2_CLK_ROOT	141	2
GPT1_CLK_ROOT	142	2
GPT2_CLK_ROOT	143	2

Table continues on the next page...

Target	SELECT index	Slice Number
GPT3_CLK_ROOT	144	2
GPT4_CLK_ROOT	145	2
GPT5_CLK_ROOT	146	2
GPT6_CLK_ROOT	147	2
FLEXSPI1_CLK_ROOT	148	2
FLEXSPI2_CLK_ROOT	149	2
CAN1_CLK_ROOT	150	2
CAN2_CLK_ROOT	151	2
CAN3_CLK_ROOT	152	0
LPUART1_CLK_ROOT	153	2
LPUART2_CLK_ROOT	154	2
LPUART3_CLK_ROOT	155	2
LPUART4_CLK_ROOT	156	2
LPUART5_CLK_ROOT	157	2
LPUART6_CLK_ROOT	158	2
LPUART7_CLK_ROOT	159	2
LPUART8_CLK_ROOT	160	2
LPUART9_CLK_ROOT	161	2
LPUART10_CLK_ROOT	162	2
LPUART11_CLK_ROOT	163	0
LPUART12_CLK_ROOT	164	0
LPI2C1_CLK_ROOT	165	2
LPI2C2_CLK_ROOT	166	2
LPI2C3_CLK_ROOT	167	2
LPI2C4_CLK_ROOT	168	2
LPI2C5_CLK_ROOT	169	0
LPI2C6_CLK_ROOT	170	0
LPSPI1_CLK_ROOT	171	2
LPSPI2_CLK_ROOT	172	2
LPSPI3_CLK_ROOT	173	2
LPSPI4_CLK_ROOT	174	2
LPSPI5_CLK_ROOT	175	0
LPSPI6_CLK_ROOT	176	0
EMV1_CLK_ROOT	177	2
EMV2_CLK_ROOT	178	2
ENET1_CLK_ROOT	179	2
ENET2_CLK_ROOT	180	2
ENET_QOS_CLK_ROOT	181	2
ENET_25M_CLK_ROOT	182	2
ENET_TIMER1_CLK_ROOT	183	2
ENET_TIMER2_CLK_ROOT	184	2

Table continues on the next page...

Target	SELECT index	Slice Number
ENET_TIMER3_CLK_ROOT	185	2
USDHC1_CLK_ROOT	186	2
USDHC2_CLK_ROOT	187	2
ASRC_CLK_ROOT	188	2
MQS_CLK_ROOT	189	2
MIC_CLK_ROOT	190	0
SPDIF_CLK_ROOT	191	2
SAI1_CLK_ROOT	192	2
SAI2_CLK_ROOT	193	2
SAI3_CLK_ROOT	194	2
SAI4_CLK_ROOT	195	0
GPU2D_CLK_ROOT	196	2
ELCDIF_CLK_ROOT	197	2
LCDIFV2_CLK_ROOT	198	2
MIPI_REF_CLK_ROOT	199	2
MIPI_ESC_CLK_ROOT	200	2
CSI2_CLK_ROOT	201	2
CSI2_ESC_CLK_ROOT	202	2
CSI2_UI_CLK_ROOT	203	2
CSI_CLK_ROOT	204	2
CCM_CKO1_CLK_ROOT	205	0
CCM_CKO2_CLK_ROOT	206	2
CM7_CORE_STCLKEN	207	4
CCM_FLEXRAM_CLK_ROOT	208	4
MIPI_DSI_TXESC	209	2
MIPI_DSI_RXESC	210	2
OSC_RC_16M	224	0
OSC_RC_48M	225	0
OSC_RC_48M_DIV2	226	0
OSC_RC_400M	227	0
OSC_24M_OUT	229	0
PLL_ARM_OUT	231	2
SYS_PLL2_OUT	233	2
SYS_PLL2_PFD0	234	2
SYS_PLL2_PFD1	235	2
SYS_PLL2_PFD2	236	2
SYS_PLL2_PFD3	237	2
SYS_PLL3_OUT	239	2
SYS_PLL3_DIV2	240	2
SYS_PLL3_PFD0	241	2
SYS_PLL3_PFD1	242	2

Table continues on the next page...

Target	SELECT index	Slice Number
SYS_PLL3_PFD2	243	2
SYS_PLL3_PFD3	244	2
SYS_PLL1_OUT	246	2
SYS_PLL1_DIV2	247	2
SYS_PLL1_DIV5	248	2
PLL_AUDIO_OUT	250	2
PLL_VIDEO_OUT	252	2

Observe slice can measure frequency on selected signals or clocks. Each time an observe signal selection changes, all measurement data will be reset, and the measurement will restart. During measurement, the register value may change, and read from registers may not be valid. To get the measurement result, software must turn OFF the observe slice before the measurement results are read. For frequency measurements, a 32KHz clock is used as a time reference. The 32KHz clock is divided down to 64Hz resolution. The measurement unit records the maximum and minimum value that has occurred. The observe signal can be directly read from the OBSERVEn_STATUS0 register. The clock root channels have 2 work modes: unassigned mode and domain mode. In both modes, the clock root setting comes from the OBSERVEn_CONTROL register. In unassigned mode, this register can be written from any domain, but it needs to pass trustzone authentication. In domain mode, the domain based authentication is activated, and only the domains listed in the Whitelist can change the value of the OBSERVEn_CONTROL register.

NOTE

The observe functionality is for debug purposes only, and the accuracy is not guaranteed. Use of this field should be limited to room temperature, and frequency limited to less than 400 MHz.

15.5.7 General Purpose Registers (GPR)

The general purpose registers provide miscellaneous controls and data saving functions. The GPRs are divided in two regions - shared and private region.

In shared region, all CPU domains access the same registers. If any CPU updates the value in the shared region, all the other CPU domains can read out the new value. The registers have an access control bit field to protect it from being changed unexpectedly.

In private region, each CPU domain has its own register. A CPU domain can only access registers belonging to itself. Since all the registers share the same address, a CPU domain will not have access to registers belonging to another CPU domain.

Shared registers have 2 work modes, Unassigned Mode and Domain Mode. Private registers work in Unassigned Mode only. Whitelist is ignored by private registers. Unassigned Mode is an implicit mode, if a shared register is not assigned to Domain Mode, the default mode is Unassigned Mode. After reset, all registers work in Unassigned Mode.

All registers are readable for any domain in any mode, but write access is restricted for protection. In Unassigned Mode, Domain access control scheme is disabled, but Trustzone access control is still active.

Type	GPR	Position	Width	Description
Shared	0	0	4	Setpoint LPCG response time. This value need be set when Setpoint controlled LPCG is controlling a clock slower than 16MHz. The value need be no less than the multiple number of the slowest clock to 16M cycle.
Private	0	0	4	Domain LPCG response time. This value need be set when CPU domain low power controlled LPCG is controlling a clock slower than 16MHz. The value need be no less than the multiple number of the slowest clock to 16M cycle.

NOTE

A larger response time will lead to Low power Mode and Setpoint transition penalty time.

15.5.8 Reset

The reset of CCM will be released along with system early reset and clocks are provided during system reset. For each power domain, early reset of the power domain in which that clock group and clock gate reside, will be fed for correct power up and reset flow.

15.6 External Signals

The table below shows the external signals associated with CCM block.

Table 15-6. External Signals

External Signal	Description
CCM_CLKO1	Clock output 1 for off-chip devices
CCM_CLKO2	Clock output 2 for off-chip devices
CCM_ENET_REF_CLK_25M	ENET Reference Clock

15.7 Analog IP (AI) Interface

The chip has various analog IP and components that are controlled via an AI interface. The AI interface provides a bridge to access the control registers within the analog IP and their respective components. The following table shows the AI interface registers relative to their respective analog components and power domains.

Table 15-7. AI registers

Domain	Analog Component	AI Registers
SOC2PLL	SYS_PLL1 (1G)	VDDSOC2PLL_AI_CTRL_1G
	AUDIO_PLL	VDDSOC2PLL_AI_WDATA_1G
	VIDEO_PLL	VDDSOC2PLL_AI_RDATA_1G
		VDDSOC2PLL_AI_CTRL_AUDIO
		VDDSOC2PLL_AI_WDATA_AUDIO
		VDDSOC2PLL_AI_RDATA_AUDIO
		VDDSOC2PLL_AI_CTRL_VIDEO
		VDDSOC2PLL_AI_WDATA_VIDEO
		VDDSOC2PLL_AI_RDATA_VIDEO
SOC	PHY LDO	VDDSOC_AI_CTRL
		VDDSOC_AI_WDATA
		VDDSOC_AI_RDATA
		PMU_LDO_PLL
LPSR	Bandgap	VDDLPSR_AI_CTRL
	RCOSC_400M	VDDLPSR_AI_WDATA
	Tempsensor	VDDLPSR_AI_RDATA_REFTOP
		PMU_REF_CTRL
		VDDLPSR_AI400M_CTRL

Table continues on the next page...

Table 15-7. AI registers (continued)

Domain	Analog Component	AI Registers
		VDDLPSR_AI400M_WDATA
		VDDLPSR_AI400M_RDATA
		VDDLPSR_AI_RDATA_TMPSNS

The following recommendations outline how to perform register accesses through the AI interface.

Steps to perform a write operation of an analog component's AI interface:

1. Write *AI_CTRL register
   - RWB value is 0
   - ADDRESS value is the analog component's relative AI control register to be accessed.
2. Write *AI_WDATA
   - Data[31:0] value is the analog component's relative AI control register value to be written
3. Write *CTRL IP Toggle
   - Toggle the TOGGLE bit in the bit position for the respective analog component. The toggle will initiate the write sequence.

Steps to perform a read operation of an analog component's AI interface:

1. Write *AI_CTRL register
   - RWB value is 1
   - ADDRESS value is the analog component's relative AI control register to be accessed.
2. Write *CTRL IP Toggle
   - Toggle the TOGGLE bit in the bit position for the respective analog component. The toggle will initiate the read sequence.
3. Read *CTRL
   - Poll this register until the respective analog component's busy bit is low, indicating the read transaction has completed.
4. Read *AI_RDATA
   - Read Data[31:0]. The contents of the Read Data register is the value of the analog component's respective AI register's contents.

NOTE

- For PLL operations, follow the PLL enable sequence first.
- Varying IP will have a corresponding identical toggle bit

Example: To open the bandgap, the following steps have to be made:

1. Perform the write on CTRL0
    - Set REFTOP_PWD value to 0
2. Wait 100us
3. Perform the write on CTRL0
    - Set REFTOP_SELFBIASOFF value to 1

15.8 Programming Guidelines

The Clock Controller Module contains the following Fractional and Integer PLLs:

- Fractional PLLs:
    - SYS_PLL1
    - AUDIO PLL
    - VIDEO PLL
- Integer PLLs:
    - SYS_PLL2
    - SYS_PLL3
    - ARM PLL

The PLLs can work in either GPC mode or Software mode (default). In GPC mode, GPC is the master and drives all the slaves and the users can configure the Setpoint registers. In Software mode, the users can configure all the registers.

The equations given below help to calculate the output frequency for PLLs based on the mode (Software or GPC) and type of PLL (Fractional or Integer).

- During Software mode (default):
    - For Fractional PLLs (SYS_PLL1, AUDIO_PLL, VIDEO_PLL):
        - Output frequency = Fref * (CTRL0[DIV_SELECT] + (NUMERATOR[NUM]/DENOMINATOR[DENOM]))
        - **NOTE**

          The range limitation for NUMERATOR[NUM]/ DENOMINATOR[DENOM] restricts to -1 to +1.

    - For Integer PLLs (SYS_PLL2, SYS_PLL3, ARM_PLL):
        - Output frequency = Fref * MFI
- During GPC mode:
    - For Fractional PLLs (SYS_PLL1, AUDIO_PLL, VIDEO_PLL):
        - Output frequency = Fref * (SYS_PLLx_DIV_SELECT[DIV_SELECT] + (SYS_PLLx_NUMERATOR[NUM]/ SYS_PLLx_DENOMINATOR[DENOM]))
    - For Integer PLLs (SYS_PLL2, SYS_PLL3, ARM_PLL):
        - Output frequency = Fref * MFI

Fref is the PLL Reference Frequency sourced from 24MHz.

NOTE

For SYS_PLL2 and SYS_PLL3, MFI is a fixed value of 22 and 20 respectively.

For ARM_PLL, MFI is equivalent to (ARM_PLL_CTRL[DIV_SELECT]/2)/ ARM_PLL_CTRL[POST_DIV_SELECT].

15.8.1 Spread Spectrum Parameters

The SYS_PLL1, SYS_PLL2, AUDIO_PLL, VIDEO_PLL and the related PFDs support spread spectrum (down spread). The equations given below help to calculate spread spectrum range, and modulation frequency based on the mode (Software or GPC) and type of PLL (Fractional or Integer).

- During Software mode (default):
  - For Fractional PLLs (SYS_PLL1, AUDIO_PLL, VIDEO_PLL):
    - Spread spectrum range = Fref * SPREAD_SPECTRUM[STOP]/ DENOMINATOR[DENOM]
    - Modulation frequency = Fref * SPREAD_SPECTRUM[STEP]/ (2*SPREAD_SPECTRUM[STOP])
  - For Integer PLL (SYS_PLL2):
    - Spread spectrum range = Fref * SYS_PLL2_SS[STOP]/ SYS_PLL2_MFD[MFD]
    - Modulation frequency= Fref * SYS_PLL2_SS[STEP]/ (SYS_PLL2_SS[STOP] *2)
- During GPC mode:
  - For Fractional PLLs (SYS_PLL1, AUDIO_PLL, VIDEO_PLL):
    - Spread spectrum range = Fref * SYS_PLLx_SS[STOP]/ SYS_PLLx_DENOMINATOR[DENOM]
    - Modulation frequency = Fref * SYS_PLLx_SS[STEP]/ (2*SYS_PLLx_SS[STOP])
  - For Integer PLL (SYS_PLL2):
    - Spread spectrum range = Fref * SYS_PLL2_SS[STOP]/ SYS_PLL2_MFD[MFD]
    - Modulation frequency= Fref * SYS_PLL2_SS[STEP]/ (SYS_PLL2_SS[STOP] *2)

Fref is the PLL Reference Frequency sourced from 24MHz.

The spread range indicates how much frequency the PLL will sweep down. For example, a spread range of 6MHz would mean that the PLL will sweep from 'Target_frequency' to 'Target_Frequency - 6MHz' and sweep back.

15.8.2 PLL Enable Sequence

The following sequence has to be followed to enable ARM_PLL, SYS_PLL2 and SYS_PLL3. The PLLs should be powered down and output should be disabled before beginning the sequence.

1. Enable PHY_LDO by setting Analog Control Register CTRL0 [LINREG_EN]. It is recommended to set LinReg output voltage = 1V [LINREG_OUTPUT_TRG] and enable LinReg current limit [LINREG_ILIMIT_EN]. This must be done via ANALOG IP interface. Please read chapter ANALOG IP Interface in RM.
2. Wait at least 100 us.
3. Enable the reference voltage by setting PMU_REF_CTRL[EN_PLL_VOL_REF_BUFFER].
4. Configure the PLL registers such as xx_DIV_SELECT/xx_MFN/xx_MFD/xx_SS described in the PLL Register Descriptions section.
5. Enable the internal LDO (ARM_PLL does not have internal LDO and this step is skipped.
6. Wait at least 30 us for LDO stability (ARM_PLL does not have internal LDO and this step is skipped).
7. Power up the PLL by enabling xx_PLL_CTRL[POWERUP] and assertxx_PLL_CTRL[HOLD_RING_OFF].
8. When the PLL lock time is halfway through, de-assert the xx_PLL_CTRL[HOLD_RING_OFF].
9. Wait till the PLL lock time is complete. This can be monitored through the xx_PLL_CTRL[xx_PLL_STABLE] bit.
10. Enable clock output through xx_PLL_CTRL[ENABLE_CLK].

The following sequence has to be followed for SYS_PLL1, AUDIO_PLL and VIDEO_PLL:

1. Enable PHY_LDO by setting Analog Control Register CTRL0 [LINREG_EN]. It is recommended to set LinReg output voltage = 1V [LINREG_OUTPUT_TRG] and enable LinReg current limit [LINREG_ILIMIT_EN]. This must be done via ANALOG IP interface. Please read chapter ANALOG IP Interface in RM.
2. Wait at least 100 µs.
3. Enable the reference voltage by setting PMU_REF_CTRL[EN_PLL_VOL_REF_BUFFER].

4. Bypass PLL by asserting BYPASS bit in Fractional PLL Control Register (CTRL0). This must be done via ANALOG IP interface. Please read chapter ANALOG IP Interface in RM There are separate registers for access SYS_PLL1 (VDDSOC2PLL_AI_CTRL_1G), AUDIO_PLL (VDDSOC2PLL_AI_CTRL_AUDIO) and VIDEO_PLL(VDDSOC2PLL_AI_CTRL_VIDEO).
5. Enable PLL clock output by setting SYS_PLL1_CTRL[ENABLE_CLK], PLL_AUDIO_CTRL[ENABLE_CLK] or PLL_VIDEO_CTRL[ENABLE_CLK].
6. Configure the PLL registers Fractional PLL Control Register (CTRL0) [POST_DIV_SEL], Fractional PLL Control Register (CTRL0)[DIV_SELECT], Fractional PLL Numerator Control Register (NUMERATOR)[NUM], Fractional PLL Denominator Control Register(DENOMINATOR)[DENOM], Fractional PLL Spread Spectrum Control Register(SPREAD_SPECTRUM)[STOP/STEP] [ENABLE]. This must be done via ANALOG IP interface. Please read chapter ANALOG IP Interface in RM. SYS_PLL1 is fixed PLL, so Fractional PLL Spread Spectrum Control Register(SPREAD_SPECTRUM) can be only configured.
7. Enable the internal LDO by setting Fractional PLL Control Register (CTRL0) [PLL_REG_EN] via ANALOG IP Interface.
8. When the PLL lock time is halfway through, de-assert the Fractional PLL Control Register (CTRL0) [HOLD_RING_OFF] via ANALOG IP interface.
9. Wait till the PLL lock time is complete. This can be monitored through the xx_PLL_CTRL[xx_PLL_STABLE] bit.
10. Disable PLL gate by deasserting SYS_PLL1_CTRL[SYS_PLL1_GATE], PLL_AUDIO_CTRL[PLL_AUDIO_GATE] or PLL_VIDEO_CTRL[PLL_VIDEO_GATE].
11. Disable PLL bypass by deasserting Fractional PLL Control Register (CTRL0) [BYPASS] via ANALOG IP Interface.

NOTE

- In order to avoid unstable clocks that might alter the PFD's state machine, all PFDs are required to be clock-gated when the PLL is not locked.
- PFD's input can be changed on-the fly, while PLL's input cannot be changed on-the-fly (except for MFN).
- The initialization for software and GPC modes is handled by Setpoints automatically.

15.9 Memory Map and register definition

This section includes the CCM module memory map and detailed descriptions of all registers.

NOTE

For related IPS Domain Slot control, please see IPS Domain Registers in Power Management Unit (PMU) chapter.

15.9.1 CCM register descriptions

15.9.1.1 CCM memory map

CCM base address: 40CC_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Clock root control (CLOCK_ROOT0_CONTROL)	32	RW	0000_0000h
4h	Clock root control (CLOCK_ROOT0_CONTROL_SET)	32	RW	0000_0000h
8h	Clock root control (CLOCK_ROOT0_CONTROL_CLR)	32	RW	0000_0000h
Ch	Clock root control (CLOCK_ROOT0_CONTROL_TOG)	32	RW	0000_0000h
20h - 2720h	Clock root working status (CLOCK_ROOT0_STATUS0 - CLOCK_ROOT78_STATUS0)	32	R	0000_0000h
24h - 2724h	Clock root low power status (CLOCK_ROOT0_STATUS1 - CLOCK_ROOT78_STATUS1)	32	R	0000_0000h
2Ch - 272Ch	Clock root configuration (CLOCK_ROOT0_CONFIG - CLOCK_ROOT78_CONFIG)	32	R	Table 15-7
30h	Clock root access control (CLOCK_ROOT0_AUTHEN)	32	RW	0000_0000h
34h	Clock root access control (CLOCK_ROOT0_AUTHEN_SET)	32	RW	0000_0000h
38h	Clock root access control (CLOCK_ROOT0_AUTHEN_CLR)	32	RW	0000_0000h
3Ch	Clock root access control (CLOCK_ROOT0_AUTHEN_TOG)	32	RW	0000_0000h
40h - 277Ch	Setpoint setting (CLOCK_ROOT0_SETPOINT0 - CLOCK_ROOT78_SETPOINT15)	32	RW	0000_0000h
80h	Clock root control (CLOCK_ROOT1_CONTROL)	32	RW	0000_0000h
84h	Clock root control (CLOCK_ROOT1_CONTROL_SET)	32	RW	0000_0000h
88h	Clock root control (CLOCK_ROOT1_CONTROL_CLR)	32	RW	0000_0000h
8Ch	Clock root control (CLOCK_ROOT1_CONTROL_TOG)	32	RW	0000_0000h
B0h	Clock root access control (CLOCK_ROOT1_AUTHEN)	32	RW	0000_0000h
B4h	Clock root access control (CLOCK_ROOT1_AUTHEN_SET)	32	RW	0000_0000h
B8h	Clock root access control (CLOCK_ROOT1_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
BCh	Clock root access control (CLOCK_ROOT1_AUTHEN_TOG)	32	RW	0000_0000h
100h	Clock root control (CLOCK_ROOT2_CONTROL)	32	RW	0000_0000h
104h	Clock root control (CLOCK_ROOT2_CONTROL_SET)	32	RW	0000_0000h
108h	Clock root control (CLOCK_ROOT2_CONTROL_CLR)	32	RW	0000_0000h
10Ch	Clock root control (CLOCK_ROOT2_CONTROL_TOG)	32	RW	0000_0000h
130h	Clock root access control (CLOCK_ROOT2_AUTHEN)	32	RW	0000_0000h
134h	Clock root access control (CLOCK_ROOT2_AUTHEN_SET)	32	RW	0000_0000h
138h	Clock root access control (CLOCK_ROOT2_AUTHEN_CLR)	32	RW	0000_0000h
13Ch	Clock root access control (CLOCK_ROOT2_AUTHEN_TOG)	32	RW	0000_0000h
180h	Clock root control (CLOCK_ROOT3_CONTROL)	32	RW	0000_0000h
184h	Clock root control (CLOCK_ROOT3_CONTROL_SET)	32	RW	0000_0000h
188h	Clock root control (CLOCK_ROOT3_CONTROL_CLR)	32	RW	0000_0000h
18Ch	Clock root control (CLOCK_ROOT3_CONTROL_TOG)	32	RW	0000_0000h
1B0h	Clock root access control (CLOCK_ROOT3_AUTHEN)	32	RW	0000_0000h
1B4h	Clock root access control (CLOCK_ROOT3_AUTHEN_SET)	32	RW	0000_0000h
1B8h	Clock root access control (CLOCK_ROOT3_AUTHEN_CLR)	32	RW	0000_0000h
1BCh	Clock root access control (CLOCK_ROOT3_AUTHEN_TOG)	32	RW	0000_0000h
200h	Clock root control (CLOCK_ROOT4_CONTROL)	32	RW	0000_0000h
204h	Clock root control (CLOCK_ROOT4_CONTROL_SET)	32	RW	0000_0000h
208h	Clock root control (CLOCK_ROOT4_CONTROL_CLR)	32	RW	0000_0000h
20Ch	Clock root control (CLOCK_ROOT4_CONTROL_TOG)	32	RW	0000_0000h
230h	Clock root access control (CLOCK_ROOT4_AUTHEN)	32	RW	0000_0000h
234h	Clock root access control (CLOCK_ROOT4_AUTHEN_SET)	32	RW	0000_0000h
238h	Clock root access control (CLOCK_ROOT4_AUTHEN_CLR)	32	RW	0000_0000h
23Ch	Clock root access control (CLOCK_ROOT4_AUTHEN_TOG)	32	RW	0000_0000h
280h	Clock root control (CLOCK_ROOT5_CONTROL)	32	RW	0000_0000h
284h	Clock root control (CLOCK_ROOT5_CONTROL_SET)	32	RW	0000_0000h
288h	Clock root control (CLOCK_ROOT5_CONTROL_CLR)	32	RW	0000_0000h
28Ch	Clock root control (CLOCK_ROOT5_CONTROL_TOG)	32	RW	0000_0000h
2B0h	Clock root access control (CLOCK_ROOT5_AUTHEN)	32	RW	0000_0000h
2B4h	Clock root access control (CLOCK_ROOT5_AUTHEN_SET)	32	RW	0000_0000h
2B8h	Clock root access control (CLOCK_ROOT5_AUTHEN_CLR)	32	RW	0000_0000h
2BCh	Clock root access control (CLOCK_ROOT5_AUTHEN_TOG)	32	RW	0000_0000h
300h	Clock root control (CLOCK_ROOT6_CONTROL)	32	RW	0000_0000h
304h	Clock root control (CLOCK_ROOT6_CONTROL_SET)	32	RW	0000_0000h
308h	Clock root control (CLOCK_ROOT6_CONTROL_CLR)	32	RW	0000_0000h
30Ch	Clock root control (CLOCK_ROOT6_CONTROL_TOG)	32	RW	0000_0000h
330h	Clock root access control (CLOCK_ROOT6_AUTHEN)	32	RW	0000_0000h
334h	Clock root access control (CLOCK_ROOT6_AUTHEN_SET)	32	RW	0000_0000h
338h	Clock root access control (CLOCK_ROOT6_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
33Ch	Clock root access control (CLOCK_ROOT6_AUTHEN_TOG)	32	RW	0000_0000h
380h	Clock root control (CLOCK_ROOT7_CONTROL)	32	RW	0000_0000h
384h	Clock root control (CLOCK_ROOT7_CONTROL_SET)	32	RW	0000_0000h
388h	Clock root control (CLOCK_ROOT7_CONTROL_CLR)	32	RW	0000_0000h
38Ch	Clock root control (CLOCK_ROOT7_CONTROL_TOG)	32	RW	0000_0000h
3B0h	Clock root access control (CLOCK_ROOT7_AUTHEN)	32	RW	0000_0000h
3B4h	Clock root access control (CLOCK_ROOT7_AUTHEN_SET)	32	RW	0000_0000h
3B8h	Clock root access control (CLOCK_ROOT7_AUTHEN_CLR)	32	RW	0000_0000h
3BCh	Clock root access control (CLOCK_ROOT7_AUTHEN_TOG)	32	RW	0000_0000h
400h	Clock root control (CLOCK_ROOT8_CONTROL)	32	RW	0000_0000h
404h	Clock root control (CLOCK_ROOT8_CONTROL_SET)	32	RW	0000_0000h
408h	Clock root control (CLOCK_ROOT8_CONTROL_CLR)	32	RW	0000_0000h
40Ch	Clock root control (CLOCK_ROOT8_CONTROL_TOG)	32	RW	0000_0000h
430h	Clock root access control (CLOCK_ROOT8_AUTHEN)	32	RW	0000_0000h
434h	Clock root access control (CLOCK_ROOT8_AUTHEN_SET)	32	RW	0000_0000h
438h	Clock root access control (CLOCK_ROOT8_AUTHEN_CLR)	32	RW	0000_0000h
43Ch	Clock root access control (CLOCK_ROOT8_AUTHEN_TOG)	32	RW	0000_0000h
480h	Clock root control (CLOCK_ROOT9_CONTROL)	32	RW	0000_0000h
484h	Clock root control (CLOCK_ROOT9_CONTROL_SET)	32	RW	0000_0000h
488h	Clock root control (CLOCK_ROOT9_CONTROL_CLR)	32	RW	0000_0000h
48Ch	Clock root control (CLOCK_ROOT9_CONTROL_TOG)	32	RW	0000_0000h
4B0h	Clock root access control (CLOCK_ROOT9_AUTHEN)	32	RW	0000_0000h
4B4h	Clock root access control (CLOCK_ROOT9_AUTHEN_SET)	32	RW	0000_0000h
4B8h	Clock root access control (CLOCK_ROOT9_AUTHEN_CLR)	32	RW	0000_0000h
4BCh	Clock root access control (CLOCK_ROOT9_AUTHEN_TOG)	32	RW	0000_0000h
500h	Clock root control (CLOCK_ROOT10_CONTROL)	32	RW	0000_0000h
504h	Clock root control (CLOCK_ROOT10_CONTROL_SET)	32	RW	0000_0000h
508h	Clock root control (CLOCK_ROOT10_CONTROL_CLR)	32	RW	0000_0000h
50Ch	Clock root control (CLOCK_ROOT10_CONTROL_TOG)	32	RW	0000_0000h
530h	Clock root access control (CLOCK_ROOT10_AUTHEN)	32	RW	0000_0000h
534h	Clock root access control (CLOCK_ROOT10_AUTHEN_SET)	32	RW	0000_0000h
538h	Clock root access control (CLOCK_ROOT10_AUTHEN_CLR)	32	RW	0000_0000h
53Ch	Clock root access control (CLOCK_ROOT10_AUTHEN_TOG)	32	RW	0000_0000h
580h	Clock root control (CLOCK_ROOT11_CONTROL)	32	RW	0000_0000h
584h	Clock root control (CLOCK_ROOT11_CONTROL_SET)	32	RW	0000_0000h
588h	Clock root control (CLOCK_ROOT11_CONTROL_CLR)	32	RW	0000_0000h
58Ch	Clock root control (CLOCK_ROOT11_CONTROL_TOG)	32	RW	0000_0000h
5B0h	Clock root access control (CLOCK_ROOT11_AUTHEN)	32	RW	0000_0000h
5B4h	Clock root access control (CLOCK_ROOT11_AUTHEN_SET)	32	RW	0000_0000h
5B8h	Clock root access control (CLOCK_ROOT11_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5BCh	Clock root access control (CLOCK_ROOT11_AUTHEN_TOG)	32	RW	0000_0000h
600h	Clock root control (CLOCK_ROOT12_CONTROL)	32	RW	0000_0000h
604h	Clock root control (CLOCK_ROOT12_CONTROL_SET)	32	RW	0000_0000h
608h	Clock root control (CLOCK_ROOT12_CONTROL_CLR)	32	RW	0000_0000h
60Ch	Clock root control (CLOCK_ROOT12_CONTROL_TOG)	32	RW	0000_0000h
630h	Clock root access control (CLOCK_ROOT12_AUTHEN)	32	RW	0000_0000h
634h	Clock root access control (CLOCK_ROOT12_AUTHEN_SET)	32	RW	0000_0000h
638h	Clock root access control (CLOCK_ROOT12_AUTHEN_CLR)	32	RW	0000_0000h
63Ch	Clock root access control (CLOCK_ROOT12_AUTHEN_TOG)	32	RW	0000_0000h
680h	Clock root control (CLOCK_ROOT13_CONTROL)	32	RW	0000_0000h
684h	Clock root control (CLOCK_ROOT13_CONTROL_SET)	32	RW	0000_0000h
688h	Clock root control (CLOCK_ROOT13_CONTROL_CLR)	32	RW	0000_0000h
68Ch	Clock root control (CLOCK_ROOT13_CONTROL_TOG)	32	RW	0000_0000h
6B0h	Clock root access control (CLOCK_ROOT13_AUTHEN)	32	RW	0000_0000h
6B4h	Clock root access control (CLOCK_ROOT13_AUTHEN_SET)	32	RW	0000_0000h
6B8h	Clock root access control (CLOCK_ROOT13_AUTHEN_CLR)	32	RW	0000_0000h
6BCh	Clock root access control (CLOCK_ROOT13_AUTHEN_TOG)	32	RW	0000_0000h
700h	Clock root control (CLOCK_ROOT14_CONTROL)	32	RW	0000_0000h
704h	Clock root control (CLOCK_ROOT14_CONTROL_SET)	32	RW	0000_0000h
708h	Clock root control (CLOCK_ROOT14_CONTROL_CLR)	32	RW	0000_0000h
70Ch	Clock root control (CLOCK_ROOT14_CONTROL_TOG)	32	RW	0000_0000h
730h	Clock root access control (CLOCK_ROOT14_AUTHEN)	32	RW	0000_0000h
734h	Clock root access control (CLOCK_ROOT14_AUTHEN_SET)	32	RW	0000_0000h
738h	Clock root access control (CLOCK_ROOT14_AUTHEN_CLR)	32	RW	0000_0000h
73Ch	Clock root access control (CLOCK_ROOT14_AUTHEN_TOG)	32	RW	0000_0000h
780h	Clock root control (CLOCK_ROOT15_CONTROL)	32	RW	0000_0000h
784h	Clock root control (CLOCK_ROOT15_CONTROL_SET)	32	RW	0000_0000h
788h	Clock root control (CLOCK_ROOT15_CONTROL_CLR)	32	RW	0000_0000h
78Ch	Clock root control (CLOCK_ROOT15_CONTROL_TOG)	32	RW	0000_0000h
7B0h	Clock root access control (CLOCK_ROOT15_AUTHEN)	32	RW	0000_0000h
7B4h	Clock root access control (CLOCK_ROOT15_AUTHEN_SET)	32	RW	0000_0000h
7B8h	Clock root access control (CLOCK_ROOT15_AUTHEN_CLR)	32	RW	0000_0000h
7BCh	Clock root access control (CLOCK_ROOT15_AUTHEN_TOG)	32	RW	0000_0000h
800h	Clock root control (CLOCK_ROOT16_CONTROL)	32	RW	0000_0000h
804h	Clock root control (CLOCK_ROOT16_CONTROL_SET)	32	RW	0000_0000h
808h	Clock root control (CLOCK_ROOT16_CONTROL_CLR)	32	RW	0000_0000h
80Ch	Clock root control (CLOCK_ROOT16_CONTROL_TOG)	32	RW	0000_0000h
830h	Clock root access control (CLOCK_ROOT16_AUTHEN)	32	RW	0000_0000h
834h	Clock root access control (CLOCK_ROOT16_AUTHEN_SET)	32	RW	0000_0000h
838h	Clock root access control (CLOCK_ROOT16_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
83Ch	Clock root access control (CLOCK_ROOT16_AUTHEN_TOG)	32	RW	0000_0000h
880h	Clock root control (CLOCK_ROOT17_CONTROL)	32	RW	0000_0000h
884h	Clock root control (CLOCK_ROOT17_CONTROL_SET)	32	RW	0000_0000h
888h	Clock root control (CLOCK_ROOT17_CONTROL_CLR)	32	RW	0000_0000h
88Ch	Clock root control (CLOCK_ROOT17_CONTROL_TOG)	32	RW	0000_0000h
8B0h	Clock root access control (CLOCK_ROOT17_AUTHEN)	32	RW	0000_0000h
8B4h	Clock root access control (CLOCK_ROOT17_AUTHEN_SET)	32	RW	0000_0000h
8B8h	Clock root access control (CLOCK_ROOT17_AUTHEN_CLR)	32	RW	0000_0000h
8BCh	Clock root access control (CLOCK_ROOT17_AUTHEN_TOG)	32	RW	0000_0000h
900h	Clock root control (CLOCK_ROOT18_CONTROL)	32	RW	0000_0000h
904h	Clock root control (CLOCK_ROOT18_CONTROL_SET)	32	RW	0000_0000h
908h	Clock root control (CLOCK_ROOT18_CONTROL_CLR)	32	RW	0000_0000h
90Ch	Clock root control (CLOCK_ROOT18_CONTROL_TOG)	32	RW	0000_0000h
930h	Clock root access control (CLOCK_ROOT18_AUTHEN)	32	RW	0000_0000h
934h	Clock root access control (CLOCK_ROOT18_AUTHEN_SET)	32	RW	0000_0000h
938h	Clock root access control (CLOCK_ROOT18_AUTHEN_CLR)	32	RW	0000_0000h
93Ch	Clock root access control (CLOCK_ROOT18_AUTHEN_TOG)	32	RW	0000_0000h
980h	Clock root control (CLOCK_ROOT19_CONTROL)	32	RW	0000_0000h
984h	Clock root control (CLOCK_ROOT19_CONTROL_SET)	32	RW	0000_0000h
988h	Clock root control (CLOCK_ROOT19_CONTROL_CLR)	32	RW	0000_0000h
98Ch	Clock root control (CLOCK_ROOT19_CONTROL_TOG)	32	RW	0000_0000h
9B0h	Clock root access control (CLOCK_ROOT19_AUTHEN)	32	RW	0000_0000h
9B4h	Clock root access control (CLOCK_ROOT19_AUTHEN_SET)	32	RW	0000_0000h
9B8h	Clock root access control (CLOCK_ROOT19_AUTHEN_CLR)	32	RW	0000_0000h
9BCh	Clock root access control (CLOCK_ROOT19_AUTHEN_TOG)	32	RW	0000_0000h
A00h	Clock root control (CLOCK_ROOT20_CONTROL)	32	RW	0000_0000h
A04h	Clock root control (CLOCK_ROOT20_CONTROL_SET)	32	RW	0000_0000h
A08h	Clock root control (CLOCK_ROOT20_CONTROL_CLR)	32	RW	0000_0000h
A0Ch	Clock root control (CLOCK_ROOT20_CONTROL_TOG)	32	RW	0000_0000h
A30h	Clock root access control (CLOCK_ROOT20_AUTHEN)	32	RW	0000_0000h
A34h	Clock root access control (CLOCK_ROOT20_AUTHEN_SET)	32	RW	0000_0000h
A38h	Clock root access control (CLOCK_ROOT20_AUTHEN_CLR)	32	RW	0000_0000h
A3Ch	Clock root access control (CLOCK_ROOT20_AUTHEN_TOG)	32	RW	0000_0000h
A80h	Clock root control (CLOCK_ROOT21_CONTROL)	32	RW	0000_0000h
A84h	Clock root control (CLOCK_ROOT21_CONTROL_SET)	32	RW	0000_0000h
A88h	Clock root control (CLOCK_ROOT21_CONTROL_CLR)	32	RW	0000_0000h
A8Ch	Clock root control (CLOCK_ROOT21_CONTROL_TOG)	32	RW	0000_0000h
AB0h	Clock root access control (CLOCK_ROOT21_AUTHEN)	32	RW	0000_0000h
AB4h	Clock root access control (CLOCK_ROOT21_AUTHEN_SET)	32	RW	0000_0000h
AB8h	Clock root access control (CLOCK_ROOT21_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
ABCh	Clock root access control (CLOCK_ROOT21_AUTHEN_TOG)	32	RW	0000_0000h
B00h	Clock root control (CLOCK_ROOT22_CONTROL)	32	RW	0000_0000h
B04h	Clock root control (CLOCK_ROOT22_CONTROL_SET)	32	RW	0000_0000h
B08h	Clock root control (CLOCK_ROOT22_CONTROL_CLR)	32	RW	0000_0000h
B0Ch	Clock root control (CLOCK_ROOT22_CONTROL_TOG)	32	RW	0000_0000h
B30h	Clock root access control (CLOCK_ROOT22_AUTHEN)	32	RW	0000_0000h
B34h	Clock root access control (CLOCK_ROOT22_AUTHEN_SET)	32	RW	0000_0000h
B38h	Clock root access control (CLOCK_ROOT22_AUTHEN_CLR)	32	RW	0000_0000h
B3Ch	Clock root access control (CLOCK_ROOT22_AUTHEN_TOG)	32	RW	0000_0000h
B80h	Clock root control (CLOCK_ROOT23_CONTROL)	32	RW	0000_0000h
B84h	Clock root control (CLOCK_ROOT23_CONTROL_SET)	32	RW	0000_0000h
B88h	Clock root control (CLOCK_ROOT23_CONTROL_CLR)	32	RW	0000_0000h
B8Ch	Clock root control (CLOCK_ROOT23_CONTROL_TOG)	32	RW	0000_0000h
BB0h	Clock root access control (CLOCK_ROOT23_AUTHEN)	32	RW	0000_0000h
BB4h	Clock root access control (CLOCK_ROOT23_AUTHEN_SET)	32	RW	0000_0000h
BB8h	Clock root access control (CLOCK_ROOT23_AUTHEN_CLR)	32	RW	0000_0000h
BBCh	Clock root access control (CLOCK_ROOT23_AUTHEN_TOG)	32	RW	0000_0000h
C00h	Clock root control (CLOCK_ROOT24_CONTROL)	32	RW	0000_0000h
C04h	Clock root control (CLOCK_ROOT24_CONTROL_SET)	32	RW	0000_0000h
C08h	Clock root control (CLOCK_ROOT24_CONTROL_CLR)	32	RW	0000_0000h
C0Ch	Clock root control (CLOCK_ROOT24_CONTROL_TOG)	32	RW	0000_0000h
C30h	Clock root access control (CLOCK_ROOT24_AUTHEN)	32	RW	0000_0000h
C34h	Clock root access control (CLOCK_ROOT24_AUTHEN_SET)	32	RW	0000_0000h
C38h	Clock root access control (CLOCK_ROOT24_AUTHEN_CLR)	32	RW	0000_0000h
C3Ch	Clock root access control (CLOCK_ROOT24_AUTHEN_TOG)	32	RW	0000_0000h
C80h	Clock root control (CLOCK_ROOT25_CONTROL)	32	RW	0000_0000h
C84h	Clock root control (CLOCK_ROOT25_CONTROL_SET)	32	RW	0000_0000h
C88h	Clock root control (CLOCK_ROOT25_CONTROL_CLR)	32	RW	0000_0000h
C8Ch	Clock root control (CLOCK_ROOT25_CONTROL_TOG)	32	RW	0000_0000h
CB0h	Clock root access control (CLOCK_ROOT25_AUTHEN)	32	RW	0000_0000h
CB4h	Clock root access control (CLOCK_ROOT25_AUTHEN_SET)	32	RW	0000_0000h
CB8h	Clock root access control (CLOCK_ROOT25_AUTHEN_CLR)	32	RW	0000_0000h
CBCh	Clock root access control (CLOCK_ROOT25_AUTHEN_TOG)	32	RW	0000_0000h
D00h	Clock root control (CLOCK_ROOT26_CONTROL)	32	RW	0000_0000h
D04h	Clock root control (CLOCK_ROOT26_CONTROL_SET)	32	RW	0000_0000h
D08h	Clock root control (CLOCK_ROOT26_CONTROL_CLR)	32	RW	0000_0000h
D0Ch	Clock root control (CLOCK_ROOT26_CONTROL_TOG)	32	RW	0000_0000h
D30h	Clock root access control (CLOCK_ROOT26_AUTHEN)	32	RW	0000_0000h
D34h	Clock root access control (CLOCK_ROOT26_AUTHEN_SET)	32	RW	0000_0000h
D38h	Clock root access control (CLOCK_ROOT26_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
D3Ch	Clock root access control (CLOCK_ROOT26_AUTHEN_TOG)	32	RW	0000_0000h
D80h	Clock root control (CLOCK_ROOT27_CONTROL)	32	RW	0000_0000h
D84h	Clock root control (CLOCK_ROOT27_CONTROL_SET)	32	RW	0000_0000h
D88h	Clock root control (CLOCK_ROOT27_CONTROL_CLR)	32	RW	0000_0000h
D8Ch	Clock root control (CLOCK_ROOT27_CONTROL_TOG)	32	RW	0000_0000h
DB0h	Clock root access control (CLOCK_ROOT27_AUTHEN)	32	RW	0000_0000h
DB4h	Clock root access control (CLOCK_ROOT27_AUTHEN_SET)	32	RW	0000_0000h
DB8h	Clock root access control (CLOCK_ROOT27_AUTHEN_CLR)	32	RW	0000_0000h
DBCh	Clock root access control (CLOCK_ROOT27_AUTHEN_TOG)	32	RW	0000_0000h
E00h	Clock root control (CLOCK_ROOT28_CONTROL)	32	RW	0000_0000h
E04h	Clock root control (CLOCK_ROOT28_CONTROL_SET)	32	RW	0000_0000h
E08h	Clock root control (CLOCK_ROOT28_CONTROL_CLR)	32	RW	0000_0000h
E0Ch	Clock root control (CLOCK_ROOT28_CONTROL_TOG)	32	RW	0000_0000h
E30h	Clock root access control (CLOCK_ROOT28_AUTHEN)	32	RW	0000_0000h
E34h	Clock root access control (CLOCK_ROOT28_AUTHEN_SET)	32	RW	0000_0000h
E38h	Clock root access control (CLOCK_ROOT28_AUTHEN_CLR)	32	RW	0000_0000h
E3Ch	Clock root access control (CLOCK_ROOT28_AUTHEN_TOG)	32	RW	0000_0000h
E80h	Clock root control (CLOCK_ROOT29_CONTROL)	32	RW	0000_0000h
E84h	Clock root control (CLOCK_ROOT29_CONTROL_SET)	32	RW	0000_0000h
E88h	Clock root control (CLOCK_ROOT29_CONTROL_CLR)	32	RW	0000_0000h
E8Ch	Clock root control (CLOCK_ROOT29_CONTROL_TOG)	32	RW	0000_0000h
EB0h	Clock root access control (CLOCK_ROOT29_AUTHEN)	32	RW	0000_0000h
EB4h	Clock root access control (CLOCK_ROOT29_AUTHEN_SET)	32	RW	0000_0000h
EB8h	Clock root access control (CLOCK_ROOT29_AUTHEN_CLR)	32	RW	0000_0000h
EBCh	Clock root access control (CLOCK_ROOT29_AUTHEN_TOG)	32	RW	0000_0000h
F00h	Clock root control (CLOCK_ROOT30_CONTROL)	32	RW	0000_0000h
F04h	Clock root control (CLOCK_ROOT30_CONTROL_SET)	32	RW	0000_0000h
F08h	Clock root control (CLOCK_ROOT30_CONTROL_CLR)	32	RW	0000_0000h
F0Ch	Clock root control (CLOCK_ROOT30_CONTROL_TOG)	32	RW	0000_0000h
F30h	Clock root access control (CLOCK_ROOT30_AUTHEN)	32	RW	0000_0000h
F34h	Clock root access control (CLOCK_ROOT30_AUTHEN_SET)	32	RW	0000_0000h
F38h	Clock root access control (CLOCK_ROOT30_AUTHEN_CLR)	32	RW	0000_0000h
F3Ch	Clock root access control (CLOCK_ROOT30_AUTHEN_TOG)	32	RW	0000_0000h
F80h	Clock root control (CLOCK_ROOT31_CONTROL)	32	RW	0000_0000h
F84h	Clock root control (CLOCK_ROOT31_CONTROL_SET)	32	RW	0000_0000h
F88h	Clock root control (CLOCK_ROOT31_CONTROL_CLR)	32	RW	0000_0000h
F8Ch	Clock root control (CLOCK_ROOT31_CONTROL_TOG)	32	RW	0000_0000h
FB0h	Clock root access control (CLOCK_ROOT31_AUTHEN)	32	RW	0000_0000h
FB4h	Clock root access control (CLOCK_ROOT31_AUTHEN_SET)	32	RW	0000_0000h
FB8h	Clock root access control (CLOCK_ROOT31_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
FBCh	Clock root access control (CLOCK_ROOT31_AUTHEN_TOG)	32	RW	0000_0000h
1000h	Clock root control (CLOCK_ROOT32_CONTROL)	32	RW	0000_0000h
1004h	Clock root control (CLOCK_ROOT32_CONTROL_SET)	32	RW	0000_0000h
1008h	Clock root control (CLOCK_ROOT32_CONTROL_CLR)	32	RW	0000_0000h
100Ch	Clock root control (CLOCK_ROOT32_CONTROL_TOG)	32	RW	0000_0000h
1030h	Clock root access control (CLOCK_ROOT32_AUTHEN)	32	RW	0000_0000h
1034h	Clock root access control (CLOCK_ROOT32_AUTHEN_SET)	32	RW	0000_0000h
1038h	Clock root access control (CLOCK_ROOT32_AUTHEN_CLR)	32	RW	0000_0000h
103Ch	Clock root access control (CLOCK_ROOT32_AUTHEN_TOG)	32	RW	0000_0000h
1080h	Clock root control (CLOCK_ROOT33_CONTROL)	32	RW	0000_0000h
1084h	Clock root control (CLOCK_ROOT33_CONTROL_SET)	32	RW	0000_0000h
1088h	Clock root control (CLOCK_ROOT33_CONTROL_CLR)	32	RW	0000_0000h
108Ch	Clock root control (CLOCK_ROOT33_CONTROL_TOG)	32	RW	0000_0000h
10B0h	Clock root access control (CLOCK_ROOT33_AUTHEN)	32	RW	0000_0000h
10B4h	Clock root access control (CLOCK_ROOT33_AUTHEN_SET)	32	RW	0000_0000h
10B8h	Clock root access control (CLOCK_ROOT33_AUTHEN_CLR)	32	RW	0000_0000h
10BCh	Clock root access control (CLOCK_ROOT33_AUTHEN_TOG)	32	RW	0000_0000h
1100h	Clock root control (CLOCK_ROOT34_CONTROL)	32	RW	0000_0000h
1104h	Clock root control (CLOCK_ROOT34_CONTROL_SET)	32	RW	0000_0000h
1108h	Clock root control (CLOCK_ROOT34_CONTROL_CLR)	32	RW	0000_0000h
110Ch	Clock root control (CLOCK_ROOT34_CONTROL_TOG)	32	RW	0000_0000h
1130h	Clock root access control (CLOCK_ROOT34_AUTHEN)	32	RW	0000_0000h
1134h	Clock root access control (CLOCK_ROOT34_AUTHEN_SET)	32	RW	0000_0000h
1138h	Clock root access control (CLOCK_ROOT34_AUTHEN_CLR)	32	RW	0000_0000h
113Ch	Clock root access control (CLOCK_ROOT34_AUTHEN_TOG)	32	RW	0000_0000h
1180h	Clock root control (CLOCK_ROOT35_CONTROL)	32	RW	0000_0000h
1184h	Clock root control (CLOCK_ROOT35_CONTROL_SET)	32	RW	0000_0000h
1188h	Clock root control (CLOCK_ROOT35_CONTROL_CLR)	32	RW	0000_0000h
118Ch	Clock root control (CLOCK_ROOT35_CONTROL_TOG)	32	RW	0000_0000h
11B0h	Clock root access control (CLOCK_ROOT35_AUTHEN)	32	RW	0000_0000h
11B4h	Clock root access control (CLOCK_ROOT35_AUTHEN_SET)	32	RW	0000_0000h
11B8h	Clock root access control (CLOCK_ROOT35_AUTHEN_CLR)	32	RW	0000_0000h
11BCh	Clock root access control (CLOCK_ROOT35_AUTHEN_TOG)	32	RW	0000_0000h
1200h	Clock root control (CLOCK_ROOT36_CONTROL)	32	RW	0000_0000h
1204h	Clock root control (CLOCK_ROOT36_CONTROL_SET)	32	RW	0000_0000h
1208h	Clock root control (CLOCK_ROOT36_CONTROL_CLR)	32	RW	0000_0000h
120Ch	Clock root control (CLOCK_ROOT36_CONTROL_TOG)	32	RW	0000_0000h
1230h	Clock root access control (CLOCK_ROOT36_AUTHEN)	32	RW	0000_0000h
1234h	Clock root access control (CLOCK_ROOT36_AUTHEN_SET)	32	RW	0000_0000h
1238h	Clock root access control (CLOCK_ROOT36_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
123Ch	Clock root access control (CLOCK_ROOT36_AUTHEN_TOG)	32	RW	0000_0000h
1280h	Clock root control (CLOCK_ROOT37_CONTROL)	32	RW	0000_0000h
1284h	Clock root control (CLOCK_ROOT37_CONTROL_SET)	32	RW	0000_0000h
1288h	Clock root control (CLOCK_ROOT37_CONTROL_CLR)	32	RW	0000_0000h
128Ch	Clock root control (CLOCK_ROOT37_CONTROL_TOG)	32	RW	0000_0000h
12B0h	Clock root access control (CLOCK_ROOT37_AUTHEN)	32	RW	0000_0000h
12B4h	Clock root access control (CLOCK_ROOT37_AUTHEN_SET)	32	RW	0000_0000h
12B8h	Clock root access control (CLOCK_ROOT37_AUTHEN_CLR)	32	RW	0000_0000h
12BCh	Clock root access control (CLOCK_ROOT37_AUTHEN_TOG)	32	RW	0000_0000h
1300h	Clock root control (CLOCK_ROOT38_CONTROL)	32	RW	0000_0000h
1304h	Clock root control (CLOCK_ROOT38_CONTROL_SET)	32	RW	0000_0000h
1308h	Clock root control (CLOCK_ROOT38_CONTROL_CLR)	32	RW	0000_0000h
130Ch	Clock root control (CLOCK_ROOT38_CONTROL_TOG)	32	RW	0000_0000h
1330h	Clock root access control (CLOCK_ROOT38_AUTHEN)	32	RW	0000_0000h
1334h	Clock root access control (CLOCK_ROOT38_AUTHEN_SET)	32	RW	0000_0000h
1338h	Clock root access control (CLOCK_ROOT38_AUTHEN_CLR)	32	RW	0000_0000h
133Ch	Clock root access control (CLOCK_ROOT38_AUTHEN_TOG)	32	RW	0000_0000h
1380h	Clock root control (CLOCK_ROOT39_CONTROL)	32	RW	0000_0000h
1384h	Clock root control (CLOCK_ROOT39_CONTROL_SET)	32	RW	0000_0000h
1388h	Clock root control (CLOCK_ROOT39_CONTROL_CLR)	32	RW	0000_0000h
138Ch	Clock root control (CLOCK_ROOT39_CONTROL_TOG)	32	RW	0000_0000h
13B0h	Clock root access control (CLOCK_ROOT39_AUTHEN)	32	RW	0000_0000h
13B4h	Clock root access control (CLOCK_ROOT39_AUTHEN_SET)	32	RW	0000_0000h
13B8h	Clock root access control (CLOCK_ROOT39_AUTHEN_CLR)	32	RW	0000_0000h
13BCh	Clock root access control (CLOCK_ROOT39_AUTHEN_TOG)	32	RW	0000_0000h
1400h	Clock root control (CLOCK_ROOT40_CONTROL)	32	RW	0000_0000h
1404h	Clock root control (CLOCK_ROOT40_CONTROL_SET)	32	RW	0000_0000h
1408h	Clock root control (CLOCK_ROOT40_CONTROL_CLR)	32	RW	0000_0000h
140Ch	Clock root control (CLOCK_ROOT40_CONTROL_TOG)	32	RW	0000_0000h
1430h	Clock root access control (CLOCK_ROOT40_AUTHEN)	32	RW	0000_0000h
1434h	Clock root access control (CLOCK_ROOT40_AUTHEN_SET)	32	RW	0000_0000h
1438h	Clock root access control (CLOCK_ROOT40_AUTHEN_CLR)	32	RW	0000_0000h
143Ch	Clock root access control (CLOCK_ROOT40_AUTHEN_TOG)	32	RW	0000_0000h
1480h	Clock root control (CLOCK_ROOT41_CONTROL)	32	RW	0000_0000h
1484h	Clock root control (CLOCK_ROOT41_CONTROL_SET)	32	RW	0000_0000h
1488h	Clock root control (CLOCK_ROOT41_CONTROL_CLR)	32	RW	0000_0000h
148Ch	Clock root control (CLOCK_ROOT41_CONTROL_TOG)	32	RW	0000_0000h
14B0h	Clock root access control (CLOCK_ROOT41_AUTHEN)	32	RW	0000_0000h
14B4h	Clock root access control (CLOCK_ROOT41_AUTHEN_SET)	32	RW	0000_0000h
14B8h	Clock root access control (CLOCK_ROOT41_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
14BCh	Clock root access control (CLOCK_ROOT41_AUTHEN_TOG)	32	RW	0000_0000h
1500h	Clock root control (CLOCK_ROOT42_CONTROL)	32	RW	0000_0000h
1504h	Clock root control (CLOCK_ROOT42_CONTROL_SET)	32	RW	0000_0000h
1508h	Clock root control (CLOCK_ROOT42_CONTROL_CLR)	32	RW	0000_0000h
150Ch	Clock root control (CLOCK_ROOT42_CONTROL_TOG)	32	RW	0000_0000h
1530h	Clock root access control (CLOCK_ROOT42_AUTHEN)	32	RW	0000_0000h
1534h	Clock root access control (CLOCK_ROOT42_AUTHEN_SET)	32	RW	0000_0000h
1538h	Clock root access control (CLOCK_ROOT42_AUTHEN_CLR)	32	RW	0000_0000h
153Ch	Clock root access control (CLOCK_ROOT42_AUTHEN_TOG)	32	RW	0000_0000h
1580h	Clock root control (CLOCK_ROOT43_CONTROL)	32	RW	0000_0000h
1584h	Clock root control (CLOCK_ROOT43_CONTROL_SET)	32	RW	0000_0000h
1588h	Clock root control (CLOCK_ROOT43_CONTROL_CLR)	32	RW	0000_0000h
158Ch	Clock root control (CLOCK_ROOT43_CONTROL_TOG)	32	RW	0000_0000h
15B0h	Clock root access control (CLOCK_ROOT43_AUTHEN)	32	RW	0000_0000h
15B4h	Clock root access control (CLOCK_ROOT43_AUTHEN_SET)	32	RW	0000_0000h
15B8h	Clock root access control (CLOCK_ROOT43_AUTHEN_CLR)	32	RW	0000_0000h
15BCh	Clock root access control (CLOCK_ROOT43_AUTHEN_TOG)	32	RW	0000_0000h
1600h	Clock root control (CLOCK_ROOT44_CONTROL)	32	RW	0000_0000h
1604h	Clock root control (CLOCK_ROOT44_CONTROL_SET)	32	RW	0000_0000h
1608h	Clock root control (CLOCK_ROOT44_CONTROL_CLR)	32	RW	0000_0000h
160Ch	Clock root control (CLOCK_ROOT44_CONTROL_TOG)	32	RW	0000_0000h
1630h	Clock root access control (CLOCK_ROOT44_AUTHEN)	32	RW	0000_0000h
1634h	Clock root access control (CLOCK_ROOT44_AUTHEN_SET)	32	RW	0000_0000h
1638h	Clock root access control (CLOCK_ROOT44_AUTHEN_CLR)	32	RW	0000_0000h
163Ch	Clock root access control (CLOCK_ROOT44_AUTHEN_TOG)	32	RW	0000_0000h
1680h	Clock root control (CLOCK_ROOT45_CONTROL)	32	RW	0000_0000h
1684h	Clock root control (CLOCK_ROOT45_CONTROL_SET)	32	RW	0000_0000h
1688h	Clock root control (CLOCK_ROOT45_CONTROL_CLR)	32	RW	0000_0000h
168Ch	Clock root control (CLOCK_ROOT45_CONTROL_TOG)	32	RW	0000_0000h
16B0h	Clock root access control (CLOCK_ROOT45_AUTHEN)	32	RW	0000_0000h
16B4h	Clock root access control (CLOCK_ROOT45_AUTHEN_SET)	32	RW	0000_0000h
16B8h	Clock root access control (CLOCK_ROOT45_AUTHEN_CLR)	32	RW	0000_0000h
16BCh	Clock root access control (CLOCK_ROOT45_AUTHEN_TOG)	32	RW	0000_0000h
1700h	Clock root control (CLOCK_ROOT46_CONTROL)	32	RW	0000_0000h
1704h	Clock root control (CLOCK_ROOT46_CONTROL_SET)	32	RW	0000_0000h
1708h	Clock root control (CLOCK_ROOT46_CONTROL_CLR)	32	RW	0000_0000h
170Ch	Clock root control (CLOCK_ROOT46_CONTROL_TOG)	32	RW	0000_0000h
1730h	Clock root access control (CLOCK_ROOT46_AUTHEN)	32	RW	0000_0000h
1734h	Clock root access control (CLOCK_ROOT46_AUTHEN_SET)	32	RW	0000_0000h
1738h	Clock root access control (CLOCK_ROOT46_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
173Ch	Clock root access control (CLOCK_ROOT46_AUTHEN_TOG)	32	RW	0000_0000h
1780h	Clock root control (CLOCK_ROOT47_CONTROL)	32	RW	0000_0000h
1784h	Clock root control (CLOCK_ROOT47_CONTROL_SET)	32	RW	0000_0000h
1788h	Clock root control (CLOCK_ROOT47_CONTROL_CLR)	32	RW	0000_0000h
178Ch	Clock root control (CLOCK_ROOT47_CONTROL_TOG)	32	RW	0000_0000h
17B0h	Clock root access control (CLOCK_ROOT47_AUTHEN)	32	RW	0000_0000h
17B4h	Clock root access control (CLOCK_ROOT47_AUTHEN_SET)	32	RW	0000_0000h
17B8h	Clock root access control (CLOCK_ROOT47_AUTHEN_CLR)	32	RW	0000_0000h
17BCh	Clock root access control (CLOCK_ROOT47_AUTHEN_TOG)	32	RW	0000_0000h
1800h	Clock root control (CLOCK_ROOT48_CONTROL)	32	RW	0000_0000h
1804h	Clock root control (CLOCK_ROOT48_CONTROL_SET)	32	RW	0000_0000h
1808h	Clock root control (CLOCK_ROOT48_CONTROL_CLR)	32	RW	0000_0000h
180Ch	Clock root control (CLOCK_ROOT48_CONTROL_TOG)	32	RW	0000_0000h
1830h	Clock root access control (CLOCK_ROOT48_AUTHEN)	32	RW	0000_0000h
1834h	Clock root access control (CLOCK_ROOT48_AUTHEN_SET)	32	RW	0000_0000h
1838h	Clock root access control (CLOCK_ROOT48_AUTHEN_CLR)	32	RW	0000_0000h
183Ch	Clock root access control (CLOCK_ROOT48_AUTHEN_TOG)	32	RW	0000_0000h
1880h	Clock root control (CLOCK_ROOT49_CONTROL)	32	RW	0000_0000h
1884h	Clock root control (CLOCK_ROOT49_CONTROL_SET)	32	RW	0000_0000h
1888h	Clock root control (CLOCK_ROOT49_CONTROL_CLR)	32	RW	0000_0000h
188Ch	Clock root control (CLOCK_ROOT49_CONTROL_TOG)	32	RW	0000_0000h
18B0h	Clock root access control (CLOCK_ROOT49_AUTHEN)	32	RW	0000_0000h
18B4h	Clock root access control (CLOCK_ROOT49_AUTHEN_SET)	32	RW	0000_0000h
18B8h	Clock root access control (CLOCK_ROOT49_AUTHEN_CLR)	32	RW	0000_0000h
18BCh	Clock root access control (CLOCK_ROOT49_AUTHEN_TOG)	32	RW	0000_0000h
1900h	Clock root control (CLOCK_ROOT50_CONTROL)	32	RW	0000_0000h
1904h	Clock root control (CLOCK_ROOT50_CONTROL_SET)	32	RW	0000_0000h
1908h	Clock root control (CLOCK_ROOT50_CONTROL_CLR)	32	RW	0000_0000h
190Ch	Clock root control (CLOCK_ROOT50_CONTROL_TOG)	32	RW	0000_0000h
1930h	Clock root access control (CLOCK_ROOT50_AUTHEN)	32	RW	0000_0000h
1934h	Clock root access control (CLOCK_ROOT50_AUTHEN_SET)	32	RW	0000_0000h
1938h	Clock root access control (CLOCK_ROOT50_AUTHEN_CLR)	32	RW	0000_0000h
193Ch	Clock root access control (CLOCK_ROOT50_AUTHEN_TOG)	32	RW	0000_0000h
1980h	Clock root control (CLOCK_ROOT51_CONTROL)	32	RW	0000_0000h
1984h	Clock root control (CLOCK_ROOT51_CONTROL_SET)	32	RW	0000_0000h
1988h	Clock root control (CLOCK_ROOT51_CONTROL_CLR)	32	RW	0000_0000h
198Ch	Clock root control (CLOCK_ROOT51_CONTROL_TOG)	32	RW	0000_0000h
19B0h	Clock root access control (CLOCK_ROOT51_AUTHEN)	32	RW	0000_0000h
19B4h	Clock root access control (CLOCK_ROOT51_AUTHEN_SET)	32	RW	0000_0000h
19B8h	Clock root access control (CLOCK_ROOT51_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
19BCh	Clock root access control (CLOCK_ROOT51_AUTHEN_TOG)	32	RW	0000_0000h
1A00h	Clock root control (CLOCK_ROOT52_CONTROL)	32	RW	0000_0000h
1A04h	Clock root control (CLOCK_ROOT52_CONTROL_SET)	32	RW	0000_0000h
1A08h	Clock root control (CLOCK_ROOT52_CONTROL_CLR)	32	RW	0000_0000h
1A0Ch	Clock root control (CLOCK_ROOT52_CONTROL_TOG)	32	RW	0000_0000h
1A30h	Clock root access control (CLOCK_ROOT52_AUTHEN)	32	RW	0000_0000h
1A34h	Clock root access control (CLOCK_ROOT52_AUTHEN_SET)	32	RW	0000_0000h
1A38h	Clock root access control (CLOCK_ROOT52_AUTHEN_CLR)	32	RW	0000_0000h
1A3Ch	Clock root access control (CLOCK_ROOT52_AUTHEN_TOG)	32	RW	0000_0000h
1A80h	Clock root control (CLOCK_ROOT53_CONTROL)	32	RW	0000_0000h
1A84h	Clock root control (CLOCK_ROOT53_CONTROL_SET)	32	RW	0000_0000h
1A88h	Clock root control (CLOCK_ROOT53_CONTROL_CLR)	32	RW	0000_0000h
1A8Ch	Clock root control (CLOCK_ROOT53_CONTROL_TOG)	32	RW	0000_0000h
1AB0h	Clock root access control (CLOCK_ROOT53_AUTHEN)	32	RW	0000_0000h
1AB4h	Clock root access control (CLOCK_ROOT53_AUTHEN_SET)	32	RW	0000_0000h
1AB8h	Clock root access control (CLOCK_ROOT53_AUTHEN_CLR)	32	RW	0000_0000h
1ABCh	Clock root access control (CLOCK_ROOT53_AUTHEN_TOG)	32	RW	0000_0000h
1B00h	Clock root control (CLOCK_ROOT54_CONTROL)	32	RW	0000_0000h
1B04h	Clock root control (CLOCK_ROOT54_CONTROL_SET)	32	RW	0000_0000h
1B08h	Clock root control (CLOCK_ROOT54_CONTROL_CLR)	32	RW	0000_0000h
1B0Ch	Clock root control (CLOCK_ROOT54_CONTROL_TOG)	32	RW	0000_0000h
1B30h	Clock root access control (CLOCK_ROOT54_AUTHEN)	32	RW	0000_0000h
1B34h	Clock root access control (CLOCK_ROOT54_AUTHEN_SET)	32	RW	0000_0000h
1B38h	Clock root access control (CLOCK_ROOT54_AUTHEN_CLR)	32	RW	0000_0000h
1B3Ch	Clock root access control (CLOCK_ROOT54_AUTHEN_TOG)	32	RW	0000_0000h
1B80h	Clock root control (CLOCK_ROOT55_CONTROL)	32	RW	0000_0000h
1B84h	Clock root control (CLOCK_ROOT55_CONTROL_SET)	32	RW	0000_0000h
1B88h	Clock root control (CLOCK_ROOT55_CONTROL_CLR)	32	RW	0000_0000h
1B8Ch	Clock root control (CLOCK_ROOT55_CONTROL_TOG)	32	RW	0000_0000h
1BB0h	Clock root access control (CLOCK_ROOT55_AUTHEN)	32	RW	0000_0000h
1BB4h	Clock root access control (CLOCK_ROOT55_AUTHEN_SET)	32	RW	0000_0000h
1BB8h	Clock root access control (CLOCK_ROOT55_AUTHEN_CLR)	32	RW	0000_0000h
1BBCh	Clock root access control (CLOCK_ROOT55_AUTHEN_TOG)	32	RW	0000_0000h
1C00h	Clock root control (CLOCK_ROOT56_CONTROL)	32	RW	0000_0000h
1C04h	Clock root control (CLOCK_ROOT56_CONTROL_SET)	32	RW	0000_0000h
1C08h	Clock root control (CLOCK_ROOT56_CONTROL_CLR)	32	RW	0000_0000h
1C0Ch	Clock root control (CLOCK_ROOT56_CONTROL_TOG)	32	RW	0000_0000h
1C30h	Clock root access control (CLOCK_ROOT56_AUTHEN)	32	RW	0000_0000h
1C34h	Clock root access control (CLOCK_ROOT56_AUTHEN_SET)	32	RW	0000_0000h
1C38h	Clock root access control (CLOCK_ROOT56_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1C3Ch	Clock root access control (CLOCK_ROOT56_AUTHEN_TOG)	32	RW	0000_0000h
1C80h	Clock root control (CLOCK_ROOT57_CONTROL)	32	RW	0000_0000h
1C84h	Clock root control (CLOCK_ROOT57_CONTROL_SET)	32	RW	0000_0000h
1C88h	Clock root control (CLOCK_ROOT57_CONTROL_CLR)	32	RW	0000_0000h
1C8Ch	Clock root control (CLOCK_ROOT57_CONTROL_TOG)	32	RW	0000_0000h
1CB0h	Clock root access control (CLOCK_ROOT57_AUTHEN)	32	RW	0000_0000h
1CB4h	Clock root access control (CLOCK_ROOT57_AUTHEN_SET)	32	RW	0000_0000h
1CB8h	Clock root access control (CLOCK_ROOT57_AUTHEN_CLR)	32	RW	0000_0000h
1CBCh	Clock root access control (CLOCK_ROOT57_AUTHEN_TOG)	32	RW	0000_0000h
1D00h	Clock root control (CLOCK_ROOT58_CONTROL)	32	RW	0000_0000h
1D04h	Clock root control (CLOCK_ROOT58_CONTROL_SET)	32	RW	0000_0000h
1D08h	Clock root control (CLOCK_ROOT58_CONTROL_CLR)	32	RW	0000_0000h
1D0Ch	Clock root control (CLOCK_ROOT58_CONTROL_TOG)	32	RW	0000_0000h
1D30h	Clock root access control (CLOCK_ROOT58_AUTHEN)	32	RW	0000_0000h
1D34h	Clock root access control (CLOCK_ROOT58_AUTHEN_SET)	32	RW	0000_0000h
1D38h	Clock root access control (CLOCK_ROOT58_AUTHEN_CLR)	32	RW	0000_0000h
1D3Ch	Clock root access control (CLOCK_ROOT58_AUTHEN_TOG)	32	RW	0000_0000h
1D80h	Clock root control (CLOCK_ROOT59_CONTROL)	32	RW	0000_0000h
1D84h	Clock root control (CLOCK_ROOT59_CONTROL_SET)	32	RW	0000_0000h
1D88h	Clock root control (CLOCK_ROOT59_CONTROL_CLR)	32	RW	0000_0000h
1D8Ch	Clock root control (CLOCK_ROOT59_CONTROL_TOG)	32	RW	0000_0000h
1DB0h	Clock root access control (CLOCK_ROOT59_AUTHEN)	32	RW	0000_0000h
1DB4h	Clock root access control (CLOCK_ROOT59_AUTHEN_SET)	32	RW	0000_0000h
1DB8h	Clock root access control (CLOCK_ROOT59_AUTHEN_CLR)	32	RW	0000_0000h
1DBCh	Clock root access control (CLOCK_ROOT59_AUTHEN_TOG)	32	RW	0000_0000h
1E00h	Clock root control (CLOCK_ROOT60_CONTROL)	32	RW	0000_0000h
1E04h	Clock root control (CLOCK_ROOT60_CONTROL_SET)	32	RW	0000_0000h
1E08h	Clock root control (CLOCK_ROOT60_CONTROL_CLR)	32	RW	0000_0000h
1E0Ch	Clock root control (CLOCK_ROOT60_CONTROL_TOG)	32	RW	0000_0000h
1E30h	Clock root access control (CLOCK_ROOT60_AUTHEN)	32	RW	0000_0000h
1E34h	Clock root access control (CLOCK_ROOT60_AUTHEN_SET)	32	RW	0000_0000h
1E38h	Clock root access control (CLOCK_ROOT60_AUTHEN_CLR)	32	RW	0000_0000h
1E3Ch	Clock root access control (CLOCK_ROOT60_AUTHEN_TOG)	32	RW	0000_0000h
1E80h	Clock root control (CLOCK_ROOT61_CONTROL)	32	RW	0000_0000h
1E84h	Clock root control (CLOCK_ROOT61_CONTROL_SET)	32	RW	0000_0000h
1E88h	Clock root control (CLOCK_ROOT61_CONTROL_CLR)	32	RW	0000_0000h
1E8Ch	Clock root control (CLOCK_ROOT61_CONTROL_TOG)	32	RW	0000_0000h
1EB0h	Clock root access control (CLOCK_ROOT61_AUTHEN)	32	RW	0000_0000h
1EB4h	Clock root access control (CLOCK_ROOT61_AUTHEN_SET)	32	RW	0000_0000h
1EB8h	Clock root access control (CLOCK_ROOT61_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1EBCh	Clock root access control (CLOCK_ROOT61_AUTHEN_TOG)	32	RW	0000_0000h
1F00h	Clock root control (CLOCK_ROOT62_CONTROL)	32	RW	0000_0000h
1F04h	Clock root control (CLOCK_ROOT62_CONTROL_SET)	32	RW	0000_0000h
1F08h	Clock root control (CLOCK_ROOT62_CONTROL_CLR)	32	RW	0000_0000h
1F0Ch	Clock root control (CLOCK_ROOT62_CONTROL_TOG)	32	RW	0000_0000h
1F30h	Clock root access control (CLOCK_ROOT62_AUTHEN)	32	RW	0000_0000h
1F34h	Clock root access control (CLOCK_ROOT62_AUTHEN_SET)	32	RW	0000_0000h
1F38h	Clock root access control (CLOCK_ROOT62_AUTHEN_CLR)	32	RW	0000_0000h
1F3Ch	Clock root access control (CLOCK_ROOT62_AUTHEN_TOG)	32	RW	0000_0000h
1F80h	Clock root control (CLOCK_ROOT63_CONTROL)	32	RW	0000_0000h
1F84h	Clock root control (CLOCK_ROOT63_CONTROL_SET)	32	RW	0000_0000h
1F88h	Clock root control (CLOCK_ROOT63_CONTROL_CLR)	32	RW	0000_0000h
1F8Ch	Clock root control (CLOCK_ROOT63_CONTROL_TOG)	32	RW	0000_0000h
1FB0h	Clock root access control (CLOCK_ROOT63_AUTHEN)	32	RW	0000_0000h
1FB4h	Clock root access control (CLOCK_ROOT63_AUTHEN_SET)	32	RW	0000_0000h
1FB8h	Clock root access control (CLOCK_ROOT63_AUTHEN_CLR)	32	RW	0000_0000h
1FBCh	Clock root access control (CLOCK_ROOT63_AUTHEN_TOG)	32	RW	0000_0000h
2000h	Clock root control (CLOCK_ROOT64_CONTROL)	32	RW	0000_0000h
2004h	Clock root control (CLOCK_ROOT64_CONTROL_SET)	32	RW	0000_0000h
2008h	Clock root control (CLOCK_ROOT64_CONTROL_CLR)	32	RW	0000_0000h
200Ch	Clock root control (CLOCK_ROOT64_CONTROL_TOG)	32	RW	0000_0000h
2030h	Clock root access control (CLOCK_ROOT64_AUTHEN)	32	RW	0000_0000h
2034h	Clock root access control (CLOCK_ROOT64_AUTHEN_SET)	32	RW	0000_0000h
2038h	Clock root access control (CLOCK_ROOT64_AUTHEN_CLR)	32	RW	0000_0000h
203Ch	Clock root access control (CLOCK_ROOT64_AUTHEN_TOG)	32	RW	0000_0000h
2080h	Clock root control (CLOCK_ROOT65_CONTROL)	32	RW	0000_0000h
2084h	Clock root control (CLOCK_ROOT65_CONTROL_SET)	32	RW	0000_0000h
2088h	Clock root control (CLOCK_ROOT65_CONTROL_CLR)	32	RW	0000_0000h
208Ch	Clock root control (CLOCK_ROOT65_CONTROL_TOG)	32	RW	0000_0000h
20B0h	Clock root access control (CLOCK_ROOT65_AUTHEN)	32	RW	0000_0000h
20B4h	Clock root access control (CLOCK_ROOT65_AUTHEN_SET)	32	RW	0000_0000h
20B8h	Clock root access control (CLOCK_ROOT65_AUTHEN_CLR)	32	RW	0000_0000h
20BCh	Clock root access control (CLOCK_ROOT65_AUTHEN_TOG)	32	RW	0000_0000h
2100h	Clock root control (CLOCK_ROOT66_CONTROL)	32	RW	0000_0000h
2104h	Clock root control (CLOCK_ROOT66_CONTROL_SET)	32	RW	0000_0000h
2108h	Clock root control (CLOCK_ROOT66_CONTROL_CLR)	32	RW	0000_0000h
210Ch	Clock root control (CLOCK_ROOT66_CONTROL_TOG)	32	RW	0000_0000h
2130h	Clock root access control (CLOCK_ROOT66_AUTHEN)	32	RW	0000_0000h
2134h	Clock root access control (CLOCK_ROOT66_AUTHEN_SET)	32	RW	0000_0000h
2138h	Clock root access control (CLOCK_ROOT66_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
213Ch	Clock root access control (CLOCK_ROOT66_AUTHEN_TOG)	32	RW	0000_0000h
2180h	Clock root control (CLOCK_ROOT67_CONTROL)	32	RW	0000_0000h
2184h	Clock root control (CLOCK_ROOT67_CONTROL_SET)	32	RW	0000_0000h
2188h	Clock root control (CLOCK_ROOT67_CONTROL_CLR)	32	RW	0000_0000h
218Ch	Clock root control (CLOCK_ROOT67_CONTROL_TOG)	32	RW	0000_0000h
21B0h	Clock root access control (CLOCK_ROOT67_AUTHEN)	32	RW	0000_0000h
21B4h	Clock root access control (CLOCK_ROOT67_AUTHEN_SET)	32	RW	0000_0000h
21B8h	Clock root access control (CLOCK_ROOT67_AUTHEN_CLR)	32	RW	0000_0000h
21BCh	Clock root access control (CLOCK_ROOT67_AUTHEN_TOG)	32	RW	0000_0000h
2200h	Clock root control (CLOCK_ROOT68_CONTROL)	32	RW	0000_0000h
2204h	Clock root control (CLOCK_ROOT68_CONTROL_SET)	32	RW	0000_0000h
2208h	Clock root control (CLOCK_ROOT68_CONTROL_CLR)	32	RW	0000_0000h
220Ch	Clock root control (CLOCK_ROOT68_CONTROL_TOG)	32	RW	0000_0000h
2230h	Clock root access control (CLOCK_ROOT68_AUTHEN)	32	RW	0000_0000h
2234h	Clock root access control (CLOCK_ROOT68_AUTHEN_SET)	32	RW	0000_0000h
2238h	Clock root access control (CLOCK_ROOT68_AUTHEN_CLR)	32	RW	0000_0000h
223Ch	Clock root access control (CLOCK_ROOT68_AUTHEN_TOG)	32	RW	0000_0000h
2280h	Clock root control (CLOCK_ROOT69_CONTROL)	32	RW	0000_0000h
2284h	Clock root control (CLOCK_ROOT69_CONTROL_SET)	32	RW	0000_0000h
2288h	Clock root control (CLOCK_ROOT69_CONTROL_CLR)	32	RW	0000_0000h
228Ch	Clock root control (CLOCK_ROOT69_CONTROL_TOG)	32	RW	0000_0000h
22B0h	Clock root access control (CLOCK_ROOT69_AUTHEN)	32	RW	0000_0000h
22B4h	Clock root access control (CLOCK_ROOT69_AUTHEN_SET)	32	RW	0000_0000h
22B8h	Clock root access control (CLOCK_ROOT69_AUTHEN_CLR)	32	RW	0000_0000h
22BCh	Clock root access control (CLOCK_ROOT69_AUTHEN_TOG)	32	RW	0000_0000h
2300h	Clock root control (CLOCK_ROOT70_CONTROL)	32	RW	0000_0000h
2304h	Clock root control (CLOCK_ROOT70_CONTROL_SET)	32	RW	0000_0000h
2308h	Clock root control (CLOCK_ROOT70_CONTROL_CLR)	32	RW	0000_0000h
230Ch	Clock root control (CLOCK_ROOT70_CONTROL_TOG)	32	RW	0000_0000h
2330h	Clock root access control (CLOCK_ROOT70_AUTHEN)	32	RW	0000_0000h
2334h	Clock root access control (CLOCK_ROOT70_AUTHEN_SET)	32	RW	0000_0000h
2338h	Clock root access control (CLOCK_ROOT70_AUTHEN_CLR)	32	RW	0000_0000h
233Ch	Clock root access control (CLOCK_ROOT70_AUTHEN_TOG)	32	RW	0000_0000h
2380h	Clock root control (CLOCK_ROOT71_CONTROL)	32	RW	0000_0000h
2384h	Clock root control (CLOCK_ROOT71_CONTROL_SET)	32	RW	0000_0000h
2388h	Clock root control (CLOCK_ROOT71_CONTROL_CLR)	32	RW	0000_0000h
238Ch	Clock root control (CLOCK_ROOT71_CONTROL_TOG)	32	RW	0000_0000h
23B0h	Clock root access control (CLOCK_ROOT71_AUTHEN)	32	RW	0000_0000h
23B4h	Clock root access control (CLOCK_ROOT71_AUTHEN_SET)	32	RW	0000_0000h
23B8h	Clock root access control (CLOCK_ROOT71_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
23BCh	Clock root access control (CLOCK_ROOT71_AUTHEN_TOG)	32	RW	0000_0000h
2400h	Clock root control (CLOCK_ROOT72_CONTROL)	32	RW	0000_0000h
2404h	Clock root control (CLOCK_ROOT72_CONTROL_SET)	32	RW	0000_0000h
2408h	Clock root control (CLOCK_ROOT72_CONTROL_CLR)	32	RW	0000_0000h
240Ch	Clock root control (CLOCK_ROOT72_CONTROL_TOG)	32	RW	0000_0000h
2430h	Clock root access control (CLOCK_ROOT72_AUTHEN)	32	RW	0000_0000h
2434h	Clock root access control (CLOCK_ROOT72_AUTHEN_SET)	32	RW	0000_0000h
2438h	Clock root access control (CLOCK_ROOT72_AUTHEN_CLR)	32	RW	0000_0000h
243Ch	Clock root access control (CLOCK_ROOT72_AUTHEN_TOG)	32	RW	0000_0000h
2480h	Clock root control (CLOCK_ROOT73_CONTROL)	32	RW	0000_0000h
2484h	Clock root control (CLOCK_ROOT73_CONTROL_SET)	32	RW	0000_0000h
2488h	Clock root control (CLOCK_ROOT73_CONTROL_CLR)	32	RW	0000_0000h
248Ch	Clock root control (CLOCK_ROOT73_CONTROL_TOG)	32	RW	0000_0000h
24B0h	Clock root access control (CLOCK_ROOT73_AUTHEN)	32	RW	0000_0000h
24B4h	Clock root access control (CLOCK_ROOT73_AUTHEN_SET)	32	RW	0000_0000h
24B8h	Clock root access control (CLOCK_ROOT73_AUTHEN_CLR)	32	RW	0000_0000h
24BCh	Clock root access control (CLOCK_ROOT73_AUTHEN_TOG)	32	RW	0000_0000h
2500h	Clock root control (CLOCK_ROOT74_CONTROL)	32	RW	0000_0000h
2504h	Clock root control (CLOCK_ROOT74_CONTROL_SET)	32	RW	0000_0000h
2508h	Clock root control (CLOCK_ROOT74_CONTROL_CLR)	32	RW	0000_0000h
250Ch	Clock root control (CLOCK_ROOT74_CONTROL_TOG)	32	RW	0000_0000h
2530h	Clock root access control (CLOCK_ROOT74_AUTHEN)	32	RW	0000_0000h
2534h	Clock root access control (CLOCK_ROOT74_AUTHEN_SET)	32	RW	0000_0000h
2538h	Clock root access control (CLOCK_ROOT74_AUTHEN_CLR)	32	RW	0000_0000h
253Ch	Clock root access control (CLOCK_ROOT74_AUTHEN_TOG)	32	RW	0000_0000h
2580h	Clock root control (CLOCK_ROOT75_CONTROL)	32	RW	0000_0000h
2584h	Clock root control (CLOCK_ROOT75_CONTROL_SET)	32	RW	0000_0000h
2588h	Clock root control (CLOCK_ROOT75_CONTROL_CLR)	32	RW	0000_0000h
258Ch	Clock root control (CLOCK_ROOT75_CONTROL_TOG)	32	RW	0000_0000h
25B0h	Clock root access control (CLOCK_ROOT75_AUTHEN)	32	RW	0000_0000h
25B4h	Clock root access control (CLOCK_ROOT75_AUTHEN_SET)	32	RW	0000_0000h
25B8h	Clock root access control (CLOCK_ROOT75_AUTHEN_CLR)	32	RW	0000_0000h
25BCh	Clock root access control (CLOCK_ROOT75_AUTHEN_TOG)	32	RW	0000_0000h
2600h	Clock root control (CLOCK_ROOT76_CONTROL)	32	RW	0000_0000h
2604h	Clock root control (CLOCK_ROOT76_CONTROL_SET)	32	RW	0000_0000h
2608h	Clock root control (CLOCK_ROOT76_CONTROL_CLR)	32	RW	0000_0000h
260Ch	Clock root control (CLOCK_ROOT76_CONTROL_TOG)	32	RW	0000_0000h
2630h	Clock root access control (CLOCK_ROOT76_AUTHEN)	32	RW	0000_0000h
2634h	Clock root access control (CLOCK_ROOT76_AUTHEN_SET)	32	RW	0000_0000h
2638h	Clock root access control (CLOCK_ROOT76_AUTHEN_CLR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
263Ch	Clock root access control (CLOCK_ROOT76_AUTHEN_TOG)	32	RW	0000_0000h
2680h	Clock root control (CLOCK_ROOT77_CONTROL)	32	RW	0000_0000h
2684h	Clock root control (CLOCK_ROOT77_CONTROL_SET)	32	RW	0000_0000h
2688h	Clock root control (CLOCK_ROOT77_CONTROL_CLR)	32	RW	0000_0000h
268Ch	Clock root control (CLOCK_ROOT77_CONTROL_TOG)	32	RW	0000_0000h
26B0h	Clock root access control (CLOCK_ROOT77_AUTHEN)	32	RW	0000_0000h
26B4h	Clock root access control (CLOCK_ROOT77_AUTHEN_SET)	32	RW	0000_0000h
26B8h	Clock root access control (CLOCK_ROOT77_AUTHEN_CLR)	32	RW	0000_0000h
26BCh	Clock root access control (CLOCK_ROOT77_AUTHEN_TOG)	32	RW	0000_0000h
2700h	Clock root control (CLOCK_ROOT78_CONTROL)	32	RW	0000_0000h
2704h	Clock root control (CLOCK_ROOT78_CONTROL_SET)	32	RW	0000_0000h
2708h	Clock root control (CLOCK_ROOT78_CONTROL_CLR)	32	RW	0000_0000h
270Ch	Clock root control (CLOCK_ROOT78_CONTROL_TOG)	32	RW	0000_0000h
2730h	Clock root access control (CLOCK_ROOT78_AUTHEN)	32	RW	0000_0000h
2734h	Clock root access control (CLOCK_ROOT78_AUTHEN_SET)	32	RW	0000_0000h
2738h	Clock root access control (CLOCK_ROOT78_AUTHEN_CLR)	32	RW	0000_0000h
273Ch	Clock root access control (CLOCK_ROOT78_AUTHEN_TOG)	32	RW	0000_0000h
4000h	Clock group control (CLOCK_GROUP0_CONTROL)	32	RW	0000_0000h
4004h	Clock group control (CLOCK_GROUP0_CONTROL_SET)	32	RW	0000_0000h
4008h	Clock group control (CLOCK_GROUP0_CONTROL_CLR)	32	RW	0000_0000h
400Ch	Clock group control (CLOCK_GROUP0_CONTROL_TOG)	32	RW	0000_0000h
4020h - 40A0h	Clock group working status (CLOCK_GROUP0_STATUS0 - CLOCK_GROUP1_STATUS0)	32	RW	0000_0000h
4024h - 40A4h	Clock group low power/extend status (CLOCK_GROUP0_STATUS1 - CLOCK_GROUP1_STATUS1)	32	R	0000_0000h
402Ch - 40ACh	Clock group configuration (CLOCK_GROUP0_CONFIG - CLOCK_GROUP1_CONFIG)	32	R	Table 15-7
4030h	Clock group access control (CLOCK_GROUP0_AUTHEN)	32	RW	0000_0000h
4034h	Clock group access control (CLOCK_GROUP0_AUTHEN_SET)	32	RW	0000_0000h
4038h	Clock group access control (CLOCK_GROUP0_AUTHEN_CLR)	32	RW	0000_0000h
403Ch	Clock group access control (CLOCK_GROUP0_AUTHEN_TOG)	32	RW	0000_0000h
4040h - 40FCh	Setpoint setting (CLOCK_GROUP0_SETPOINT0 - CLOCK_GROUP1_SETPOINT15)	32	RW	0000_0000h
4080h	Clock group control (CLOCK_GROUP1_CONTROL)	32	RW	0000_0000h
4084h	Clock group control (CLOCK_GROUP1_CONTROL_SET)	32	RW	0000_0000h
4088h	Clock group control (CLOCK_GROUP1_CONTROL_CLR)	32	RW	0000_0000h
408Ch	Clock group control (CLOCK_GROUP1_CONTROL_TOG)	32	RW	0000_0000h
40B0h	Clock group access control (CLOCK_GROUP1_AUTHEN)	32	RW	0000_0000h
40B4h	Clock group access control (CLOCK_GROUP1_AUTHEN_SET)	32	RW	0000_0000h
40B8h	Clock group access control (CLOCK_GROUP1_AUTHEN_CLR)	32	RW	0000_0000h
40BCh	Clock group access control (CLOCK_GROUP1_AUTHEN_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4800h	General Purpose Register (GPR_SHARED0)	32	RW	0000_0000h
4804h	General Purpose Register (GPR_SHARED0_SET)	32	RW	0000_0000h
4808h	General Purpose Register (GPR_SHARED0_CLR)	32	RW	0000_0000h
480Ch	General Purpose Register (GPR_SHARED0_TOG)	32	RW	0000_0000h
4810h	GPR access control (GPR_SHARED0_AUTHEN)	32	RW	0000_0000h
4814h	GPR access control (GPR_SHARED0_AUTHEN_SET)	32	RW	0000_0000h
4818h	GPR access control (GPR_SHARED0_AUTHEN_CLR)	32	RW	0000_0000h
481Ch	GPR access control (GPR_SHARED0_AUTHEN_TOG)	32	RW	0000_0000h
4820h	General Purpose Register (GPR_SHARED1)	32	RW	0000_0000h
4824h	General Purpose Register (GPR_SHARED1_SET)	32	RW	0000_0000h
4828h	General Purpose Register (GPR_SHARED1_CLR)	32	RW	0000_0000h
482Ch	General Purpose Register (GPR_SHARED1_TOG)	32	RW	0000_0000h
4830h	GPR access control (GPR_SHARED1_AUTHEN)	32	RW	0000_0000h
4834h	GPR access control (GPR_SHARED1_AUTHEN_SET)	32	RW	0000_0000h
4838h	GPR access control (GPR_SHARED1_AUTHEN_CLR)	32	RW	0000_0000h
483Ch	GPR access control (GPR_SHARED1_AUTHEN_TOG)	32	RW	0000_0000h
4840h	General Purpose Register (GPR_SHARED2)	32	RW	0000_0000h
4844h	General Purpose Register (GPR_SHARED2_SET)	32	RW	0000_0000h
4848h	General Purpose Register (GPR_SHARED2_CLR)	32	RW	0000_0000h
484Ch	General Purpose Register (GPR_SHARED2_TOG)	32	RW	0000_0000h
4850h	GPR access control (GPR_SHARED2_AUTHEN)	32	RW	0000_0000h
4854h	GPR access control (GPR_SHARED2_AUTHEN_SET)	32	RW	0000_0000h
4858h	GPR access control (GPR_SHARED2_AUTHEN_CLR)	32	RW	0000_0000h
485Ch	GPR access control (GPR_SHARED2_AUTHEN_TOG)	32	RW	0000_0000h
4860h	General Purpose Register (GPR_SHARED3)	32	RW	0000_0000h
4864h	General Purpose Register (GPR_SHARED3_SET)	32	RW	0000_0000h
4868h	General Purpose Register (GPR_SHARED3_CLR)	32	RW	0000_0000h
486Ch	General Purpose Register (GPR_SHARED3_TOG)	32	RW	0000_0000h
4870h	GPR access control (GPR_SHARED3_AUTHEN)	32	RW	0000_0000h
4874h	GPR access control (GPR_SHARED3_AUTHEN_SET)	32	RW	0000_0000h
4878h	GPR access control (GPR_SHARED3_AUTHEN_CLR)	32	RW	0000_0000h
487Ch	GPR access control (GPR_SHARED3_AUTHEN_TOG)	32	RW	0000_0000h
4880h	General Purpose Register (GPR_SHARED4)	32	RW	0000_0000h
4884h	General Purpose Register (GPR_SHARED4_SET)	32	RW	0000_0000h
4888h	General Purpose Register (GPR_SHARED4_CLR)	32	RW	0000_0000h
488Ch	General Purpose Register (GPR_SHARED4_TOG)	32	RW	0000_0000h
4890h	GPR access control (GPR_SHARED4_AUTHEN)	32	RW	0000_0000h
4894h	GPR access control (GPR_SHARED4_AUTHEN_SET)	32	RW	0000_0000h
4898h	GPR access control (GPR_SHARED4_AUTHEN_CLR)	32	RW	0000_0000h
489Ch	GPR access control (GPR_SHARED4_AUTHEN_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
48A0h	General Purpose Register (GPR_SHARED5)	32	RW	0000_0000h
48A4h	General Purpose Register (GPR_SHARED5_SET)	32	RW	0000_0000h
48A8h	General Purpose Register (GPR_SHARED5_CLR)	32	RW	0000_0000h
48ACh	General Purpose Register (GPR_SHARED5_TOG)	32	RW	0000_0000h
48B0h	GPR access control (GPR_SHARED5_AUTHEN)	32	RW	0000_0000h
48B4h	GPR access control (GPR_SHARED5_AUTHEN_SET)	32	RW	0000_0000h
48B8h	GPR access control (GPR_SHARED5_AUTHEN_CLR)	32	RW	0000_0000h
48BCh	GPR access control (GPR_SHARED5_AUTHEN_TOG)	32	RW	0000_0000h
48C0h	General Purpose Register (GPR_SHARED6)	32	RW	0000_0000h
48C4h	General Purpose Register (GPR_SHARED6_SET)	32	RW	0000_0000h
48C8h	General Purpose Register (GPR_SHARED6_CLR)	32	RW	0000_0000h
48CCh	General Purpose Register (GPR_SHARED6_TOG)	32	RW	0000_0000h
48D0h	GPR access control (GPR_SHARED6_AUTHEN)	32	RW	0000_0000h
48D4h	GPR access control (GPR_SHARED6_AUTHEN_SET)	32	RW	0000_0000h
48D8h	GPR access control (GPR_SHARED6_AUTHEN_CLR)	32	RW	0000_0000h
48DCh	GPR access control (GPR_SHARED6_AUTHEN_TOG)	32	RW	0000_0000h
48E0h	General Purpose Register (GPR_SHARED7)	32	RW	0000_0000h
48E4h	General Purpose Register (GPR_SHARED7_SET)	32	RW	0000_0000h
48E8h	General Purpose Register (GPR_SHARED7_CLR)	32	RW	0000_0000h
48ECh	General Purpose Register (GPR_SHARED7_TOG)	32	RW	0000_0000h
48F0h	GPR access control (GPR_SHARED7_AUTHEN)	32	RW	0000_0000h
48F4h	GPR access control (GPR_SHARED7_AUTHEN_SET)	32	RW	0000_0000h
48F8h	GPR access control (GPR_SHARED7_AUTHEN_CLR)	32	RW	0000_0000h
48FCh	GPR access control (GPR_SHARED7_AUTHEN_TOG)	32	RW	0000_0000h
4C20h	General Purpose Register (GPR_PRIVATE1)	32	RW	0000_0000h
4C24h	General Purpose Register (GPR_PRIVATE1_SET)	32	RW	0000_0000h
4C28h	General Purpose Register (GPR_PRIVATE1_CLR)	32	RW	0000_0000h
4C2Ch	General Purpose Register (GPR_PRIVATE1_TOG)	32	RW	0000_0000h
4C30h	GPR access control (GPR_PRIVATE1_AUTHEN)	32	RW	0000_0000h
4C34h	GPR access control (GPR_PRIVATE1_AUTHEN_SET)	32	RW	0000_0000h
4C38h	GPR access control (GPR_PRIVATE1_AUTHEN_CLR)	32	RW	0000_0000h
4C3Ch	GPR access control (GPR_PRIVATE1_AUTHEN_TOG)	32	RW	0000_0000h
4C40h	General Purpose Register (GPR_PRIVATE2)	32	RW	0000_0000h
4C44h	General Purpose Register (GPR_PRIVATE2_SET)	32	RW	0000_0000h
4C48h	General Purpose Register (GPR_PRIVATE2_CLR)	32	RW	0000_0000h
4C4Ch	General Purpose Register (GPR_PRIVATE2_TOG)	32	RW	0000_0000h
4C50h	GPR access control (GPR_PRIVATE2_AUTHEN)	32	RW	0000_0000h
4C54h	GPR access control (GPR_PRIVATE2_AUTHEN_SET)	32	RW	0000_0000h
4C58h	GPR access control (GPR_PRIVATE2_AUTHEN_CLR)	32	RW	0000_0000h
4C5Ch	GPR access control (GPR_PRIVATE2_AUTHEN_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4C60h	General Purpose Register (GPR_PRIVATE3)	32	RW	0000_0000h
4C64h	General Purpose Register (GPR_PRIVATE3_SET)	32	RW	0000_0000h
4C68h	General Purpose Register (GPR_PRIVATE3_CLR)	32	RW	0000_0000h
4C6Ch	General Purpose Register (GPR_PRIVATE3_TOG)	32	RW	0000_0000h
4C70h	GPR access control (GPR_PRIVATE3_AUTHEN)	32	RW	0000_0000h
4C74h	GPR access control (GPR_PRIVATE3_AUTHEN_SET)	32	RW	0000_0000h
4C78h	GPR access control (GPR_PRIVATE3_AUTHEN_CLR)	32	RW	0000_0000h
4C7Ch	GPR access control (GPR_PRIVATE3_AUTHEN_TOG)	32	RW	0000_0000h
4C80h	General Purpose Register (GPR_PRIVATE4)	32	RW	0000_0000h
4C84h	General Purpose Register (GPR_PRIVATE4_SET)	32	RW	0000_0000h
4C88h	General Purpose Register (GPR_PRIVATE4_CLR)	32	RW	0000_0000h
4C8Ch	General Purpose Register (GPR_PRIVATE4_TOG)	32	RW	0000_0000h
4C90h	GPR access control (GPR_PRIVATE4_AUTHEN)	32	RW	0000_0000h
4C94h	GPR access control (GPR_PRIVATE4_AUTHEN_SET)	32	RW	0000_0000h
4C98h	GPR access control (GPR_PRIVATE4_AUTHEN_CLR)	32	RW	0000_0000h
4C9Ch	GPR access control (GPR_PRIVATE4_AUTHEN_TOG)	32	RW	0000_0000h
4CA0h	General Purpose Register (GPR_PRIVATE5)	32	RW	0000_0000h
4CA4h	General Purpose Register (GPR_PRIVATE5_SET)	32	RW	0000_0000h
4CA8h	General Purpose Register (GPR_PRIVATE5_CLR)	32	RW	0000_0000h
4CACh	General Purpose Register (GPR_PRIVATE5_TOG)	32	RW	0000_0000h
4CB0h	GPR access control (GPR_PRIVATE5_AUTHEN)	32	RW	0000_0000h
4CB4h	GPR access control (GPR_PRIVATE5_AUTHEN_SET)	32	RW	0000_0000h
4CB8h	GPR access control (GPR_PRIVATE5_AUTHEN_CLR)	32	RW	0000_0000h
4CBCh	GPR access control (GPR_PRIVATE5_AUTHEN_TOG)	32	RW	0000_0000h
4CC0h	General Purpose Register (GPR_PRIVATE6)	32	RW	0000_0000h
4CC4h	General Purpose Register (GPR_PRIVATE6_SET)	32	RW	0000_0000h
4CC8h	General Purpose Register (GPR_PRIVATE6_CLR)	32	RW	0000_0000h
4CCCh	General Purpose Register (GPR_PRIVATE6_TOG)	32	RW	0000_0000h
4CD0h	GPR access control (GPR_PRIVATE6_AUTHEN)	32	RW	0000_0000h
4CD4h	GPR access control (GPR_PRIVATE6_AUTHEN_SET)	32	RW	0000_0000h
4CD8h	GPR access control (GPR_PRIVATE6_AUTHEN_CLR)	32	RW	0000_0000h
4CDCh	GPR access control (GPR_PRIVATE6_AUTHEN_TOG)	32	RW	0000_0000h
4CE0h	General Purpose Register (GPR_PRIVATE7)	32	RW	0000_0000h
4CE4h	General Purpose Register (GPR_PRIVATE7_SET)	32	RW	0000_0000h
4CE8h	General Purpose Register (GPR_PRIVATE7_CLR)	32	RW	0000_0000h
4CECh	General Purpose Register (GPR_PRIVATE7_TOG)	32	RW	0000_0000h
4CF0h	GPR access control (GPR_PRIVATE7_AUTHEN)	32	RW	0000_0000h
4CF4h	GPR access control (GPR_PRIVATE7_AUTHEN_SET)	32	RW	0000_0000h
4CF8h	GPR access control (GPR_PRIVATE7_AUTHEN_CLR)	32	RW	0000_0000h
4CFCh	GPR access control (GPR_PRIVATE7_AUTHEN_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5000h - 5380h	Clock source direct control (OSCPLL0_DIRECT - OSCPLL28_DIRECT)	32	RW	0000_0001h
5004h - 5384h	Clock source domain control (OSCPLL0_DOMAIN - OSCPLL28_DOMAIN)	32	RW	0001_0001h
5008h - 5388h	Clock source Setpoint setting (OSCPLL0_SETPOINT - OSCPLL28_SETPOINT)	32	RW	0000_0000h
5010h - 5390h	Clock source working status (OSCPLL0_STATUS0 - OSCPLL28_STATUS0)	32	R	0000_0000h
5014h - 5394h	Clock source low power status (OSCPLL0_STATUS1 - OSCPLL28_STATUS1)	32	R	0000_0000h
5018h - 5398h	Clock source configuration (OSCPLL0_CONFIG - OSCPLL28_CONFIG)	32	R	Table 15-8
501Ch - 539Ch	Clock source access control (OSCPLL0_AUTHEN - OSCPLL28_AUTHEN)	32	RW	0000_0000h
6000h - 7120h	LPCG direct control (LPCG0_DIRECT - LPCG137_DIRECT)	32	RW	0000_0001h
6004h - 7124h	LPCG domain control (LPCG0_DOMAIN - LPCG137_DOMAIN)	32	RW	0001_0001h
6010h - 7130h	LPCG working status (LPCG0_STATUS0 - LPCG137_STATUS0)	32	R	0000_0000h
6014h - 7134h	LPCG low power status (LPCG0_STATUS1 - LPCG137_STATUS1)	32	R	0000_0000h
6018h - 7138h	LPCG configuration (LPCG0_CONFIG - LPCG137_CONFIG)	32	R	Table 15-9
601Ch - 713Ch	LPCG access control (LPCG0_AUTHEN - LPCG137_AUTHEN)	32	RW	0000_0000h
6048h - 6608h	LPCG Setpoint setting (LPCG2_SETPOINT - LPCG48_SETPOINT)	32	RW	0000_0000h

15.9.1.2 Clock root control (CLOCK_ROOT0_CONTROL - CLOCK_ROOT78_CONTROL)

15.9.1.2.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 78:

Register	Offset	Description
CLOCK_ROOTa_CONTROL	0h + (a × 80h)	Clock root control
CLOCK_ROOTa_CONTROL_SET	4h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_CONTROL is 1

Table continues on the next page...

Register	Offset	Description
CLOCK_ROOTa_CONTROL_CLR	8h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_CONTROL is 0
CLOCK_ROOTa_CONTROL_TOG	Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_ROOTa_CONTROL

15.9.1.2.2 Function

This register controls clock root generation.

15.9.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							OFF	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					MUX			DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.2.4 Fields

Field	Function
31-25 —	Reserved
24 OFF	OFF Shutdown clock root 0b - Turn on clock 1b - Turn off clock
23-16 —	Reserved
15-11 —	Reserved
10-8 MUX	Clock multiplexer Select clock from 8 clock sources.
7-0 DIV	Clock divider Divider selected clock by DIV + 1.

15.9.1.3 Clock root working status (CLOCK_ROOT0_STATUS0 - CLOCK_ROOT78_STATUS0)

15.9.1.3.1 Offset

For a = 0 to 78:

Register	Offset
CLOCK_ROOTa_STATUS0	20h + (a × 80h)

15.9.1.3.2 Function

This register shows current clock root running status.

15.9.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CHANGING	UPDATE_REVERSE	UPDATE_FORWARD	SLICE_BUSY	POWERDOWN	Reserved		OFF	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					MUX			DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.3.4 Fields

Field	Function
31 CHANGING	Internal updating in clock root Indication for clock root logic is internal updating. This status is combination of UPDATE_FORWARD, UPDATE_REVERSE, and SLICE_BUSY. 0b - Clock Status is not updating currently

Table continues on the next page...

Field	Function
	1b - Clock generation logic is updating currently
30 UPDATE_REVERSE	Internal status synchronization from clock generation logic Indication for internal status synchronizing to clock generation logic. 0b - Synchronization not in process 1b - Synchronization in process
29 UPDATE_FORWARD	Internal status synchronization to clock generation logic Indication for clock status is synchronizing for clock root. 0b - Synchronization not in process 1b - Synchronization in process
28 SLICE_BUSY	Internal updating in generation logic Indication for clock generation logic is applying new setting. 0b - Clock generation logic is not busy 1b - Clock generation logic is applying the new setting
27 POWERDOWN	Current clock root POWERDOWN setting Current running state of POWERDOWN field for clock root. 0b - Clock root is running 1b - Clock root is Powered Down
26-25 —	Reserved
24 OFF	Current clock root OFF setting Current running state of OFF field for clock root. 0b - Clock is running 1b - Clock is disabled/off
23-16 —	Reserved
15-11 —	Reserved
10-8 MUX	Current clock root MUX setting Current running state of MUX field for clock root.
7-0 DIV	Current clock root DIV setting Current running state of DIV field for clock root.

15.9.1.4 Clock root low power status (CLOCK_ROOT0_STATUS1 - CLOCK_ROOT78_STATUS1)

15.9.1.4.1 Offset

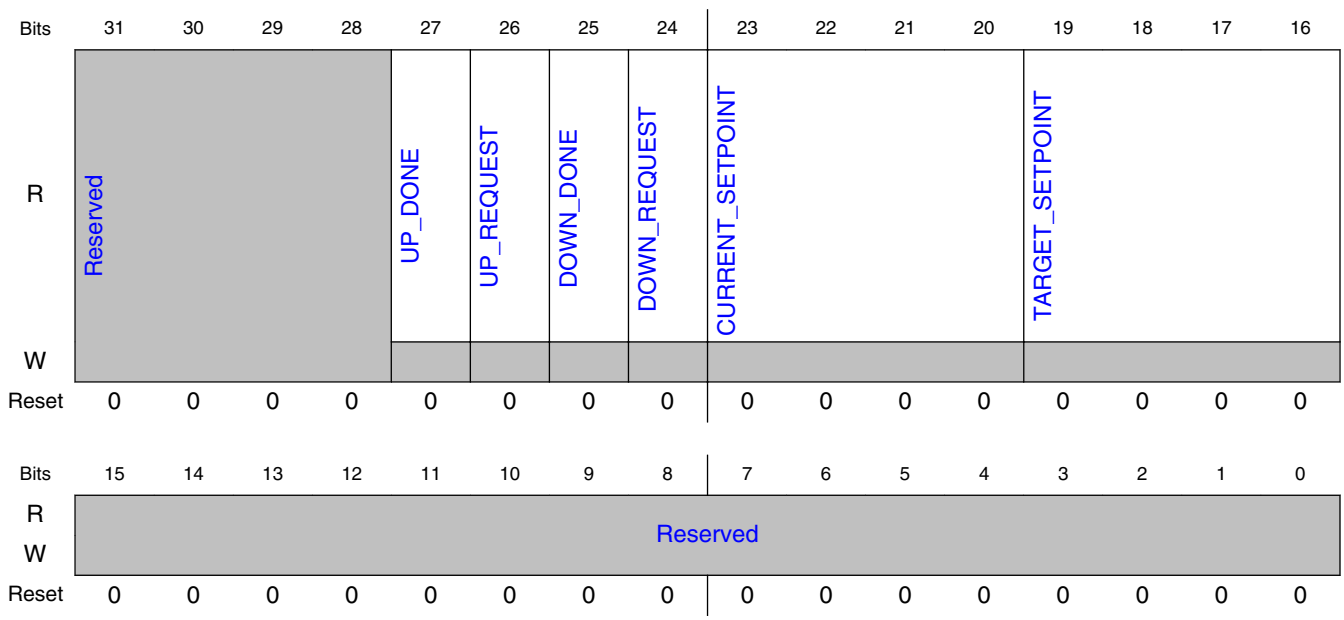
For a = 0 to 78:

Register	Offset
CLOCK_ROOTa_STATUS1	24h + (a × 80h)

15.9.1.4.2 Function

This register shows low power status of clock root.

15.9.1.4.3 Diagram



15.9.1.4.4 Fields

Field	Function
31-28 —	Reserved
27 UP_DONE	Clock frequency increase finish Handshake signal with GPC status indicating frequency increase finish. 0b - Frequency increase not completed 1b - Frequency increase completed
26 UP_REQUEST	Clock frequency increase request Handshake signal with GPC status indicating frequency increase is requesting. 0b - Frequency increase not requested 1b - Frequency increase requested
25 DOWN_DONE	Clock frequency decrease finish Handshake signal with GPC status indicating frequency decrease finish.

Table continues on the next page...

Field	Function
	0b - Frequency decrease not completed 1b - Frequency decrease completed
24 DOWN_REQUEST	Clock frequency decrease request Handshake signal with GPC status indicating frequency decrease is requesting. 0b - Frequency decrease not requested 1b - Frequency decrease requested
23-20 CURRENT_SETPOINT	Current Setpoint This is the Setpoint value the Soc is current working in.
19-16 TARGET_SETPOINT	Target Setpoint This is the Setpoint value the SoC will switch to
15-0 —	Reserved

15.9.1.5 Clock root configuration (CLOCK_ROOT0_CONFIG - CLOCK_ROOT78_CONFIG)

15.9.1.5.1 Offset

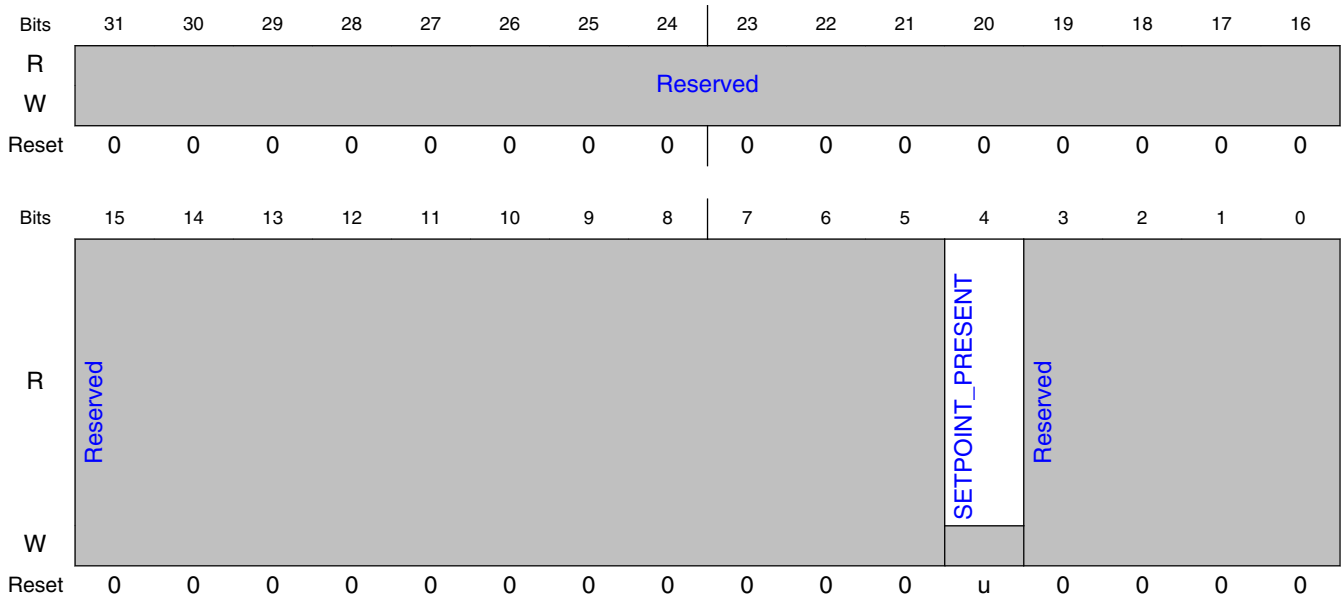
For a = 0 to 78:

Register	Offset
CLOCK_ROOTa_CONFIG	2Ch + (a × 80h)

15.9.1.5.2 Function

This register shows integration parameters

15.9.1.5.3 Diagram



15.9.1.5.4 Fields

Field	Function
31-5 —	Reserved
4 SETPOINT_PRESENT	Setpoint present This bit indicate whether this clock root implement Setpoint control scheme. 0b - Setpoint is not implemented. 1b - Setpoint is implemented.
3-0 —	Reserved

15.9.1.6 Clock root access control (CLOCK_ROOT0_AUTHEN - CLOCK_ROOT78_AUTHEN)

15.9.1.6.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

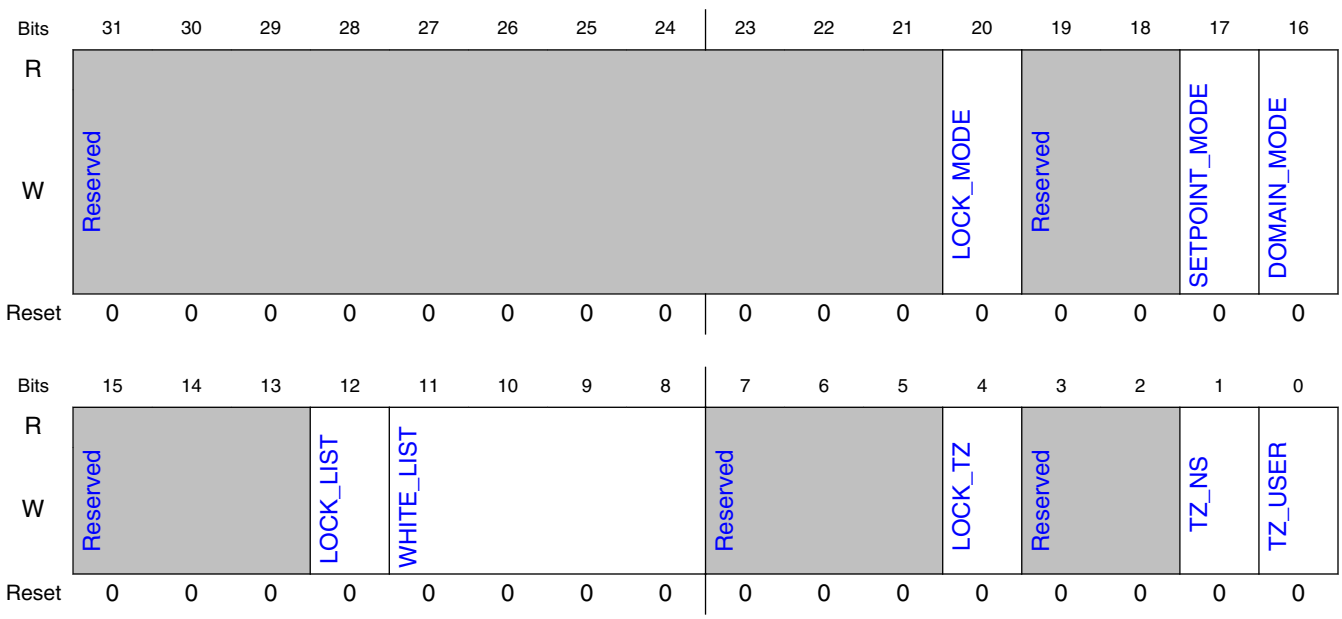
For a = 0 to 78:

Register	Offset	Description
CLOCK_ROOTa_AUTHEN	30h + (a × 80h)	Clock root access control
CLOCK_ROOTa_AUTHEN_SET	34h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_AUTHEN is 1
CLOCK_ROOTa_AUTHEN_CLR	38h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_ROOTa_AUTHEN is 0
CLOCK_ROOTa_AUTHEN_TOG	3Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_ROOTa_AUTHEN

15.9.1.6.2 Function

This register that manages clock root access control.

15.9.1.6.3 Diagram



15.9.1.6.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked 1b - MODE is locked

Table continues on the next page...

Field	Function
19-18 —	Reserved
17 SETPOINT_MODE	Low power and access control by Setpoint Clock root works in Setpoint controlled Mode 0b - Clock does NOT work in Setpoint Mode 1b - Clock works in Setpoint Mode
16 DOMAIN_MODE	Low power and access control by domain Clock root works in domain controlled mode 0b - Clock does NOT work in Domain Mode 1b - Clock works in Domain Mode
15-13 —	Reserved
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, CLOCK_ROOTx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked 1b - Whitelist is locked
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this clock root. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively. 0000b - This domain is NOT allowed to change clock 0001b - This domain is allowed to change clock
7-5 —	Reserved
4 LOCK_TZ	Lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked 1b - Trustzone setting is locked
3-2 —	Reserved
1 TZ_NS	Non-secure access This clock root can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode 1b - Can be changed in Non-secure mode
0 TZ_USER	User access This clock root can be changed when CPU is in user mode (in CMx core). 0b - Clock cannot be changed in user mode 1b - Clock can be changed in user mode

15.9.1.7 Setpoint setting (CLOCK_ROOT0_SETPOINT0 - CLOCK_ROOT78_SETPOINT15)

15.9.1.7.1 Offset

Register	Offset
CLOCK_ROOT0_SETPOINT0	40h
CLOCK_ROOT0_SETPOINT1	44h
CLOCK_ROOT0_SETPOINT2	48h
CLOCK_ROOT0_SETPOINT3	4Ch
CLOCK_ROOT0_SETPOINT4	50h
CLOCK_ROOT0_SETPOINT5	54h
CLOCK_ROOT0_SETPOINT6	58h
CLOCK_ROOT0_SETPOINT7	5Ch
CLOCK_ROOT0_SETPOINT8	60h
CLOCK_ROOT0_SETPOINT9	64h
CLOCK_ROOT0_SETPOINT10	68h
CLOCK_ROOT0_SETPOINT11	6Ch
CLOCK_ROOT0_SETPOINT12	70h
CLOCK_ROOT0_SETPOINT13	74h
CLOCK_ROOT0_SETPOINT14	78h
CLOCK_ROOT0_SETPOINT15	7Ch
CLOCK_ROOT1_SETPOINT0	C0h
CLOCK_ROOT1_SETPOINT1	C4h
CLOCK_ROOT1_SETPOINT2	C8h
CLOCK_ROOT1_SETPOINT3	CCh

Table continues on the next page...

Register	Offset
CLOCK_ROOT1_SETPOINT4	D0h
CLOCK_ROOT1_SETPOINT5	D4h
CLOCK_ROOT1_SETPOINT6	D8h
CLOCK_ROOT1_SETPOINT7	DCh
CLOCK_ROOT1_SETPOINT8	E0h
CLOCK_ROOT1_SETPOINT9	E4h
CLOCK_ROOT1_SETPOINT10	E8h
CLOCK_ROOT1_SETPOINT11	ECh
CLOCK_ROOT1_SETPOINT12	F0h
CLOCK_ROOT1_SETPOINT13	F4h
CLOCK_ROOT1_SETPOINT14	F8h
CLOCK_ROOT1_SETPOINT15	FCh
CLOCK_ROOT2_SETPOINT0	140h
CLOCK_ROOT2_SETPOINT1	144h
CLOCK_ROOT2_SETPOINT2	148h
CLOCK_ROOT2_SETPOINT3	14Ch
CLOCK_ROOT2_SETPOINT4	150h
CLOCK_ROOT2_SETPOINT5	154h
CLOCK_ROOT2_SETPOINT6	158h
CLOCK_ROOT2_SETPOINT7	15Ch
CLOCK_ROOT2_SETPOINT8	160h
CLOCK_ROOT2_SETPOINT9	164h
CLOCK_ROOT2_SETPOINT10	168h
CLOCK_ROOT2_SETPOINT11	16Ch

Table continues on the next page...

Register	Offset
CLOCK_ROOT2_SETPOINT12	170h
CLOCK_ROOT2_SETPOINT13	174h
CLOCK_ROOT2_SETPOINT14	178h
CLOCK_ROOT2_SETPOINT15	17Ch
CLOCK_ROOT3_SETPOINT0	1C0h
CLOCK_ROOT3_SETPOINT1	1C4h
CLOCK_ROOT3_SETPOINT2	1C8h
CLOCK_ROOT3_SETPOINT3	1CCh
CLOCK_ROOT3_SETPOINT4	1D0h
CLOCK_ROOT3_SETPOINT5	1D4h
CLOCK_ROOT3_SETPOINT6	1D8h
CLOCK_ROOT3_SETPOINT7	1DCh
CLOCK_ROOT3_SETPOINT8	1E0h
CLOCK_ROOT3_SETPOINT9	1E4h
CLOCK_ROOT3_SETPOINT10	1E8h
CLOCK_ROOT3_SETPOINT11	1ECh
CLOCK_ROOT3_SETPOINT12	1F0h
CLOCK_ROOT3_SETPOINT13	1F4h
CLOCK_ROOT3_SETPOINT14	1F8h
CLOCK_ROOT3_SETPOINT15	1FCh
CLOCK_ROOT4_SETPOINT0	240h
CLOCK_ROOT4_SETPOINT1	244h
CLOCK_ROOT4_SETPOINT2	248h
CLOCK_ROOT4_SETPOINT3	24Ch

Table continues on the next page...

Register	Offset
CLOCK_ROOT4_SETPOINT4	250h
CLOCK_ROOT4_SETPOINT5	254h
CLOCK_ROOT4_SETPOINT6	258h
CLOCK_ROOT4_SETPOINT7	25Ch
CLOCK_ROOT4_SETPOINT8	260h
CLOCK_ROOT4_SETPOINT9	264h
CLOCK_ROOT4_SETPOINT10	268h
CLOCK_ROOT4_SETPOINT11	26Ch
CLOCK_ROOT4_SETPOINT12	270h
CLOCK_ROOT4_SETPOINT13	274h
CLOCK_ROOT4_SETPOINT14	278h
CLOCK_ROOT4_SETPOINT15	27Ch
CLOCK_ROOT20_SETPOINT0	A40h
CLOCK_ROOT20_SETPOINT1	A44h
CLOCK_ROOT20_SETPOINT2	A48h
CLOCK_ROOT20_SETPOINT3	A4Ch
CLOCK_ROOT20_SETPOINT4	A50h
CLOCK_ROOT20_SETPOINT5	A54h
CLOCK_ROOT20_SETPOINT6	A58h
CLOCK_ROOT20_SETPOINT7	A5Ch
CLOCK_ROOT20_SETPOINT8	A60h
CLOCK_ROOT20_SETPOINT9	A64h
CLOCK_ROOT20_SETPOINT10	A68h
CLOCK_ROOT20_SETPOINT11	A6Ch

Table continues on the next page...

Register	Offset
CLOCK_ROOT20_SETPOINT12	A70h
CLOCK_ROOT20_SETPOINT13	A74h
CLOCK_ROOT20_SETPOINT14	A78h
CLOCK_ROOT20_SETPOINT15	A7Ch
CLOCK_ROOT21_SETPOINT0	AC0h
CLOCK_ROOT21_SETPOINT1	AC4h
CLOCK_ROOT21_SETPOINT2	AC8h
CLOCK_ROOT21_SETPOINT3	ACCh
CLOCK_ROOT21_SETPOINT4	AD0h
CLOCK_ROOT21_SETPOINT5	AD4h
CLOCK_ROOT21_SETPOINT6	AD8h
CLOCK_ROOT21_SETPOINT7	ADCh
CLOCK_ROOT21_SETPOINT8	AE0h
CLOCK_ROOT21_SETPOINT9	AE4h
CLOCK_ROOT21_SETPOINT10	AE8h
CLOCK_ROOT21_SETPOINT11	AECh
CLOCK_ROOT21_SETPOINT12	AF0h
CLOCK_ROOT21_SETPOINT13	AF4h
CLOCK_ROOT21_SETPOINT14	AF8h
CLOCK_ROOT21_SETPOINT15	AFCh
CLOCK_ROOT77_SETPOINT0	26C0h
CLOCK_ROOT77_SETPOINT1	26C4h
CLOCK_ROOT77_SETPOINT2	26C8h
CLOCK_ROOT77_SETPOINT3	26CCh

Table continues on the next page...

Register	Offset
CLOCK_ROOT77_SETPOINT4	26D0h
CLOCK_ROOT77_SETPOINT5	26D4h
CLOCK_ROOT77_SETPOINT6	26D8h
CLOCK_ROOT77_SETPOINT7	26DCh
CLOCK_ROOT77_SETPOINT8	26E0h
CLOCK_ROOT77_SETPOINT9	26E4h
CLOCK_ROOT77_SETPOINT10	26E8h
CLOCK_ROOT77_SETPOINT11	26ECh
CLOCK_ROOT77_SETPOINT12	26F0h
CLOCK_ROOT77_SETPOINT13	26F4h
CLOCK_ROOT77_SETPOINT14	26F8h
CLOCK_ROOT77_SETPOINT15	26FCh
CLOCK_ROOT78_SETPOINT0	2740h
CLOCK_ROOT78_SETPOINT1	2744h
CLOCK_ROOT78_SETPOINT2	2748h
CLOCK_ROOT78_SETPOINT3	274Ch
CLOCK_ROOT78_SETPOINT4	2750h
CLOCK_ROOT78_SETPOINT5	2754h
CLOCK_ROOT78_SETPOINT6	2758h
CLOCK_ROOT78_SETPOINT7	275Ch
CLOCK_ROOT78_SETPOINT8	2760h
CLOCK_ROOT78_SETPOINT9	2764h
CLOCK_ROOT78_SETPOINT10	2768h
CLOCK_ROOT78_SETPOINT11	276Ch

Table continues on the next page...

Register	Offset
CLOCK_ROOT78_SETPOINT12	2770h
CLOCK_ROOT78_SETPOINT13	2774h
CLOCK_ROOT78_SETPOINT14	2778h
CLOCK_ROOT78_SETPOINT15	277Ch

15.9.1.7.2 Function

These registers define 16 clock root setting which will take effect when the system switches to a corresponding Setpoint.

These 16 registers provide clock root generation settings for the Setpoint. Corresponding settings will be automatically loaded when system low power mode transition takes place.

The following clock roots support Setpoints:

- M7_CLK_ROOT
- M4_CLK_ROOT
- BUS_CLK_ROOT
- BUS_LPSR_CLK_ROOT
- SEMC_CLK_ROOT
- FLEXSPI1_CLK_ROOT
- FLEXSPI2_CLK_ROOT
- CCM_CLK01_CLK_ROOT
- CCM_CLKO2_CLK_ROOT

15.9.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GRADE				Reserved			OFF	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					MUX			DIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.7.4 Fields

Field	Function
31-28 GRADE	Grade Indicate speed grade for each Setpoint setting. This value MUST be set to make Setpoint work properly. Values will be used CCM internally to determine clock frequency relation ship between Setpoint settings and generate proper sequence. Smaller value means higher clock speed.
27-25 —	Reserved
24 OFF	OFF OFF value in Setpoint 0b - ON 1b - OFF
23-11 —	Reserved
10-8 MUX	Clock multiplexer MUX value in Setpoint.
7-0 DIV	Clock divider DIV value in Setpoint.

15.9.1.8 Clock group control (CLOCK_GROUP0_CONTROL - CLOCK_GROUP1_CONTROL)

15.9.1.8.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

For a = 0 to 1:

Register	Offset	Description
CLOCK_GROUPa_CONTROL	4000h + (a × 80h)	Clock group control
CLOCK_GROUPa_CONTROL_SET	4004h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_GROUPa_CONTROL is 1
CLOCK_GROUPa_CONTROL_CLR	4008h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_GROUPa_CONTROL is 0

Table continues on the next page...

Register	Offset	Description
CLOCK_GROUPa_CONTROL_TOG	400Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_GROUPa_CONTROL

15.9.1.8.2 Function

This register controls synchronous clock group clocks generation.

15.9.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							OFF	RSTDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												DIV0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.8.4 Fields

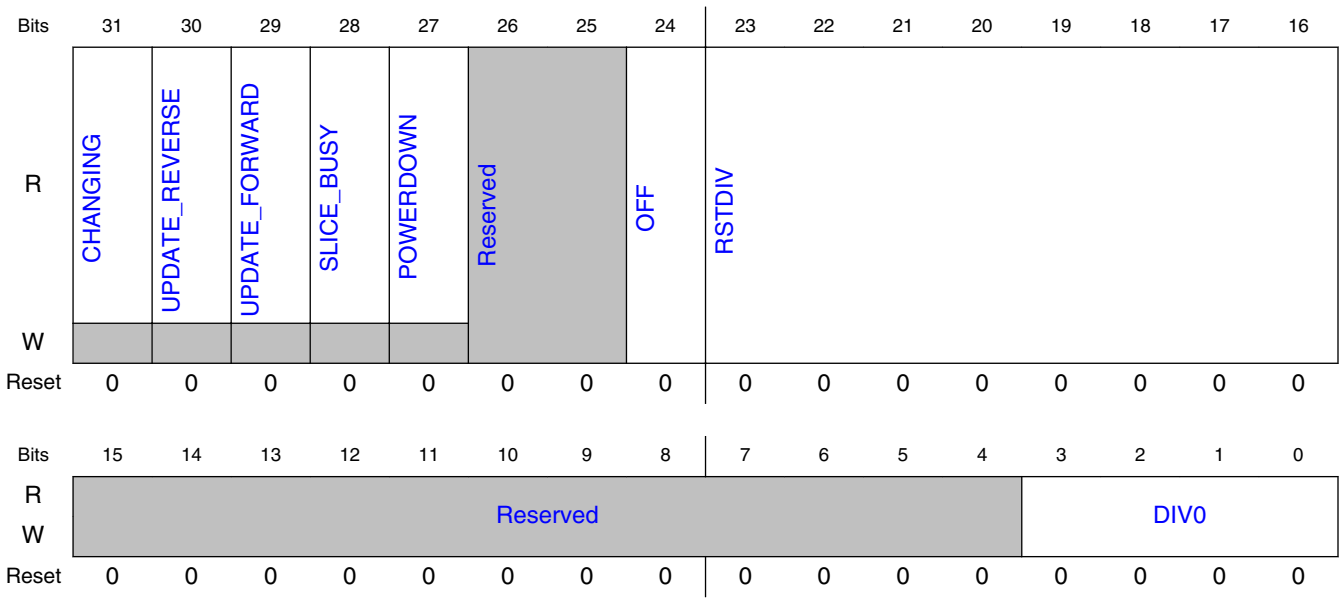
Field	Function
31-25 —	Reserved
24 OFF	OFF Shutdown all clocks in clock group 0b - Clock is running 1b - Turn off clock
23-16 RSTDIV	Clock group global restart count Clock group will restart when this divider overflows. This field must be common multiple of all dividers.
15-4 —	Reserved
3-0 DIV0	Clock divider0 Divider clock root by DIV0 + 1.

15.9.1.9 Clock group working status (CLOCK_GROUP0_STATUS0 - CLOCK_GROUP1_STATUS0)

15.9.1.9.1 Offset

Register	Offset
CLOCK_GROUP0_STATUS0	4020h
CLOCK_GROUP1_STATUS0	40A0h

15.9.1.9.2 Diagram



15.9.1.9.3 Fields

Field	Function
31 CHANGING	Internal updating in clock group Indication for clock group logic is internal updating. This status is combination of UPDATE_FORWARD, UPDATE_REVERSE, and SLICE_BUSY. 0b - Clock root is not updating currently 1b - Clock root logic is updating currently
30 UPDATE_REVERSE	Internal status synchronization from clock generation logic Indication for internal status synchronizing to clock generation logic. 0b - Synchronization not in process 1b - Synchronization in process

Table continues on the next page...

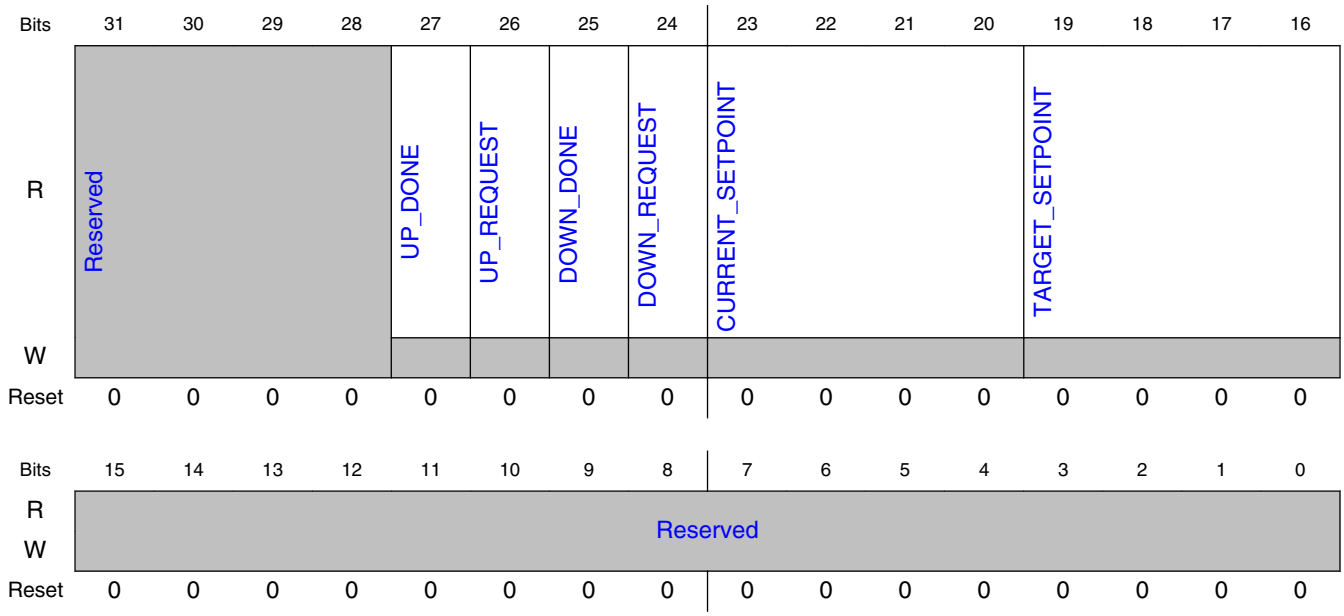
Field	Function
29 UPDATE_FORWARD	Internal status synchronization to clock generation logic Indication for clock status is synchronizing fro clock generation logic. 0b - Synchronization not in process 1b - Synchronization in process
28 SLICE_BUSY	Internal updating in generation logic Indication for clock generation logic is applying new setting. 0b - Clock generation logic is not busy 1b - Clock generation logic is applying the new setting
27 POWERDOWN	Current clock root POWERDOWN setting Current running state of POWERDOWN field for clock root. 0b - Clock root is running 1b - Clock root is Powered Down
26-25 —	Reserved
24 OFF	OFF Current running state of OFF field for clock group. 0b - Clock is running. 1b - Turn off clock.
23-16 RSTDIV	Clock divider Current running state of RSTDIV field for clock group.
15-4 —	Reserved
3-0 DIV0	Clock divider Current running state of DIV0 field for clock group.

15.9.1.10 Clock group low power/extend status (CLOCK_GROUP0_STATUS1 - CLOCK_GROUP1_STATUS1)

15.9.1.10.1 Offset

Register	Offset
CLOCK_GROUP0_STATUS1	4024h
CLOCK_GROUP1_STATUS1	40A4h

15.9.1.10.2 Diagram



15.9.1.10.3 Fields

Field	Function
31-28 —	Reserved
27 UP_DONE	Clock frequency increase complete Handshake signal with GPC status indicating frequency increase is complete. 0b - Handshake signal with GPC status indicating frequency increase is not complete 1b - Handshake signal with GPC status indicating frequency increase is complete
26 UP_REQUEST	Clock frequency increase request Handshake signal with GPC status indicating frequency increase is requested. 0b - No handshake signal is not requested 1b - Handshake signal with GPC status indicating frequency increase is requested
25 DOWN_DONE	Clock frequency decrease complete Handshake signal with GPC status indicating frequency decrease is complete. 0b - Handshake signal with GPC status indicating frequency decrease is not complete 1b - Handshake signal with GPC status indicating frequency decrease is complete
24 DOWN_REQUEST	Clock frequency decrease request 0b - No handshake signal is not requested 1b - Handshake signal with GPC status indicating frequency decrease is requested
23-20 CURRENT_SETPOINT	Current Setpoint This is the Setpoint value the Soc is current working in
19-16	Next Setpoint to change to This is the Setpoint value the SoC will switch to

Table continues on the next page...

Field	Function
TARGET_SETPOINT	
15-0 —	Reserved

15.9.1.11 Clock group configuration (CLOCK_GROUP0_CONFIG - CLOCK_GROUP1_CONFIG)

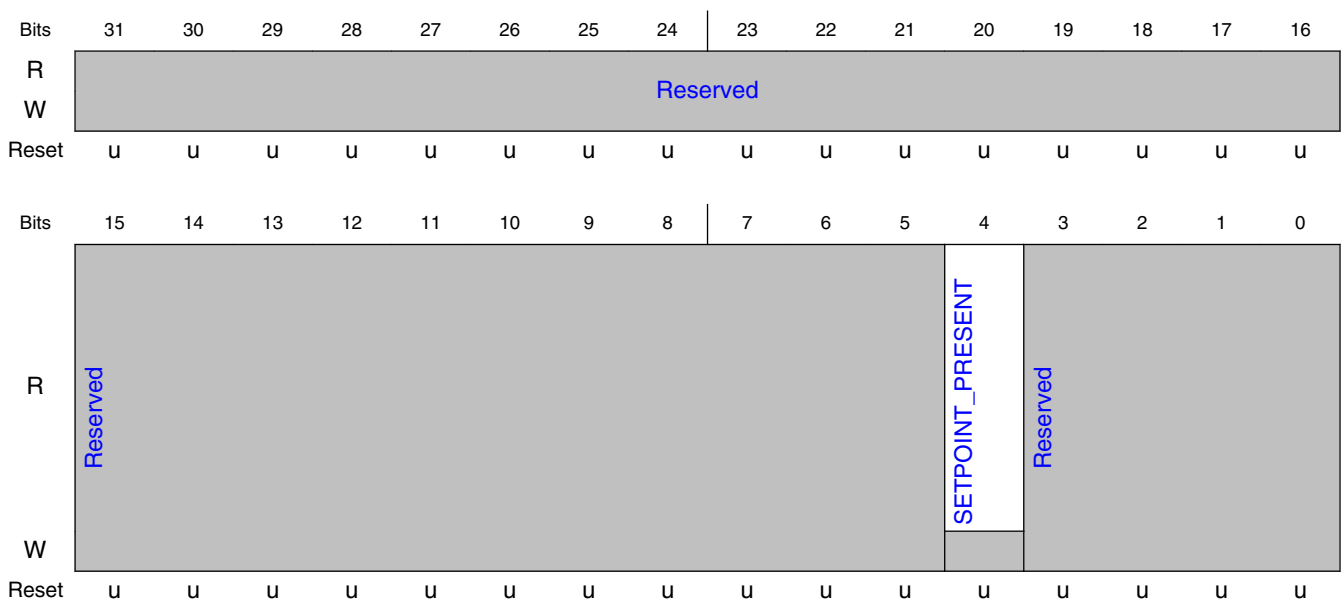
15.9.1.11.1 Offset

Register	Offset
CLOCK_GROUP0_CONFIG	402Ch
CLOCK_GROUP1_CONFIG	40ACh

15.9.1.11.2 Function

This register shows integration parameters.

15.9.1.11.3 Diagram



15.9.1.11.4 Fields

Field	Function
31-5 —	Reserved
4 SETPOINT_PRESENT	Setpoint present This bit indicate whether this clock root implement Setpoint control scheme. 0b - Setpoint is not implemented. 1b - Setpoint is implemented.
3-0 —	Reserved

15.9.1.12 Clock group access control (CLOCK_GROUP0_AUTHEN - CLOCK_GROUP1_AUTHEN)

15.9.1.12.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

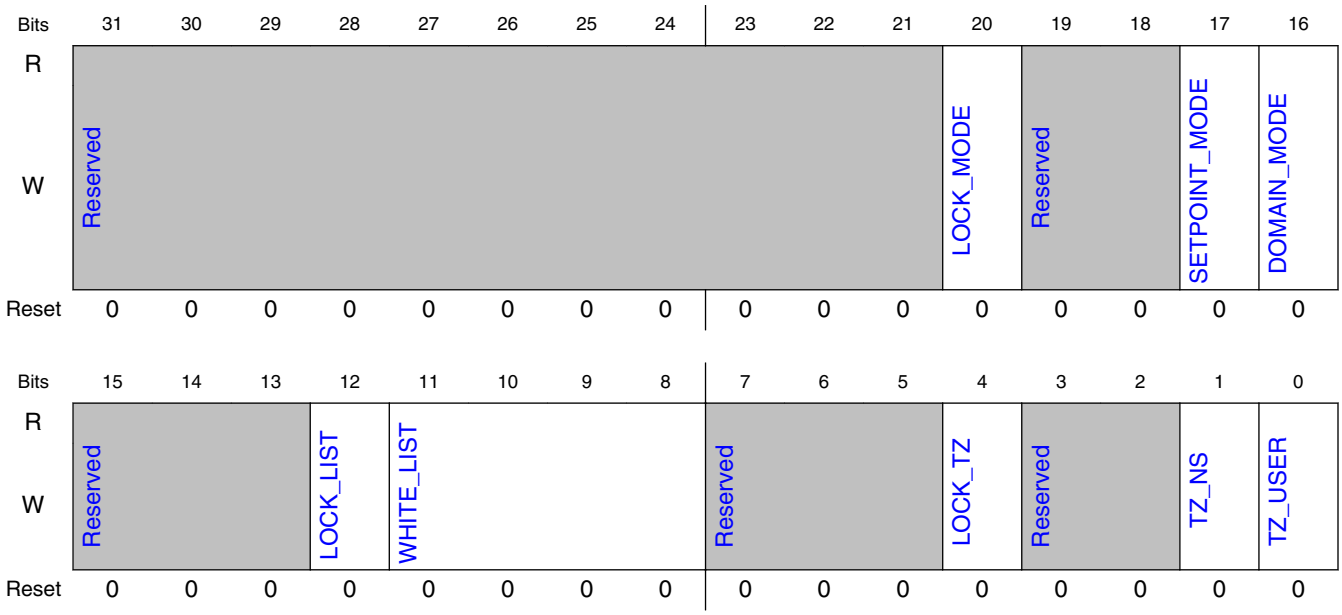
For a = 0 to 1:

Register	Offset	Description
CLOCK_GROUPa_AUTHEN	4030h + (a × 80h)	Clock group access control
CLOCK_GROUPa_AUTHEN_SET	4034h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_GROUPa_AUTHEN is 1
CLOCK_GROUPa_AUTHEN_CLR	4038h + (a × 80h)	Writing 1 to a bit in this register ensures that the corresponding bit in CLOCK_GROUPa_AUTHEN is 0
CLOCK_GROUPa_AUTHEN_TOG	403Ch + (a × 80h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in CLOCK_GROUPa_AUTHEN

15.9.1.12.2 Function

This register manages clock group access control.

15.9.1.12.3 Diagram



15.9.1.12.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked. 1b - MODE is locked.
19-18 —	Reserved
17 SETPOINT_MODE	Low power and access control by Setpoint Clock group works in Setpoint controlled Mode.
16 DOMAIN_MODE	Low power and access control by domain Clock group works in Domain controlled Mode. 0b - Clock does not work in Domain Mode. 1b - Clock works in Domain Mode.
15-13 —	Reserved
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, CLOCK_GROUPx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked.

Table continues on the next page...

Field	Function
	1b - Whitelist is locked.
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this clock root. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively.
7-5 —	Reserved
4 LOCK_TZ	Lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked. 1b - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This clock root can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode. 1b - Can be changed in Non-secure mode.
0 TZ_USER	User access This clock group can be changed when CPU is in user mode (in CMx core). 0b - Clock cannot be changed in user mode. 1b - Clock can be changed in user mode.

15.9.1.13 Setpoint setting (CLOCK_GROUP0_SETPOINT0 - CLOCK_GROUP1_SETPOINT15)

15.9.1.13.1 Offset

For a = 0 to 1; c = 0 to 15:

Register	Offset
CLOCK_GROUPa_SETPOINTc	4040h + (a × 80h) + (c × 4h)

15.9.1.13.2 Function

Register that provide clock root generation setting for Setpoint.

Corresponding setting will be automatically loaded when system low power mode shift.

15.9.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GRADE				Reserved			OFF	RSTDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												DIV0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.13.4 Fields

Field	Function
31-28 GRADE	Grade Indicate speed grade for each Setpoint setting. This value MUST be set to make Setpoint work properly. Values will be used CCM internally to determine clock frequency relation ship between Setpoint settings and generate proper sequence. Smaller value means higher clock speed.
27-25 —	Reserved
24 OFF	OFF Shutdown all locks in clock group. 0b - Clock is running. 1b - Turn off clock.
23-16 RSTDIV	Clock group global restart count Clock group will restart when this divider overflows. This field must be common multiple of all dividers.
15-4 —	Reserved
3-0 DIV0	Clock divider Divider selected clock by DIV + 1. 0000b - Direct output. 0001b - Divide by 2. 0010b - Divide by 3. 0011b - Divide by 4. 1111b - Divide by 16.

15.9.1.14 General Purpose Register (GPR_SHARED0 - GPR_SHARED7)

15.9.1.14.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

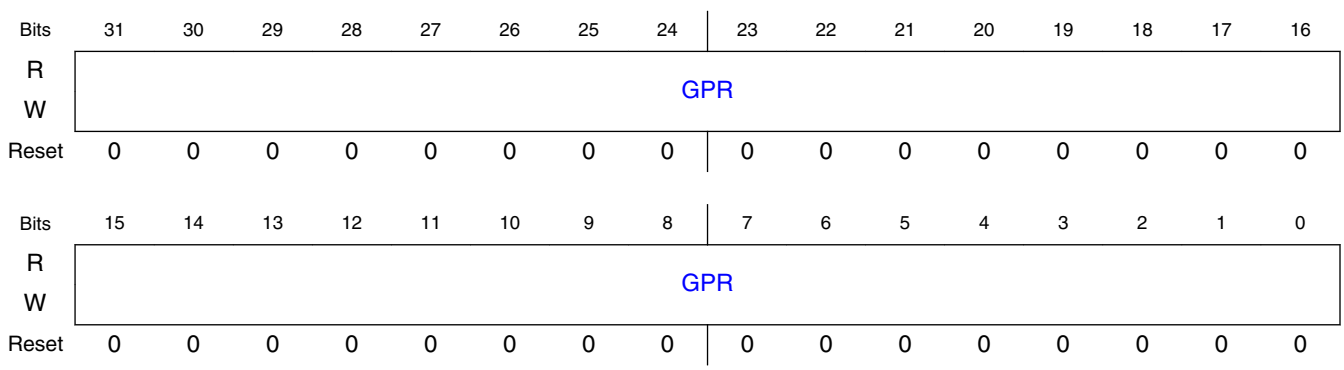
For a = 0 to 7:

Register	Offset	Description
GPR_SHAREDa	4800h + (a × 20h)	General Purpose Register
GPR_SHAREDa_SET	4804h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa is 1
GPR_SHAREDa_CLR	4808h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa is 0
GPR_SHAREDa_TOG	480Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHAREDa

15.9.1.14.2 Function

These registers do not have any dedicated functions. They can be set by the user

15.9.1.14.3 Diagram



15.9.1.14.4 Fields

Field	Function
31-0 GPR	GP register This register is shared for all CPU domains.

15.9.1.15 GPR access control (GPR_SHARED0_AUTHEN - GPR_SHARED7_AUTHEN)

15.9.1.15.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

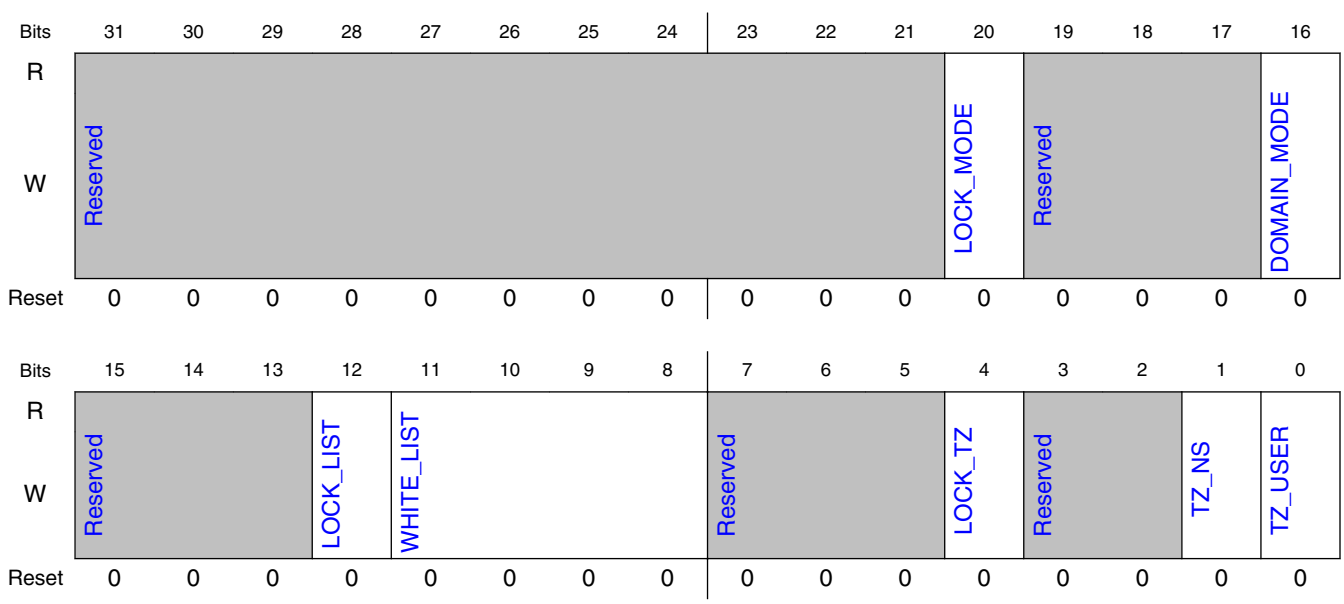
For a = 0 to 7:

Register	Offset	Description
GPR_SHAREDa_AUTHEN	4810h + (a × 20h)	GPR access control
GPR_SHAREDa_AUTHEN_SET	4814h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa_AUTHEN is 1
GPR_SHAREDa_AUTHEN_CLR	4818h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_SHAREDa_AUTHEN is 0
GPR_SHAREDa_AUTHEN_TOG	481Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_SHAREDa_AUTHEN

15.9.1.15.2 Function

This register manages GPR access control.

15.9.1.15.3 Diagram



15.9.1.15.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked. 1b - MODE is locked.
19-17 —	Reserved
16 DOMAIN_MODE	Low power and access control by domain Register works in Domain controlled Mode. 0b - Clock does NOT work in Domain Mode. 1b - Clock works in Domain Mode.
15-13 —	Reserved
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, GPR_SHAREDx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked. 1b - Whitelist is locked.
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this register. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively. 0000b - This domain is NOT allowed to change clock. 0001b - This domain is allowed to change clock.
7-5 —	Reserved
4 LOCK_TZ	Lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked. 1b - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This register can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode. 1b - Can be changed in Non-secure mode.
0 TZ_USER	User access This register can be changed when CPU is in user mode (in CMx core). 0b - Clock cannot be changed in user mode. 1b - Clock can be changed in user mode.

15.9.1.16 General Purpose Register (GPR_PRIVATE1 - GPR_PRIVATE7)

15.9.1.16.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

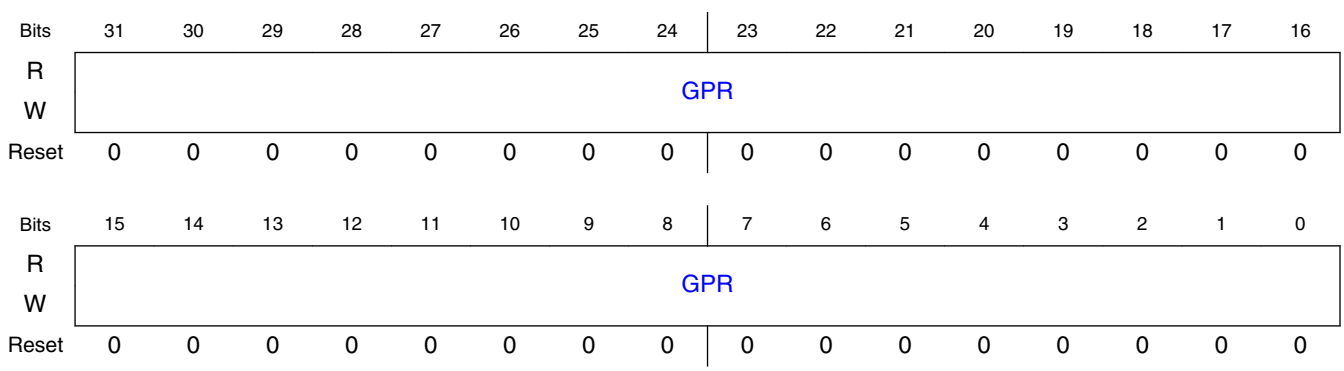
For a = 1 to 7:

Register	Offset	Description
GPR_PRIVATEa	4C00h + (a × 20h)	General Purpose Register
GPR_PRIVATEa_SET	4C04h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa is 1
GPR_PRIVATEa_CLR	4C08h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa is 0
GPR_PRIVATEa_TOG	4C0Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_PRIVATEa

15.9.1.16.2 Function

These registers is general purpose register.

15.9.1.16.3 Diagram



15.9.1.16.4 Fields

Field	Function
31-0 GPR	GP register General purpose register. This register has dedicate bits for each domain.

15.9.1.17 GPR access control (GPR_PRIVATE1_AUTHEN - GPR_PRIVATE7_AUTHEN)

15.9.1.17.1 Offset

This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

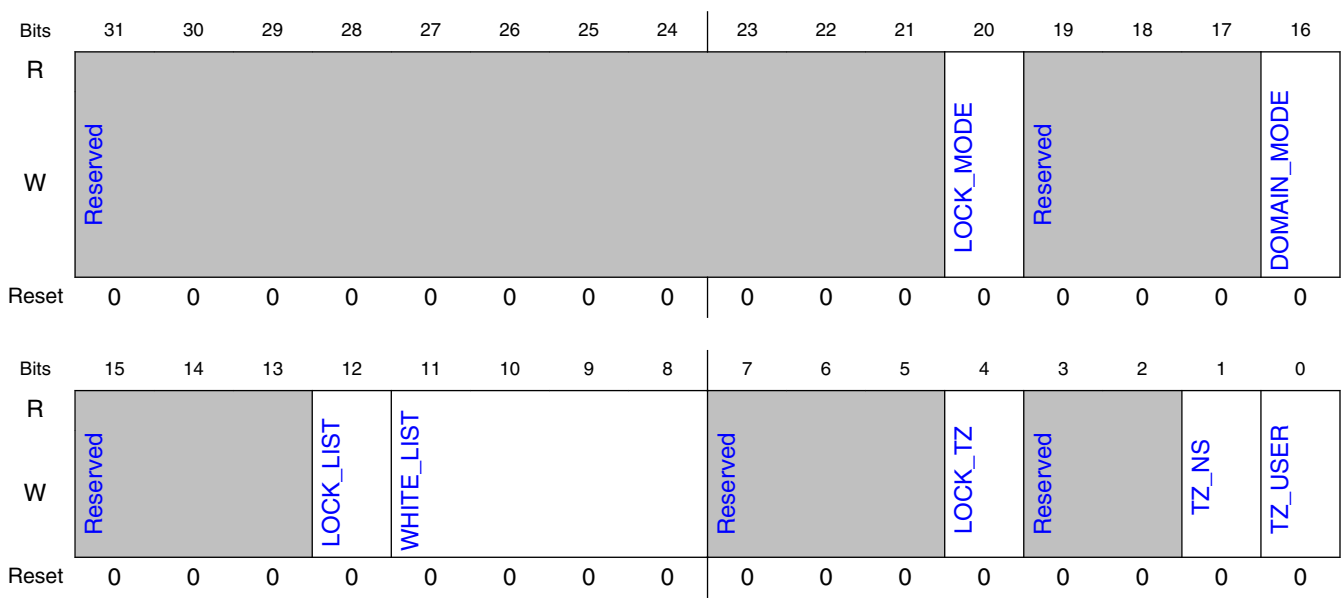
For a = 1 to 7:

Register	Offset	Description
GPR_PRIVATEa_AUTHEN	4C10h + (a × 20h)	GPR access control
GPR_PRIVATEa_AUTHEN_SET	4C14h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa_AUTHEN is 1
GPR_PRIVATEa_AUTHEN_CLR	4C18h + (a × 20h)	Writing 1 to a bit in this register ensures that the corresponding bit in GPR_PRIVATEa_AUTHEN is 0
GPR_PRIVATEa_AUTHEN_TOG	4C1Ch + (a × 20h)	Writing 1 to a bit in this register inverts the value of the corresponding bit in GPR_PRIVATEa_AUTHEN

15.9.1.17.2 Function

This register that manages clock root access control.

15.9.1.17.3 Diagram



15.9.1.17.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked. 1b - MODE is locked.
19-17 —	Reserved
16 DOMAIN_MODE	Low power and access control by Domain Register works in Domain controlled Mode. 0b - Clock does NOT work in Domain Mode. 1b - Clock works in Domain Mode.
15-13 —	Reserved
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, GPR_PRIVATEx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked. 1b - Whitelist is locked.
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this register. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively. 0000b - This domain is NOT allowed to change clock. 0001b - This domain is allowed to change clock.
7-5 —	Reserved
4 LOCK_TZ	Lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked. 1b - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This register can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode. 1b - Can be changed in Non-secure mode.
0 TZ_USER	User access This register can be changed when CPU is in user mode (in CMx core). 0b - Clock cannot be changed in user mode. 1b - Clock can be changed in user mode.

15.9.1.18 Clock source direct control (OSCPLL0_DIRECT - OSCPLL28_DIRECT)

15.9.1.18.1 Offset

For a = 0 to 28:

Register	Offset
OSCPLLa_DIRECT	5000h + (a × 20h)

15.9.1.18.2 Function

This register controls clock source on and off when clock source works in Unassigned Mode and Domain Mode.

15.9.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															ON
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

15.9.1.18.4 Fields

Field	Function
31-1 —	Reserved
0 ON	turn on clock source This bit controls clock source. 0b - OSCPLL is OFF 1b - OSCPLL is ON

15.9.1.19 Clock source domain control (OSCPLL0_DOMAIN - OSCPLL28_DOMAIN)

15.9.1.19.1 Offset

For a = 0 to 28:

Register	Offset
OSCPLLa_DOMAIN	5004h + (a × 20h)

15.9.1.19.2 Function

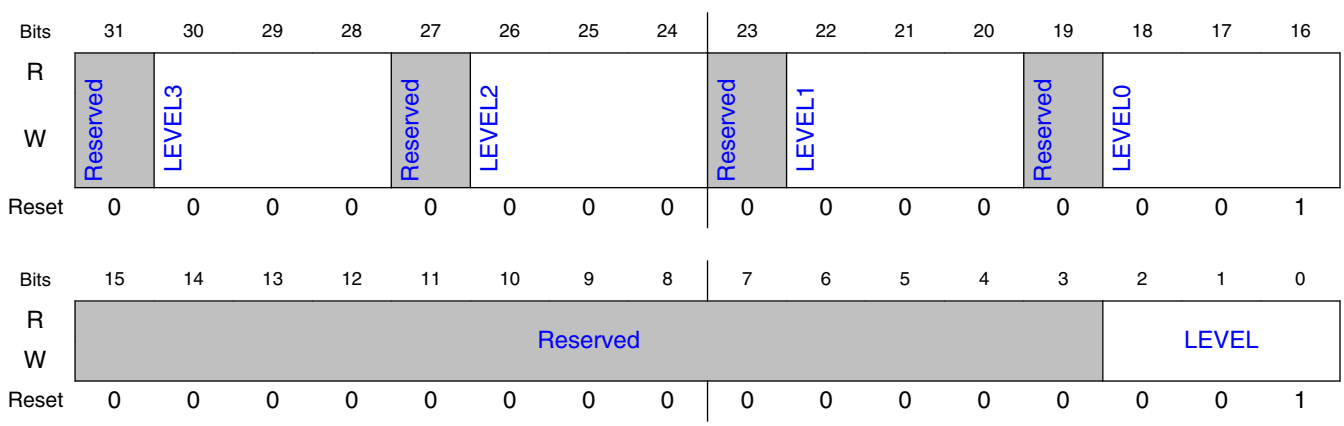
This register controls clock source on and off when Clock source works in CPU Low Power Mode.

During CPULPM mode, write to bit field OSCPLLn_DOMAIN[LEVEL], and during Unassigned mode, write to OSCPLLn_DOMAIN[LEVELx], where x = 0,1,2 or 3. See the table below for more details.

Table 15-8. CCM Mode and access types

CCM Mode/ Access	Unassigned Mode		CPU Low Power Mode		Other Modes	
	Read	Write	Read	Write	Read	Write
LEVEL	Y	N	Y	Y	Y	N
LEVEL0	Y	Y	Y	N	Y	N
LEVEL1	Y	Y	Y	N	Y	N
LEVEL2	Y	Y	Y	N	Y	N
LEVEL3	Y	Y	Y	N	Y	N

15.9.1.19.3 Diagram



15.9.1.19.4 Fields

Field	Function
31 —	Reserved
30-28 LEVEL3	Depend level Depend level of this clock source for DOMAIN3. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
27 —	Reserved
26-24 LEVEL2	Depend level Depend level of this clock source for DOMAIN2. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
23 —	Reserved
22-20 LEVEL1	Depend level Depend level of this clock source for DOMAIN1. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
19 —	Reserved
18-16 LEVEL0	Dependence level Dependence level of this clock source for DOMAIN0 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
15-3 —	Reserved
2-0 LEVEL	Current dependence level Dependence level of this clock source for the current accessing domain 000b - This clock source is not needed in any mode, and can be turned off

Field	Function
	001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved

15.9.1.20 Clock source Setpoint setting (OSCPLL0_SETPOINT - OSCPLL28_SETPOINT)

15.9.1.20.1 Offset

For a = 0 to 28:

Register	Offset
OSCPLLa_SETPOINT	5008h + (a × 20h)

15.9.1.20.2 Function

This register defines 16 Setpoint values when clock source works in Setpoint Mode.

15.9.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STANDBY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SETPOINT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.20.4 Fields

Field	Function
31-16 STANDBY	Standby This field defines 16 Setpoint standby values. Bit0~Bit15 hold value for Setpoint 0~16 standby respectively.

Table continues on the next page...

Field	Function
	A bitfield value of 0 implies the OSC or PLL will be shutdown during standby. A bitfield value of 1 represent OSC or PLL will keep Setpoint setting during standby.
15-0 SETPOINT	Setpoint This field defines 16 Setpoint values. Bit0~Bit15 hold value for Setpoint 0~16 respectively. A bitfield value of 0 implies OSC or PLL will be shutdown in this Setpoint. A bitfield value of 1 implies OSC or PLL will be turn on in this Setpoint.

15.9.1.21 Clock source working status (OSCPLL0_STATUS0 - OSCPLL28_STATUS0)

15.9.1.21.1 Offset

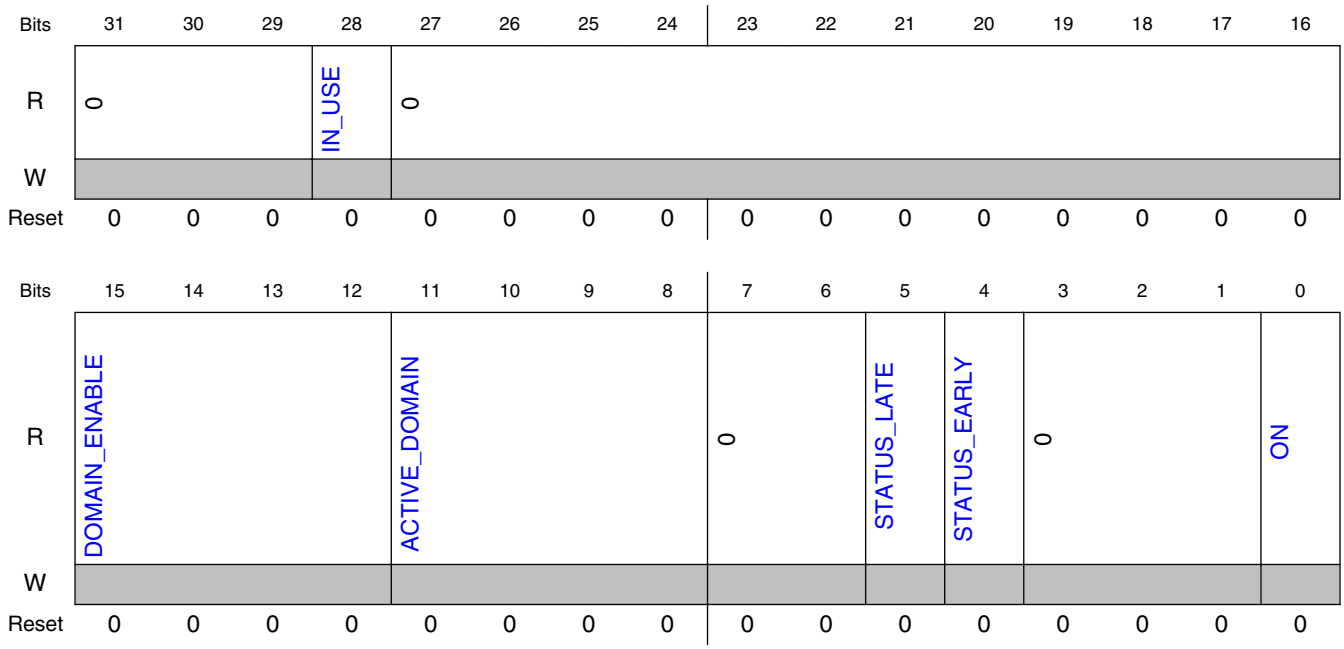
For a = 0 to 28:

Register	Offset
OSCPLLa_STATUS0	5010h + (a × 20h)

15.9.1.21.2 Function

This register shows current clock source running status.

15.9.1.21.3 Diagram



15.9.1.21.4 Fields

Field	Function
31-29 —	Reserved
28 IN_USE	In use This status bit indicates whether the clock source is being used by active clock roots 0b - Clock source is not being used by clock roots 1b - Clock source is being used by clock roots
27-16 —	Reserved
15-12 DOMAIN_ENABLE	Enable status from each domain Enable status from domains, each bit represent one domain. 0000b - No domain request 0001b - Request from Domain0 0010b - Request from Domain1 0011b - Request from Domain0 and Domain1 0100b - Request from Domain2 0101b - Request from Domain0 and Domain2 0110b - Request from Domain1 and Domain2 0111b - Request from Domain0, Domain1 and Domain 2 1000b - Request from Domain3 1001b - Request from Domain0 and Domain3 1010b - Request from Domain1 and Domain3 1011b - Request from Domain2 and Domain3 1100b - Request from Domain0, Domain 1, and Domain3 1101b - Request from Domain0, Domain 2, and Domain3

Table continues on the next page...

Field	Function
	1110b - Request from Domain1, Domain 2, and Domain3 1111b - Request from all domains
11-8 ACTIVE_DOMAIN	Domains that own this clock source Domains that own this clock source according to Whitelist. 0000b - Clock not owned by any domain 0001b - Clock owned by Domain0 0010b - Clock owned by Domain1 0011b - Clock owned by Domain0 and Domain1 0100b - Clock owned by Domain2 0101b - Clock owned by Domain0 and Domain2 0110b - Clock owned by Domain1 and Domain2 0111b - Clock owned by Domain0, Domain1 and Domain 2 1000b - Clock owned by Domain3 1001b - Clock owned by Domain0 and Domain3 1010b - Clock owned by Domain1 and Domain3 1011b - Clock owned by Domain2 and Domain3 1100b - Clock owned by Domain0, Domain 1, and Domain3 1101b - Clock owned by Domain0, Domain 2, and Domain3 1110b - Clock owned by Domain1, Domain 2, and Domain3 1111b - Clock owned by all domains
7-6 —	Reserved
5 STATUS_LATE	Clock source ready This status bit indicating clock source is ready to use. 0b - Clock source is not ready to use 1b - Clock source is ready to use
4 STATUS_EARLY	Clock source active This status bit indicating clock source is active. 0b - Clock source is not active 1b - Clock source is active
3-1 —	Reserved
0 ON	Clock source current state Clock source running status 0b - Clock source is OFF 1b - Clock source is ON

15.9.1.22 Clock source low power status (OSCPLL0_STATUS1 - OSCPLL28_STATUS1)

15.9.1.22.1 Offset

For a = 0 to 28:

Register	Offset
OSCPLLa_STATUS1	5014h + (a × 20h)

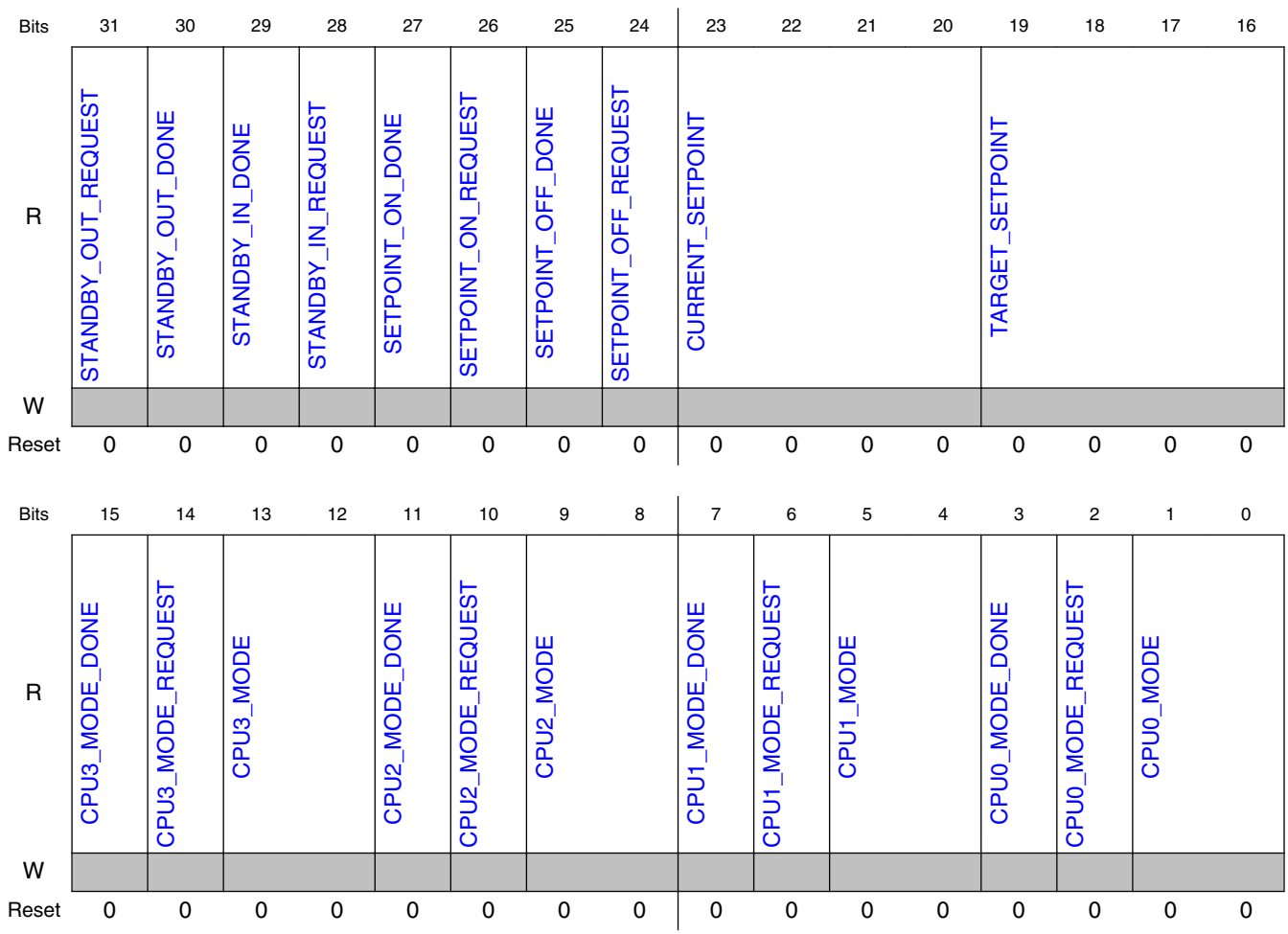
15.9.1.22.2 Function

This register shows low power status of clock source.

NOTE

The CPU0-3 references are to the CPU domain assignment, not the number of supported CPU platforms. Each CPU platform can be assigned to any domain, including the same domain.

15.9.1.22.3 Diagram



15.9.1.22.4 Fields

Field	Function
31 STANDBY_OUT_REQUEST	Clock gate turn on request from GPC standby Status bit indication clock gate is requested to turned on from GPC. 0b - No request 1b - Clock gate requested to be turned on
30 STANDBY_OUT_DONE	Clock gate turn on finish from GPC standby Status bit indication clock gate is requested to turned on from GPC. 0b - Request to turn on Clock gate is not complete 1b - Request to turn on Clock gate is complete
29 STANDBY_IN_DONE	Clock source turn off finish from GPC standby Status bit indication clock source is turned off according to GPC request. 0b - Clock source is not turned off 1b - Clock source is turned off
28 STANDBY_IN_REQUEST	Clock gate turn off request from GPC standby Status bit indication clock gate is requested to turned off from GPC. 0b - No request 1b - Clock gate requested to be turned off
27 SETPOINT_ON_DONE	Clock gate turn on finish from GPC Setpoint Status bit indication clock gate is requested to turned on from GPC. 0b - No request 1b - Request to turn on clock gate
26 SETPOINT_ON_REQUEST	Clock gate turn on request from GPC Setpoint Status bit indication clock gate is requested to turned on from GPC. 0b - No request 1b - Clock gate requested to be turned on
25 SETPOINT_OFF_DONE	Clock source turn off finish from GPC Setpoint Status bit indication clock source is turned off according to GPC request. 0b - Clock source is not turned off 1b - Clock source is turned off
24 SETPOINT_OFF_REQUEST	Clock gate turn off request from GPC Setpoint Status bit indication clock gate is requested to turned off from GPC. 0b - No request 1b - Clock gate requested to be turned off
23-20 CURRENT_SETPOINT	Current Setpoint This is the Setpoint value the Soc is current working in
19-16 TARGET_SETPOINT	Next Setpoint to change to This is the Setpoint value the SoC will switch to
15 CPU3_MODE_DONE	Domain3 Low Power Mode task done Domain3 response signal status to GPC to indicate clock was gated-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
14	Domain3 request enter Low Power Mode

Table continues on the next page...

Field	Function
CPU3_MODE_REQUEST	Domain3 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
13-12 CPU3_MODE	Domain3 Low Power Mode Domain3 will enter in Low Power Mode. 00b - Run 01b - Wait 10b - Stop 11b - Suspend
11 CPU2_MODE_DONE	Domain2 Low Power Mode task done Domain2 response signal status to GPC to indicate clock was gate-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
10 CPU2_MODE_REQUEST	Domain2 request enter Low Power Mode Domain2 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
9-8 CPU2_MODE	Domain2 Low Power Mode Domain2 will enter in Low Power Mode. 00b - Run 01b - Wait 10b - Stop 11b - Suspend
7 CPU1_MODE_DONE	Domain1 Low Power Mode task done Domain1 response signal status to GPC to indicate clock was gated-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
6 CPU1_MODE_REQUEST	Domain1 request enter Low Power Mode Domain1 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
5-4 CPU1_MODE	Domain1 Low Power Mode Domain1 will enter in Low Power Mode 00b - Run 01b - Wait 10b - Stop 11b - Suspend
3 CPU0_MODE_DONE	Domain0 Low Power Mode task done Domain0 response signal status to GPC to indicate clock was gate-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
2 CPU0_MODE_REQUEST	Domain0 request enter Low Power Mode Domain0 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
1-0	Domain0 Low Power Mode

Field	Function
CPU0_MODE	Domain0 will enter in Low Power Mode 00b - Run 01b - Wait 10b - Stop 11b - Suspend

15.9.1.23 Clock source configuration (OSCPLL0_CONFIG - OSCPLL28_CONFIG)

15.9.1.23.1 Offset

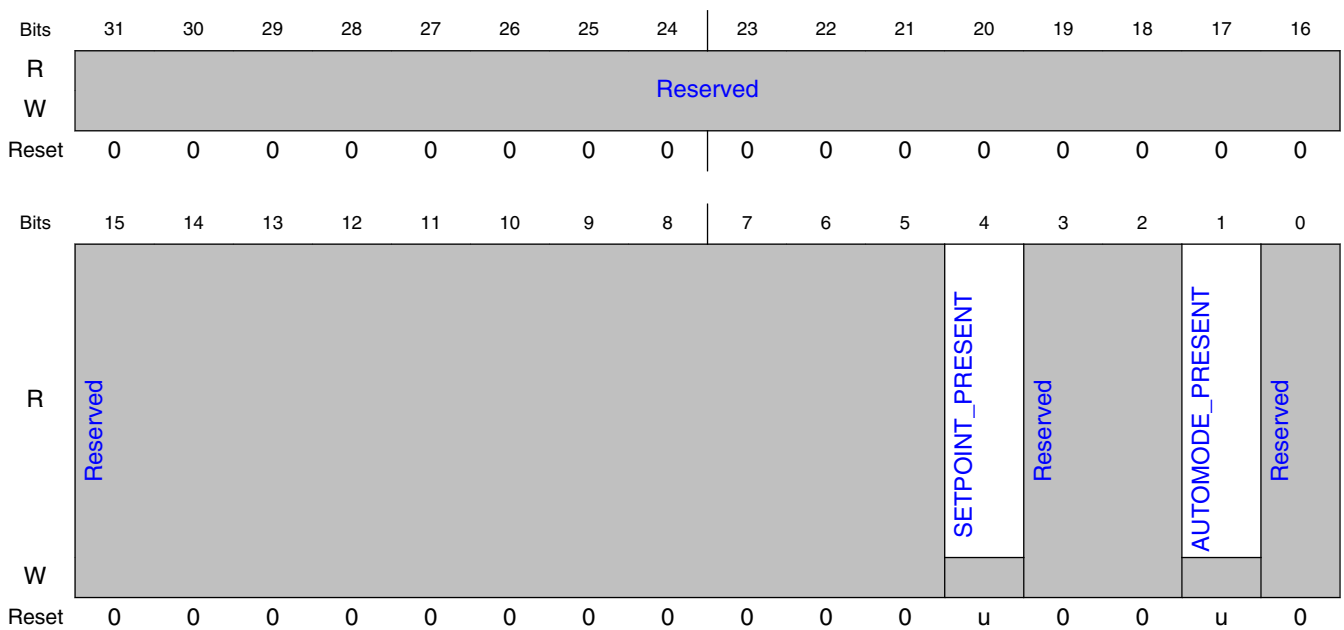
For a = 0 to 28:

Register	Offset
OSCPLLa_CONFIG	5018h + (a × 20h)

15.9.1.23.2 Function

This register shows integration parameters.

15.9.1.23.3 Diagram



15.9.1.23.4 Fields

Field	Function
31-5 —	Reserved
4 SETPOINT_PRESENT	Setpoint present This bit indicate whether this clock source implement Setpoint control scheme. 0b - Setpoint is not implemented. 1b - Setpoint is implemented.
3-2 —	Reserved
1 AUTOMODE_PRESENT	Automode Present 0b - Not present 1b - Present
0 —	Reserved

15.9.1.24 Clock source access control (OSCPLL0_AUTHEN - OSCPLL28_AUTHEN)

15.9.1.24.1 Offset

For a = 0 to 28:

Register	Offset
OSCPLLa_AUTHEN	501Ch + (a × 20h)

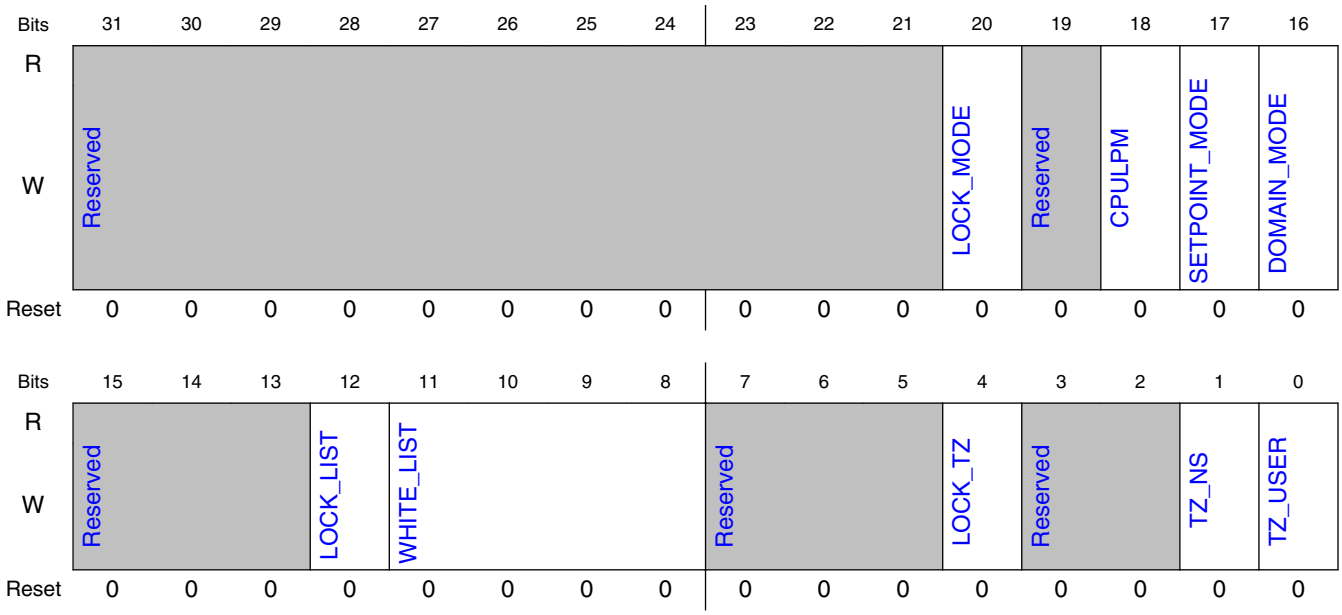
15.9.1.24.2 Function

This register that manages clock source access control.

NOTE

SETPOINT_MODE, DOMAIN_MODE, and CPULPM cannot be set to '1' at the same time.

15.9.1.24.3 Diagram



15.9.1.24.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked. 1b - MODE is locked.
19 —	Reserved
18 CPULPM	CPU Low Power Mode PLL works in CPU Low Power Mode 0b - PLL does not function in Low power Mode 1b - PLL functions in Low Power Mode
17 SETPOINT_MODE	LPCG works in Setpoint controlled Mode. Clock source works in Setpoint controlled Mode.
16 DOMAIN_MODE	Low power and access control by domain Clock source works in Domain controlled Mode. 0b - Clock does not work in Domain Mode. 1b - Clock works in Domain Mode.
15-13 —	Reserved

Table continues on the next page...

Field	Function
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, OSCPLLx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked. 1b - Whitelist is locked.
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this clock source. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively.
7-5 —	Reserved
4 LOCK_TZ	lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked. 1b - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This clock source can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode. 1b - Can be changed in Non-secure mode.
0 TZ_USER	User access This clock source can be changed when CPU is in user mode (in CMx core). 0b - Clock cannot be changed in user mode. 1b - Clock can be changed in user mode.

15.9.1.25 LPCG direct control (LPCG0_DIRECT - LPCG137_DIRECT)

15.9.1.25.1 Offset

For a = 0 to 137:

Register	Offset
LPCGa_DIRECT	6000h + (a × 20h)

15.9.1.25.2 Function

This register controls LPCG on and off when LPCG works in Unassigned Mode and Domain Mode.

15.9.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															ON
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

15.9.1.25.4 Fields

Field	Function
31-1 —	Reserved
0 ON	LPCG on This bit controls LPCG. 0b - LPCG is OFF. 1b - LPCG is ON.

15.9.1.26 LPCG domain control (LPCG0_DOMAIN - LPCG137_DOMAIN)

15.9.1.26.1 Offset

For a = 0 to 137:

Register	Offset
LPCGa_DOMAIN	6004h + (a × 20h)

15.9.1.26.2 Function

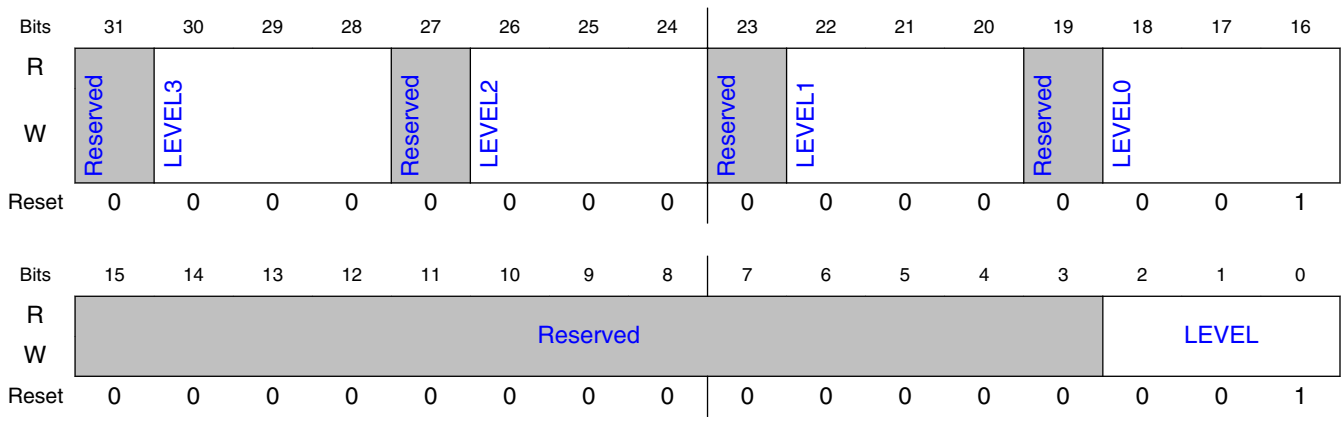
This register controls LPCG on and off when LPCG works in CPU Low Power Mode.

During CPULPM mode, write to bit field LPCGn_DOMAIN[LEVEL], and during Unassigned mode, write to LPCGn_DOMAIN[LEVELx], where x = 0,1,2 or 3. See the table below for more details.

Table 15-9. CCM Mode and access types

CCM Mode/ Access	Unassigned Mode		CPU Low Power Mode		Other Modes	
	Read	Write	Read	Write	Read	Write
LEVEL	Y	N	Y	Y	Y	N
LEVEL0	Y	Y	Y	N	Y	N
LEVEL1	Y	Y	Y	N	Y	N
LEVEL2	Y	Y	Y	N	Y	N
LEVEL3	Y	Y	Y	N	Y	N

15.9.1.26.3 Diagram



15.9.1.26.4 Fields

Field	Function
31 —	Reserved
30-28 LEVEL3	Depend level Depend level of this LPCG for DOMAIN3. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
27 —	Reserved

Table continues on the next page...

Field	Function
26-24 LEVEL2	Depend level Depend level of this LPCG for DOMAIN2. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
23 —	Reserved
22-20 LEVEL1	Depend level Depend level of this LPCG for DOMAIN1. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
19 —	Reserved
18-16 LEVEL0	Depend level Depend level of this LPCG for DOMAIN0. 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved
15-3 —	Reserved
2-0 LEVEL	Current dependence level Dependence level of this clock source for the current accessing domain 000b - This clock source is not needed in any mode, and can be turned off 001b - This clock source is needed in RUN mode, but not needed in WAIT, STOP mode 010b - This clock source is needed in RUN and WAIT mode, but not needed in STOP mode 011b - This clock source is needed in RUN, WAIT and STOP mode 100b - This clock source is always on in any mode (including SUSPEND) 101b,110b,111b - Reserved

15.9.1.27 LPCG working status (LPCG0_STATUS0 - LPCG137_STATUS0)

15.9.1.27.1 Offset

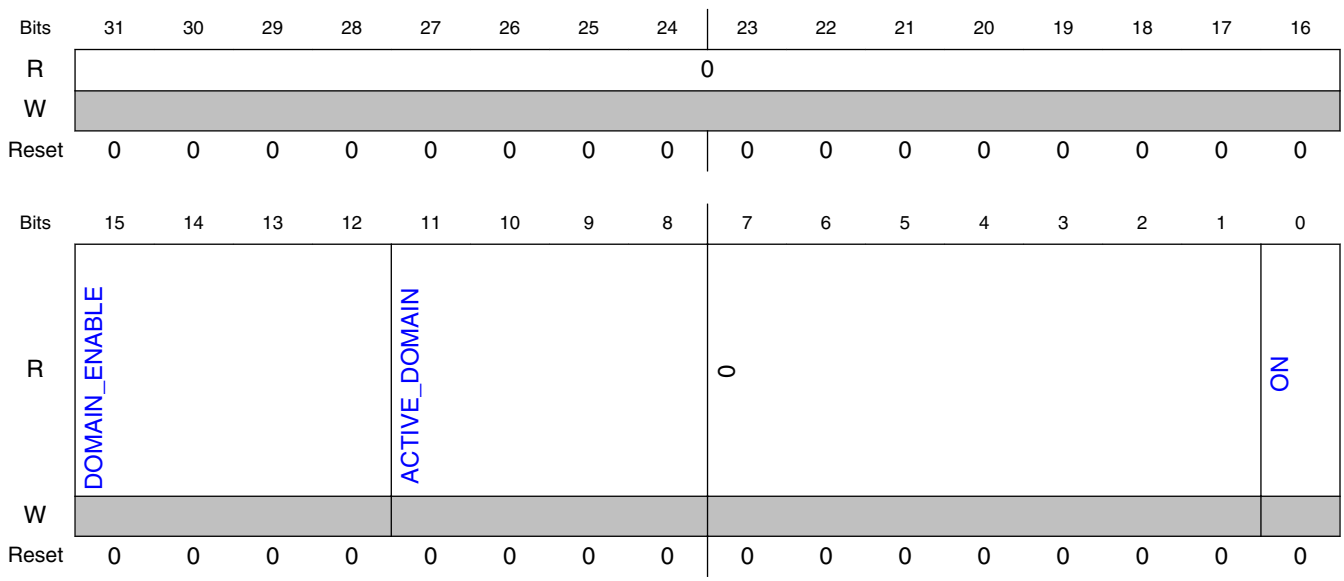
For a = 0 to 137:

Register	Offset
LPCGa_STATUS0	6010h + (a × 20h)

15.9.1.27.2 Function

This register shows current LPCG running status.

15.9.1.27.3 Diagram



15.9.1.27.4 Fields

Field	Function
31-16 —	Reserved
15-12 DOMAIN_ENABLE	Enable status from each domain Enable status from domains, each bit represent one domain. 0000b - No domain request 0001b - Request from Domain0 0010b - Request from Domain1 0011b - Request from Domain0 and Domain1 0100b - Request from Domain2 0101b - Request from Domain0 and Domain2 0110b - Request from Domain1 and Domain2 0111b - Request from Domain0, Domain1 and Domain 2 1000b - Request from Domain3 1001b - Request from Domain0 and Domain3 1010b - Request from Domain1 and Domain3 1011b - Request from Domain2 and Domain3

Table continues on the next page...

Field	Function
	1100b - Request from Domain0, Domain 1, and Domain3 1101b - Request from Domain0, Domain 2, and Domain3 1110b - Request from Domain1, Domain 2, and Domain3 1111b - Request from all domains
11-8 ACTIVE_DOMAIN	Domains that own this clock gate Domains that own this clock gate according to Whitelist. 0000b - Clock not owned by any domain 0001b - Clock owned by Domain0 0010b - Clock owned by Domain1 0011b - Clock owned by Domain0 and Domain1 0100b - Clock owned by Domain2 0101b - Clock owned by Domain0 and Domain2 0110b - Clock owned by Domain1 and Domain2 0111b - Clock owned by Domain0, Domain1 and Domain 2 1000b - Clock owned by Domain3 1001b - Clock owned by Domain0 and Domain3 1010b - Clock owned by Domain1 and Domain3 1011b - Clock owned by Domain2 and Domain3 1100b - Clock owned by Domain0, Domain 1, and Domain3 1101b - Clock owned by Domain0, Domain 2, and Domain3 1110b - Clock owned by Domain1, Domain 2, and Domain3 1111b - Clock owned by all domains
7-1 —	Reserved
0 ON	LPCG current state LPCG running status. 0b - LPCG is OFF. 1b - LPCG is ON.

15.9.1.28 LPCG low power status (LPCG0_STATUS1 - LPCG137_STATUS1)

15.9.1.28.1 Offset

For a = 0 to 137:

Register	Offset
LPCGa_STATUS1	6014h + (a × 20h)

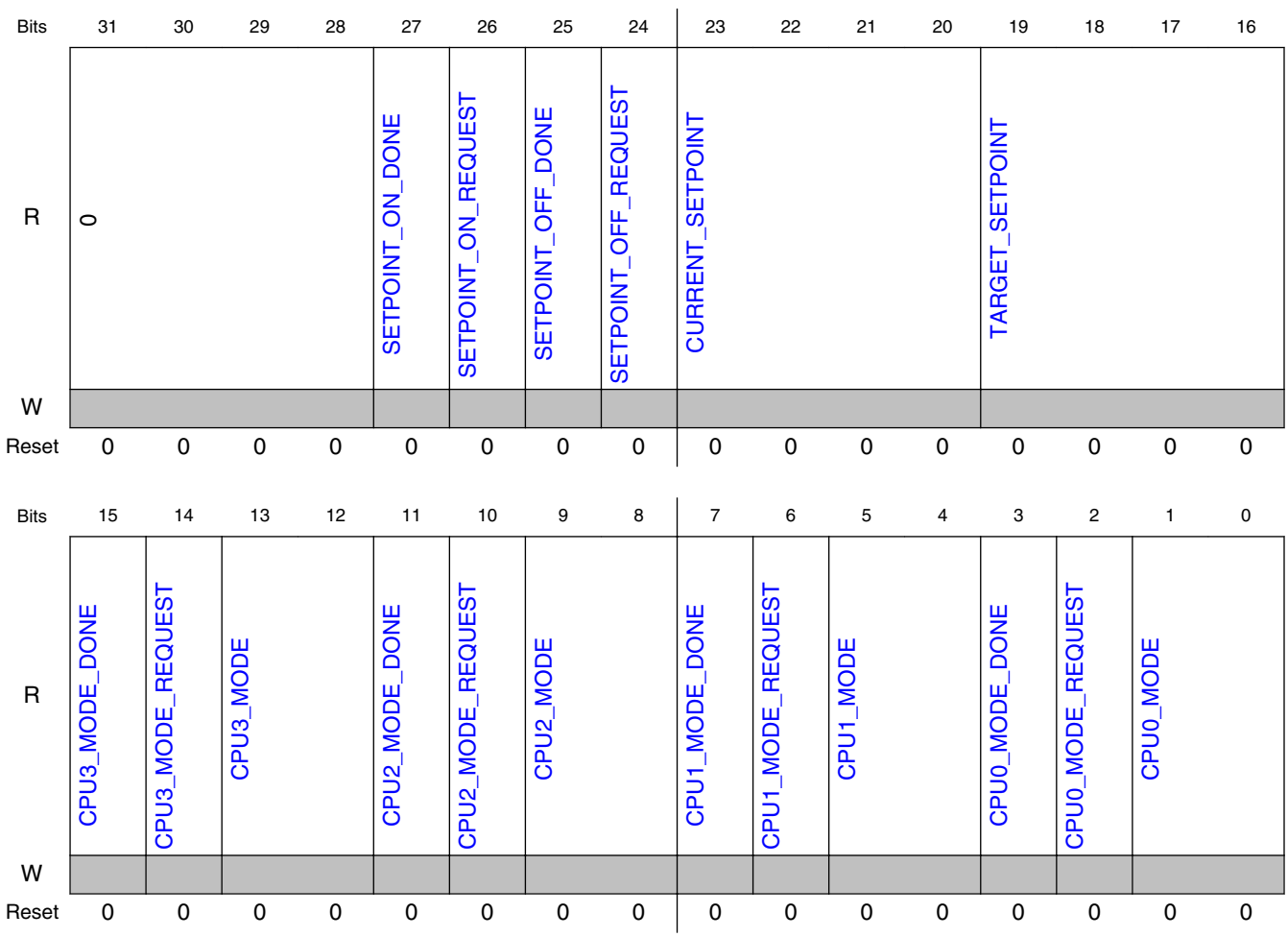
15.9.1.28.2 Function

This register shows low power status of LPCG.

NOTE

The CPU0-3 references are to the CPU domain assignment, not the number of supported CPU platforms. Each CPU platform can be assigned to any domain, including the same domain.

15.9.1.28.3 Diagram



15.9.1.28.4 Fields

Field	Function
31-28 —	Reserved
27 SETPOINT_ON_DONE	Clock gate turn on finish from GPC Setpoint Status bit indication clock gate is turned on according to GPC request. 0b - Clock gate is not turned on 1b - Clock gate is turned on

Table continues on the next page...

Field	Function
26 SETPOINT_ON_REQUEST	Clock gate turn on request from GPC Setpoint Status bit indication clock gate is requested to turned on from GPC. 0b - No request 1b - Clock gate requested to be turned on
25 SETPOINT_OFF_DONE	Clock gate turn off finish from GPC Setpoint Status bit indication clock gate is turned off according to GPC request. 0b - Clock gate is not turned off 1b - Clock gate is turned off
24 SETPOINT_OFF_REQUEST	Clock gate turn off request from GPC Setpoint Status bit indication clock gate is requested to turned off from GPC. 0b - No request 1b - Clock gate requested to be turned off
23-20 CURRENT_SETPOINT	Current Setpoint This is the Setpoint value the Soc is current working in
19-16 TARGET_SETPOINT	Next Setpoint to change to This is the Setpoint value the SoC will switch to
15 CPU3_MODE_DONE	Domain3 Low Power Mode task done Domain3 response signal status to GPC to indicate clock was gate-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
14 CPU3_MODE_REQUEST	Domain3 request enter Low Power Mode Domain3 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
13-12 CPU3_MODE	Domain3 Low Power Mode Domain3 will enter in Low Power Mode. 00b - Run 01b - Wait 10b - Stop 11b - Suspend
11 CPU2_MODE_DONE	Domain2 Low Power Mode task done Domain2 response signal status to GPC to indicate clock was gate-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
10 CPU2_MODE_REQUEST	Domain2 request enter Low Power Mode Domain2 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
9-8 CPU2_MODE	Domain2 Low Power Mode Domain2 will enter in Low Power Mode 00b - Run 01b - Wait 10b - Stop 11b - Suspend

Table continues on the next page...

Field	Function
7 CPU1_MODE_DONE	Domain1 Low Power Mode task done Domain1 response signal status to GPC to indicate clock was gate-off if needed to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
6 CPU1_MODE_REQUEST	Domain1 request enter Low Power Mode Domain1 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
5-4 CPU1_MODE	Domain1 Low Power Mode Domain1 will enter in Low Power Mode 00b - Run 01b - Wait 10b - Stop 11b - Suspend
3 CPU0_MODE_DONE	Domain0 Low Power Mode task done Domain0 response signal status to GPC to indicate clock was gate-off to enter Low Power Mode 0b - Clock is not gated 1b - Clock is gated-off
2 CPU0_MODE_REQUEST	Domain0 request enter Low Power Mode Domain0 request signal status to enter Low Power Mode 0b - No request 1b - Request from domain to enter Low Power Mode
1-0 CPU0_MODE	Domain0 Low Power Mode Domain0 will enter in Low Power Mode 00b - Run 01b - Wait 10b - Stop 11b - Suspend

15.9.1.29 LPCG configuration (LPCG0_CONFIG - LPCG137_CONFIG)

15.9.1.29.1 Offset

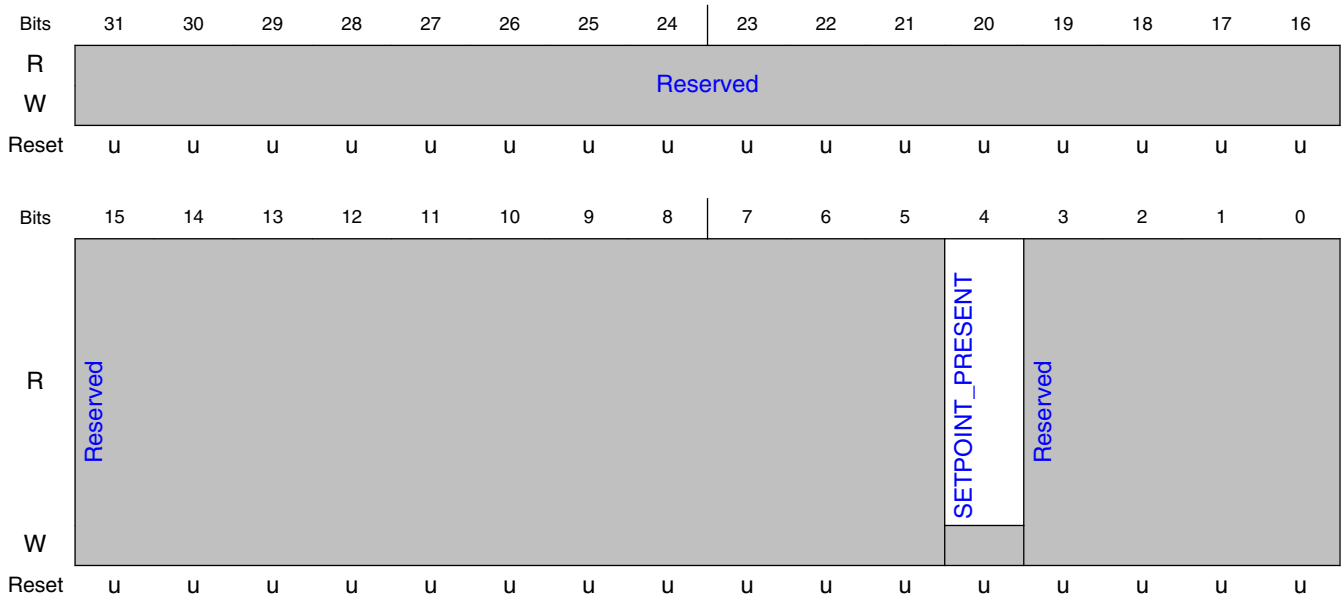
For a = 0 to 137:

Register	Offset
LPCGa_CONFIG	6018h + (a × 20h)

15.9.1.29.2 Function

This register shows integration parameters.

15.9.1.29.3 Diagram



15.9.1.29.4 Fields

Field	Function
31-5 —	Reserved
4 SETPOINT_PRESENT	Setpoint present This bit indicate whether this clock root implement Setpoint control scheme. 0b - Setpoint is not implemented. 1b - Setpoint is implemented.
3-0 —	Reserved

15.9.1.30 LPCG access control (LPCG0_AUTHEN - LPCG137_AUTHEN)

15.9.1.30.1 Offset

For a = 0 to 137:

Register	Offset
LPCGa_AUTHEN	601Ch + (a × 20h)

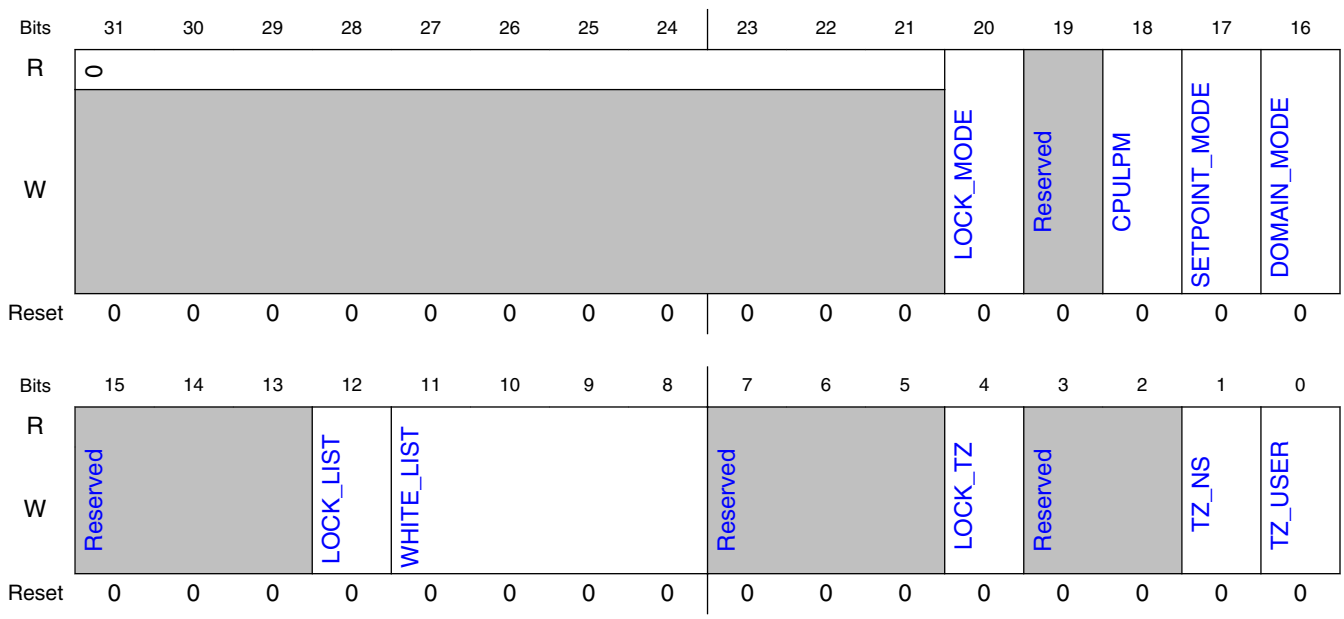
15.9.1.30.2 Function

This register that manages LPCG access control.

NOTE

SETPOINT_MODE, DOMAIN_MODE, and CPULPM cannot be set to '1' at the same time.

15.9.1.30.3 Diagram



15.9.1.30.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, bits 16-20 cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - MODE is not locked. 1b - MODE is locked.
19 —	Reserved
18 CPULPM	CPU Low Power Mode LPCG works in CPU Low Power Mode 0b - LPCG is not functioning in Low power Mode

Table continues on the next page...

Field	Function
	1b - LPCG is functioning in Low Power Mode
17 SETPOINT_MODE	Low power and access control by Setpoint 0b - LPCG is not functioning in Setpoint controlled Mode 1b - LPCG is functioning in Setpoint controlled Mode
16 DOMAIN_MODE	Low power and access control by domain LPCG works in Domain controlled Mode. 0b - Clock does not work in Domain Mode 1b - Clock works in Domain Mode
15-13 —	Reserved
12 LOCK_LIST	Lock Whitelist This bit lock Whitelist. When this bit is set, LPCGx_AUTHEN[WHITE_LIST] cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Whitelist is not locked. 1b - Whitelist is locked.
11-8 WHITE_LIST	Whitelist Domains that on the Whitelist can change this clock root. Each field in this field represent for one domain. Bit8~Bit11 represent for DOMAIN0~DOMAIN3 respectively.
7-5 —	Reserved
4 LOCK_TZ	lock truszone setting This bit lock Trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0b - Trustzone setting is not locked. 1b - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This LPCG can be changed when CPU is in Non-secure mode. 0b - Cannot be changed in Non-secure mode. 1b - Can be changed in Non-secure mode.
0 TZ_USER	User access This LPCG can be changed when CPU is in user mode (in CMx core). 0b - LPCG cannot be changed in user mode. 1b - LPCG can be changed in user mode.

15.9.1.31 LPCG Setpoint setting (LPCG2_SETPOINT - LPCG48_SETPOINT)

15.9.1.31.1 Offset

Register	Offset
LPCG2_SETPOINT	6048h
LPCG3_SETPOINT	6068h
LPCG4_SETPOINT	6088h
LPCG5_SETPOINT	60A8h
LPCG6_SETPOINT	60C8h
LPCG7_SETPOINT	60E8h
LPCG8_SETPOINT	6108h
LPCG9_SETPOINT	6128h
LPCG10_SETPOINT	6148h
LPCG11_SETPOINT	6168h
LPCG12_SETPOINT	6188h
LPCG14_SETPOINT	61C8h
LPCG15_SETPOINT	61E8h
LPCG16_SETPOINT	6208h
LPCG17_SETPOINT	6228h
LPCG18_SETPOINT	6248h
LPCG19_SETPOINT	6268h
LPCG24_SETPOINT	6308h
LPCG25_SETPOINT	6328h
LPCG26_SETPOINT	6348h
LPCG27_SETPOINT	6368h
LPCG28_SETPOINT	6388h
LPCG29_SETPOINT	63A8h
LPCG30_SETPOINT	63C8h
LPCG31_SETPOINT	63E8h
LPCG32_SETPOINT	6408h
LPCG33_SETPOINT	6428h
LPCG34_SETPOINT	6448h
LPCG35_SETPOINT	6468h
LPCG36_SETPOINT	6488h
LPCG37_SETPOINT	64A8h
LPCG38_SETPOINT	64C8h
LPCG39_SETPOINT	64E8h
LPCG40_SETPOINT	6508h
LPCG43_SETPOINT	6568h
LPCG44_SETPOINT	6588h
LPCG45_SETPOINT	65A8h
LPCG46_SETPOINT	65C8h
LPCG47_SETPOINT	65E8h
LPCG48_SETPOINT	6608h

15.9.1.31.2 Function

This register defines 16 Setpoint values when LPCG works in Setpoint Mode.

15.9.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STANDBY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SETPOINT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.1.31.4 Fields

Field	Function
31-16 STANDBY	Standby This field defines 16 Setpoint standby values. Bit0~Bit15 hold value for Setpoint 0~16 standby respectively. A bitfield value of 0 implies the LPCG will be shutdown during standby. A bitfield value of 1 represent LPCG will keep Setpoint setting during standby.
15-0 SETPOINT	Setpoints This field defines 16 Setpoint values. Bit0~Bit15 hold value for Setpoint 0~16 respectively. A value of 0 implies the clock is off in Setpoint Mode A value of 1 implies the clock is on in Setpoint Mode

15.9.2 CCM_OBS register descriptions

15.9.2.1 D_IP_IMXRT_CCM_OBS_LN28FDSOI memory map

CCM_OBS base address: 4015_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Observe control (OBSERVE0_CONTROL)	32	RW	0000_0000
4	Observe control (OBSERVE0_CONTROL_SET)	32	RW	0000_0000
8	Observe control (OBSERVE0_CONTROL_CLR)	32	RW	0000_0000
C	Observe control (OBSERVE0_CONTROL_TOG)	32	RW	0000_0000
20	Observe status (OBSERVE0_STATUS0)	32	RO	0000_0000
30	Observe access control (OBSERVE0_AUTHEN)	32	RW	0000_0000
34	Observe access control (OBSERVE0_AUTHEN_SET)	32	RW	0000_0000
38	Observe access control (OBSERVE0_AUTHEN_CLR)	32	RW	0000_0000
3C	Observe access control (OBSERVE0_AUTHEN_TOG)	32	RW	0000_0000
40	Current frequency detected (OBSERVE0_FREQUENCY_CURRENT)	32	RO	0000_0000
44	Minimum frequency detected (OBSERVE0_FREQUENCY_MIN)	32	RO	FFFF_FFC0
48	Maximum frequency detected (OBSERVE0_FREQUENCY_MAX)	32	RO	0000_0000
80	Observe control (OBSERVE1_CONTROL)	32	RW	0000_0000
84	Observe control (OBSERVE1_CONTROL_SET)	32	RW	0000_0000
88	Observe control (OBSERVE1_CONTROL_CLR)	32	RW	0000_0000
8C	Observe control (OBSERVE1_CONTROL_TOG)	32	RW	0000_0000
A0	Observe status (OBSERVE1_STATUS0)	32	RO	0000_0000
B0	Observe access control (OBSERVE1_AUTHEN)	32	RW	0000_0000
B4	Observe access control (OBSERVE1_AUTHEN_SET)	32	RW	0000_0000
B8	Observe access control (OBSERVE1_AUTHEN_CLR)	32	RW	0000_0000
BC	Observe access control (OBSERVE1_AUTHEN_TOG)	32	RW	0000_0000
C0	Current frequency detected (OBSERVE1_FREQUENCY_CURRENT)	32	RO	0000_0000
C4	Minimum frequency detected (OBSERVE1_FREQUENCY_MIN)	32	RO	FFFF_FFC0
C8	Maximum frequency detected (OBSERVE1_FREQUENCY_MAX)	32	RO	0000_0000
100	Observe control (OBSERVE2_CONTROL)	32	RW	0000_0000
104	Observe control (OBSERVE2_CONTROL_SET)	32	RW	0000_0000
108	Observe control (OBSERVE2_CONTROL_CLR)	32	RW	0000_0000
10C	Observe control (OBSERVE2_CONTROL_TOG)	32	RW	0000_0000
120	Observe status (OBSERVE2_STATUS0)	32	RO	0000_0000
130	Observe access control (OBSERVE2_AUTHEN)	32	RW	0000_0000
134	Observe access control (OBSERVE2_AUTHEN_SET)	32	RW	0000_0000
138	Observe access control (OBSERVE2_AUTHEN_CLR)	32	RW	0000_0000
13C	Observe access control (OBSERVE2_AUTHEN_TOG)	32	RW	0000_0000
140	Current frequency detected (OBSERVE2_FREQUENCY_CURRENT)	32	RO	0000_0000
144	Minimum frequency detected (OBSERVE2_FREQUENCY_MIN)	32	RO	FFFF_FFC0
148	Maximum frequency detected (OBSERVE2_FREQUENCY_MAX)	32	RO	0000_0000
180	Observe control (OBSERVE3_CONTROL)	32	RW	0000_0000
184	Observe control (OBSERVE3_CONTROL_SET)	32	RW	0000_0000
188	Observe control (OBSERVE3_CONTROL_CLR)	32	RW	0000_0000
18C	Observe control (OBSERVE3_CONTROL_TOG)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
1A0	Observe status (OBSERVE3_STATUS0)	32	RO	0000_0000
1B0	Observe access control (OBSERVE3_AUTHEN)	32	RW	0000_0000
1B4	Observe access control (OBSERVE3_AUTHEN_SET)	32	RW	0000_0000
1B8	Observe access control (OBSERVE3_AUTHEN_CLR)	32	RW	0000_0000
1BC	Observe access control (OBSERVE3_AUTHEN_TOG)	32	RW	0000_0000
1C0	Current frequency detected (OBSERVE3_FREQUENCY_CURRENT)	32	RO	0000_0000
1C4	Minimum frequency detected (OBSERVE3_FREQUENCY_MIN)	32	RO	FFFF_FFC0
1C8	Maximum frequency detected (OBSERVE3_FREQUENCY_MAX)	32	RO	0000_0000
200	Observe control (OBSERVE4_CONTROL)	32	RW	0000_0000
204	Observe control (OBSERVE4_CONTROL_SET)	32	RW	0000_0000
208	Observe control (OBSERVE4_CONTROL_CLR)	32	RW	0000_0000
20C	Observe control (OBSERVE4_CONTROL_TOG)	32	RW	0000_0000
220	Observe status (OBSERVE4_STATUS0)	32	RO	0000_0000
230	Observe access control (OBSERVE4_AUTHEN)	32	RW	0000_0000
234	Observe access control (OBSERVE4_AUTHEN_SET)	32	RW	0000_0000
238	Observe access control (OBSERVE4_AUTHEN_CLR)	32	RW	0000_0000
23C	Observe access control (OBSERVE4_AUTHEN_TOG)	32	RW	0000_0000
240	Current frequency detected (OBSERVE4_FREQUENCY_CURRENT)	32	RO	0000_0000
244	Minimum frequency detected (OBSERVE4_FREQUENCY_MIN)	32	RO	FFFF_FFC0
248	Maximum frequency detected (OBSERVE4_FREQUENCY_MAX)	32	RO	0000_0000
280	Observe control (OBSERVE5_CONTROL)	32	RW	0000_0000
284	Observe control (OBSERVE5_CONTROL_SET)	32	RW	0000_0000
288	Observe control (OBSERVE5_CONTROL_CLR)	32	RW	0000_0000
28C	Observe control (OBSERVE5_CONTROL_TOG)	32	RW	0000_0000
2A0	Observe status (OBSERVE5_STATUS0)	32	RO	0000_0000
2B0	Observe access control (OBSERVE5_AUTHEN)	32	RW	0000_0000
2B4	Observe access control (OBSERVE5_AUTHEN_SET)	32	RW	0000_0000
2B8	Observe access control (OBSERVE5_AUTHEN_CLR)	32	RW	0000_0000
2BC	Observe access control (OBSERVE5_AUTHEN_TOG)	32	RW	0000_0000
2C0	Current frequency detected (OBSERVE5_FREQUENCY_CURRENT)	32	RO	0000_0000
2C4	Minimum frequency detected (OBSERVE5_FREQUENCY_MIN)	32	RO	FFFF_FFC0
2C8	Maximum frequency detected (OBSERVE5_FREQUENCY_MAX)	32	RO	0000_0000

15.9.2.2 Observe control (OBSERVE0_CONTROL - OBSERVE5_CONTROL)

Register that control observe.

15.9.2.2.1 Offset

For a = 0 to 5:

Register	Offset	Description
OBSERVEa_CONTROL	0h + (a × 80h)	Observe control
OBSERVEa_CONTROL_SET	4h + (a × 80h)	Writing a 1 to a bit in this register sets the corresponding bit in OBSERVEa_CONTROL
OBSERVEa_CONTROL_CLR	8h + (a × 80h)	Writing a 1 to a bit in this register clears the corresponding bit in OBSERVEa_CONTROL
OBSERVEa_CONTROL_TOG	Ch + (a × 80h)	Writing a 1 to a bit in this register toggles the corresponding bit in OBSERVEa_CONTROL

15.9.2.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							OFF	DIVIDE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RESET	Reserved	INV	RAW	Reserved			SELECT								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.2.2.3 Fields

Field	Description
31-25 —	Reserved
24 OFF	Turn off Turn off slice to save power. 0 - observe slice is on 1 - observe slice is off
23-16 DIVIDE	Divider for observe signal Divider before measurement or send out to IO. Divider by DIVIDE+1. **NOTE:** If divider selected, first several toggles may not be observed. Divider is typically be used when observe signal is a continuous clock.
15	Reset observe divider

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RESET</td><td>A change from 0 to 1 will reset the divider.
0 - No reset
1 - Reset observe divider</td></tr>
<tr><td>14
—</td><td>Reserved</td></tr>
<tr><td>13
INV</td><td>Invert
Invert input signal phase.
0 - Clock phase remain same.
1 - Invert clock phase before measurement or send to IO.</td></tr>
<tr><td>12
RAW</td><td>Observe raw signal
Select from raw signal and divided signal.
0 - Select divided signal.
1 - Select raw signal.</td></tr>
<tr><td>11-9
—</td><td>Reserved</td></tr>
<tr><td>8-0
SELECT</td><td>Observe signal selector
Selector that controls which observe signals will be selected and observed.
<table><tr><th>Target</th><th>SELECT index</th><th>Slice Number</th></tr><tr><td>M7_CLK_ROOT</td><td>128</td><td>4</td></tr><tr><td>M4_CLK_ROOT</td><td>129</td><td>0</td></tr><tr><td>BUS_CLK_ROOT</td><td>130</td><td>2</td></tr><tr><td>BUS_LPSR_CLK_ROOT</td><td>131</td><td>0</td></tr><tr><td>SEMC_CLK_ROOT</td><td>132</td><td>2</td></tr><tr><td>CSSYS_CLK_ROOT</td><td>133</td><td>2</td></tr><tr><td>CSTRACE_CLK_ROOT</td><td>134</td><td>2</td></tr><tr><td>M4_SYSTICK_CLK_ROOT</td><td>135</td><td>0</td></tr><tr><td>M7_SYSTICK_CLK_ROOT</td><td>136</td><td>2</td></tr><tr><td>ADC1_CLK_ROOT</td><td>137</td><td>2</td></tr><tr><td>ADC2_CLK_ROOT</td><td>138</td><td>2</td></tr><tr><td>ACMP_CLK_ROOT</td><td>139</td><td>2</td></tr><tr><td>FLEXIO1_CLK_ROOT</td><td>140</td><td>2</td></tr><tr><td>FLEXIO2_CLK_ROOT</td><td>141</td><td>2</td></tr><tr><td>GPT1_CLK_ROOT</td><td>142</td><td>2</td></tr><tr><td>GPT2_CLK_ROOT</td><td>143</td><td>2</td></tr><tr><td>GPT3_CLK_ROOT</td><td>144</td><td>2</td></tr><tr><td>GPT4_CLK_ROOT</td><td>145</td><td>2</td></tr><tr><td>GPT5_CLK_ROOT</td><td>146</td><td>2</td></tr><tr><td>GPT6_CLK_ROOT</td><td>147</td><td>2</td></tr><tr><td>FLEXSPI1_CLK_ROOT</td><td>148</td><td>2</td></tr></table></td></tr>
</table>

Field	Description		
	Target	**SELECT index**	**Slice Number**
	FLEXSPI2_CLK_ROOT	149	2
	CAN1_CLK_ROOT	150	2
	CAN2_CLK_ROOT	151	2
	CAN3_CLK_ROOT	152	0
	LPUART1_CLK_ROOT	153	2
	LPUART2_CLK_ROOT	154	2
	LPUART3_CLK_ROOT	155	2
	LPUART4_CLK_ROOT	156	2
	LPUART5_CLK_ROOT	157	2
	LPUART6_CLK_ROOT	158	2
	LPUART7_CLK_ROOT	159	2
	LPUART8_CLK_ROOT	160	2
	LPUART9_CLK_ROOT	161	2
	LPUART10_CLK_ROOT	162	2
	LPUART11_CLK_ROOT	163	0
	LPUART12_CLK_ROOT	164	0
	LPI2C1_CLK_ROOT	165	2
	LPI2C2_CLK_ROOT	166	2
	LPI2C3_CLK_ROOT	167	2
	LPI2C4_CLK_ROOT	168	2
	LPI2C5_CLK_ROOT	169	0
	LPI2C6_CLK_ROOT	170	0
	LPSPI1_CLK_ROOT	171	2
	LPSPI2_CLK_ROOT	172	2
	LPSPI3_CLK_ROOT	173	2
	LPSPI4_CLK_ROOT	174	2
	LPSPI5_CLK_ROOT	175	0
	LPSPI6_CLK_ROOT	176	0
	EMV1_CLK_ROOT	177	2
	EMV2_CLK_ROOT	178	2
	ENET1_CLK_ROOT	179	2
	ENET2_CLK_ROOT	180	2
	ENET_QOS_CLK_ROOT	181	2
	ENET_25M_CLK_ROOT	182	2
	ENET_TIMER1_CLK_ROOT	183	2
	ENET_TIMER2_CLK_ROOT	184	2
	ENET_TIMER3_CLK_ROOT	185	2
	USDHC1_CLK_ROOT	186	2
	USDHC2_CLK_ROOT	187	2
	ASRC_CLK_ROOT	188	2

Field	Description		
	Target	**SELECT index**	**Slice Number**
	MQS_CLK_ROOT	189	2
	MIC_CLK_ROOT	190	0
	SPDIF_CLK_ROOT	191	2
	SAI1_CLK_ROOT	192	2
	SAI2_CLK_ROOT	193	2
	SAI3_CLK_ROOT	194	2
	SAI4_CLK_ROOT	195	0
	GPU2D_CLK_ROOT	196	2
	ELCDIF_CLK_ROOT	197	2
	LCDIFV2_CLK_ROOT	198	2
	MIPI_REF_CLK_ROOT	199	2
	MIPI_ESC_CLK_ROOT	200	2
	CSI2_CLK_ROOT	201	2
	CSI2_ESC_CLK_ROOT	202	2
	CSI2_UI_CLK_ROOT	203	2
	CSI_CLK_ROOT	204	2
	CCM_CKO1_CLK_ROOT	205	0
	CCM_CKO2_CLK_ROOT	206	2
	CM7_CORE_STCLKEN	207	4
	CCM_FLEXRAM_CLK_ROOT	208	4
	MIPI_DSI_TXESC	209	2
	MIPI_DSI_RXESC	210	2
	OSC_RC_16M	224	0
	OSC_RC_48M	225	0
	OSC_RC_48M_DIV2	226	0
	OSC_RC_400M	227	0
	OSC_24M_OUT	229	0
	PLL_ARM_OUT	231	2
	SYS_PLL2_OUT	233	2
	SYS_PLL2_PFD0	234	2
	SYS_PLL2_PFD1	235	2
	SYS_PLL2_PFD2	236	2
	SYS_PLL2_PFD3	237	2
	SYS_PLL3_OUT	239	2
	SYS_PLL3_DIV2	240	2
	SYS_PLL3_PFD0	241	2
	SYS_PLL3_PFD1	242	2
	SYS_PLL3_PFD2	243	2
	SYS_PLL3_PFD3	244	2
	SYS_PLL1_OUT	246	2

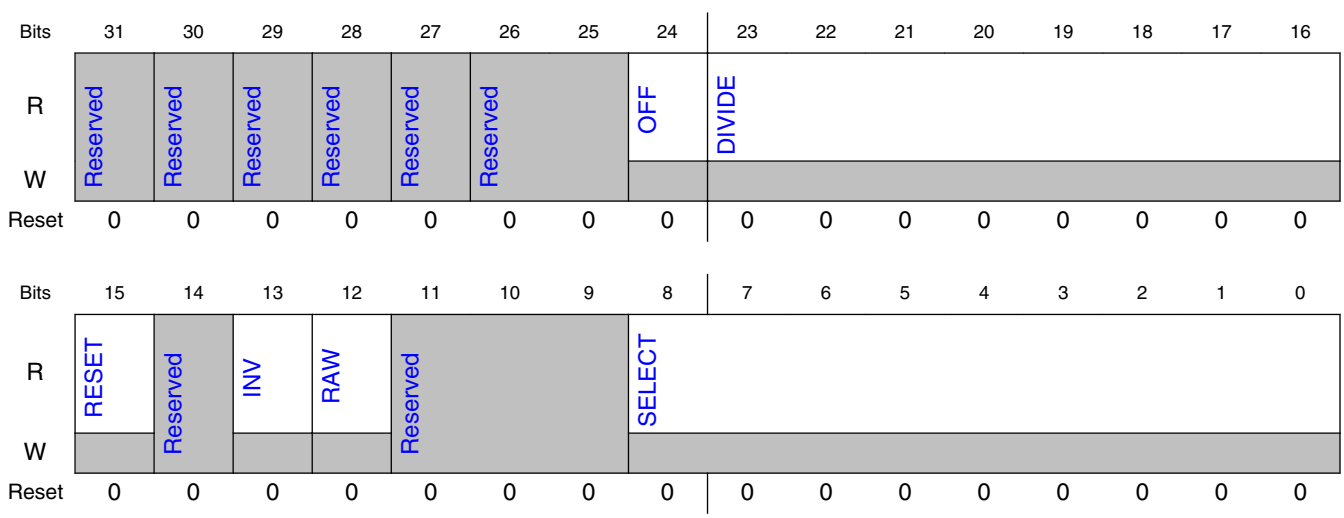
Field	Description		
	Target	**SELECT index**	**Slice Number**
	SYS_PLL1_DIV2	247	2
	SYS_PLL1_DIV5	248	2
	PLL_AUDIO_OUT	250	2
	PLL_VIDEO_OUT	252	2

15.9.2.3 Observe status (OBSERVE0_STATUS0 - OBSERVE5_STATUS0)

15.9.2.3.1 Offset

Register	Offset
OBSERVE0_STATUS0	20h
OBSERVE1_STATUS0	A0h
OBSERVE2_STATUS0	120h
OBSERVE3_STATUS0	1A0h
OBSERVE4_STATUS0	220h
OBSERVE5_STATUS0	2A0h

15.9.2.3.2 Diagram



15.9.2.3.3 Fields

Field	Description
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26-25 —	Reserved
24 OFF	Turn off slice 0 - observe slice is on 1 - observe slice is off
23-16 DIVIDE	Divide value status. The clock will be divided by DIVIDE + 1.
15 RESET	Reset state 0 - Observe divider is not in reset state 1 - Observe divider is in reset state
14 —	Reserved
13 INV	Polarity of the observe target 0 - Polarity is not inverted 1 - Polarity of the observe target is inverted
12 RAW	Observe raw signal 0 - Divided signal is selected 1 - Raw signal is selected
11-9 —	Reserved
8-0 SELECT	Select value

Target	SELECT index	Slice Number
M7_CLK_ROOT	128	4
M4_CLK_ROOT	129	0
BUS_CLK_ROOT	130	2
BUS_LPSR_CLK_ROOT	131	0

Field	Description		
	Target	**SELECT index**	**Slice Number**
	SEMC_CLK_ROOT	132	2
	CSSYS_CLK_ROOT	133	2
	CSTRACE_CLK_ROOT	134	2
	M4_SYSTICK_CLK_ROOT	135	0
	M7_SYSTICK_CLK_ROOT	136	2
	ADC1_CLK_ROOT	137	2
	ADC2_CLK_ROOT	138	2
	ACMP_CLK_ROOT	139	2
	FLEXIO1_CLK_ROOT	140	2
	FLEXIO2_CLK_ROOT	141	2
	GPT1_CLK_ROOT	142	2
	GPT2_CLK_ROOT	143	2
	GPT3_CLK_ROOT	144	2
	GPT4_CLK_ROOT	145	2
	GPT5_CLK_ROOT	146	2
	GPT6_CLK_ROOT	147	2
	FLEXSPI1_CLK_ROOT	148	2
	FLEXSPI2_CLK_ROOT	149	2
	CAN1_CLK_ROOT	150	2
	CAN2_CLK_ROOT	151	2
	CAN3_CLK_ROOT	152	0
	LPUART1_CLK_ROOT	153	2
	LPUART2_CLK_ROOT	154	2
	LPUART3_CLK_ROOT	155	2
	LPUART4_CLK_ROOT	156	2
	LPUART5_CLK_ROOT	157	2
	LPUART6_CLK_ROOT	158	2
	LPUART7_CLK_ROOT	159	2
	LPUART8_CLK_ROOT	160	2
	LPUART9_CLK_ROOT	161	2
	LPUART10_CLK_ROOT	162	2
	LPUART11_CLK_ROOT	163	0
	LPUART12_CLK_ROOT	164	0
	LPI2C1_CLK_ROOT	165	2
	LPI2C2_CLK_ROOT	166	2
	LPI2C3_CLK_ROOT	167	2
	LPI2C4_CLK_ROOT	168	2
	LPI2C5_CLK_ROOT	169	0
	LPI2C6_CLK_ROOT	170	0
	LPSPI1_CLK_ROOT	171	2

Field	Description		
	Target	**SELECT index**	**Slice Number**
	LPSPI2_CLK_ROOT	172	2
	LPSPI3_CLK_ROOT	173	2
	LPSPI4_CLK_ROOT	174	2
	LPSPI5_CLK_ROOT	175	0
	LPSPI6_CLK_ROOT	176	0
	EMV1_CLK_ROOT	177	2
	EMV2_CLK_ROOT	178	2
	ENET1_CLK_ROOT	179	2
	ENET2_CLK_ROOT	180	2
	ENET_QOS_CLK_ROOT	181	2
	ENET_25M_CLK_ROOT	182	2
	ENET_TIMER1_CLK_ROOT	183	2
	ENET_TIMER2_CLK_ROOT	184	2
	ENET_TIMER3_CLK_ROOT	185	2
	USDHC1_CLK_ROOT	186	2
	USDHC2_CLK_ROOT	187	2
	ASRC_CLK_ROOT	188	2
	MQS_CLK_ROOT	189	2
	MIC_CLK_ROOT	190	0
	SPDIF_CLK_ROOT	191	2
	SAI1_CLK_ROOT	192	2
	SAI2_CLK_ROOT	193	2
	SAI3_CLK_ROOT	194	2
	SAI4_CLK_ROOT	195	0
	GPU2D_CLK_ROOT	196	2
	ELCDIF_CLK_ROOT	197	2
	LCDIFV2_CLK_ROOT	198	2
	MIPI_REF_CLK_ROOT	199	2
	MIPI_ESC_CLK_ROOT	200	2
	CSI2_CLK_ROOT	201	2
	CSI2_ESC_CLK_ROOT	202	2
	CSI2_UI_CLK_ROOT	203	2
	CSI_CLK_ROOT	204	2
	CCM_CKO1_CLK_ROOT	205	0
	CCM_CKO2_CLK_ROOT	206	2
	CM7_CORE_STCLKEN	207	4
	CCM_FLEXRAM_CLK_ROOT	208	4
	MIPI_DSI_TXESC	209	2
	MIPI_DSI_RXESC	210	2
	OSC_RC_16M	224	0

Field	Description		
	Target	**SELECT index**	**Slice Number**
	OSC_RC_48M	225	0
	OSC_RC_48M_DIV2	226	0
	OSC_RC_400M	227	0
	OSC_24M_OUT	229	0
	PLL_ARM_OUT	231	2
	SYS_PLL2_OUT	233	2
	SYS_PLL2_PFD0	234	2
	SYS_PLL2_PFD1	235	2
	SYS_PLL2_PFD2	236	2
	SYS_PLL2_PFD3	237	2
	SYS_PLL3_OUT	239	2
	SYS_PLL3_DIV2	240	2
	SYS_PLL3_PFD0	241	2
	SYS_PLL3_PFD1	242	2
	SYS_PLL3_PFD2	243	2
	SYS_PLL3_PFD3	244	2
	SYS_PLL1_OUT	246	2
	SYS_PLL1_DIV2	247	2
	SYS_PLL1_DIV5	248	2
	PLL_AUDIO_OUT	250	2
	PLL_VIDEO_OUT	252	2

15.9.2.4 Observe access control (OBSERVE0_AUTHEN - OBSERVE5_AUTHEN)

This register that manages observe access control.

15.9.2.4.1 Offset

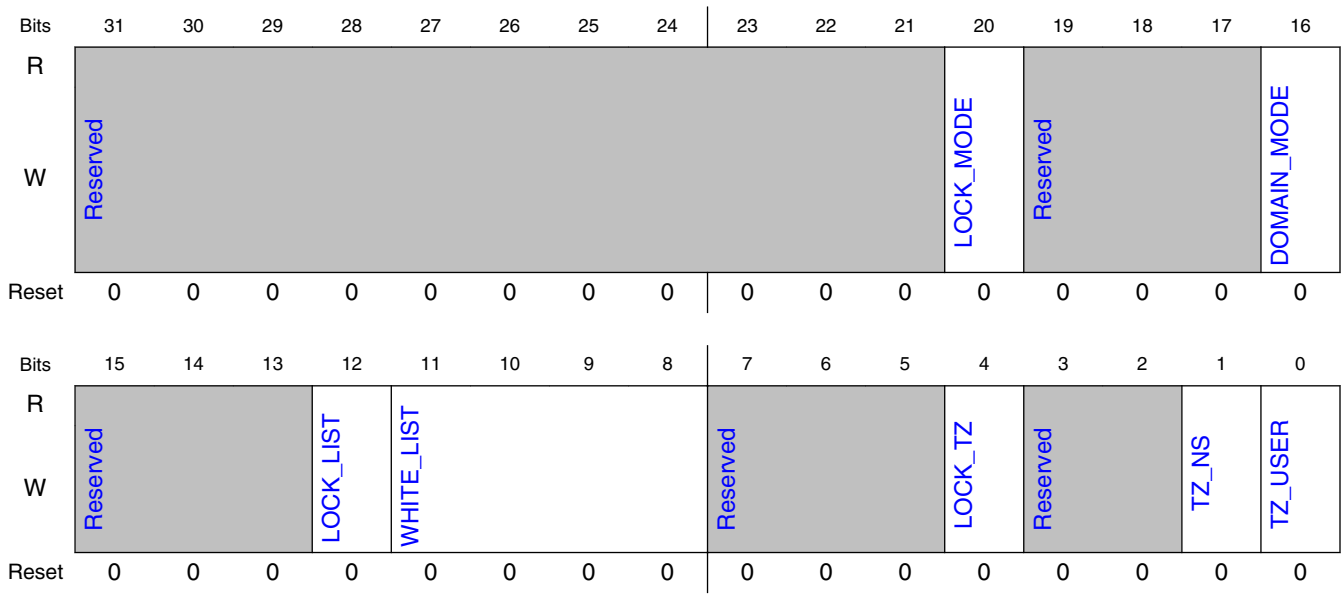
For a = 0 to 5:

Register	Offset	Description
OBSERVEa_AUTHEN	30h + (a × 80h)	Observe access control
OBSERVEa_AUTHEN_SET	34h + (a × 80h)	Writing a 1 to a bit in this register sets the corresponding bit in OBSERVEa_AUTHEN
OBSERVEa_AUTHEN_CLR	38h + (a × 80h)	Writing a 1 to a bit in this register clears the corresponding bit in OBSERVEa_AUTHEN

Table continues on the next page...

Register	Offset	Description
OBSERVEa_AUTHEN_TOG	3Ch + (a × 80h)	Writing a 1 to a bit in this register toggles the corresponding bit in OBSERVEa_AUTHEN

15.9.2.4.2 Diagram



15.9.2.4.3 Fields

Field	Description
31-21 —	Reserved
20 LOCK_MODE	Lock low power and access mode This bit lock low power and access control mode. When this bit is set, DOMAIN_MODE cannot change. Once this bit is set, it cannot be cleared, until next system reset. 0 - MODE is not locked. 1 - MODE is locked.
19-17 —	Reserved
16 DOMAIN_MODE	Low power and access control by domain Observe works in domain controlled mode. 0 - Clock does not work in domain mode. 1 - Clock works in domain mode.
15-13 —	Reserved

Table continues on the next page...

Field	Description
12 LOCK_LIST	Lock white list This bit lock white list. When this bit is set, WHITE_LIST cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - White list is not locked. 1 - White list is locked.
11-8 WHITE_LIST	White list Domains that on the white list can change this observe. Each field in this field represent for one domain. Bit8~Bit12 represent for DOMAIN0~DOMAIN3 respectively. 0000 - No domain can change. 0001 - Domain 0 can change. 0010 - Domain 1 can change. 0011 - Domain 0 and domain 1 can change. 0100 - Domain 2 can change. 1111 - All domain can change.
7-5 —	Reserved
4 LOCK_TZ	Lock truszone setting This bit lock trustzone setting. When this bit is set, TRUSTZONE_USER and TRUSTZONE_NONSECURE cannot be changed. Once this bit is set, it cannot be cleared, until next system reset. 0 - Trustzone setting is not locked. 1 - Trustzone setting is locked.
3-2 —	Reserved
1 TZ_NS	Non-secure access This observe can be changed when CPU is in Non-secure mode. 0 - Cannot be changed in Non-secure mode. 1 - Can be changed in Non-secure mode.
0 TZ_USER	User access This observe can be changed when CPU is in user mode. 0 - Clock cannot be changed in user mode. 1 - Clock can be changed in user mode.

15.9.2.5 Current frequency detected (OBSERVE0_FREQUENCY_CURRENT - OBSERVE5_FREQUENCY_CURRENT)

This register shows the current frequency detected since last update or reset. After the measurement begins, wait for 20ms before reading this register.

NOTE

Frequency related measurement requires 32K crystal clock.

15.9.2.5.1 Offset

Register	Offset
OBSERVE0_FREQUENCY_CURRENT	40h
OBSERVE1_FREQUENCY_CURRENT	C0h
OBSERVE2_FREQUENCY_CURRENT	140h
OBSERVE3_FREQUENCY_CURRENT	1C0h
OBSERVE4_FREQUENCY_CURRENT	240h
OBSERVE5_FREQUENCY_CURRENT	2C0h

15.9.2.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FREQUENCY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FREQUENCY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.2.5.3 Fields

Field	Description
31-0 FREQUENCY	Frequency Current frequency of observed signal in Hz.

15.9.2.6 Minimum frequency detected (OBSERVE0_FREQUENCY_MIN - OBSERVE5_FREQUENCY_MIN)

This register shows the minimum frequency detected since last update or reset. After the measurement begins, wait for 20ms before reading this register.

15.9.2.6.1 Offset

Register	Offset
OBSERVE0_FREQUENCY_MIN	44h
OBSERVE1_FREQUENCY_MIN	C4h
OBSERVE2_FREQUENCY_MIN	144h
OBSERVE3_FREQUENCY_MIN	1C4h
OBSERVE4_FREQUENCY_MIN	244h
OBSERVE5_FREQUENCY_MIN	2C4h

15.9.2.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FREQUENCY															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FREQUENCY															
W																
Reset	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

15.9.2.6.3 Fields

Field	Description
31-0 FREQUENCY	Frequency Minimum frequency of observed signal in Hz.

15.9.2.7 Maximum frequency detected (OBSERVE0_FREQUENCY_MAX - OBSERVE5_FREQUENCY_MAX)

This register shows the maximum frequency detected since last update or reset. After the measurement begins, wait for 20ms before reading this register.

15.9.2.7.1 Offset

Register	Offset
OBSERVE0_FREQUENCY_MAX	48h
OBSERVE1_FREQUENCY_MAX	C8h
OBSERVE2_FREQUENCY_MAX	148h
OBSERVE3_FREQUENCY_MAX	1C8h
OBSERVE4_FREQUENCY_MAX	248h
OBSERVE5_FREQUENCY_MAX	2C8h

15.9.2.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FREQUENCY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FREQUENCY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.2.7.3 Fields

Field	Description
31-0 FREQUENCY	Frequency Maximum frequency of observed signal in Hz.

15.9.3 MISC register descriptions

15.9.3.1 Miscellaneous Registers memory map

For details on how to program these registers, refer to Analog IP (AI) Interface.

misc base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
800	Chip Silicon Version Register (MISC_DIFPROG)	32	RO	0011_70B0
820	VDDSOC_AI_CTRL_REGISTER (VDDSOC_AI_CTRL)	32	RW	0000_0000
830	VDDSOC_AI_WDATA_REGISTER (VDDSOC_AI_WDATA)	32	RW	0000_0000
840	VDDSOC_AI_RDATA_REGISTER (VDDSOC_AI_RDATA)	32	RO	0000_0000
850	VDDSOC2PLL_AI_CTRL_1G_REGISTER (VDDSOC2PLL_AI_CTRL_1G)	32	RW	0000_0000
860	VDDSOC2PLL_AI_WDATA_1G_REGISTER (VDDSOC2PLL_AI_WDATA_1G)	32	RW	0000_0000
870	VDDSOC2PLL_AI_RDATA_1G_REGISTER (VDDSOC2PLL_AI_RDATA_1G)	32	RO	0000_0000
880	VDDSOC_AI_CTRL_AUDIO_REGISTER (VDDSOC2PLL_AI_CTRL_AUDIO)	32	RW	0000_0000
890	VDDSOC_AI_WDATA_AUDIO_REGISTER (VDDSOC2PLL_AI_WDATA_AUDIO)	32	RW	0000_0000
8A0	VDDSOC2PLL_AI_RDATA_REGISTER (VDDSOC2PLL_AI_RDATA_AUDIO)	32	RO	0000_0000
8B0	VDDSOC2PLL_AI_CTRL_VIDEO_REGISTER (VDDSOC2PLL_AI_CTRL_VIDEO)	32	RW	0000_0000
8C0	VDDSOC2PLL_AI_WDATA_VIDEO_REGISTER (VDDSOC2PLL_AI_WDATA_VIDEO)	32	RW	0000_0000
8D0	VDDSOC2PLL_AI_RDATA_VIDEO_REGISTER (VDDSOC2PLL_AI_RDATA_VIDEO)	32	RO	0000_0000
8E0	VDDSOC_AI_CTRL_REGISTER (VDDLPSR_AI_CTRL)	32	RW	0000_0000
8F0	VDDLPSR_AI_WDATA_REGISTER (VDDLPSR_AI_WDATA)	32	RW	0000_0000
900	VDDLPSR_AI_RDATA_REFTOP_REGISTER (VDDLPSR_AI_RDATA_REFTOP)	32	RO	0000_0000
910	VDDLPSR_AI_RDATA_TMPSNS_REGISTER (VDDLPSR_AI_RDATA_TMPSNS)	32	RO	0000_0000
920	VDDLPSR_AI400M_CTRL_REGISTER (VDDLPSR_AI400M_CTRL)	32	RW	0000_0000
930	VDDLPSR_AI400M_WDATA_REGISTER (VDDLPSR_AI400M_WDATA)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
940	VDDLPSR_AI400M_RDATA_REGISTER (VDDLPSR_AI400M_RDATA)	32	RO	0000_0000

15.9.3.2 Chip Silicon Version Register (MISC_DIFPROG)

Chip Silicon Version Register

15.9.3.2.1 Offset

Register	Offset
MISC_DIFPROG	800h

15.9.3.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CHIPID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHIPID															
W																
Reset	0	1	1	1	0	0	0	0	u	u	u	u	0	0	0	0

NOTE

- For i.MX RT1170A chip, bit[7:4] is 1011. For i.MX RT1170B, bit[7:4] is 1100.

15.9.3.2.3 Fields

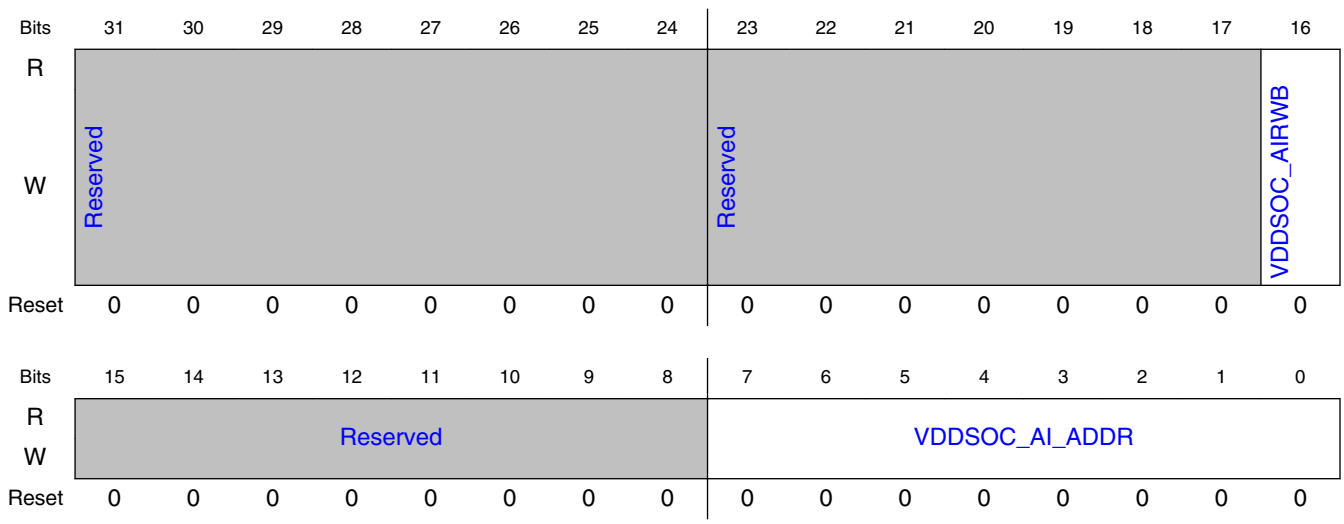
Field	Description
31-0 CHIPID	Chip ID

15.9.3.3 VDDSOC_AI_CTRL_REGISTER (VDDSOC_AI_CTRL)

15.9.3.3.1 Offset

Register	Offset
VDDSOC_AI_CTRL	820h

15.9.3.3.2 Diagram



15.9.3.3.3 Fields

Field	Description
31-24 —	Always set to zero (0). Always set to zero (0).
23-17 —	Reserved
16 VDDSOC_AIRWB	VDDSOC_AIRWB Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
15-8 —	Always set to zero (0). Always set to zero (0).
7-0	VDDSOC_AI_ADDR

Field	Description
VDDSOC_AI_ADDR	Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.4 VDDSOC_AI_WDATA_REGISTER (VDDSOC_AI_WDATA)

15.9.3.4.1 Offset

Register	Offset
VDDSOC_AI_WDATA	830h

15.9.3.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDSOC_AI_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDSOC_AI_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.4.3 Fields

Field	Description
31-0 VDDSOC_AI_WDATA	VDDSOC_AI_WDATA Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.5 VDDSOC_AI_RDATA_REGISTER (VDDSOC_AI_RDATA)

15.9.3.5.1 Offset

Register	Offset
VDDSOC_AI_RDATA	840h

15.9.3.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDSOC_AI_RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDSOC_AI_RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.5.3 Fields

Field	Description
31-0 VDDSOC_AI_RDATA	VDDSOC_AI_RDATA Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

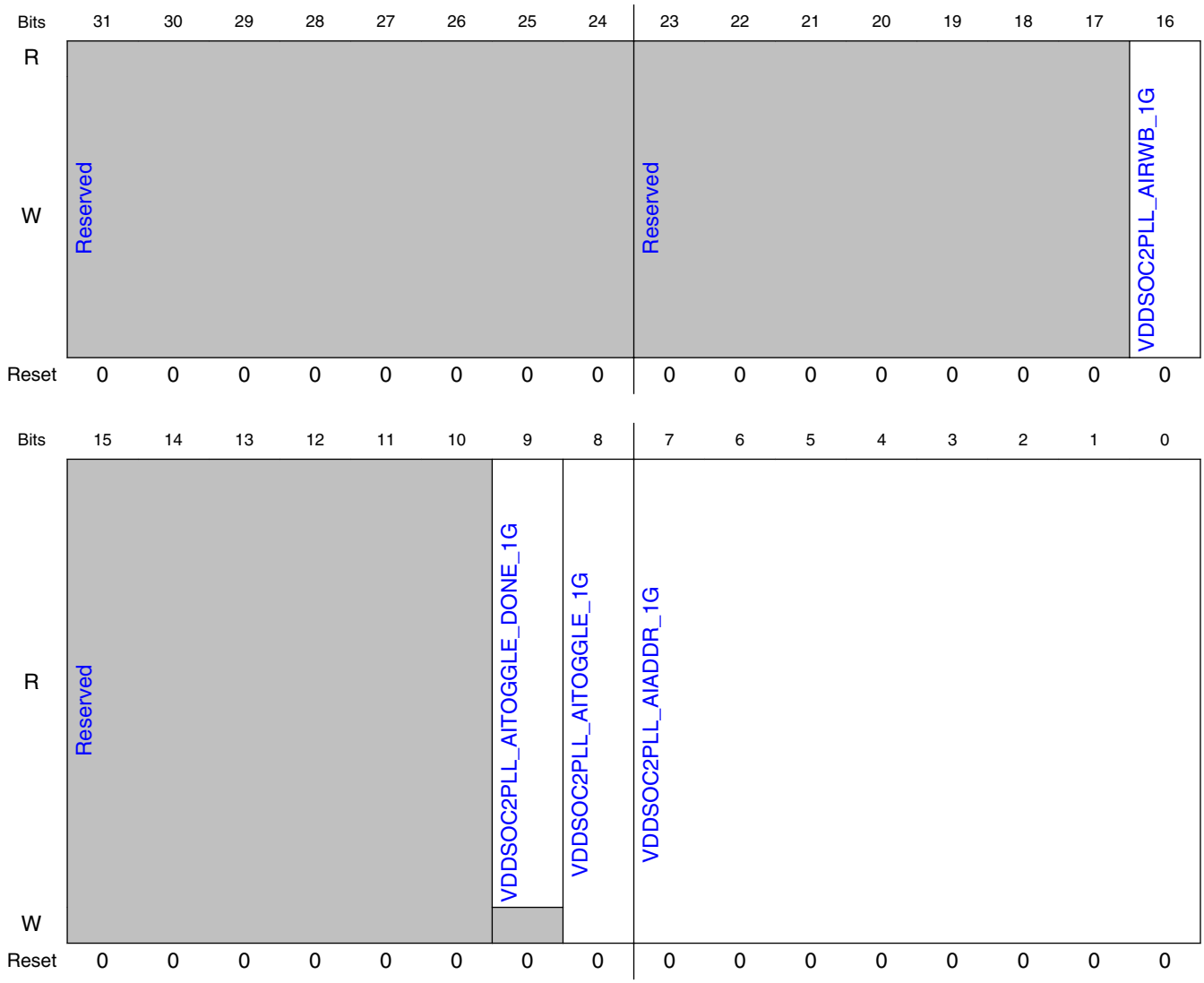
15.9.3.6 VDDSOC2PLL_AI_CTRL_1G_REGISTER (VDDSOC2PLL_AI_CTRL_1G)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.6.1 Offset

Register	Offset
VDDSOC2PLL_AI_CTRL_1G	850h

15.9.3.6.2 Diagram



15.9.3.6.3 Fields

Field	Description
31-24 —	Reserved
23-17 —	Reserved
16 VDDSOC2PLL_AIRWB_1G	VDDSOC2PLL_AIRWB_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
15-10	Reserved

Table continues on the next page...

Field	Description
—	
9 VDDSOC2PLL_AITOGGLE_DONE_1G	VDDSOC2PLL_AITOGGLE_DONE_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description..
8 VDDSOC2PLL_AITOGGLE_1G	VDDSOC2PLL_AITOGGLE_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description..
7-0 VDDSOC2PLL_AIADDR_1G	VDDSOC2PLL_AIADDR_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.7 VDDSOC2PLL_AI_WDATA_1G_REGISTER (VDDSOC2PLL_AI_WDATA_1G)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.7.1 Offset

Register	Offset
VDDSOC2PLL_AI_WDATA_1G	860h

15.9.3.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDSOC2PLL_AI_WDATA_1G															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDSOC2PLL_AI_WDATA_1G															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.7.3 Fields

Field	Description
31-0 VDDSOC2PLL_AI_WDATA_1G	VDDSOC2PLL_AI_WDATA_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.8 VDDSOC2PLL_AI_RDATA_1G_REGISTER (VDDSOC2PLL_AI_RDATA_1G)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.8.1 Offset

Register	Offset
VDDSOC2PLL_AI_RDATA_1G	870h

15.9.3.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDSOC2PLL_AI_RDATA_1G															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDSOC2PLL_AI_RDATA_1G															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.8.3 Fields

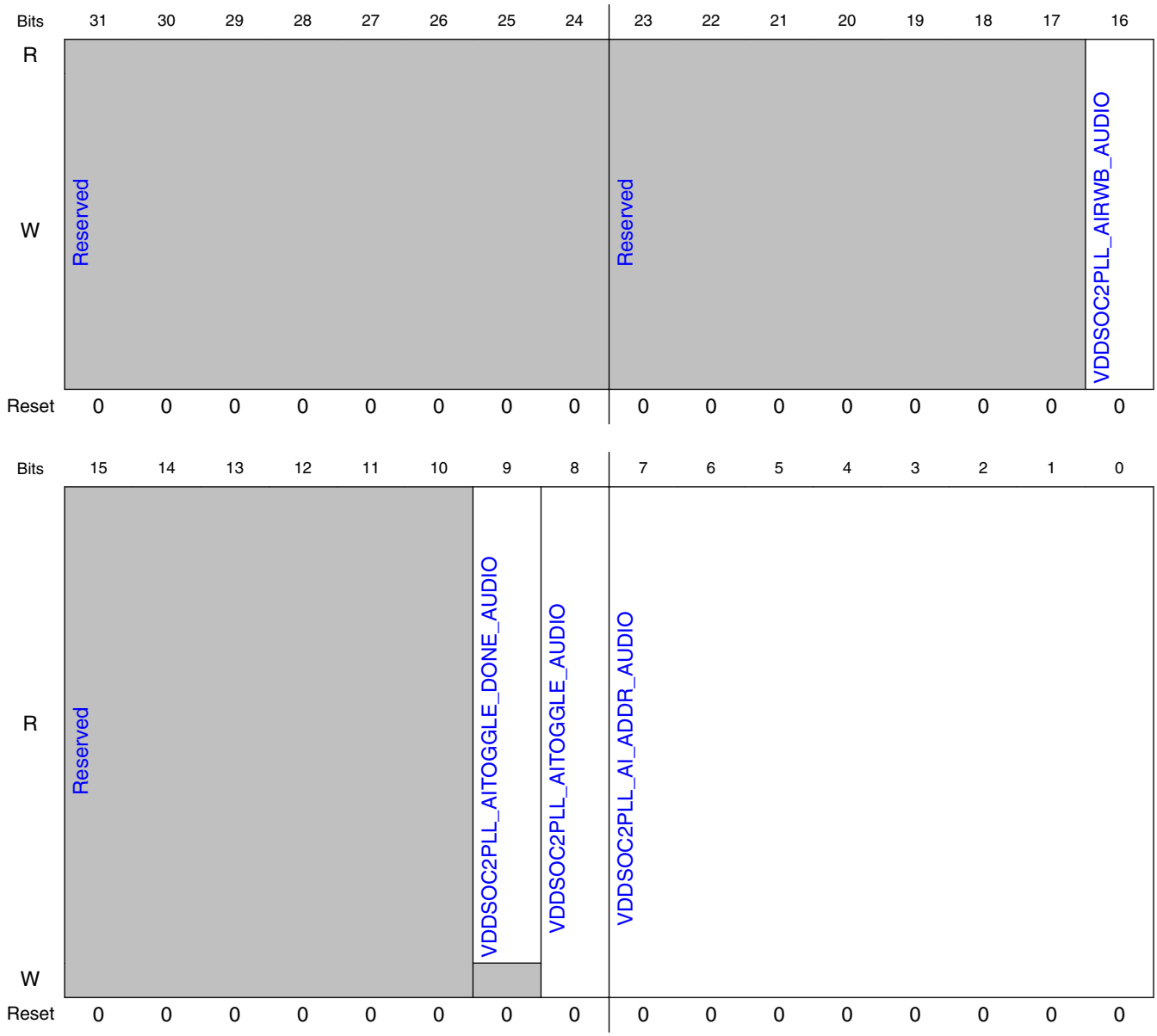
Field	Description
31-0 VDDSOC2PLL_AI_RDATA_1G	VDDSOC2PLL_AI_RDATA_1G Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.9 VDDSOC_AI_CTRL_AUDIO_REGISTER (VDDSOC2PLL_AI_CTRL_AUDIO)

15.9.3.9.1 Offset

Register	Offset
VDDSOC2PLL_AI_CTRL_AUDIO	880h

15.9.3.9.2 Diagram



15.9.3.9.3 Fields

Field	Description
31-24 —	Always set to zero (0). Always set to zero (0).
23-17 —	Reserved
16	VDDSOC_AIRWB Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

Table continues on the next page...

Field	Description
VDDSOC2PLL_AIRWB_AUDIO	
15-10 —	Always set to zero (0). Always set to zero (0).
9 VDDSOC2PLL_AITOGGLE_DONE_AUDIO	VDDSOC2PLL_AITOGGLE_DONE_AUDIO Please refer to 'Analog IP (AI) Interface' topic for the detailed description..
8 VDDSOC2PLL_AITOGGLE_AUDIO	VDDSOC2PLL_AITOGGLE_AUDIO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
7-0 VDDSOC2PLL_AI_ADDR_AUDIO	VDDSOC2PLL_AI_ADDR_AUDIO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.10 VDDSOC_AI_WDATA_AUDIO_REGISTER (VDDSOC2PLL_AI_WDATA_AUDIO)

15.9.3.10.1 Offset

Register	Offset
VDDSOC2PLL_AI_WDATA_AUDIO	890h

15.9.3.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDSOC2PLL_AI_WDATA_AUDIO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDSOC2PLL_AI_WDATA_AUDIO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.10.3 Fields

Field	Description
31-0 VDDSOC2PLL_AI_WDATA_AUDIO	VDDSOC2PLL_AI_WDATA_AUDIO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.11 VDDSOC2PLL_AI_RDATA_REGISTER (VDDSOC2PLL_AI_RDATA_AUDIO)

15.9.3.11.1 Offset

Register	Offset
VDDSOC2PLL_AI_RDATA_AUDIO	8A0h

15.9.3.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDSOC2PLL_AI_RDATA_AUDIO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDSOC2PLL_AI_RDATA_AUDIO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.11.3 Fields

Field	Description
31-0 VDDSOC2PLL_AI_RDATA_AUDIO	VDDSOC2PLL_AI_RDATA_AUDIO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

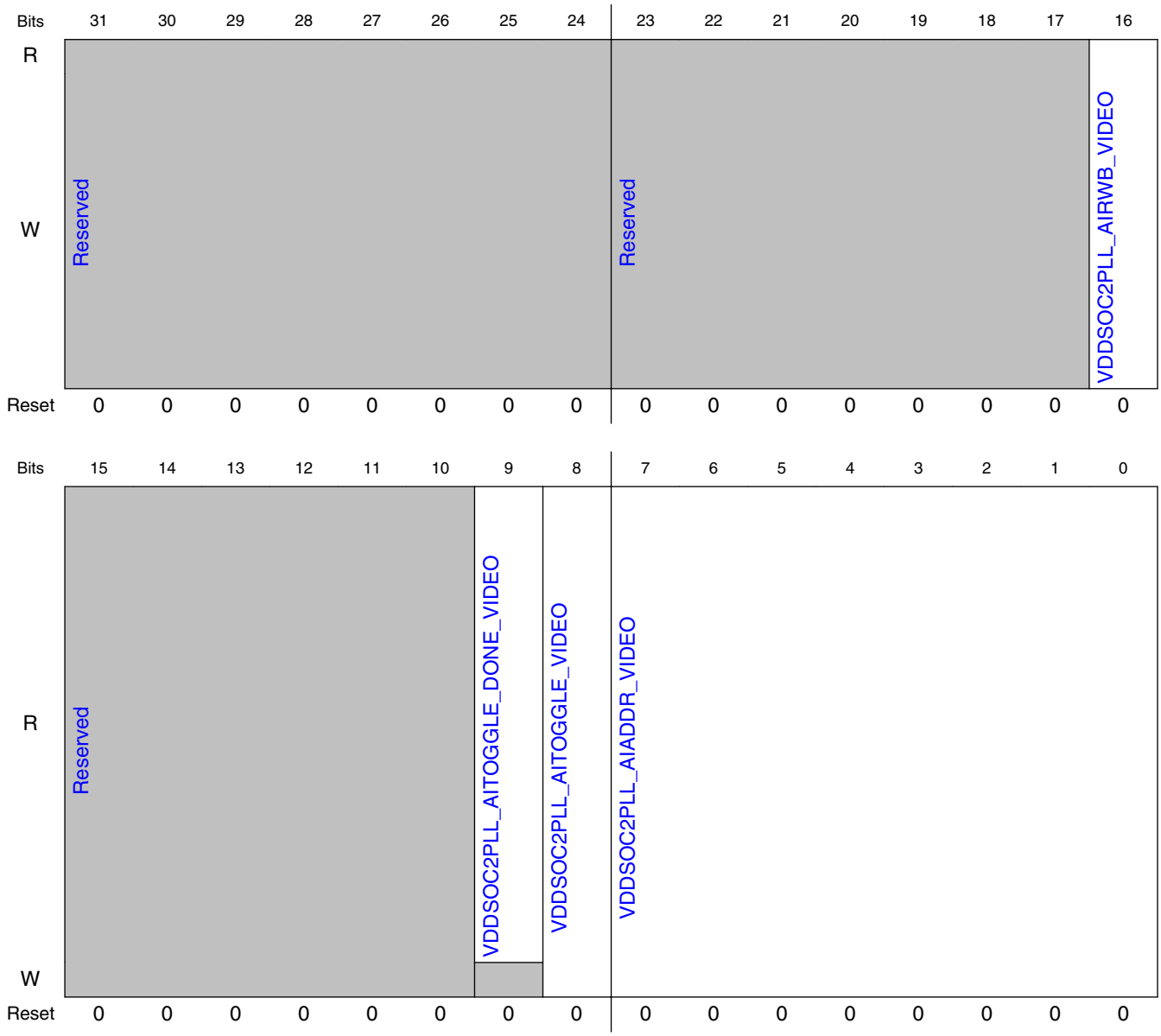
15.9.3.12 VDDSOC2PLL_AI_CTRL_VIDEO_REGISTER (VDDSOC2PLL_AI_CTRL_VIDEO)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.12.1 Offset

Register	Offset
VDDSOC2PLL_AI_CTRL_VIDEO	8B0h

15.9.3.12.2 Diagram



15.9.3.12.3 Fields

Field	Description
31-24 —	Reserved
23-17 —	Reserved
16	VDDSOC2PLL_AIRWB_VIDEO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

Table continues on the next page...

Field	Description
VDDSOC2PLL_AIRWB_VIDEO	
15-10 —	Reserved
9 VDDSOC2PLL_AITOGGLE_DONE_VIDEO	VDDSOC2PLL_AITOGGLE_DONE_VIDEO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
8 VDDSOC2PLL_AITOGGLE_VIDEO	VDDSOC2PLL_AITOGGLE_VIDEO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
7-0 VDDSOC2PLL_AIADDR_VIDEO	VDDSOC2PLL_AIADDR_VIDEO VDDSOC2PLL_AIADDR_VIDEO

15.9.3.13 VDDSOC2PLL_AI_WDATA_VIDEO_REGISTER (VDDSOC2PLL_AI_WDATA_VIDEO)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.13.1 Offset

Register	Offset
VDDSOC2PLL_AI_WDATA_VIDEO	8C0h

15.9.3.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDSOC2PLL_AI_WDATA_VIDEO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDSOC2PLL_AI_WDATA_VIDEO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.13.3 Fields

Field	Description
31-0 VDDSOC2PLL_AI_WDATA_VIDEO	VDDSOC2PLL_AI_WDATA_VIDEO Please refer to 'Analog IP (AI) Interface' topic for the detailed description..

15.9.3.14 VDDSOC2PLL_AI_RDATA_VIDEO_REGISTER (VDDSOC2PLL_AI_RDATA_VIDEO)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.14.1 Offset

Register	Offset
VDDSOC2PLL_AI_RDATA_VIDEO	8D0h

15.9.3.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDSOC2PLL_AI_RDATA_VIDEO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDSOC2PLL_AI_RDATA_VIDEO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.14.3 Fields

Field	Description
31-0	VDDSOC2PLL_AI_RDATA_VIDEO Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

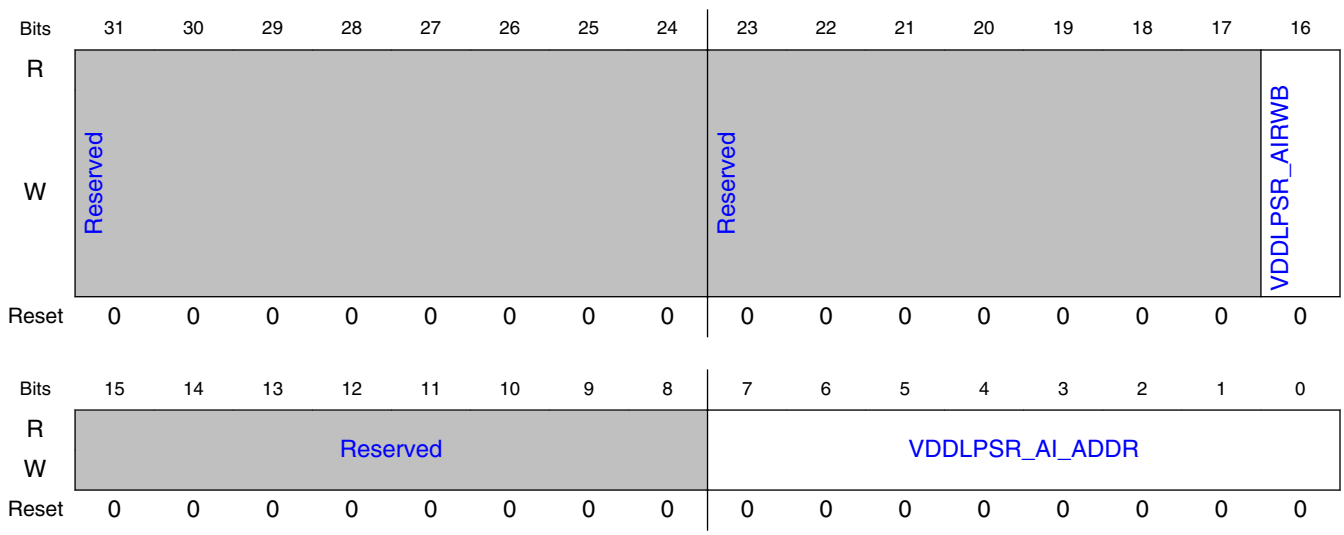
Field	Description
VDDSOC2PLL_AI_RDATA_VIDEO	

15.9.3.15 VDDSOC_AI_CTRL_REGISTER (VDDLPSR_AI_CTRL)

15.9.3.15.1 Offset

Register	Offset
VDDLPSR_AI_CTRL	8E0h

15.9.3.15.2 Diagram



15.9.3.15.3 Fields

Field	Description
31-24 —	Always set to zero (0). Always set to zero (0).
23-17 —	Reserved

Table continues on the next page...

Field	Description
16 VDDLPSR_AIRWB	VDDLPSR_AIRWB Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
15-8 —	Always set to zero (0). Always set to zero (0).
7-0 VDDLPSR_AI_ADDR	VDDLPSR_AI_ADDR Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.16 VDDLPSR_AI_WDATA_REGISTER (VDDLPSR_AI_WDATA)

15.9.3.16.1 Offset

Register	Offset
VDDLPSR_AI_WDATA	8F0h

15.9.3.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDLPSR_AI_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDLPSR_AI_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.16.3 Fields

Field	Description
31-0 VDDLPSR_AI_WDATA	VDD_LPSR_AI_WDATA Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.17 VDDLPSR_AI_RDATA_REFTOP_REGISTER (VDDLPSR_AI_RDATA_REFTOP)

15.9.3.17.1 Offset

Register	Offset
VDDLPSR_AI_RDATA_REFTOP	900h

15.9.3.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDLPSR_AI_RDATA_REFTOP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDLPSR_AI_RDATA_REFTOP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.17.3 Fields

Field	Description
31-0 VDDLPSR_AI_RDATA_REFTOP	VDDLPSR_AI_RDATA_REFTOP Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.18 VDDLPSR_AI_RDATA_TMPSNS_REGISTER (VDDLPSR_AI_RDATA_TMPSNS)

15.9.3.18.1 Offset

Register	Offset
VDDLPSR_AI_RDATA_TMPSNS	910h

15.9.3.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDLPSR_AI_RDATA_TMPSNS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDLPSR_AI_RDATA_TMPSNS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.18.3 Fields

Field	Description
31-0 VDDLPSR_AI_RDATA_TMPSNS	VDDLPSR_AI_RDATA_TMPSNS Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

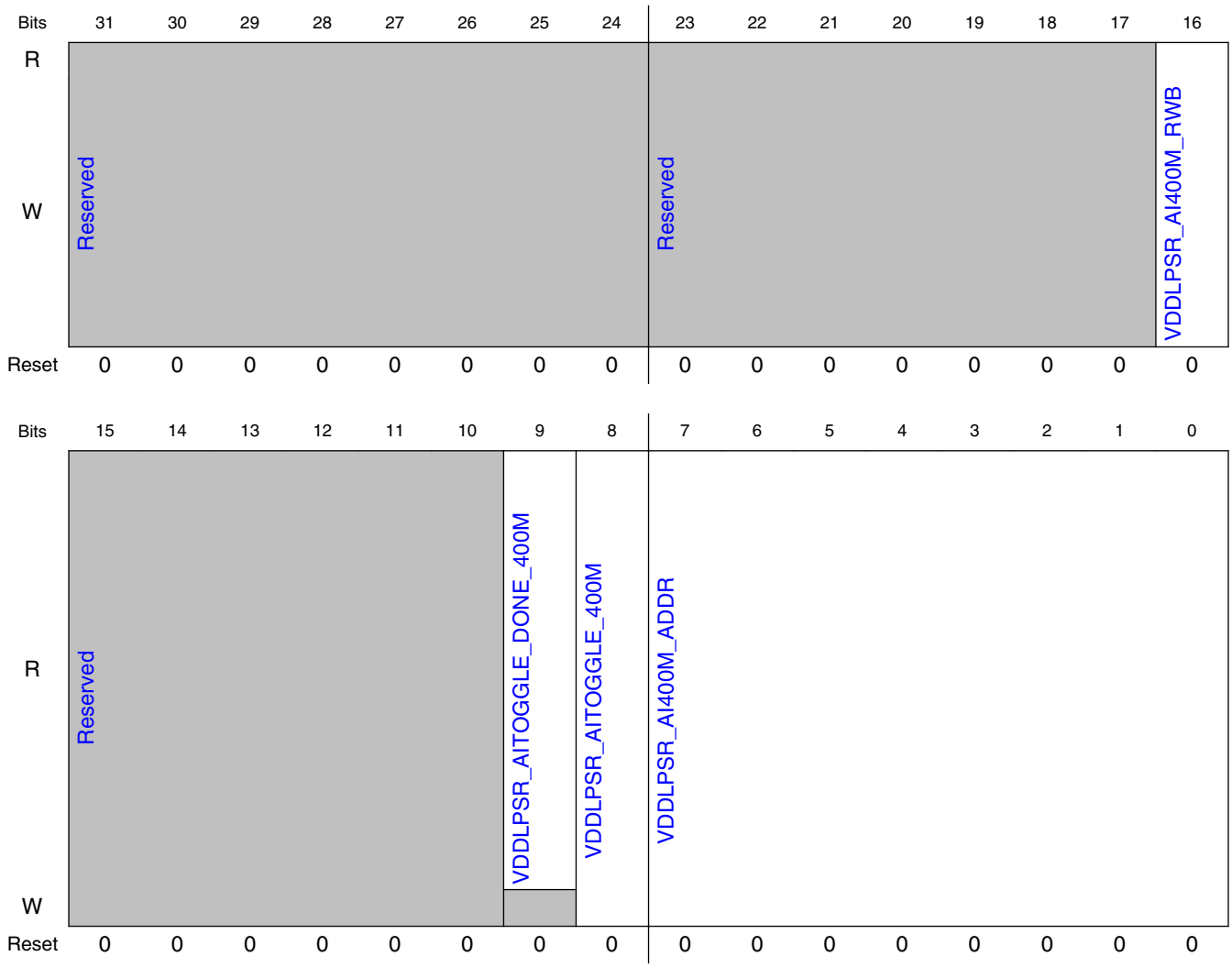
15.9.3.19 VDDLPSR_AI400M_CTRL_REGISTER (VDDLPSR_AI400M_CTRL)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.19.1 Offset

Register	Offset
VDDLPSR_AI400M_CTRL	920h

15.9.3.19.2 Diagram



15.9.3.19.3 Fields

Field	Description
31-24 —	Reserved
23-17 —	Reserved
16 VDDLPSR_AI400M_RWB	VDDLPSR_AI400M_RWB Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
15-10	Reserved

Table continues on the next page...

Field	Description
—	
9 VDDLPSR_AITOGGLE_DONE_400M	VDDLPSR_AITOGGLE_DONE_400M Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
8 VDDLPSR_AITOGGLE_400M	VDDLPSR_AITOGGLE_400M Please refer to 'Analog IP (AI) Interface' topic for the detailed description.
7-0 VDDLPSR_AI400M_ADDR	VDDLPSR_AI400M_ADDR Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.20 VDDLPSR_AI400M_WDATA_REGISTER (VDDLPSR_AI400M_WDATA)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.20.1 Offset

Register	Offset
VDDLPSR_AI400M_WDATA	930h

15.9.3.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDDLPSR_AI400M_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDDLPSR_AI400M_WDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.20.3 Fields

Field	Description
31-0 VDDLPSR_AI400M_WDATA	VDDLPSR_AI400M_WDATA Please refer to 'Analog IP (AI) Interface' topic for the detailed description.

15.9.3.21 VDDLPSR_AI400M_RDATA_REGISTER (VDDLPSR_AI400M_RDATA)

Please refer to 'Analog IP (AI) Interface' topic for the detailed description. It should follow the sequence provided.

15.9.3.21.1 Offset

Register	Offset
VDDLPSR_AI400M_RDATA	940h

15.9.3.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VDDLPSR_AI400M_RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDDLPSR_AI400M_RDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.3.21.3 Fields

Field	Description
31-0 VDDLPSR_AI400M_RDATA	VDDLPSR_AI400M_RDATA Please refer to 'Analog IP (AI) Interface' topic for the detailed description..

15.9.4 PLL register descriptions

15.9.4.1 PLL memory map

pll base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
200	ARM_PLL_CTRL_REGISTER (ARM_PLL_CTRL)	32	RW	4000_00A6
210	SYS_PLL3_CTRL_REGISTER (SYS_PLL3_CTRL)	32	RW	4000_0003
220	SYS_PLL3_UPDATE_REGISTER (SYS_PLL3_UPDATE)	32	RW	0000_0000
230	SYS_PLL3_PFD_REGISTER (SYS_PLL3_PFD)	32	RW	8CA0_918D
240	SYS_PLL2_CTRL_REGISTER (SYS_PLL2_CTRL)	32	RW	4000_0000
250	SYS_PLL2_UPDATE_REGISTER (SYS_PLL2_UPDATE)	32	RW	0000_0000
260	SYS_PLL2_SS_REGISTER (SYS_PLL2_SS)	32	RW	0000_0000
270	SYS_PLL2_PFD_REGISTER (SYS_PLL2_PFD)	32	RW	A098_909B
2A0	SYS_PLL2_MFD_REGISTER (SYS_PLL2_MFD)	32	RW	0FFF_FFFF
2B0	SYS_PLL1_SS_REGISTER (SYS_PLL1_SS)	32	RW	0000_0000
2C0	SYS_PLL1_CTRL_REGISTER (SYS_PLL1_CTRL)	32	RW	0000_4000
2D0	SYS_PLL1_DENOMINATOR_REGISTER (SYS_PLL1_DENOMINATOR)	32	RW	2FFF_FFFE
2E0	SYS_PLL1_NUMERATOR_REGISTER (SYS_PLL1_NUMERATOR)	32	RW	1FFF_FFFF
2F0	SYS_PLL1_DIV_SELECT_REGISTER (SYS_PLL1_DIV_SELECT)	32	RW	0000_001D
300	PLL_AUDIO_CTRL_REGISTER (PLL_AUDIO_CTRL)	32	RW	0000_4000
310	PLL_AUDIO_SS_REGISTER (PLL_AUDIO_SS)	32	RW	0000_0000
320	PLL_AUDIO_DENOMINATOR_REGISTER (PLL_AUDIO_DENOMINATOR)	32	RW	2FFF_FFFE
330	PLL_AUDIO_NUMERATOR_REGISTER (PLL_AUDIO_NUMERATOR)	32	RW	1FFF_FFFF
340	PLL_AUDIO_DIV_SELECT_REGISTER (PLL_AUDIO_DIV_SELECT)	32	RW	0000_0029
350	PLL_VIDEO_CTRL_REGISTER (PLL_VIDEO_CTRL)	32	RW	0000_4000
360	PLL_VIDEO_SS_REGISTER (PLL_VIDEO_SS)	32	RW	0000_0000
370	PLL_VIDEO_DENOMINATOR_REGISTER (PLL_VIDEO_DENOMINATOR)	32	RW	2FFF_FFFE
380	PLL_VIDEO_NUMERATOR_REGISTER (PLL_VIDEO_NUMERATOR)	32	RW	1FFF_FFFF
390	PLL_VIDEO_DIV_SELECT_REGISTER (PLL_VIDEO_DIV_SELECT)	32	RW	0000_0029

15.9.4.2 ARM_PLL_CTRL_REGISTER (ARM_PLL_CTRL)

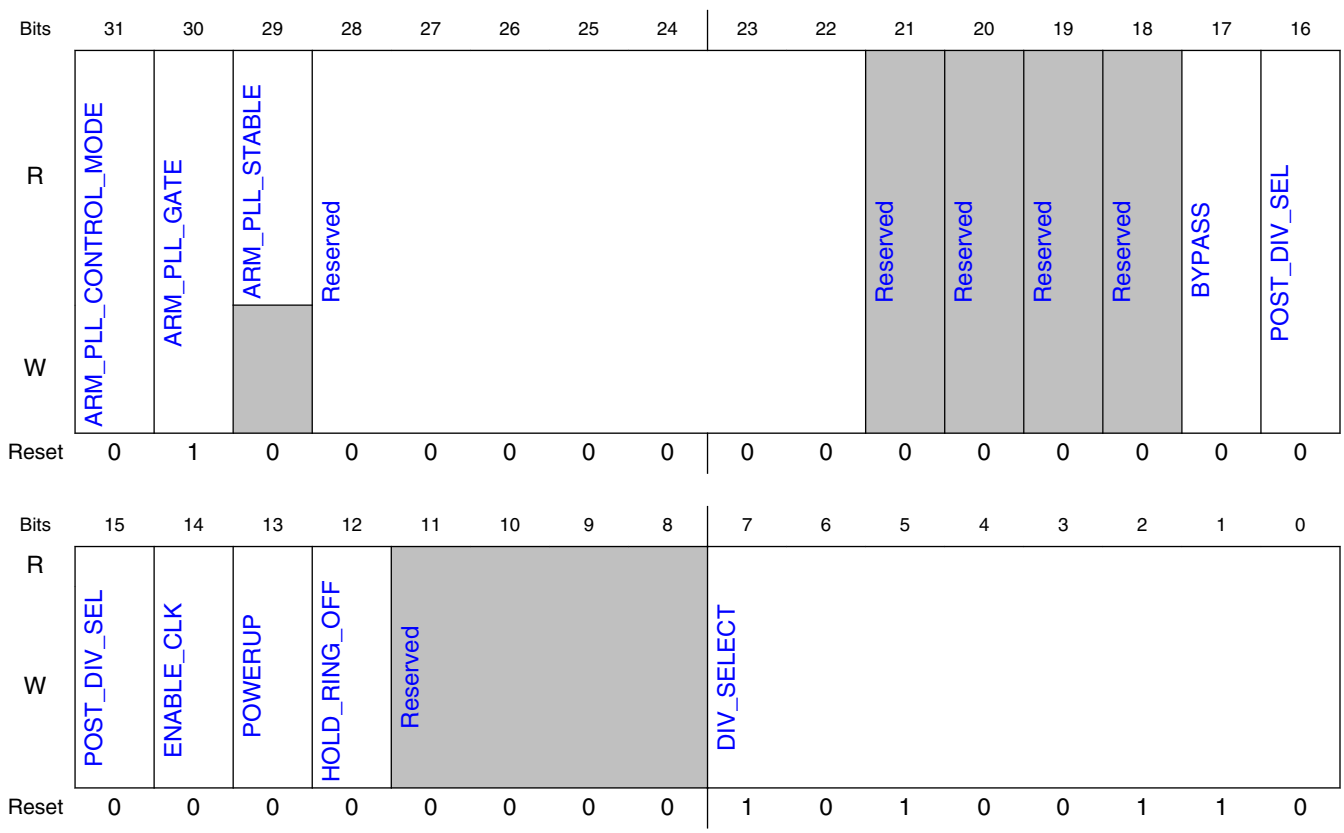
ARM_PLL control register

The control register provides control for the ARM PLL.

15.9.4.2.1 Offset

Register	Offset
ARM_PLL_CTRL	200h

15.9.4.2.2 Diagram



15.9.4.2.3 Fields

Field	Description
31 ARM_PLL_CONTROL_MODE	pll_arm_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
30 ARM_PLL_GATE	ARM_PLL_GATE default value is 1'b1 0 - Clock is not gated 1 - Clock is gated
29 ARM_PLL_STABLE	ARM_PLL_STABLE The stable indicate bit. Normally in Software mode, after the power up and enable sequence for arm pll, arm pll will not be in stable state immediately. This bit is used by software to monitor the locking status for arm pll. GPC mode do not need to take care of this bit. Hardware will handle it automatically. 0 - ARM PLL is not stable 1 - ARM PLL is stable
28-22 —	Always set to zero (0). Always set to zero (0).
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17 BYPASS	Bypass the pll. Bypass the pll enable bit. This bit could be used to bypass the PLL output to the source of reference clock 0 - Function mode 1 - Bypass Mode
16-15 POST_DIV_SEL	POST_DIV_SEL 00 - Divide by 2 01 - Divide by 4 10 - Divide by 8 11 - Divide by 1
14 ENABLE_CLK	Enable the clock output. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Disable the clock 1 - Enable the clock
13	Powers up the PLL.

Table continues on the next page...

Field	Description
POWERUP	Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Power down the PLL 1 - Power Up the PLL
12 HOLD_RING_OFF	PLL Start up initialization This field should be set to 1 every time during PLL lock, and cleared after a certain delay. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Normal operation 1 - Initialize PLL start up
11-8 —	Reserved
7-0 DIV_SELECT	DIV_SELECT This field controls the pll loop divider. Valid range for divider value: 104-208. Fout = Fin * div_select/2.0

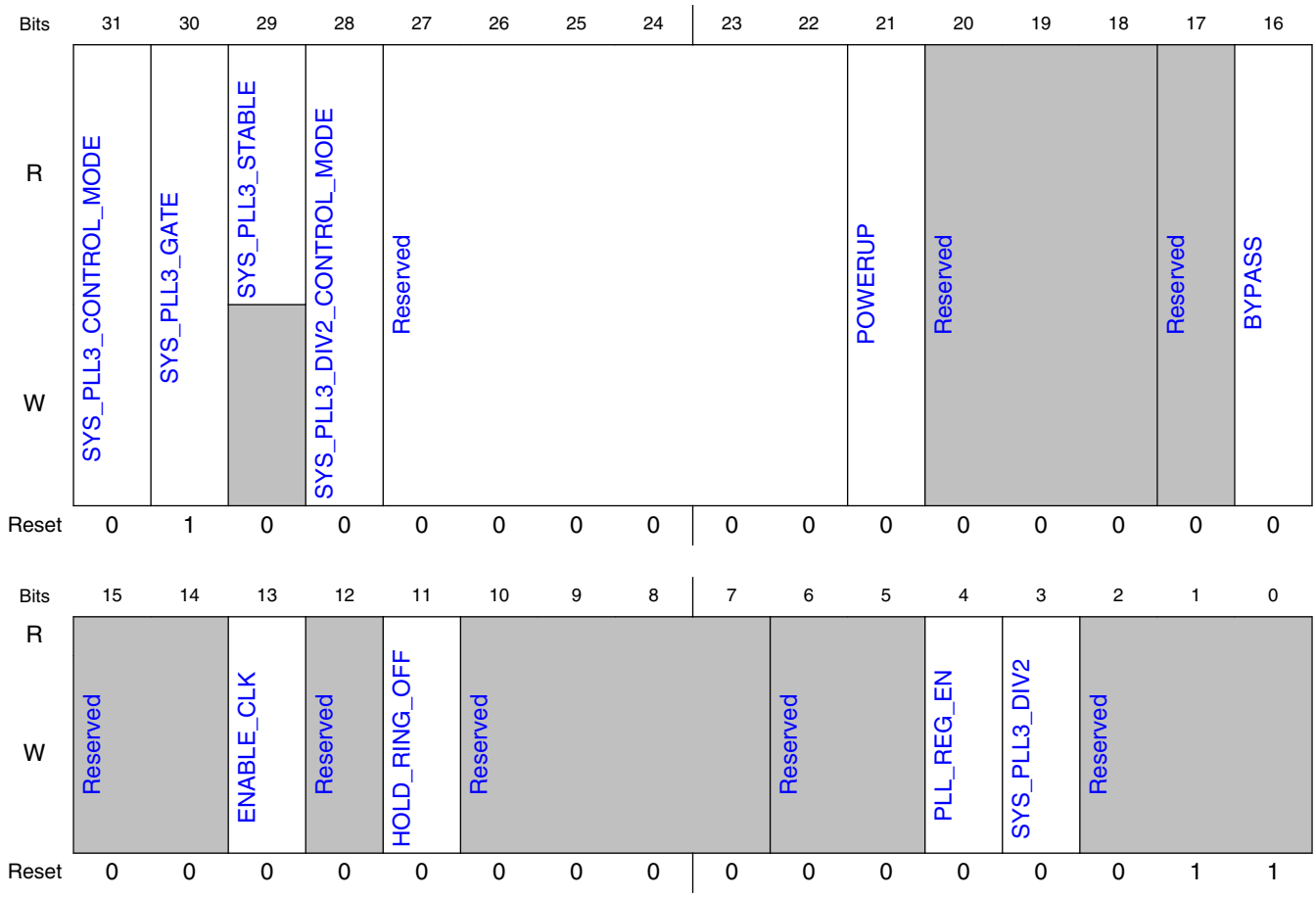
15.9.4.3 SYS_PLL3_CTRL_REGISTER (SYS_PLL3_CTRL)

The control register provides control for the 480 PLL.

15.9.4.3.1 Offset

Register	Offset
SYS_PLL3_CTRL	210h

15.9.4.3.2 Diagram



15.9.4.3.3 Fields

Field	Description
31 SYS_PLL3_CONTROL_MODE	SYS_PLL3_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
30 SYS_PLL3_GATE	SYS_PLL3_GATE default value is 1'b1 0 - Clock is not gated 1 - Clock is gated
29 SYS_PLL3_STABLE	SYS_PLL3_STABLE The stable indicate bit. Normally in Software mode, after the power up and enable sequence for SYS_PLL3, arm pll will not be in stable state immediately. This bit is used by software to monitor the locking status for SYS_PLL3.
28	SYS_PLL3_DIV2_CONTROL_MODE

Table continues on the next page...

Field	Description
SYS_PLL3_DIV2_CONTROL_MODE	Enable mode 0 - Software Mode (Default) 1 - GPC Mode
27-22 —	Always set to zero (0). Always set to zero (0).
21 POWERUP	Powers up the PLL. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Power down the PLL 1 - Power Up the PLL
20-18 —	Reserved
17 —	Reserved
16 BYPASS	BYPASS Bypass the pll enable bit. This bit could be used to bypass the PLL output to the source of reference clock 0 - Function mode 1 - Bypass Mode
15-14 —	Reserved
13 ENABLE_CLK	Enable the clock output. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Disable the clock 1 - Enable the clock
12 —	Reserved
11 HOLD_RING_OFF	PLL Start up initialization This field should be set to 1 every time during PLL lock, and cleared after a certain delay. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Normal operation 1 - Initialize PLL start up
10-7 —	Reserved
6-5 —	Reserved
4 PLL_REG_EN	Enable Internal PLL Regulator Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement.
3 SYS_PLL3_DIV2	SYS PLL3 DIV2 gate Enable SYS_PLL3_DIV2 clock which is sourced from SYS_PLL3 0 : disable

Table continues on the next page...

Field	Description
	1 : enable
2-0 —	Reserved

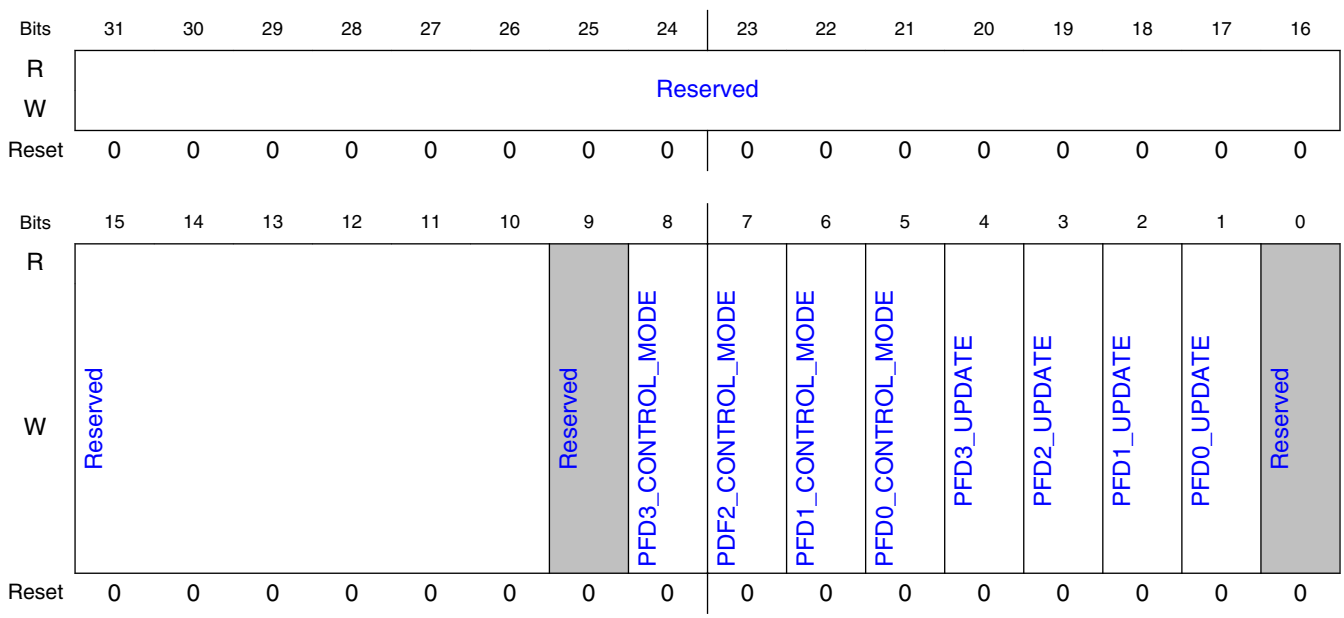
15.9.4.4 SYS_PLL3_UPDATE_REGISTER (SYS_PLL3_UPDATE)

The control register provides control for the 480 PLL.

15.9.4.4.1 Offset

Register	Offset
SYS_PLL3_UPDATE	220h

15.9.4.4.2 Diagram



15.9.4.4.3 Fields

Field	Description
31-10	Always set to zero (0).

Table continues on the next page...

Field	Description
—	Always set to zero (0).
9 —	Reserved
8 PFD3_CONTROL_MODE	pfd3_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
7 PDF2_CONTROL_MODE	pdf2_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
6 PFD1_CONTROL_MODE	pfd1_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
5 PFD0_CONTROL_MODE	pfd0_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
4 PFD3_UPDATE	PFD3_UPDATE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
3 PFD2_UPDATE	PFD2_OVERRIDE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
2 PFD1_UPDATE	PFD1_OVERRIDE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
1 PFD0_UPDATE	PFD0_OVERRIDE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
0 —	Reserved

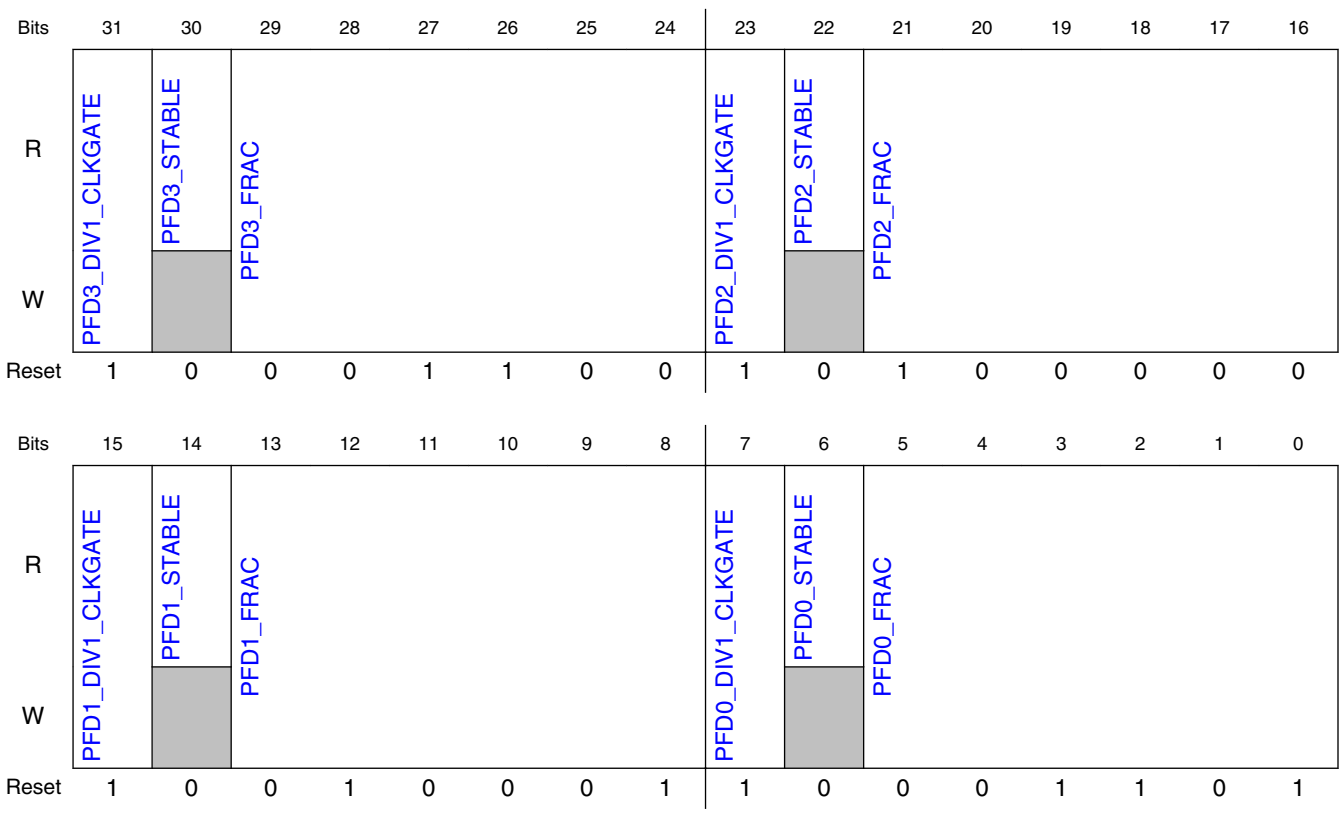
15.9.4.5 SYS_PLL3_PFD_REGISTER (SYS_PLL3_PFD)

This register contains the numerator of SYS PLL3 fractional loop divider.

15.9.4.5.1 Offset

Register	Offset
SYS_PLL3_PFD	230h

15.9.4.5.2 Diagram



15.9.4.5.3 Fields

Field	Description
31 PFD3_DIV1_CLKGATE	PFD3_DIV1_CLKGATE 0 - ref_pfd3 fractional divider clock is enabled 1 - Fractional divider clock (reference ref_pfd3) is off (power savings)
30 PFD3_STABLE	PFD3_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.

Table continues on the next page...

Field	Description
29-24 PFD3_FRAC	PFD3_FRAC This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD3_FRAC where PFD3_FRAC is in the range 13-35. **NOTE:** The PFD value can be set to 12, however the lowest recommended setting for PFD is 13.
23 PFD2_DIV1_CLKGATE	PFD2_DIV1_CLKGATE 0 - ref_pfd2 fractional divider clock is enabled 1 - Fractional divider clock (reference ref_pfd2) is off (power savings)
22 PFD2_STABLE	PFD2_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
21-16 PFD2_FRAC	PFD2_FRAC This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD2_FRAC where PFD2_FRAC is in the range 13-35.
15 PFD1_DIV1_CLKGATE	PFD1_DIV1_CLKGATE 0 - ref_pfd1 fractional divider clock is enabled 1 - Fractional divider clock (reference ref_pfd1) is off (power savings)
14 PFD1_STABLE	PFD1_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
13-8 PFD1_FRAC	PFD1_FRAC This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD1_FRAC where PFD1_FRAC is in the range 13-35.
7 PFD0_DIV1_CLKGATE	PFD0_DIV1_CLKGATE 0 - ref_pfd0 fractional divider clock is enabled 1 - Fractional divider clock (reference ref_pfd0) is off (power savings
6 PFD0_STABLE	PFD0_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
5-0 PFD0_FRAC	PFD0_FRAC This field controls the fractional divide value. The resulting frequency shall be 480*18/PFD0_FRAC where PFD0_FRAC is in the range 13-35.

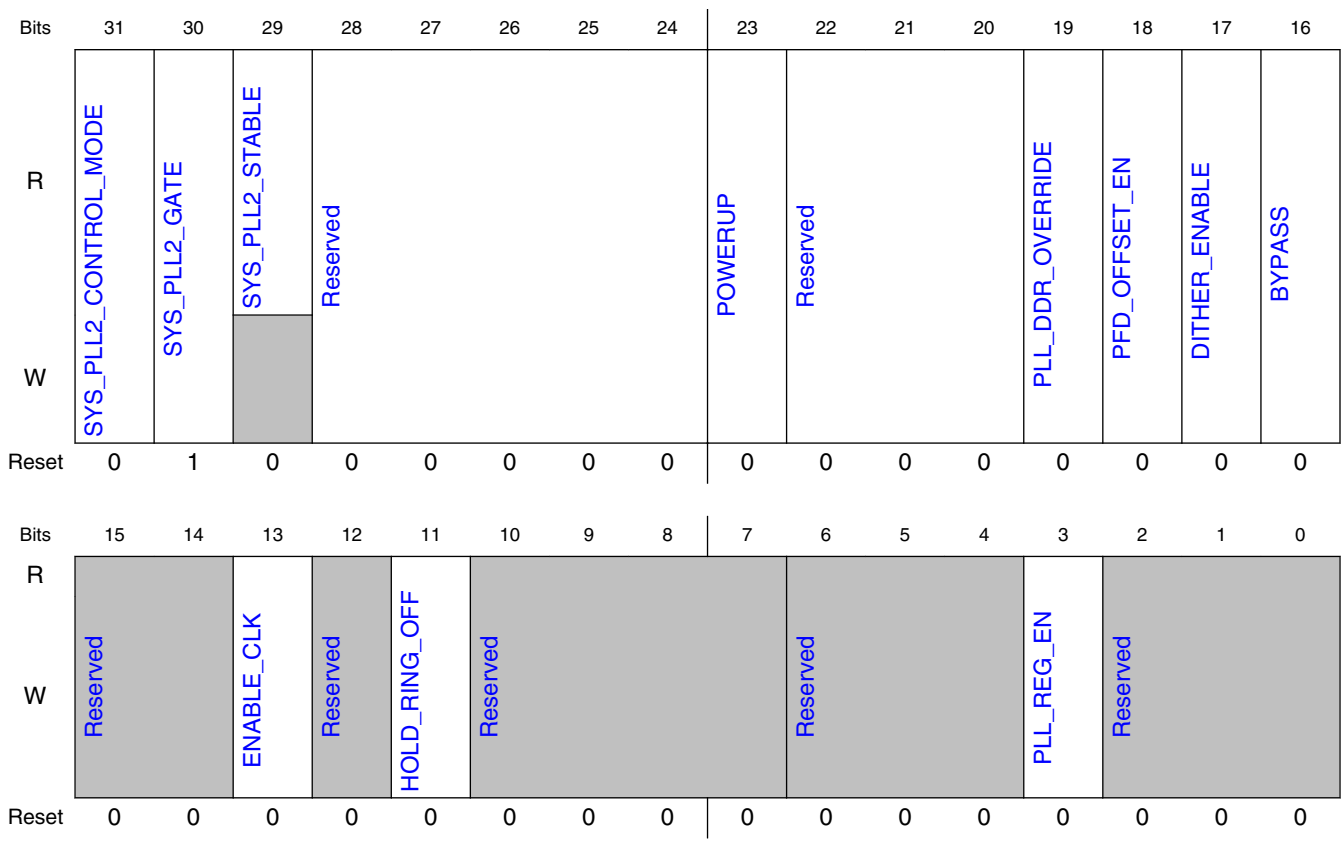
15.9.4.6 SYS_PLL2_CTRL_REGISTER (SYS_PLL2_CTRL)

The control register provides control for the SYS_PLL2

15.9.4.6.1 Offset

Register	Offset
SYS_PLL2_CTRL	240h

15.9.4.6.2 Diagram



15.9.4.6.3 Fields

Field	Description
31 SYS_PLL2_CONTROL_MODE	SYS_PLL2_control_mode SYS_PLL2 has two mode to for enable. One is software mode, the other is GPC mode(Setpoint). 0 - Software Mode (Default)

Table continues on the next page...

Field	Description
	1 - GPC Mode
30 SYS_PLL2_GATE	SYS_PLL2_GATE default value is 1'b1 0 - Clock is not gated 1 - Clock is gated
29 SYS_PLL2_STABLE	SYS_PLL2_STABLE The stable indicate bit. Normally in Software mode, after the power up and enable sequence for SYS_PLL2, ARM_PLL will not be in stable state immediately. This bit is used by software to monitor the locking status for SYS_PLL2.
28-24 —	Always set to zero (0). Always set to zero (0).
23 POWERUP	Powers up the PLL. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Power down the PLL 1 - Power Up the PLL
22-20 —	Always set to zero (0). Always set to zero (0).
19 PLL_DDR_OVERRIDE	PLL_DDR_OVERRIDE The OVERRIDE bit allows the clock control module to automatically override portions of the register. PLL_DDR_OVERRIDE = 0x0: ENABLE_CLK bits and the POWERDOWN bit controlled by this register. PLL_DDR_OVERRIDE = 0x1: CCM based hardware bits override the ENABLE_CLK bits and the POWERDOWN bit in this register.
18 PFD_OFFSET_EN	PFD_OFFSET_EN Enables an offset in the phase frequency detector.
17 DITHER_ENABLE	DITHER_ENABLE Enables dither in the fractional modulator calculation. 0 - Disable Dither 1 - Enable Dither
16 BYPASS	Bypass the pll. Control bit to Bypass the pll to the reference clock 0 - Function mode 1 - Bypass Mode
15-14 —	Reserved
13 ENABLE_CLK	Enable the clock output. Enable the clock output. 0 - Disable the clock 1 - Enable the clock
12 —	Reserved

Table continues on the next page...

Field	Description
11 HOLD_RING_OFF	PLL Start up initialization This field should be set to 1 every time during PLL lock, and cleared after a certain delay. Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement. 0 - Normal operation 1 - Initialize PLL start up
10-7 —	Reserved
6-4 —	Reserved
3 PLL_REG_EN	Enable Internal PLL Regulator Please refer to PLL Enable Sequence for more details on how to use this bit during the PLL enablement.
2-0 —	Reserved

15.9.4.7 SYS_PLL2_UPDATE_REGISTER (SYS_PLL2_UPDATE)

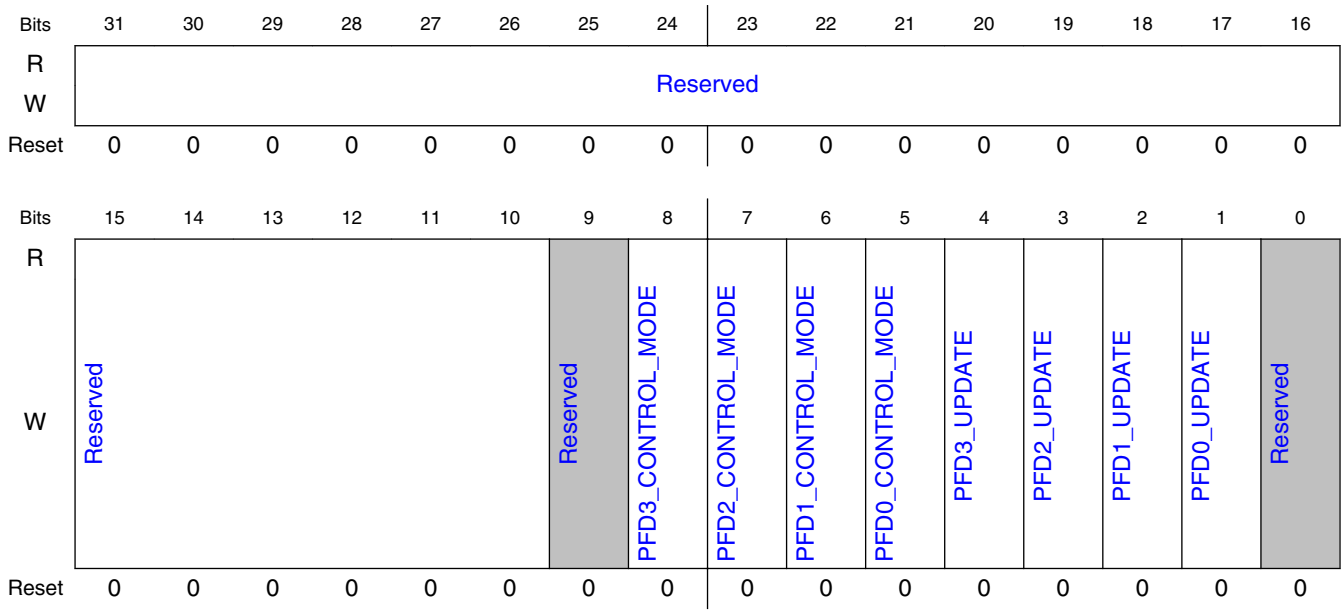
SYS_PLL2 PFD control register

The control register provides control for the 528 PLL.

15.9.4.7.1 Offset

Register	Offset
SYS_PLL2_UPDATE	250h

15.9.4.7.2 Diagram



15.9.4.7.3 Fields

Field	Description
31-10 —	Always set to zero (0). Always set to zero (0).
9 —	Reserved
8 PFD3_CONTROL_MODE	pfd3_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
7 PFD2_CONTROL_MODE	pfd2_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
6 PFD1_CONTROL_MODE	pfd1_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
5 PFD0_CONTROL_MODE	pfd0_control_mode Enable mode 0 - Software Mode (Default)

Table continues on the next page...

Field	Description
	1 - GPC Mode
4 PFD3_UPDATE	PFD3_UPDATE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
3 PFD2_UPDATE	PFD2_UPDATE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
2 PFD1_UPDATE	PFD1_UPDATE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
1 PFD0_UPDATE	PFD0_UPDATE PFD update This bit is used to update the pfd value. Toggle it will make the update function.
0 —	Reserved

15.9.4.8 SYS_PLL2_SS_REGISTER (SYS_PLL2_SS)

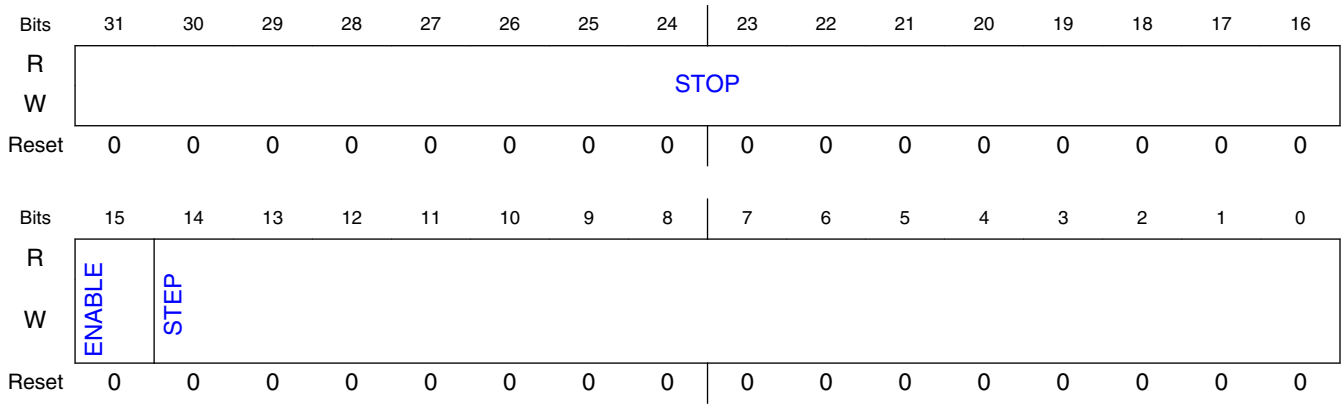
This register contains the SYS PLL2 spread spectrum controls

Refer to Spread Spectrum Parameters for more details.

15.9.4.8.1 Offset

Register	Offset
SYS_PLL2_SS	260h

15.9.4.8.2 Diagram



15.9.4.8.3 Fields

Field	Description
31-16 STOP	STOP Frequency change = stop/MFD*24MHz.
15 ENABLE	ENABLE This bit enables the spread spectrum modulation 0 - Disable Spread Spectrum. In this setting, the values for STEP and STOP are ignored 1 - Enable Spread Spectrum
14-0 STEP	STEP The max frequency change = step/MFD*24MHz.

15.9.4.9 SYS_PLL2_PFD_REGISTER (SYS_PLL2_PFD)

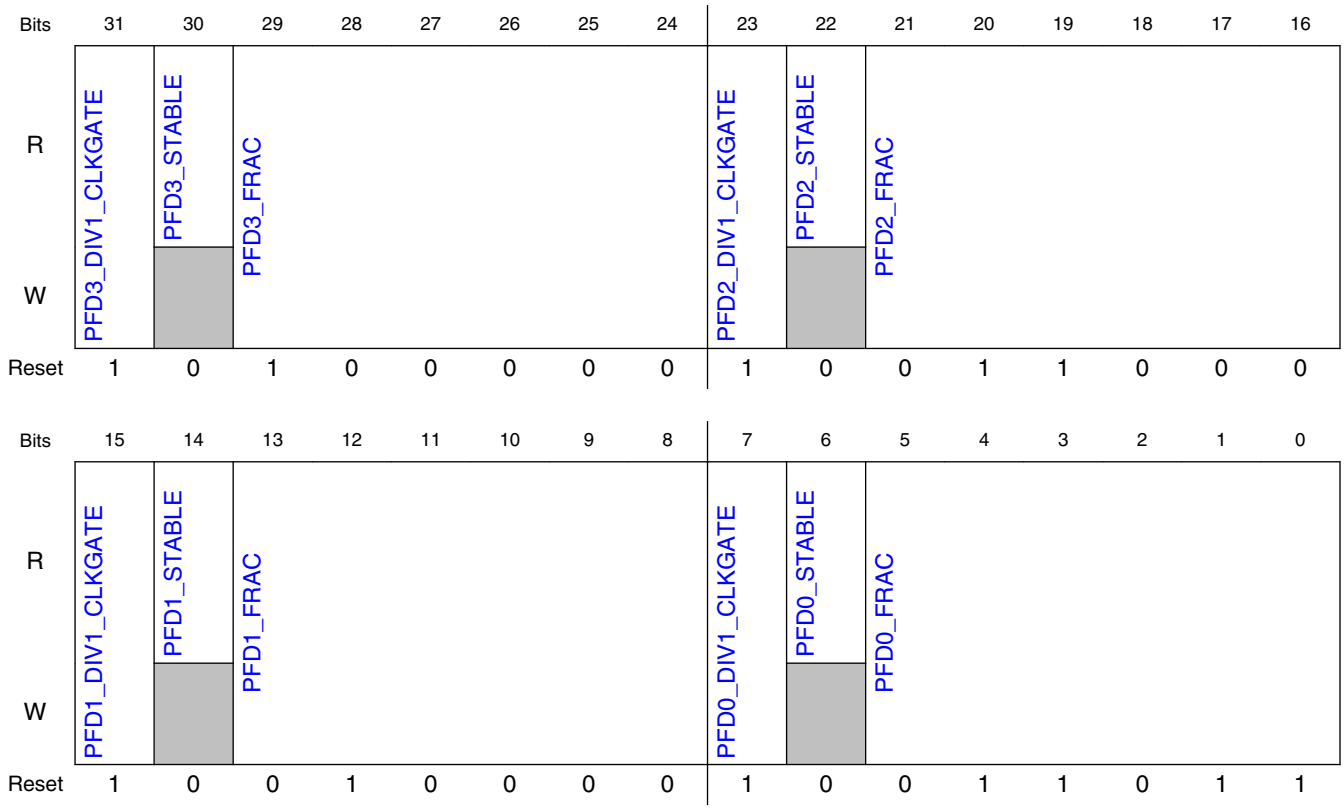
SYS_PLL2 fractional loop divider

This register contains the numerator of DDR PLL fractional loop divider.

15.9.4.9.1 Offset

Register	Offset
SYS_PLL2_PFD	270h

15.9.4.9.2 Diagram



15.9.4.9.3 Fields

Field	Description
31 PFD3_DIV1_CLKGATE	PFD3_DIV1_CLKGATE If set to 1, the 3rd fractional divider clock (reference ref_pfd3) is off (power savings). 0: ref_pfd3 fractional divider clock is enabled.
30 PFD3_STABLE	PFD3_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
29-24 PFD3_FRAC	PFD3_FRAC This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD3_FRAC where PFD3_FRAC is in the range 13-35.
23 PFD2_DIV1_CLKGATE	PFD2_DIV1_CLKGATE If set to 1, the 2 fractional divider clock (reference ref_pfd2) is off (power savings). 0: ref_pfd2 fractional divider clock is enabled.
22 PFD2_STABLE	PFD2_STABLE

Table continues on the next page...

Field	Description
	This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
21-16 PFD2_FRAC	PFD2_FRAC This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD2_FRAC where PFD2_FRAC is in the range 13-35.
15 PFD1_DIV1_CLKGATE	PFD1_DIV1_CLKGATE If set to 1, the 1 fractional divider clock (reference ref_pfd1) is off (power savings). 0: ref_pfd1 fractional divider clock is enabled.
14 PFD1_STABLE	PFD1_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
13-8 PFD1_FRAC	PFD1_FRAC This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD1_FRAC where PFD1_FRAC is in the range 13-35.
7 PFD0_DIV1_CLKGATE	PFD0_DIV1_CLKGATE If set to 1, the 0 fractional divider clock (reference ref_pfd0) is off (power savings). 0: ref_pfd0 fractional divider clock is enabled.
6 PFD0_STABLE	PFD0_STABLE This read-only bitfield is for DIAGNOSTIC PURPOSES ONLY since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bit inverts, the phase divider clock output is stable. Note that the value will not invert when the fractional divider is taken out of or placed into clock-gated state.
5-0 PFD0_FRAC	PFD0_FRAC This field controls the fractional divide value. The resulting frequency shall be 528*18/PFD0_FRAC where PFD0_FRAC is in the range 13-35.

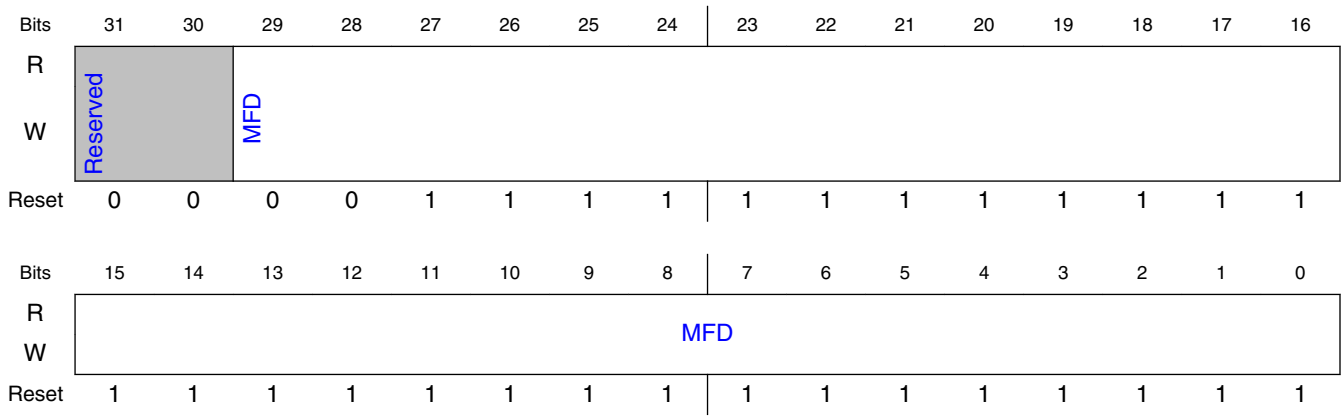
15.9.4.10 SYS_PLL2_MFD_REGISTER (SYS_PLL2_MFD)

Spread Spectrum Frequency Range is calculated as follows: Range = (STOP/MFD) * Fref

15.9.4.10.1 Offset

Register	Offset
SYS_PLL2_MFD	2A0h

15.9.4.10.2 Diagram



15.9.4.10.3 Fields

Field	Description
31-30 —	Reserved
29-0 MFD	Denominator Refer to Programming Guidelines for more details.

15.9.4.11 SYS_PLL1_SS_REGISTER (SYS_PLL1_SS)

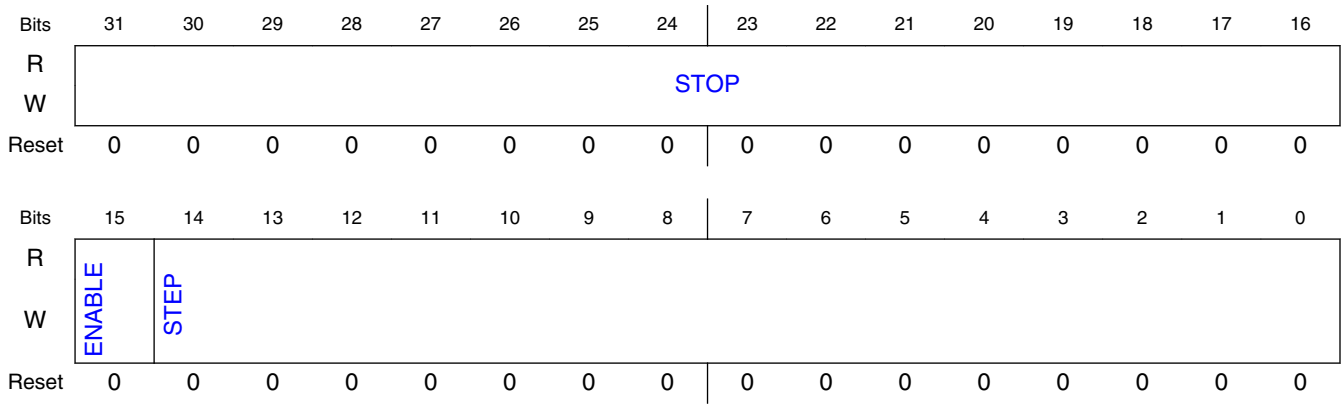
This register contains the 1G PLL spread spectrum controls

Refer to Spread Spectrum Parameters for more details.

15.9.4.11.1 Offset

Register	Offset
SYS_PLL1_SS	2B0h

15.9.4.11.2 Diagram



15.9.4.11.3 Fields

Field	Description
31-16 STOP	STOP Frequency change = stop/MFD*24MHz.
15 ENABLE	ENABLE This bit enables the spread spectrum modulation 0 - Disable Spread Spectrum. In this setting, the values for STEP and STOP are ignored 1 - Enable Spread Spectrum
14-0 STEP	STEP The max frequency change = step/MFD*24MHz.

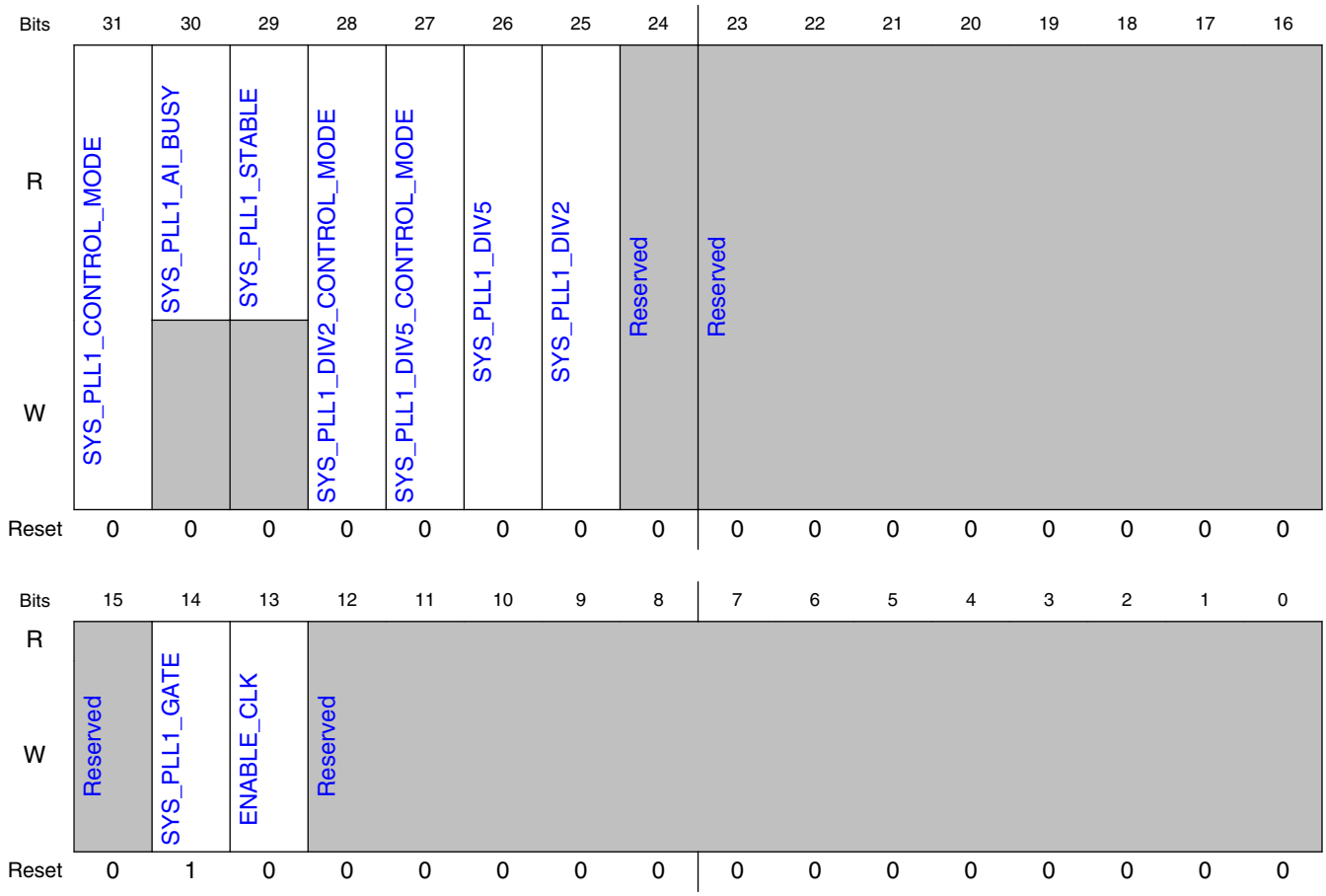
15.9.4.12 SYS_PLL1_CTRL_REGISTER (SYS_PLL1_CTRL)

The control register provides control for the 1G PLL.

15.9.4.12.1 Offset

Register	Offset
SYS_PLL1_CTRL	2C0h

15.9.4.12.2 Diagram



15.9.4.12.3 Fields

Field	Description
31 SYS_PLL1_CONTROL_MODE	SYS_PLL1_CONTROL_MODE Enable mode 0 - Software Mode (Default) 1 - GPC Mode
30 SYS_PLL1_AI_BUSY	SYS_PLL1_AI_BUSY For this AI bridge register field, this bit is used along with AI bus. Function is for AI busy monitor. There will be a detail description on how to use it. For debug purpose. Reserved.
29 SYS_PLL1_STABLE	SYS_PLL1_STABLE SYS_PLL1_STABLE Normally in Software mode, after the power up and enable sequence for pll, arm pll will not be in stable state immediately. This bit is used by software to monitor the locking status for pll. GPC mode do not need to take care of this bit. Hardware will handle it automatically.
28	SYS_PLL1_DIV2_CONTROL_MODE

Table continues on the next page...

Field	Description
SYS_PLL1_DIV2_CONTROL_MODE	Enable mode 0 - Software Mode (Default) 1 - GPC Mode
27 SYS_PLL1_DIV5_CONTROL_MODE	SYS_PLL1_DIV5_CONTROL_MODE Enable mode 0 - Software Mode (Default) 1 - GPC Mode
26 SYS_PLL1_DIV5	SYS_PLL1_DIV5 SYS_PLL1_DIV5 The PLL 1g has the source which is DIV5 by the actual frequency of PLL 1G, this bit control of the enable of this divider.
25 SYS_PLL1_DIV2	SYS_PLL1_DIV2 The PLL 1g has the source which is DIV2 by the actual frequency of PLL 1G, this bit control of the enable of this divider.
24 —	Reserved
23-15 —	Reserved
14 SYS_PLL1_GATE	SYS_PLL1_GATE SYS_PLL1_GATE There is a gate in top level to gate the pll output for power saving purpose,this field it to control this gate. 0 - No gate 1 - Gate the output
13 ENABLE_CLK	ENABLE_CLK Enable the clock output. This field is used to align the software mode and GPC mode. The usage of this bit is to describe the last PLL mode(enable or disable) before entering GPC mode. User need to make sure the pll state is matching first Setpoint will be entered.
12-0 —	Reserved

15.9.4.13 SYS_PLL1_DENOMINATOR_REGISTER (SYS_PLL1_DENOMINATOR)

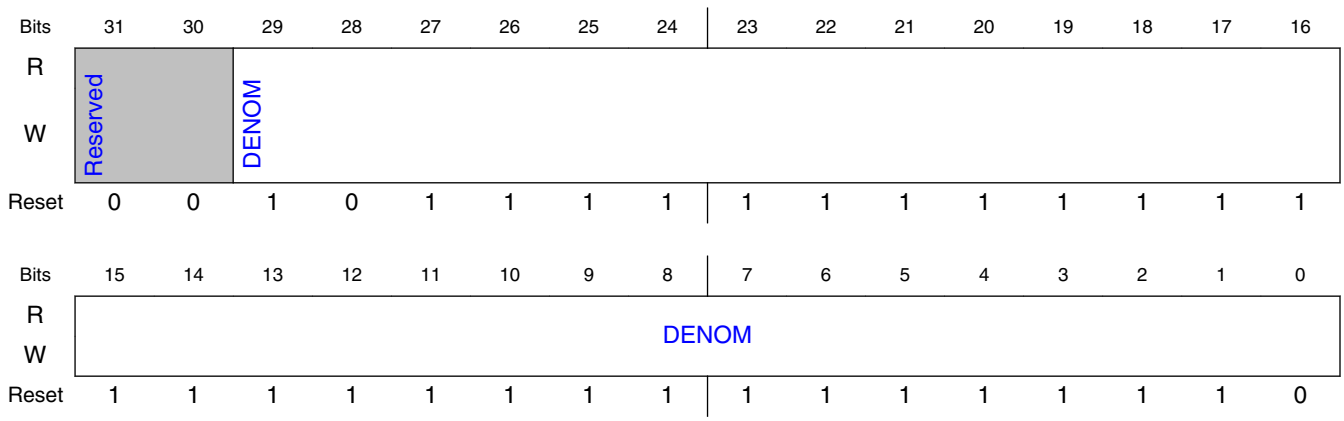
As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

Output frequency=Fref*(MFI+MFN/MFD)

15.9.4.13.1 Offset

Register	Offset
SYS_PLL1_DENOMINATOR	2D0h

15.9.4.13.2 Diagram



15.9.4.13.3 Fields

Field	Description
31-30 —	Reserved
29-0 DENOM	DENOM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

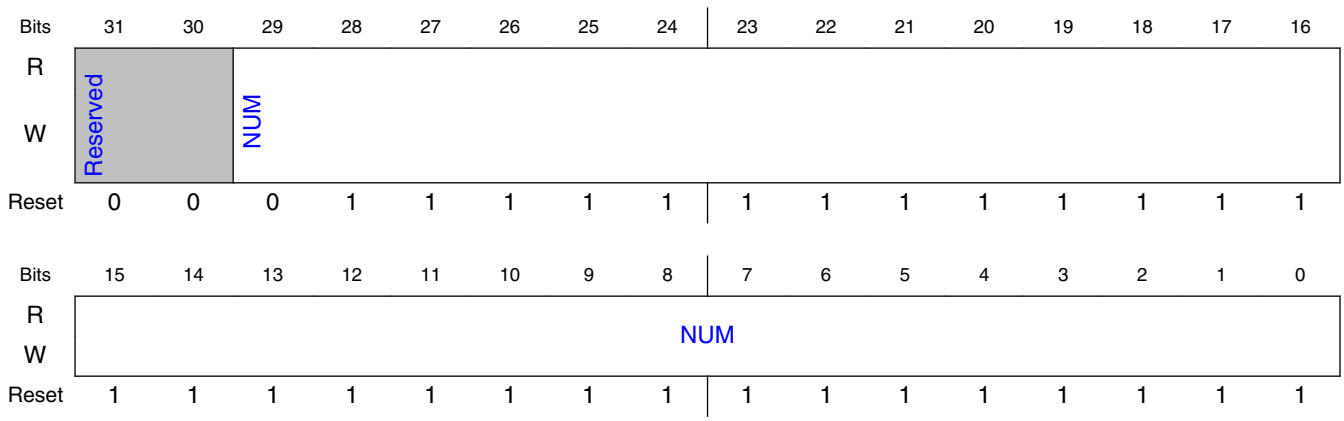
15.9.4.14 SYS_PLL1_NUMERATOR_REGISTER (SYS_PLL1_NUMERATOR)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.14.1 Offset

Register	Offset
SYS_PLL1_NUMERATOR	2E0h

15.9.4.14.2 Diagram



15.9.4.14.3 Fields

Field	Description
31-30 —	Reserved
29-0 NUM	NUM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

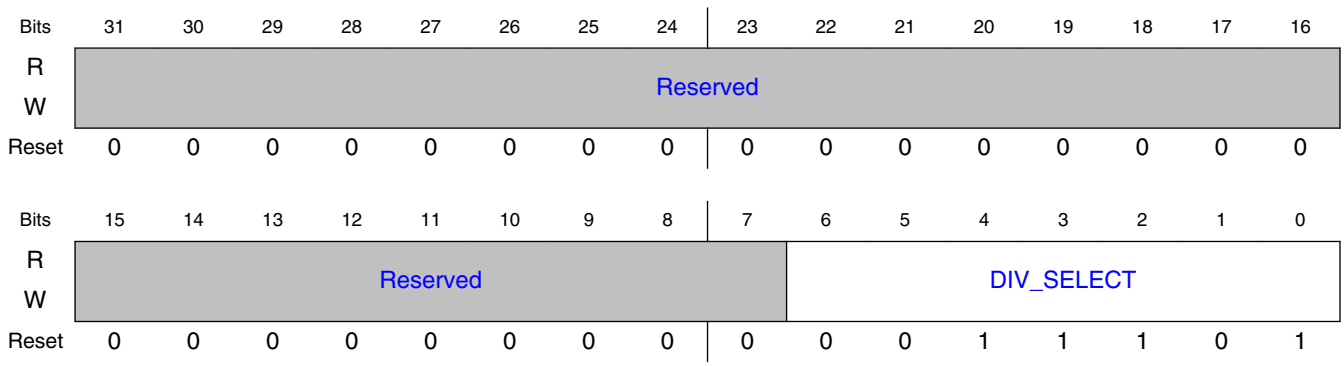
15.9.4.15 SYS_PLL1_DIV_SELECT_REGISTER (SYS_PLL1_DIV_SELECT)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.15.1 Offset

Register	Offset
SYS_PLL1_DIV_SELECT	2F0h

15.9.4.15.2 Diagram



15.9.4.15.3 Fields

Field	Description
31-7 —	Reserved
6-0 DIV_SELECT	DIV_SELECT As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

15.9.4.16 PLL_AUDIO_CTRL_REGISTER (PLL_AUDIO_CTRL)

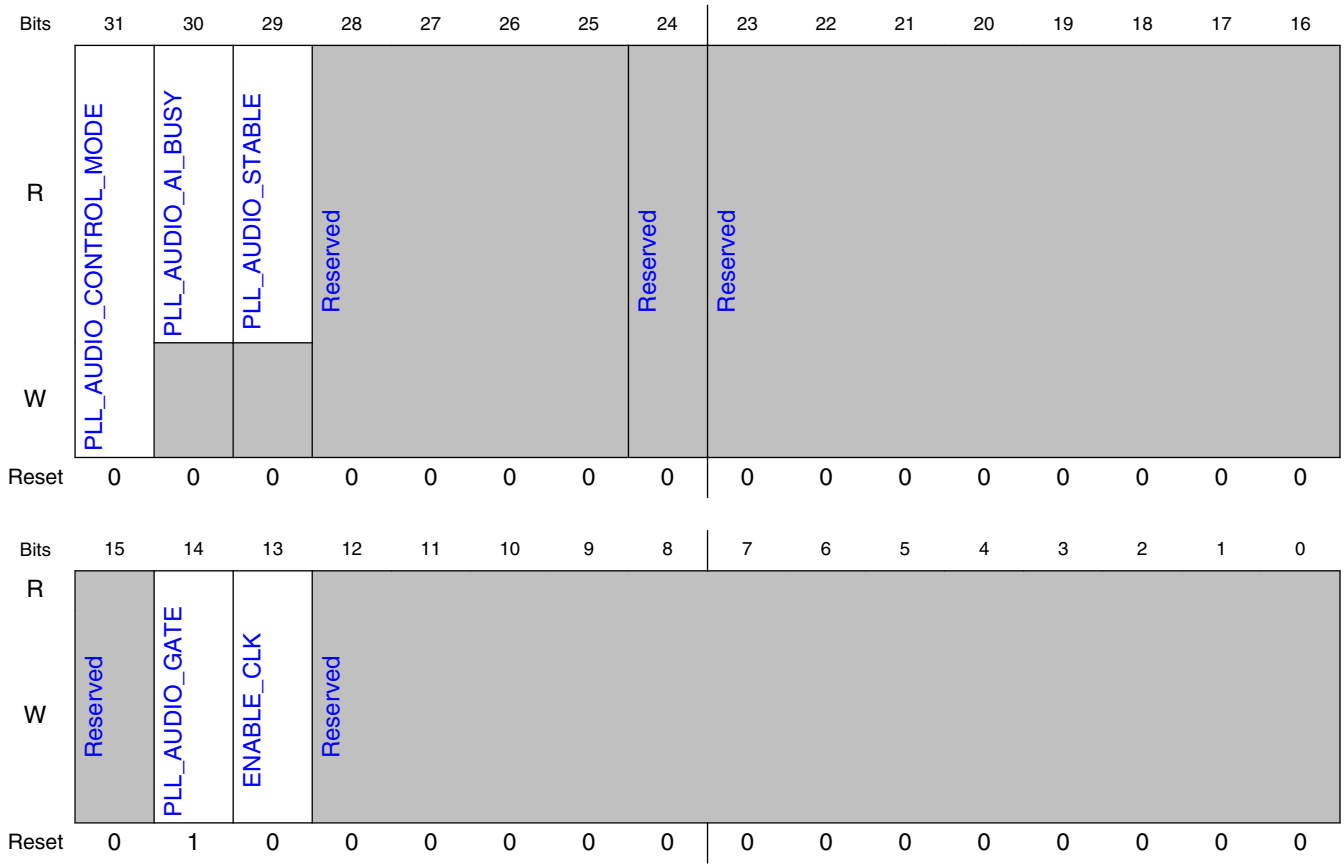
PLL_AUDIO_control_register

The control register provides control for the AUDIO PLL.

15.9.4.16.1 Offset

Register	Offset
PLL_AUDIO_CTRL	300h

15.9.4.16.2 Diagram



15.9.4.16.3 Fields

Field	Description
31 PLL_AUDIO_CONTROL_MODE	pll_audio_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
30 PLL_AUDIO_AI_BUSY	pll_audio_ai_busy For this AI bridge register field, this bit is used along with AI bus. Function is for AI busy monitor. there will be a detail description on how to use it. For debug purpose. Reserved.
29 PLL_AUDIO_STABLE	PLL_AUDIO_STABLE PLL_AUDIO_STABLE Normally in Software mode, after the power up and enable sequence for pll, arm pll will not be in stable state immediately. This bit is used by software to monitor the locking status for pll. GPC mode do not need to take care of this bit. Harfware will handle it automaticly.
28-25	Always set to zero (0).

Table continues on the next page...

Field	Description
—	Always set to zero (0).
24 —	Reserved
23-15 —	Reserved
14 PLL_AUDIO_GATE	PLL_AUDIO_GATE PLL_AUDIO_GATE There is a gate in top level to gate the pll output for power saving purpose,this field it to control this gate. 0 - No gate 1 - Gate the output
13 ENABLE_CLK	ENABLE_CLK Enable the clock output. This field is used to align the software mode and GPC mode. The usage of this bit is to describe the last PLL mode(enable or disable) before entering GPC mode. User need to make sure the pll state is matching first Setpoint will be entered.
12-0 —	Reserved

15.9.4.17 PLL_AUDIO_SS_REGISTER (PLL_AUDIO_SS)

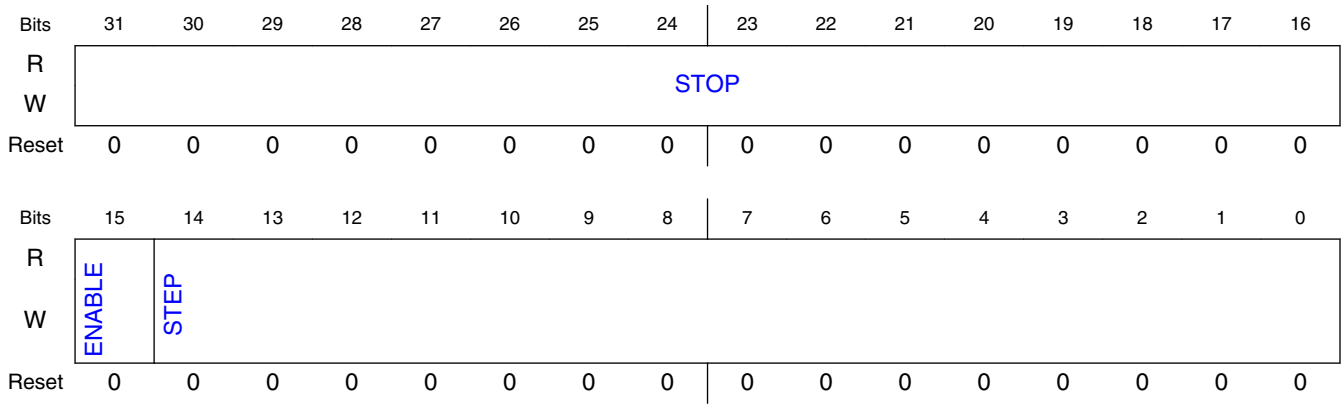
This register contains the AUDIO PLL spread spectrum controls

Refer to Spread Spectrum Parameters for more details.

15.9.4.17.1 Offset

Register	Offset
PLL_AUDIO_SS	310h

15.9.4.17.2 Diagram



15.9.4.17.3 Fields

Field	Description
31-16 STOP	STOP Frequency change = stop/MFD*24MHz.
15 ENABLE	ENABLE This bit enables the spread spectrum modulation 0 - Disable Spread Spectrum. In this setting, the values for STEP and STOP are ignored 1 - Enable Spread Spectrum
14-0 STEP	STEP The max frequency change = step/MFD*24MHz.

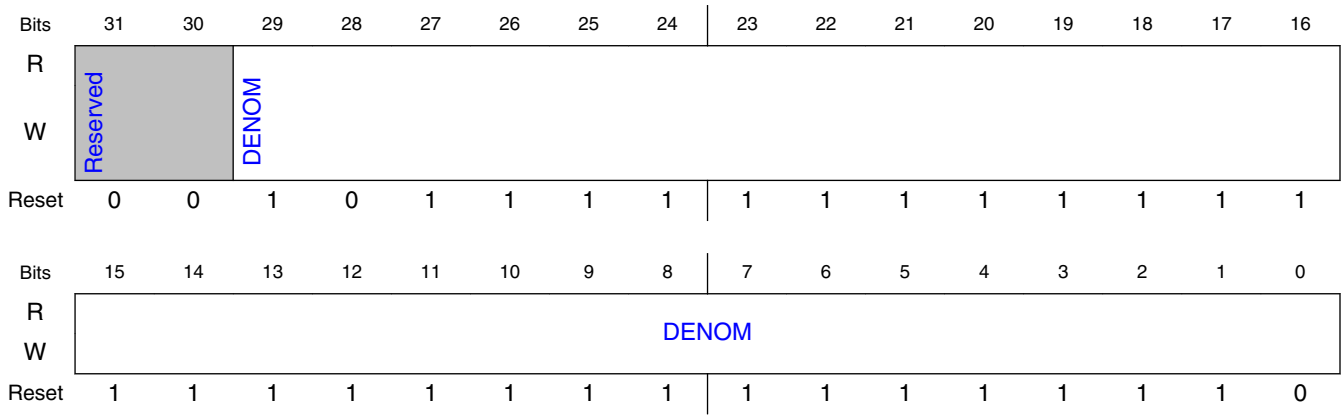
15.9.4.18 PLL_AUDIO_DENOMINATOR_REGISTER (PLL_AUDIO_DENOMINATOR)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.18.1 Offset

Register	Offset
PLL_AUDIO_DENOMINATOR	320h

15.9.4.18.2 Diagram



15.9.4.18.3 Fields

Field	Description
31-30 —	Reserved
29-0 DENOM	DENOM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

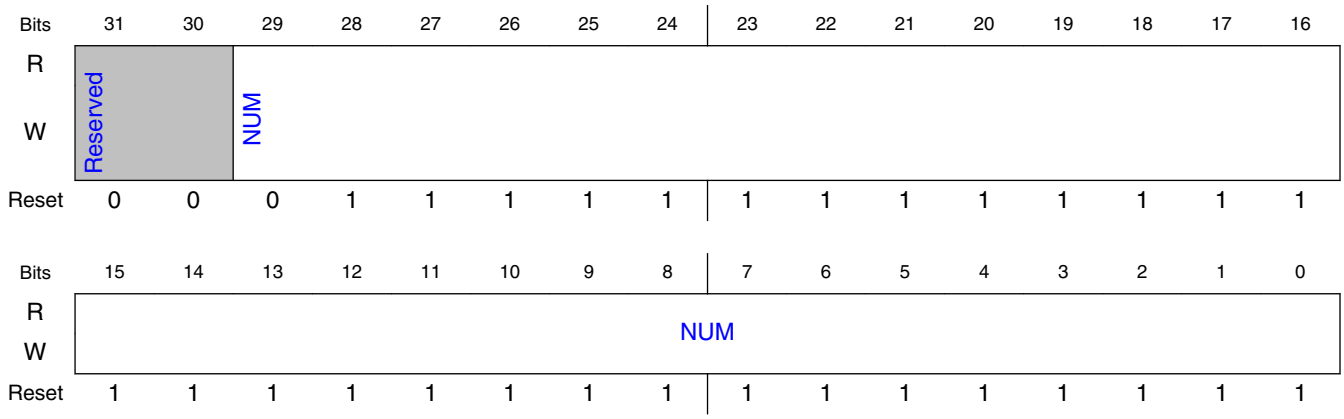
15.9.4.19 PLL_AUDIO_NUMERATOR_REGISTER (PLL_AUDIO_NUMERATOR)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.19.1 Offset

Register	Offset
PLL_AUDIO_NUMERATOR	330h

15.9.4.19.2 Diagram



15.9.4.19.3 Fields

Field	Description
31-30 —	Reserved
29-0 NUM	NUM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

15.9.4.20 PLL_AUDIO_DIV_SELECT_REGISTER (PLL_AUDIO_DIV_SELECT)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.20.1 Offset

Register	Offset
PLL_AUDIO_DIV_SELECT	340h

15.9.4.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PLL_AUDIO_DIV_SELECT							
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1

15.9.4.20.3 Fields

Field	Description
31-7 —	Reserved
6-0 PLL_AUDIO_DIV_SELECT	PLL_AUDIO_DIV_SELECT As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

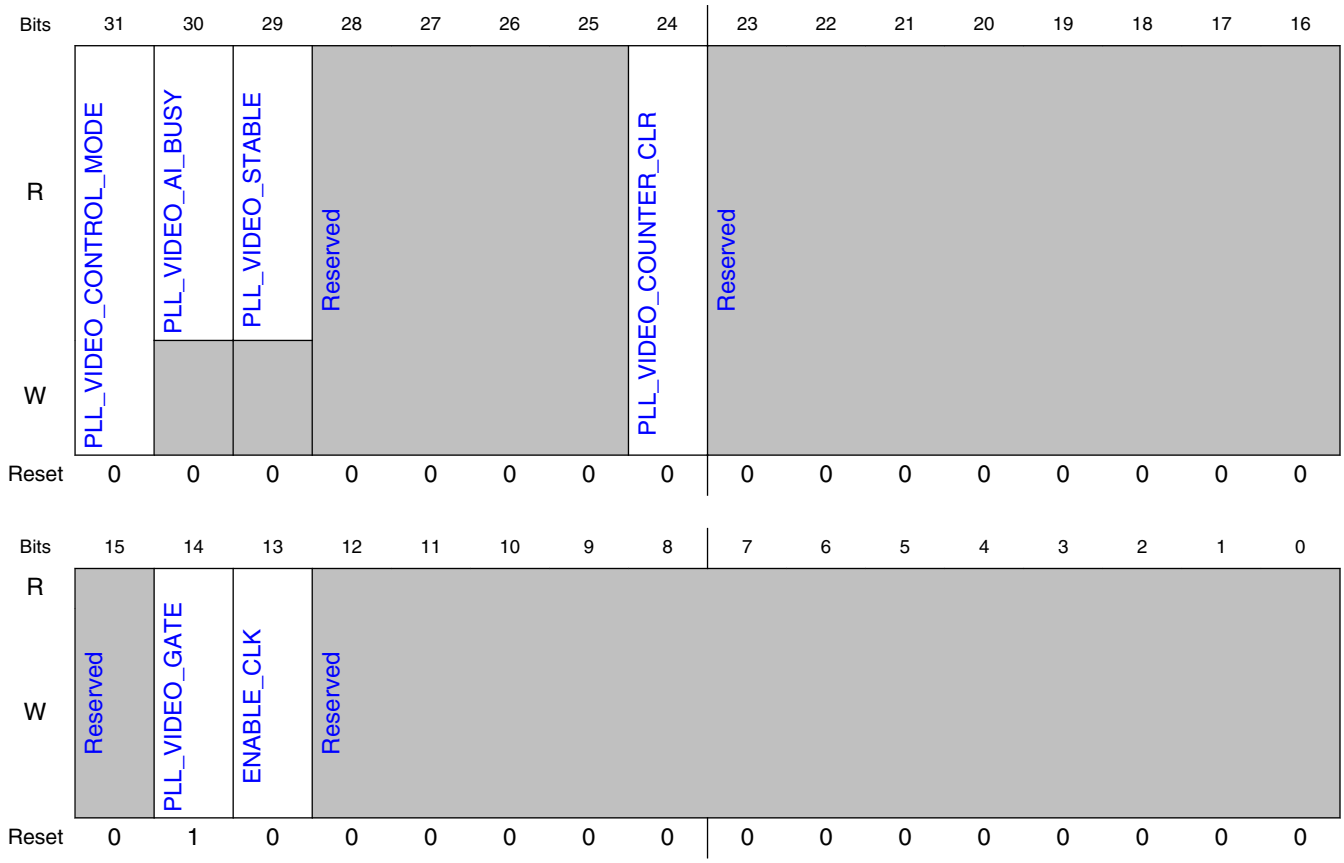
15.9.4.21 PLL_VIDEO_CTRL_REGISTER (PLL_VIDEO_CTRL)

The control register provides control for the VIDEO PLL.

15.9.4.21.1 Offset

Register	Offset
PLL_VIDEO_CTRL	350h

15.9.4.21.2 Diagram



15.9.4.21.3 Fields

Field	Description
31 PLL_VIDEO_CONTROL_MODE	pll_video_control_mode Enable mode 0 - Software Mode (Default) 1 - GPC Mode
30 PLL_VIDEO_AI_BUSY	pll_video_ai_busy For this AI bridge register field, this bit is used along with AI bus. Function is for AI busy monitor.there will be a detail description on how to use it. For debug purpose. Reserved.
29 PLL_VIDEO_STABLE	PLL_VIDEO_STABLE PLL_VIDEO_STABLE Normally in Software mode, after the power up and enable sequence for pll, arm pll will not be in stable state immediately. This bit is used by software to monitor the locking status for pll. GPC mode do not need to take care of this bit. Harfware will handle it automaticly.
28-25 —	Always set to zero (0). Always set to zero (0).

Table continues on the next page...

Field	Description
24 PLL_VIDEO_COUNTER_CLR	pll_video_counter_clr reserved
23-15 —	Reserved
14 PLL_VIDEO_GATE	PLL_VIDEO_GATE PLL_VIDEO_GATE There is a gate in top level to gate the pll output for power saving purpose,this field it to control this gate. 0 - No gate 1 - Gate the output
13 ENABLE_CLK	ENABLE_CLK Enable the clock output. This field is used to align the software mode and GPC mode. The usage of this bit is to describe the last PLL mode(enable or disable) before entering GPC mode. User need to make sure the pll state is matching first Setpoint will be entered.
12-0 —	Reserved

15.9.4.22 PLL_VIDEO_SS_REGISTER (PLL_VIDEO_SS)

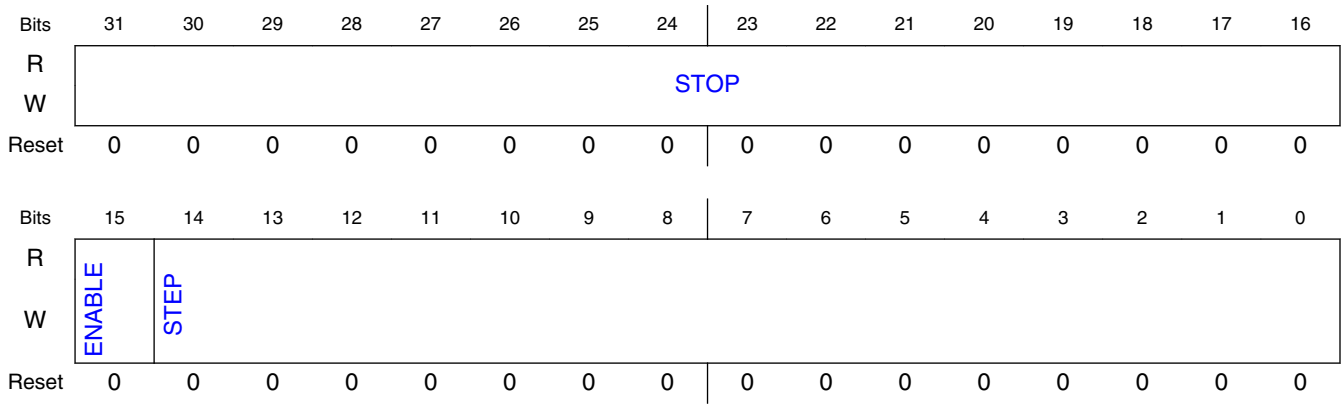
This register contains the VIDEO PLL spread spectrum controls

Refer to Spread Spectrum Parameters for more details.

15.9.4.22.1 Offset

Register	Offset
PLL_VIDEO_SS	360h

15.9.4.22.2 Diagram



15.9.4.22.3 Fields

Field	Description
31-16 STOP	STOP Frequency change = stop/MFD*24MHz.
15 ENABLE	ENABLE This bit enables the spread spectrum modulation 0 - Disable Spread Spectrum. In this setting, the values for STEP and STOP are ignored 1 - Enable Spread Spectrum
14-0 STEP	STEP .The max frequency change = step/MFD*24MHz.

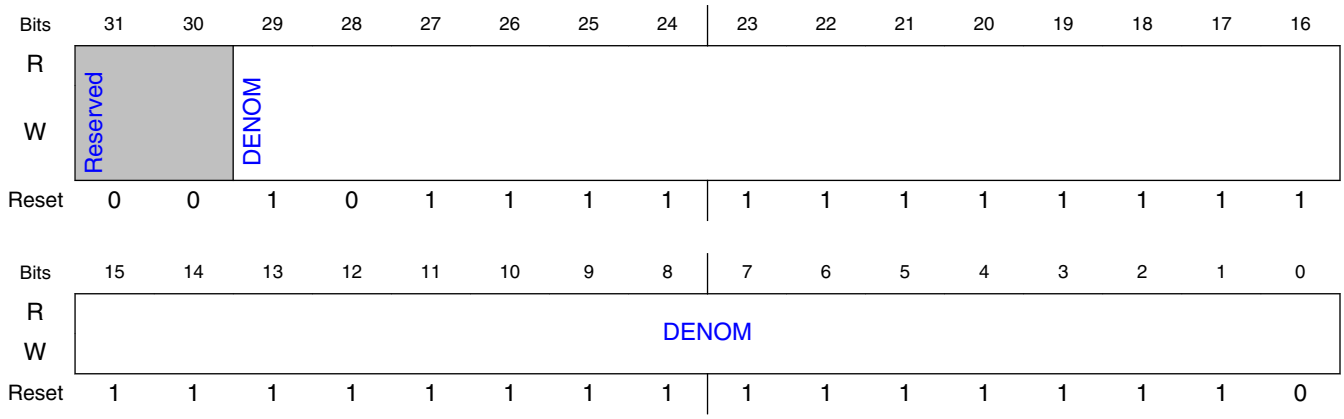
15.9.4.23 PLL_VIDEO_DENOMINATOR_REGISTER (PLL_VIDEO_DENOMINATOR)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.23.1 Offset

Register	Offset
PLL_VIDEO_DENOMINATOR	370h

15.9.4.23.2 Diagram



15.9.4.23.3 Fields

Field	Description
31-30 —	Reserved
29-0 DENOM	DENOM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

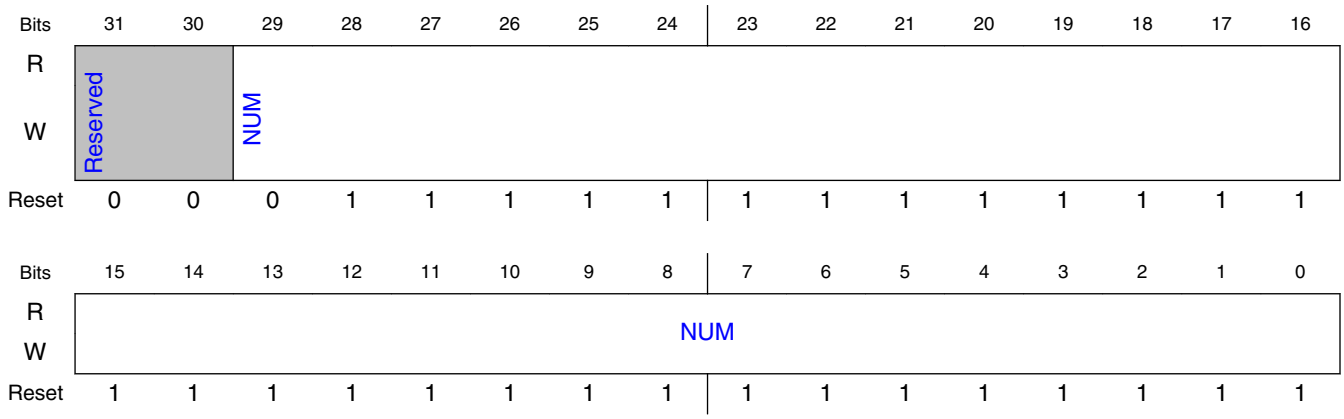
15.9.4.24 PLL_VIDEO_NUMERATOR_REGISTER (PLL_VIDEO_NUMERATOR)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.24.1 Offset

Register	Offset
PLL_VIDEO_NUMERATOR	380h

15.9.4.24.2 Diagram



15.9.4.24.3 Fields

Field	Description
31-30 —	Reserved
29-0 NUM	NUM As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

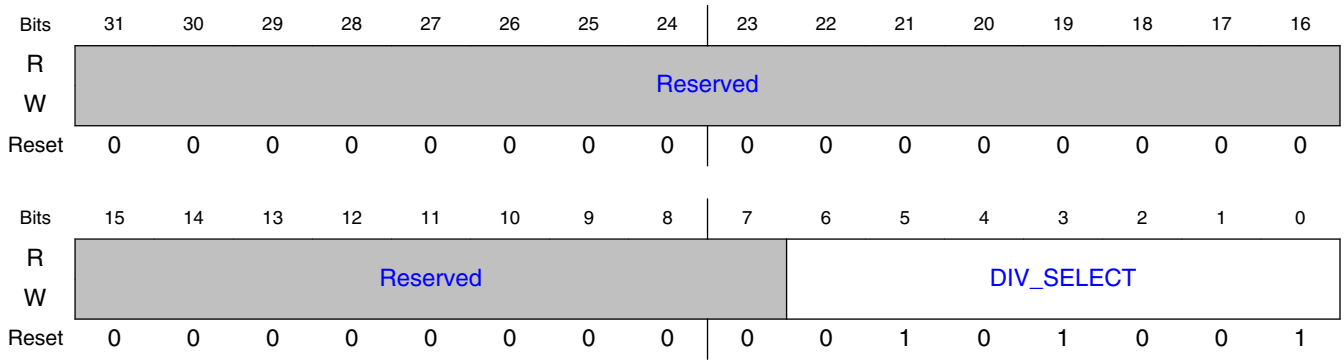
15.9.4.25 PLL_VIDEO_DIV_SELECT_REGISTER (PLL_VIDEO_DIV_SELECT)

As this AI interface PLL is used both in Software mode and GPC mode. This field is an entry to GPC mode for the corresponding field.

15.9.4.25.1 Offset

Register	Offset
PLL_VIDEO_DIV_SELECT	390h

15.9.4.25.2 Diagram



15.9.4.25.3 Fields

Field	Description
31-7 —	Reserved
6-0 DIV_SELECT	DIV_SELECT As this is AI interface PLL, there will be two modes support for this PLL, software mode and GPC mode. And this file is providing the entry to the corresponding field in GPC mode.

15.9.5 Fractional PLL register descriptions

NOTE

The CCM Fractional PLL registers are accessed through the Analog IP (AI) Interface.

15.9.5.1 FRACTIONAL_PLL memory map

frac_pll base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Fractional PLL Control Register (CTRL0)	32	RW	0000_0000
4	Fractional PLL Control Register (CTRL0_SET)	32	RW	0000_0000
8	Fractional PLL Control Register (CTRL0_CLR)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C	Fractional PLL Control Register (CTRL0_TOG)	32	RW	0000_0000
10	Fractional PLL Spread Spectrum Control Register (SPREAD_SPECTRUM)	32	RW	0000_0000
14	Fractional PLL Spread Spectrum Control Register (SPREAD_SPECTRUM_SET)	32	RW	0000_0000
18	Fractional PLL Spread Spectrum Control Register (SPREAD_SPECTRUM_CLR)	32	RW	0000_0000
1C	Fractional PLL Spread Spectrum Control Register (SPREAD_SPECTRUM_TOG)	32	RW	0000_0000
20	Fractional PLL Numerator Control Register (NUMERATOR)	32	RW	0000_0000
24	Fractional PLL Numerator Control Register (NUMERATOR_SET)	32	RW	0000_0000
28	Fractional PLL Numerator Control Register (NUMERATOR_CLR)	32	RW	0000_0000
2C	Fractional PLL Numerator Control Register (NUMERATOR_TOG)	32	RW	0000_0000
30	Fractional PLL Denominator Control Register (DENOMINATOR)	32	RW	0000_0000
34	Fractional PLL Denominator Control Register (DENOMINATOR_SET)	32	RW	0000_0000
38	Fractional PLL Denominator Control Register (DENOMINATOR_CLR)	32	RW	0000_0000
3C	Fractional PLL Denominator Control Register (DENOMINATOR_TOG)	32	RW	0000_0000

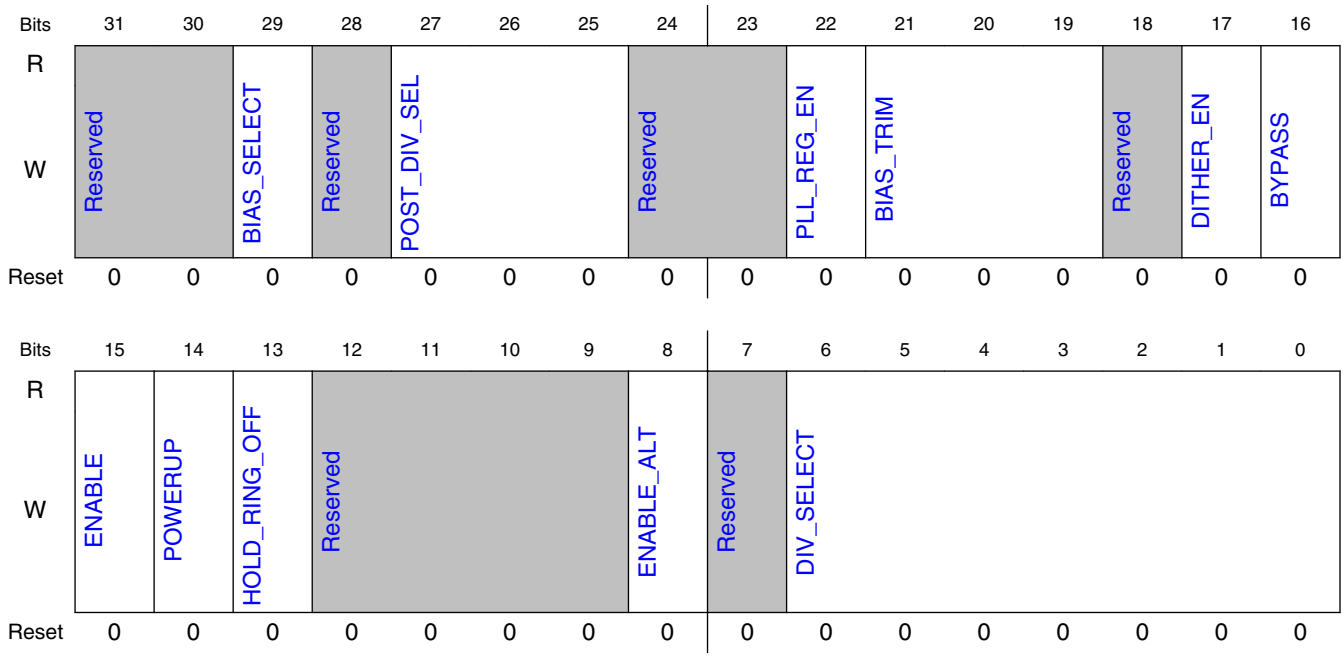
15.9.5.2 Fractional PLL Control Register (CTRL0)

This register contains fractional pll control registers.

15.9.5.2.1 Offset

Register	Offset	Description
CTRL0	0h	Fractional PLL Control Register
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

15.9.5.2.2 Diagram



15.9.5.2.3 Fields

Field	Description
31-30 —	Reserved
29 BIAS_SELECT	BIAS_SELECT Selects which input bias current is expected by the PLL. 0 - Used in SoCs with a bias current of 10uA 1 - Used in SoCs with a bias current of 2uA
28 —	Reserved
27-25 POST_DIV_SEL	Post Divide Select Fractional PLL Post Divider Select (glitchless clock mux) 000 - Divide by 1 001 - Divide by 2 010 - Divide by 4 011 - Divide by 8 100 - Divide by 16 101 - Divide by 32
24-23 —	Reserved

Table continues on the next page...

Field	Description
22 PLL_REG_EN	PLL_REG_EN Enables the internal PLL regulator.
21-19 BIAS_TRIM	BIAS_TRIM Not currently used.
18 —	Reserved
17 DITHER_EN	DITHER_EN Enables dither in the fractional modulator calculation. 0 - Disable Dither 1 - Enable Dither
16 BYPASS	BYPASS Byapss the PLL. Output is the PLL reference clock source 0 - No Bypass 1 - Bypass the PLL
15 ENABLE	ENABLE 0 - Disable the clock output 1 - Enable the clock output
14 POWERUP	POWERUP 0 - Power down the PLL 1 - Power Up the PLL
13 HOLD_RING_OFF	PLL Start up initialization This field should be set to 1 every time during PLL lock, and cleared after a certain delay. 0 - Normal operation 1 - Initialize PLL start up
12-9 —	Reserved
8 ENABLE_ALT	ENABLE_ALT 0 - Disable the alternate clock output 1 - Enable the alternate clock output which is the output of the post_divider, and cannot be bypassed
7 —	Reserved
6-0 DIV_SELECT	DIV_SELECT This field controls the pll loop divider. Valid range for DIV_SELECT divider value: 27-54 decimal.

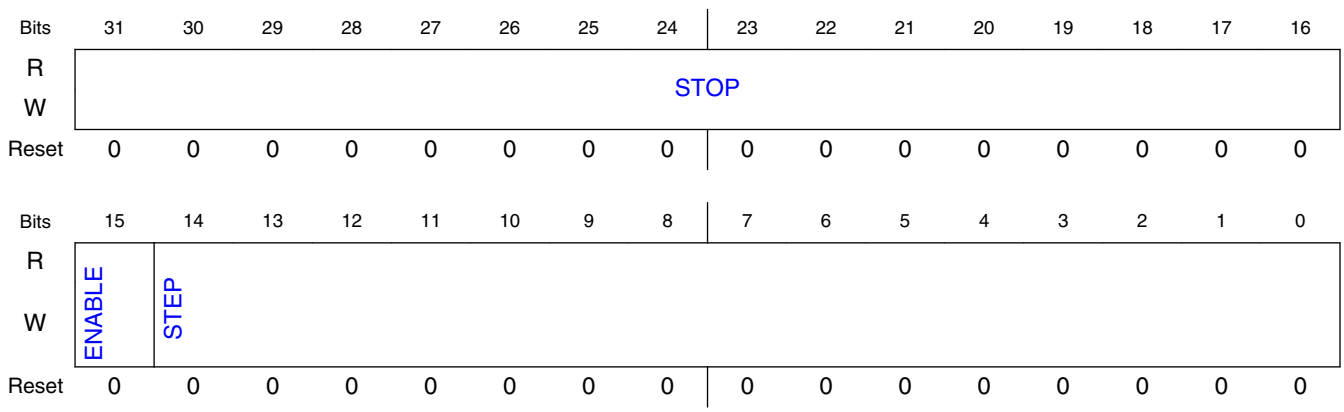
15.9.5.3 Fractional PLL Spread Spectrum Control Register (SPREAD_SPECTRUM)

This register contains fractional pll spread spectrum control registers.

15.9.5.3.1 Offset

Register	Offset	Description
SPREAD_SPECTRUM	10h	Fractional PLL Spread Spectrum Control Register
SPREAD_SPECTRUM_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in SPREAD_SPECTRUM
SPREAD_SPECTRUM_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in SPREAD_SPECTRUM
SPREAD_SPECTRUM_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in SPREAD_SPECTRUM

15.9.5.3.2 Diagram



15.9.5.3.3 Fields

Field	Description
31-16	Stop
STOP	Spread Spectrum Stop Register. Frequency change = step/MFD*24MHz.
15	Enable
ENABLE	This bit enables the spread spectrum modulation.
14-0	Step
STEP	Spread Spectrum Step The max frequency change = stop/MFD*24MHz.

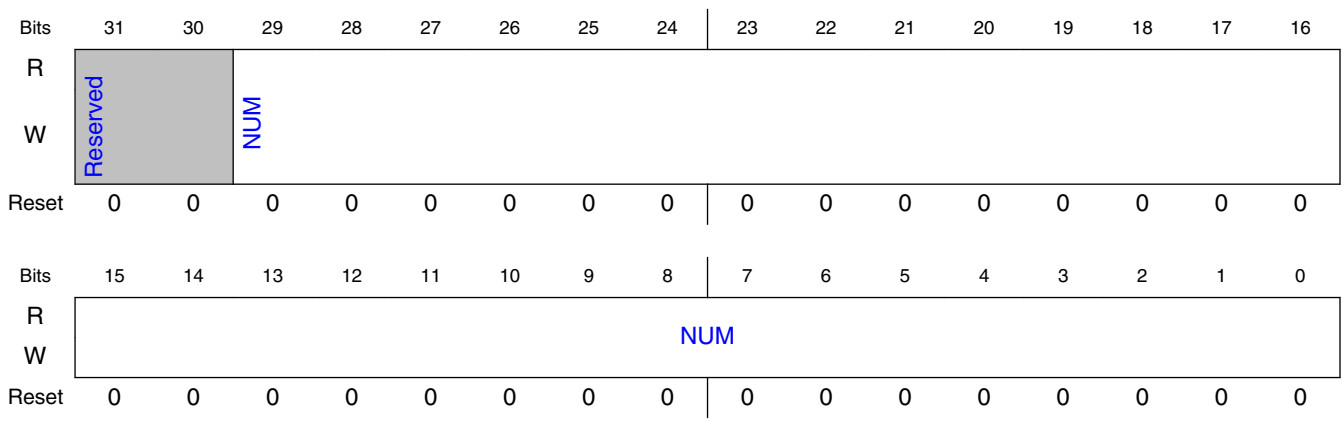
15.9.5.4 Fractional PLL Numerator Control Register (NUMERATOR)

This register contains fractional pll numerator control registers.

15.9.5.4.1 Offset

Register	Offset	Description
NUMERATOR	20h	Fractional PLL Numerator Control Register
NUMERATOR_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in NUMERATOR
NUMERATOR_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in NUMERATOR
NUMERATOR_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in NUMERATOR

15.9.5.4.2 Diagram



15.9.5.4.3 Fields

Field	Description
31-30 —	Reserved
29-0 NUM	Numerator Numerator of PLL Fractional Loop Divider. Signed number.

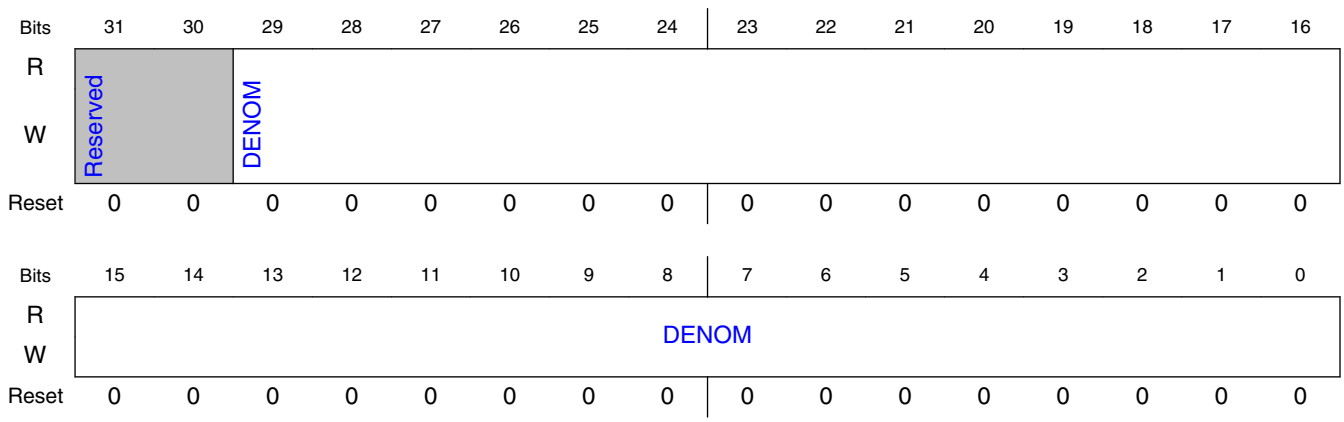
15.9.5.5 Fractional PLL Denominator Control Register (DENOMINATOR)

This register contains fractional pll denominator control registers.

15.9.5.5.1 Offset

Register	Offset	Description
DENOMINATOR	30h	Fractional PLL Denominator Control Register
DENOMINATOR_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in DENOMINATOR
DENOMINATOR_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in DENOMINATOR
DENOMINATOR_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in DENOMINATOR

15.9.5.5.2 Diagram



15.9.5.5.3 Fields

Field	Description
31-30 —	Reserved
29-0 DENOM	Denominator Denominator of Fractional Loop Divider. Unsigned number.

Chapter 16
Crystal Oscillator (XTALOSC)

16.1 Chip-specific XTALOSC information

Table 16-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

16.2 Overview

This block details the function and control of the oscillators on the chip.

This chip contains two Crystal Oscillators for clock generation - 24MHz, and 32KHz. In addition, it also contains four RC oscillators - 48MHz, 32KHz, 16MHz, and 400MHz.

16.2.1 Block Diagram

The Crystal Oscillator block diagram is shown below.

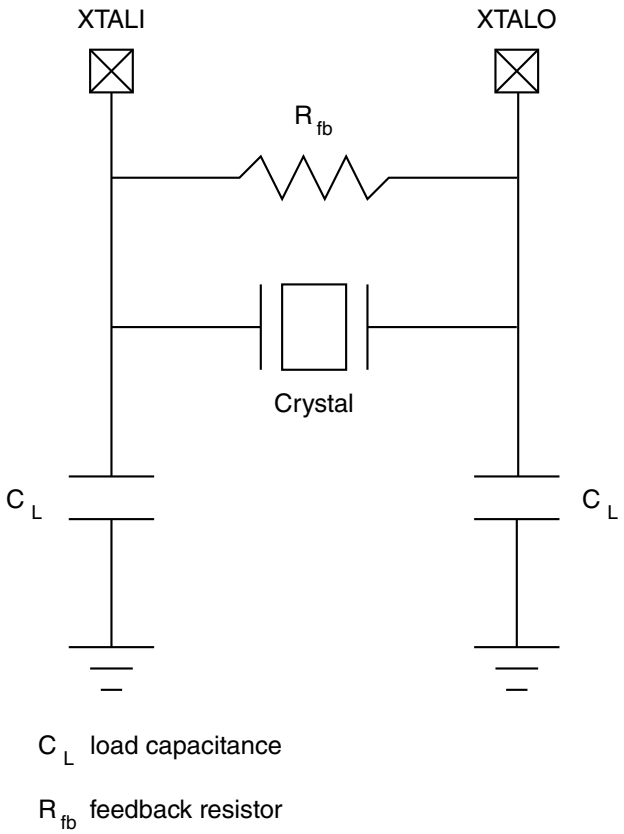


Figure 16-1. Block diagram

The Rfb resistor is used in high gain mode. It is not used in low power mode.

16.3 Functional Description

This section details the oscillator clock sources to the chip.

16.3.1 Crystal Oscillators

The XTALOSC has two external crystal oscillators:

- **24MHz XTAL oscillator**
  - Primary clock source for all PLLs that generate the clocks for the CPU, Bus, and high-speed interfaces.
  - This clock source can be bypassed by the user, and replaced with another external clock source.
- **32KHz XTAL oscillator**
  - Primary clock source for RTC as well as low-speed clock source for CCM/SRC/GPC.

16.3.1.1 Bypass configuration (24MHz XTAL oscillator)

If it is desired to drive the chip with an external clock source, then it should be connected to XTALI and OSC_24M_CTRL[BYPASS_EN] should be set to 1. XTALO should be left externally floating.

16.3.2 RC Oscillators

The XTALOSC is supported by RC Oscillators (RCOSC). This chip has four RCOSC:

- **48MHz RC oscillator**
  - Generates the 24MHz clock source for the chip during start-up when the external 24MHz crystal is not ready.
  - Alternative 24MHz clock source in low power mode when the crystal oscillator is turned off, or in system which does not have any crystal oscillator.
- **32KHz RC oscillator**
  - 32KHz clock source for the chip during start-up when the external 32KHz crystal is not ready.
  - Alternative 32KHz clock source when the external 32KHz crystal oscillator is not stable, or in system which does not have any crystal oscillator.
- **16MHz RC oscillator**
  - Clock source in low power mode.
- **400MHz RC oscillator**
  - Clock source during chip boot up before PLL is enabled or in low-power mode when PLL is turned off.

NOTE

- The switch from external 32KHz XTALOSC to internal 32KHz RCOSC is controlled by hardware and occurs automatically when the system detects a loss of 32KHz crystal clock.
- To properly switch between 24MHz XTAL and 48MHz internal RC, ensure the PLL is powered down before the switch. After switching the clock, power up PLL.

16.3.3 Analog IP (AI) Interface

The chip has various analog IP and components that are controlled via an AI interface. The AI control registers are used to indirectly program the analog components on the AI bus. The following table shows the AI interface registers relative to their respective analog components and power domains.

Power Domain	Analog Component	AI Registers
SOC2PLL	SYS_PLL1 (1G)	VDDSOC2PLL_AI_CTRL_1G
	AUDIO_PLL	VDDSOC2PLL_AI_WDATA_1G
	VIDEO_PLL	VDDSOC2PLL_AI_RDATA_1G
		VDDSOC2PLL_AI_CTRL_AUDIO
		VDDSOC2PLL_AI_WDATA_AUDIO
		VDDSOC2PLL_AI_RDATA_AUDIO
		VDDSOC2PLL_AI_CTRL_VIDEO
		VDDSOC2PLL_AI_WDATA_VIDEO
		VDDSOC2PLL_AI_RDATA_VIDEO
SOC	PHY LDO	VDDSOC_AI_CTRL
		VDDSOC_AI_WDATA
		VDDSOC_AI_RDATA
		PMU_LDO_PLL
LPSR	Bandgap	VDDLPSR_AI_CTRL
	Tempsensor	VDDLPSR_AI_WDATA
	RCOSC_400M	VDDLPSR_AI_RDATA_REFTOP
		PMU_REF_CTRL
		VDDLPSR_AI400M_CTRL
		VDDLPSR_AI400M_WDATA
		VDDLPSR_AI400M_RDATA
		VDDLPSR_AI_RDATA_TMPSNS

Follow the steps below to perform register accesses through the AI interface.

Steps to perform a write operation of an analog component's AI interface:

1. Write *AI_CTRL register
   - Set RWB value to 0
   - Set ADDR value to the analog component's relative AI control register to be accessed
2. Write *AI_WDATA
   - Set Data[31:0] value to the analog component's relative AI control register value to be written
3. Write *CTRL IP Toggle

- Set TOGGLE value to 1 in the bit position for the respective analog component. The toggle will initiate the write sequence.

Steps to perform a read operation of an analog component's AI interface:

1. Write *AI_CTRL register
   - Set RWB value to 1
   - Set ADDR value to the analog component's relative AI control register to be accessed
2. Write *CTRL IP Toggle
   - Set TOGGLE value to 1 in the bit position for the respective analog component. The toggle will initiate the read sequence
3. Read *CTRL
   - Poll this register until the respective analog component's busy bit is low, indicating the read transaction has completed.
4. Read *AI_RDATA
   - Read Data[31:0]. The contents of the Read Data register is the value of the analog component's respective AI register's contents.

NOTE

Varying IP will have a corresponding identical toggle bit

NOTE

For PLL operations, follow the PLL enable sequence first.

16.4 External Signals

The table found here describes the external signals of XTALOSC:

Table 16-2. XTALOSC External Signals

Signal	Description	Pad	Mode	Direction
RTC_XTALI	Real-time clock crystal oscillator input	RTC_XTALI	No muxing	I
RTC_XTALO	Real-time clock crystal oscillator output	RTC_XTALO	No muxing	O
XTALI	Crystal oscillator input signal	XTALI	No muxing	I
XTALO	Crystal oscillator output signal	XTALO	No muxing	O

16.5 Memory Map and register definition

This section includes the module memory map and detailed descriptions of all registers.

NOTE

For related IPS Domain Slot control, please see IPS Domain Registers in Power Management Unit (PMU) chapter.

16.5.1 OSC register descriptions

16.5.1.1 OSC memory map

osc base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
10	48MHz RCOSC Control Register (OSC_48M_CTRL)	32	RW	0179_01F2
20	24MHz OSC Control Register (OSC_24M_CTRL)	32	RW	0000_0080
40	400MHz RCOSC Control0 Register (OSC_400M_CTRL0)	32	RW	0000_0000
50	400MHz RCOSC Control1 Register (OSC_400M_CTRL1)	32	RW	0000_0001
60	400MHz RCOSC Control2 Register (OSC_400M_CTRL2)	32	RW	0000_0000
C0	16MHz RCOSC Control Register (OSC_16M_CTRL)	32	RW	0000_0007

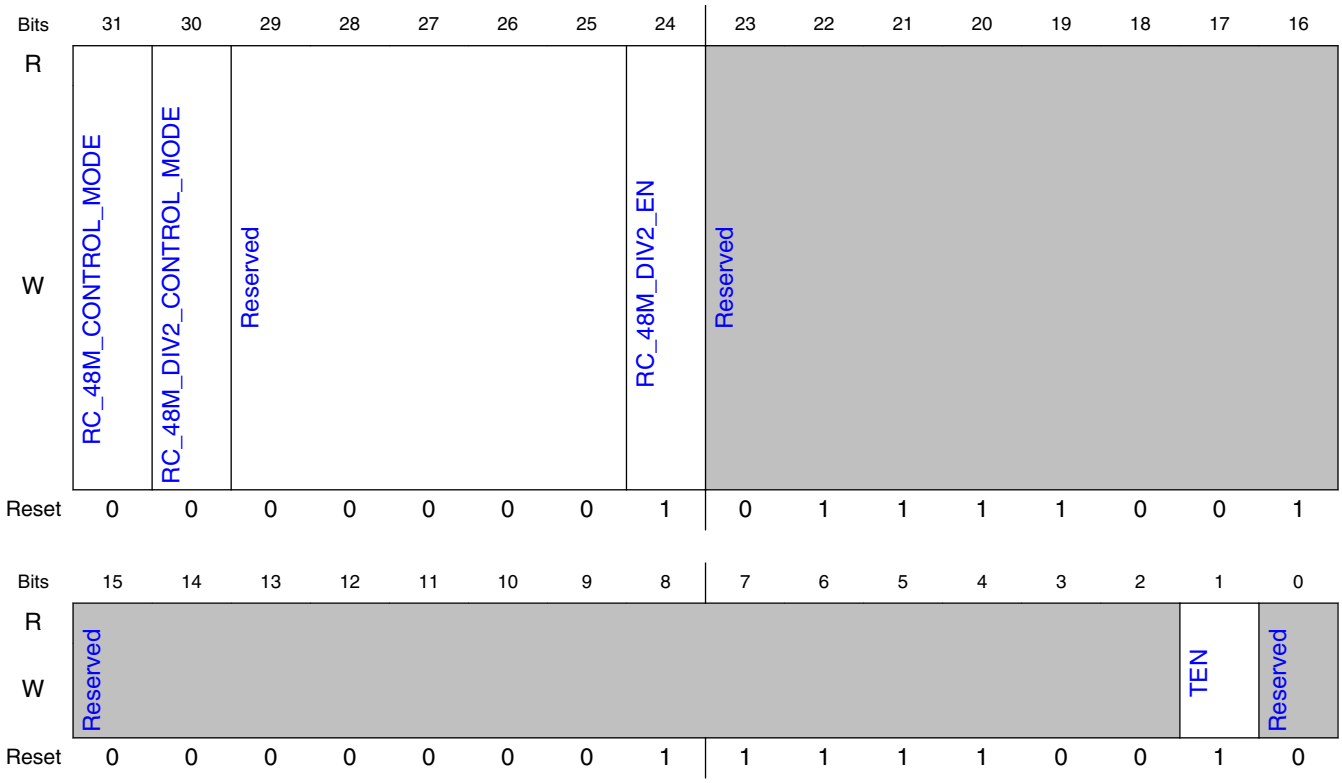
16.5.1.2 48MHzRCOSC Control Register (OSC_48M_CTRL)

This register is used to control the 48MHz RC oscillator.

16.5.1.2.1 Offset

Register	Offset
OSC_48M_CTRL	10h

16.5.1.2.2 Diagram



16.5.1.2.3 Fields

Field	Description
31 RC_48M_CONTROL_MODE	48MHzRCOSC Control Mode This field selects between GPC mode and Software mode 0 - Software mode (default) 1 - GPC mode (Setpoint)
30 RC_48M_DIV2_CONTROL_MODE	RCOSC_48M_DIV2 Control Mode This field selects between GPC mode and Software mode 0 - Software mode (default) 1 - GPC mode (Setpoint)
29-25 —	Always set to zero
24 RC_48M_DIV2_EN	RCOSC_48M_DIV2 Enable This field enables/disables the 24MHz clock sourced from 48MHz RCOSC 0 - Disable 1 - Enable
23-2	Reserved

Table continues on the next page...

Field	Description
—	
1 TEN	48MHzRCOSC Enable This field powers up or powers down the 48MHz RCOSC 0 - Power down 1 - Power up
0 —	Reserved

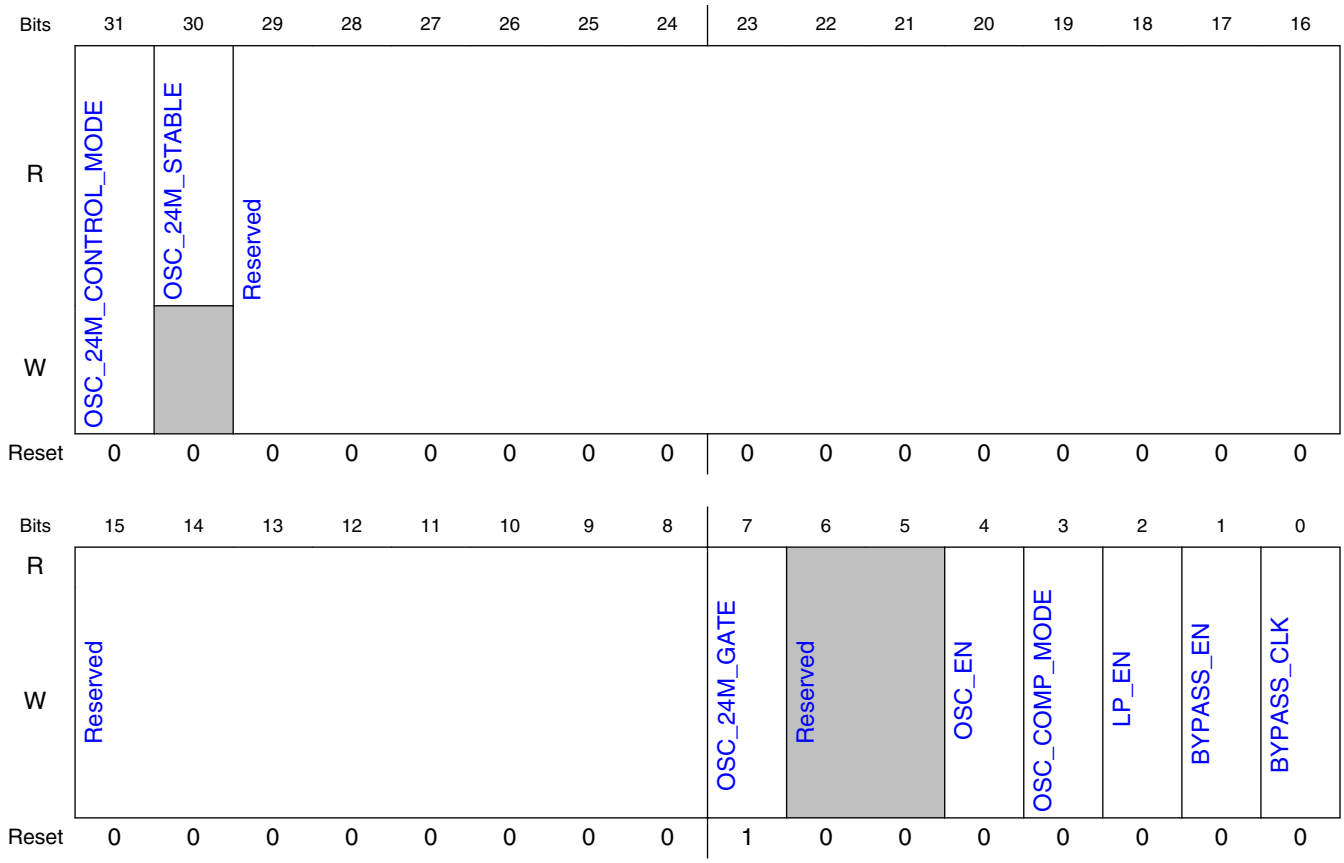
16.5.1.3 24MHz OSC Control Register (OSC_24M_CTRL)

This register is used to control the 24MHz Oscillator.

16.5.1.3.1 Offset

Register	Offset
OSC_24M_CTRL	20h

16.5.1.3.2 Diagram



16.5.1.3.3 Fields

Field	Description
31 OSC_24M_CONTROL_MODE	24MHz OSC Control Mode This field selects between GPC mode and Software mode 0 - Software mode (default) 1 - GPC mode (Setpoint)
30 OSC_24M_STABLE	24MHz OSC Stable This field indicates whether 24MHzOSC is stable (24MHz OSC will not be stable the time it is enabled) 0 - Not Stable 1 - Stable
29-8 —	Always set to zero
7 OSC_24M_GATE	24MHz OSC Gate Control This field gates the 24MHz OSC output for saving power 0 - Not Gated

Table continues on the next page...

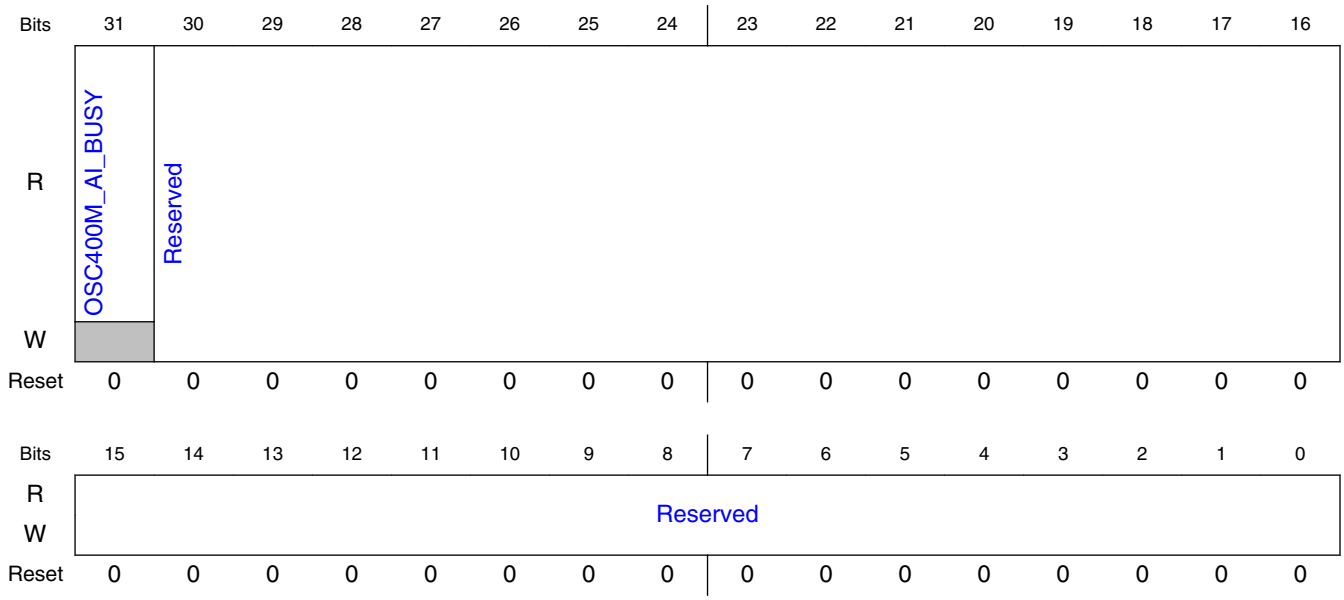
Field	Description
	1 - Gated
6-5 —	Reserved
4 OSC_EN	24MHzOSC Enable This field enables the 24MHz XTALOSC 0 - Disable 1 - Enable
3 OSC_COMP_MODE	24MHz OSC Comparator Mode This field selects the ac-coupling comparator operation-mode 0 - Single-ended mode (default) 1 - Differential mode (test mode)
2 LP_EN	24MH OSC Low-Power Mode Enable 0 - High Gain mode (HP) 1 - Low-power mode (LP)
1 BYPASS_EN	24MHz OSC Bypass Enable Enable signal for external bypass clock (mostly connect to XTAL directly in SOC level). Users can use this filed to enable an external source of XTAL. 0 - Disable 1 - Enable
0 BYPASS_CLK	24MHz OSC Bypass Clock External Bypass Clock from pad (1.8v)

16.5.1.4 400MHz RCOSC Control0 Register (OSC_400M_CTRL0)

16.5.1.4.1 Offset

Register	Offset
OSC_400M_CTRL0	40h

16.5.1.4.2 Diagram



16.5.1.4.3 Fields

Field	Description
31 OSC400M_AI_BUSY	400MHz OSC AI BUSY This field is associated with AI busy monitor function in AI bus. Please see 'Analog IP (AI) Interface' topic for more details.
30-0 —	Always set to zero

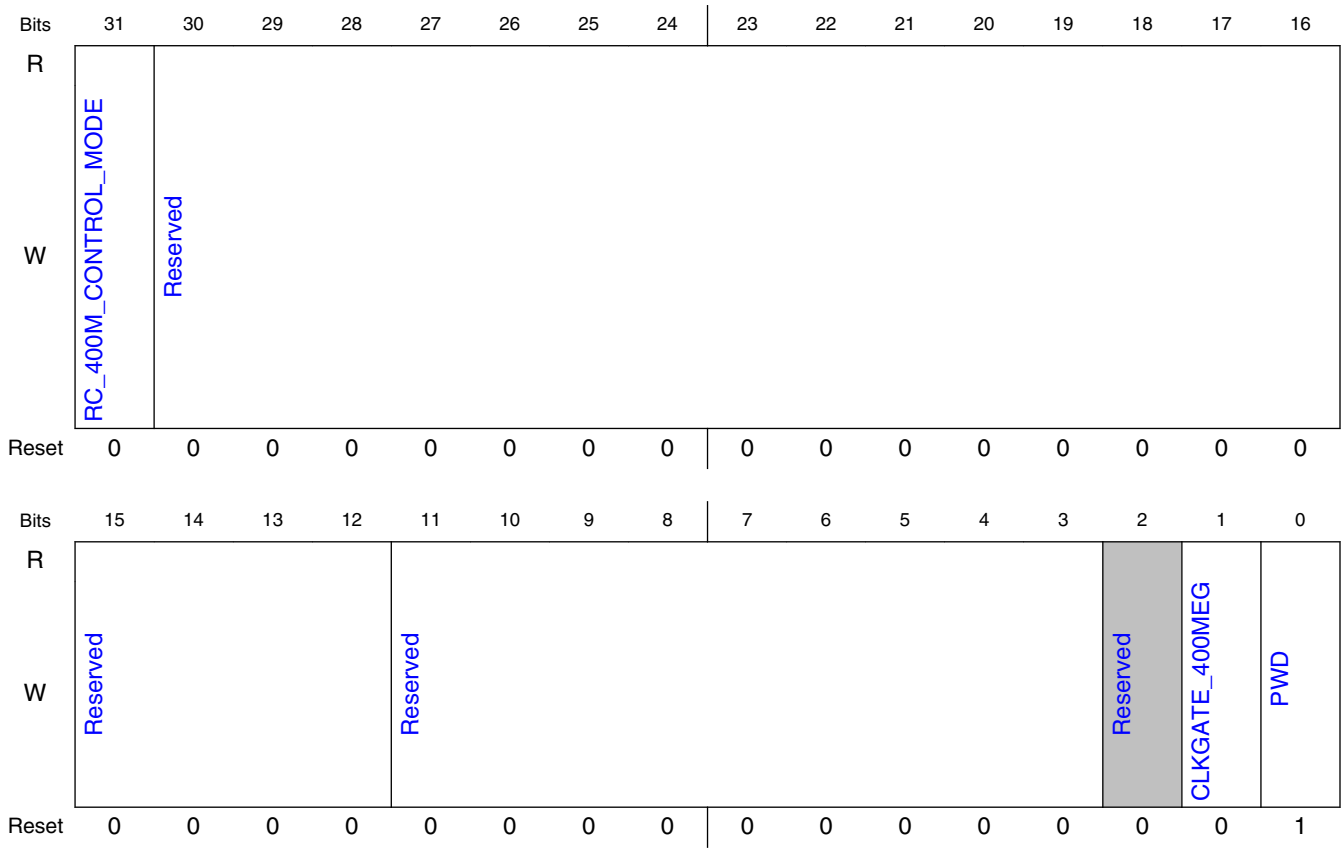
16.5.1.5 400MHz RCOSC Control1 Register (OSC_400M_CTRL1)

This register provides the controls for clock gate, power down, and 400MHz RCOSC mode selects.

16.5.1.5.1 Offset

Register	Offset
OSC_400M_CTRL1	50h

16.5.1.5.2 Diagram



16.5.1.5.3 Fields

Field	Description
31 RC_400M_CONTROL_MODE	rc_400m_control_mode This field selects between GPC mode and Software mode 0 - Software mode (default) 1 - GPC mode (Setpoint)
30-12 —	Always set to zero
11-3 —	Always set to zero
2 —	Reserved
1 CLKGATE_400MEG	Clock gate control for 400MHz RCOSC 0 - Not Gated 1 - Gated
0	Power down control for 400MHz RCOSC

Field	Description
PWD	400MHz RCOSC can be powered down directly with this bit. There is no need to follow AI bus sequence 0 - No Power down 1 - Power down

16.5.1.6 400MHz RCOSC Control2 Register (OSC_400M_CTRL2)

This register provides the OSC tune value and clock enable control used for HW mode.

16.5.1.6.1 Offset

Register	Offset
OSC_400M_CTRL2	60h

16.5.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSC_TUNE_VAL								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved			Reserved	TUNE_BYP	Reserved		Reserved							ENABLE_CLK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.1.6.3 Fields

Field	Description
31-24 OSC_TUNE_VAL	Oscillator Tune Value These bits determine the frequency of oscillator when tuning is not enabled. Frequency varies with PVT. 0 - Lowest frequency 255 - Highest frequency
23-15	Always set to zero

Table continues on the next page...

Field	Description
—	
14-12 —	Always set to zero
11 —	Always set to zero
10 TUNE_BYP	Bypass tuning logic 0 - Use the output of tuning logic to run the oscillator 1 - Bypass the tuning logic and use the programmed OSC_TUNE_VAL to run the oscillator
9-8 —	Always set to zero
7-1 —	Always set to zero
0 ENABLE_CLK	Clock enable This bit is used for the switching between Software mode and GPC mode. ENABLE_CLK is to indicate the GPC mode, the last Software mode RC400M state. It must be programed correspondingly. 0 - Clock is disabled before entering GPC mode 1 - Clock is enabled before entering GPC mode

16.5.1.7 16MHz RCOSC Control Register (OSC_16M_CTRL)

This register is used to control the 16MHz RC oscillator.

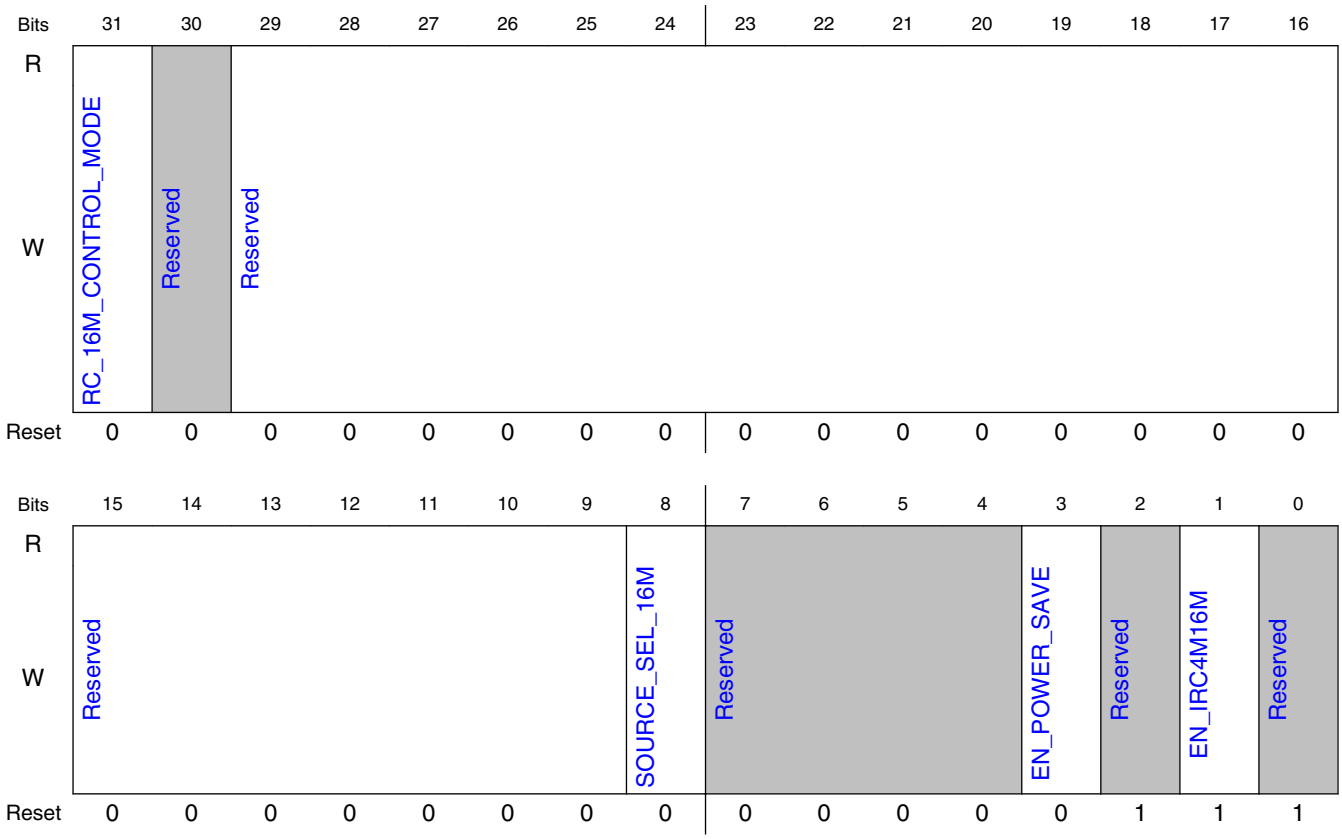
NOTE

The EN_IRC4M16M bit is used to disable/kill the output clocks while keeping the oscillator working. The oscillator should be disabled before powering it down.

16.5.1.7.1 Offset

Register	Offset
OSC_16M_CTRL	C0h

16.5.1.7.2 Diagram



16.5.1.7.3 Fields

Field	Description
31 RC_16M_CONTROL_MODE	Control Mode for 16MHz Oscillator This field selects between GPC mode and Software mode 0 - Software mode (default) 1 - GPC mode (Setpoint)
30 —	Reserved
29-9 —	Always set to zero
8 SOURCE_SEL_16M	Source select This is the source select for 16MHz RC Oscillator. The user can select between the 16MHz RCOSC or 24MHz OSC. 0 - 16MHz Oscillator 1 - 24MHz Oscillator
7-4	Reserved

Table continues on the next page...

Field	Description
—	
3 EN_POWER_SAVE	Power Save Enable Enable the power save mode function at 16MHz 0 - Disable 1 - Enable
2 -	Reserved
1 EN_IRC4M16M	Enable Clock Output This field enables clock output for 16MHz RCOSC (rc4m_en_irc4M16M_1p8v) 0 - Disable 1 - Enable
0 —	Reserved

16.5.2 OSC_RC_400M register descriptions

16.5.2.1 OSC_RC_400M memory map

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control Register 0 (CTRL0)	32	RW	0000_0000
4	Control Register 0 (CTRL0_SET)	32	RW	0000_0000
8	Control Register 0 (CTRL0_CLR)	32	RW	0000_0000
C	Control Register 0 (CTRL0_TOG)	32	RW	0000_0000
10	Control Register 1 (CTRL1)	32	RW	0000_0000
14	Control Register 1 (CTRL1_SET)	32	RW	0000_0000
18	Control Register 1 (CTRL1_CLR)	32	RW	0000_0000
1C	Control Register 1 (CTRL1_TOG)	32	RW	0000_0000
20	Control Register 2 (CTRL2)	32	RW	0000_0000
24	Control Register 2 (CTRL2_SET)	32	RW	0000_0000
28	Control Register 2 (CTRL2_CLR)	32	RW	0000_0000
2C	Control Register 2 (CTRL2_TOG)	32	RW	0000_0000
30	Control Register 3 (CTRL3)	32	RW	0000_0000
34	Control Register 3 (CTRL3_SET)	32	RW	0000_0000
38	Control Register 3 (CTRL3_CLR)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3C	Control Register 3 (CTRL3_TOG)	32	RW	0000_0000
50	Status Register 0 (STAT0)	32	RO	0000_0000
54	Status Register 0 (STAT0_SET)	32	RO	0000_0000
58	Status Register 0 (STAT0_CLR)	32	RO	0000_0000
5C	Status Register 0 (STAT0_TOG)	32	RO	0000_0000
60	Status Register 1 (STAT1)	32	RO	0000_0000
64	Status Register 1 (STAT1_SET)	32	RO	0000_0000
68	Status Register 1 (STAT1_CLR)	32	RO	0000_0000
6C	Status Register 1 (STAT1_TOG)	32	RO	0000_0000
70	Status Register 2 (STAT2)	32	RO	0000_0000
74	Status Register 2 (STAT2_SET)	32	RO	0000_0000
78	Status Register 2 (STAT2_CLR)	32	RO	0000_0000
7C	Status Register 2 (STAT2_TOG)	32	RO	0000_0000

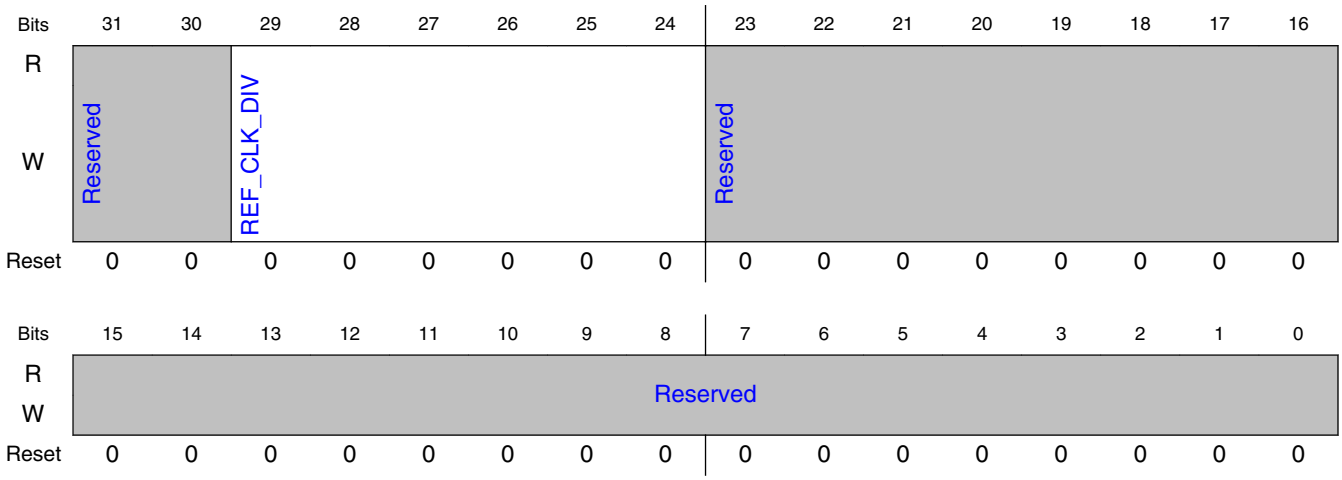
16.5.2.2 Control Register 0 (CTRL0)

This register contains analog control bits.

16.5.2.2.1 Offset

Register	Offset	Description
CTRL0	0h	Control Register 0
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

16.5.2.2.2 Diagram



16.5.2.2.3 Fields

Field	Description
31-30 —	Reserved
29-24 REF_CLK_DIV	Divide value for ref_clk to generate slow_clk (used inside this IP) 0: divide by 1 1-63: corresponding division by 1-63. The recommended divide value is 24.
23-0 —	Reserved This read-only field is reserved and always has the value 0.

16.5.2.3 Control Register 1 (CTRL1)

This register contains analog control bits.

16.5.2.3.1 Offset

Register	Offset	Description
CTRL1	10h	Control Register 1
CTRL1_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL1
CTRL1_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL1

Table continues on the next page...

Register	Offset	Description
CTRL1_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL1

16.5.2.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TARGET_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				HYST_PLUS				Reserved				HYST_MINUS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.2.3.3 Fields

Field	Description
31-16 TARGET_COUNT	Target count for the fast clock This field is used to set the desired target for the fast clock (osc_out_400m). Value in no. of clock cycles of the fast_clk (osc_out_400m) per divided ref_clk.
15-12 —	Reserved This read-only field is reserved and always has the value 0.
11-8 HYST_PLUS	Positive hysteresis value for the tuned clock Set this value after the clock is tuned. Value in no. of clock cycles of the fast clock(osc_out_400m).
7-4 —	Reserved
3-0 HYST_MINUS	Negative hysteresis value for the tuned clock Set this value after the clock is tuned. Value in no. of clock cycles of the fast clock(osc_out_400m).

16.5.2.4 Control Register 2 (CTRL2)

This register contains analog control bits.

16.5.2.4.1 Offset

Register	Offset	Description
CTRL2	20h	Control Register 2
CTRL2_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL2
CTRL2_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL2
CTRL2_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL2

16.5.2.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSC_TUNE_VAL								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	TUNE_START	Reserved	TUNE_EN	Reserved	TUNE_BYP	Reserved	Reserved	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.2.4.3 Fields

Field	Description
31-24 OSC_TUNE_VAL	Program the oscillator frequency These bits determine the frequency of oscillator when tuning is not enabled Frequency varies with PVT. 0 - Lowest frequency. 255 - Highest frequency.
23-15 —	Reserved
14 TUNE_START	Start/Stop tuning 0 - Stop tuning and reset the tuning logic. Oscillator runs using programmed OSC_TUNE_VAL 1 - Start tuning
13	Reserved

Table continues on the next page...

Field	Description
—	This read-only field is reserved and always has the value 0.
12 TUNE_EN	Freeze/Unfreeze the tuning value 0 - Freezes the tuning at the current tuned value. Oscillator runs at the frozen tuning value 1 - Unfreezes and continues the tuning operation
11 —	Reserved This read-only field is reserved and always has the value 0.
10 TUNE_BYP	Bypass the tuning logic 0 - Use the output of tuning logic to run the oscillator 1 - Bypass the tuning logic and use the programmed OSC_TUNE_VAL to run the oscillator
9 —	Reserved This read-only field is reserved and always has the value 0.
8 —	Reserved
7-0 —	Reserved This read-only field is reserved and always has the value 0.

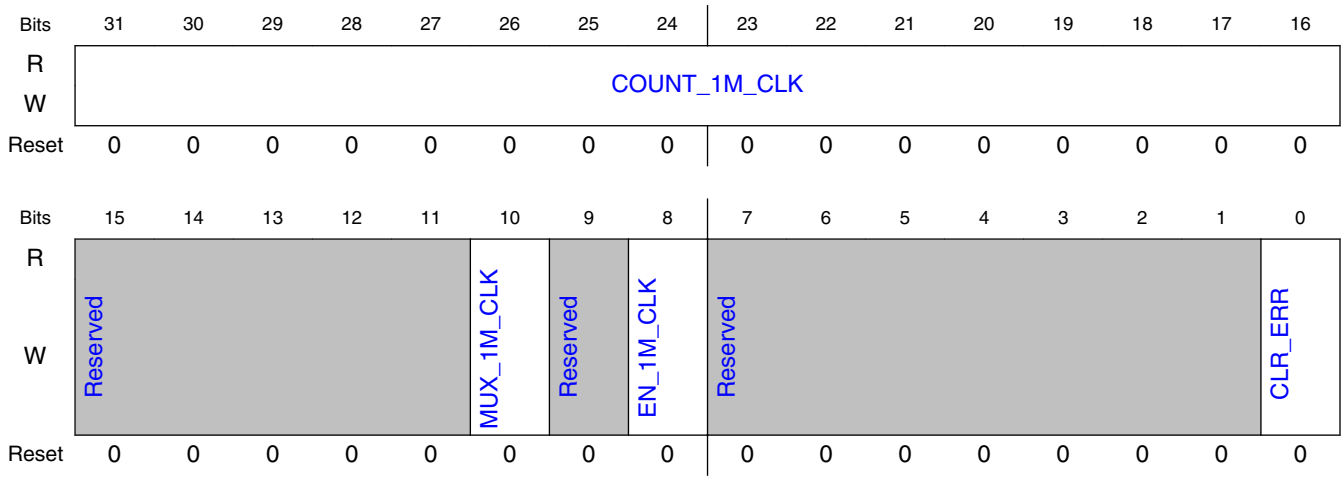
16.5.2.5 Control Register 3 (CTRL3)

This register contains analog control bits.

16.5.2.5.1 Offset

Register	Offset	Description
CTRL3	30h	Control Register 3
CTRL3_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL3
CTRL3_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL3
CTRL3_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL3

16.5.2.5.2 Diagram



16.5.2.5.3 Fields

Field	Description
31-16 COUNT_1M_CLK	Count for the locked clk_1m_out This bit field is used to set the desired target for the locked clk_1m_out. Value in no. of clock cycles of the fast_clk (osc_out_400m) per divided ref_clk. **NOTE:** This value should be 4 to 8 counts less than CTRL1[TARGET_COUNT] - CTRL1[HYST_MINUS].
15-11 —	Reserved This read-only field is reserved and always has the value 0.
10 MUX_1M_CLK	Select free/locked 1MHz output 0 - Select free-running 1MHz to be put out on clk_1m_out 1 - Select locked 1MHz to be put out on clk_1m_out
9 —	Reserved This read-only field is reserved and always has the value 0.
8 EN_1M_CLK	Enable 1MHz output Clock Though the name is EN_1M_CLK, it's function is inverted in analog to make this clock enabled by default. This clock is used to keep time. 0 - Enable the output (clk_1m_out) 1 - Disable the output (clk_1m_out)
7-1 —	Reserved
0 CLR_ERR	Clear the error flag CLK1M_ERR 0 - No effect 1 - Clears the error flag CLK1M_ERR in status register STAT0

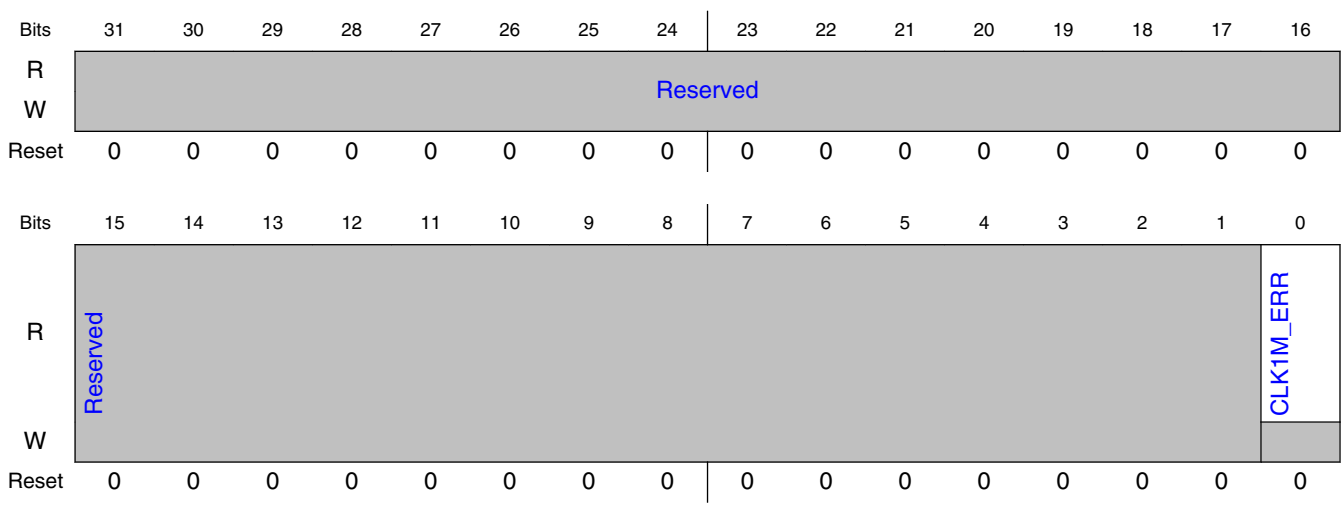
16.5.2.6 Status Register 0 (STAT0)

This register contains analog status bits.

16.5.2.6.1 Offset

Register	Offset	Description
STAT0	50h	Status Register 0
STAT0_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in STAT0
STAT0_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in STAT0
STAT0_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in STAT0

16.5.2.6.2 Diagram



16.5.2.6.3 Fields

Field	Description
31-1 —	Reserved This read-only field is reserved and always has the value 0.
0 CLK1M_ERR	Error flag for clk_1m_locked Flag indicates that the count_1m count wasn't reached within one divided ref_clk period. 0 - No effect 1 - The count value has been reached within one divided ref_clk period

16.5.2.7 Status Register 1 (STAT1)

This register contains analog status bits.

16.5.2.7.1 Offset

Register	Offset	Description
STAT1	60h	Status Register 1
STAT1_SET	64h	Writing a 1 to a bit in this register sets the corresponding bit in STAT1
STAT1_CLR	68h	Writing a 1 to a bit in this register clears the corresponding bit in STAT1
STAT1_TOG	6Ch	Writing a 1 to a bit in this register toggles the corresponding bit in STAT1

16.5.2.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURR_COUNT_VAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.2.7.3 Fields

Field	Description
31-16 CURR_COUNT_VAL	Current count for the fast clock This field holds the current count during the tuning process. This value converges to TARGET_COUNT as tuning progresses.
15-0 —	Reserved This read-only field is reserved and always has the value 0.

16.5.2.8 Status Register 2 (STAT2)

This register contains analog status bits.

16.5.2.8.1 Offset

Register	Offset	Description
STAT2	70h	Status Register 2
STAT2_SET	74h	Writing a 1 to a bit in this register sets the corresponding bit in STAT2
STAT2_CLR	78h	Writing a 1 to a bit in this register clears the corresponding bit in STAT2
STAT2_TOG	7Ch	Writing a 1 to a bit in this register toggles the corresponding bit in STAT2

16.5.2.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURR_OSC_TUNE_VAL								Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.2.8.3 Fields

Field	Description
31-24 CURR_OSC_TUNE_VAL	Current tuning value used by oscillator This bit field holds the current OSC_TUNE_VAL value which is being used by the oscillator. This value changes as the tuning progresses.
23-0 —	Reserved This read-only field is reserved and always has the value 0.

Chapter 17
Power Management Unit (PMU)

17.1 Chip-specific PMU information

Table 17-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

17.2 Overview

The Power Management Unit (PMU) is designed to simplify the external power interface.

17.2.1 Block Diagram

The power architecture of the chip is illustrated in the figure below, which shows the typical use case of a single 3.3V power supply and a coin cell battery. Depending on the application, different power supply schemes can apply (e.g. DCDC bypassed).

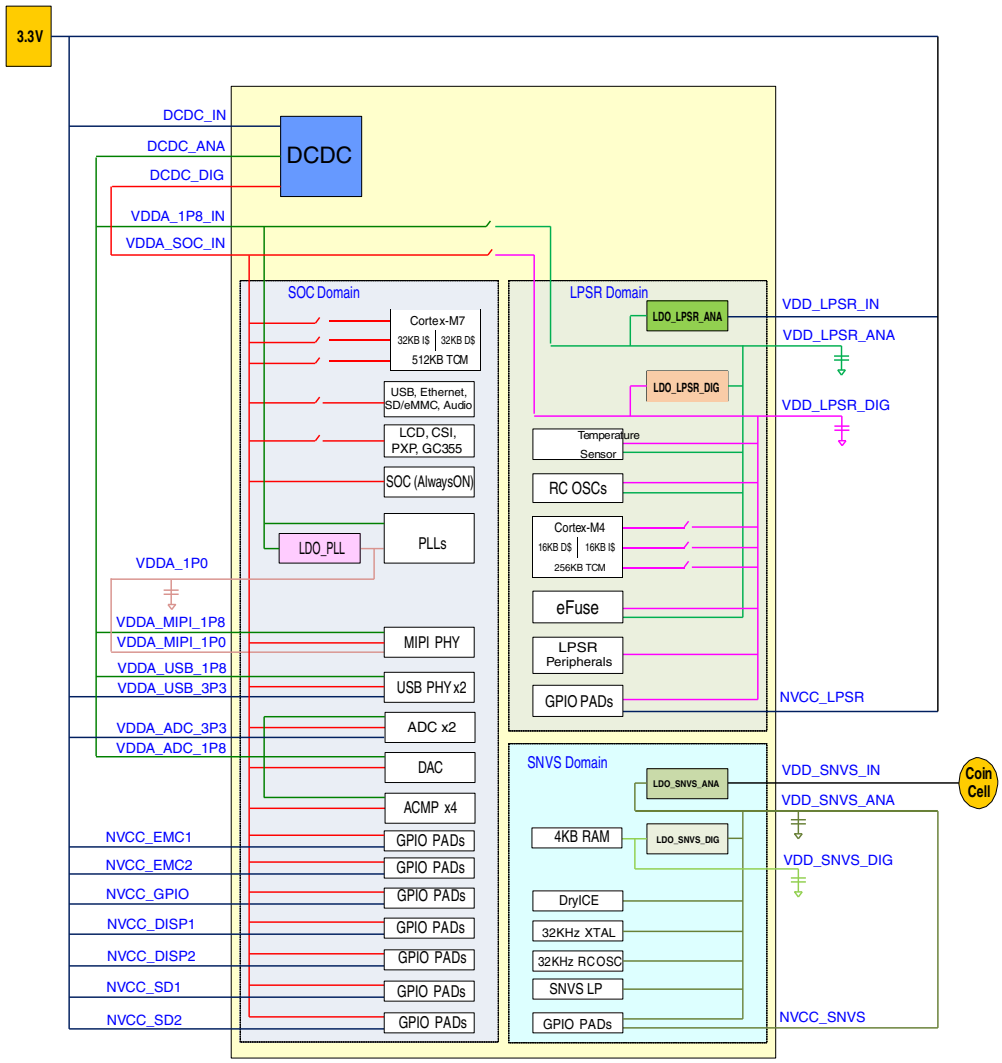


Figure 17-1. Power System Overview

17.2.2 Features

The PMU has the following components integrated for power management:

- One DCDC to generate core power supply, with dynamic voltage scaling capability
- LDOs to generate power for internal logics
- Multiple Power Switches for sophisticated power mode management

17.3 Functional Description

The PMU operations are detailed in the sections below.

17.3.1 Well-Bias configuration

The well-bias configuration is shown below. The NWELL is an adaptive, quiescent consumption, negative voltage output, switched regulator. The PWELL is an adaptive, quiescent consumption, positive voltage tracker, switched regulator. The NWELL and PWELL regulators are biased together, as either FBB or RBB, not both at the same time. Both regulators are connected to RVT and LVT transistors.

- For low-voltage threshold (LVT), FBB is supported. It provides enhanced SoC core performance at a cost of higher leakage.
- For regular-voltage threshold (RVT), RBB is supported. It reduces SoC core consumption with lower leakage.

When the system enters low-power mode, it normally decreases the power supply voltage, then turns on RBB. The VDDA_SOC_IN and VDD_LPSR_ANA supplies need to be settled before the well-bias regulator is turned on (WB_EN). Both supplies need to be settled for FBB as well.

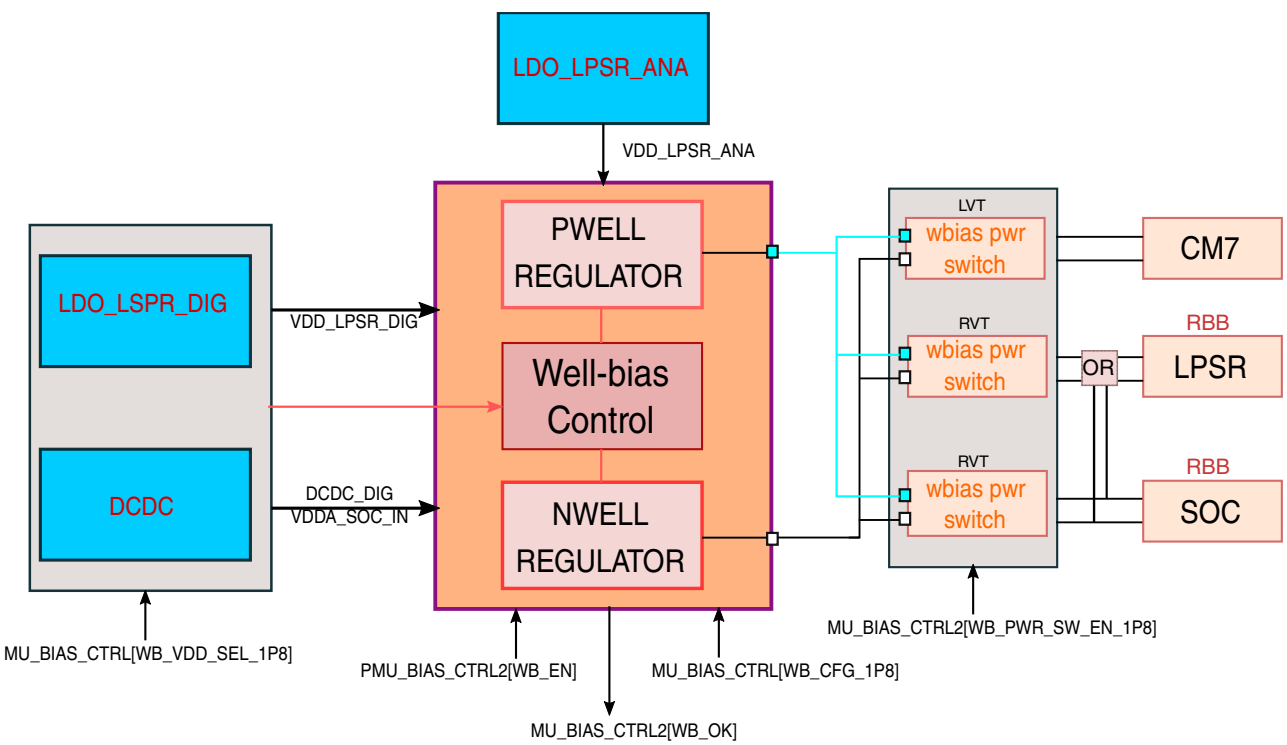


Figure 17-2. Well-Biasing Block Diagram

NOTE

The VDD selection (WB_VDD_SEL_1P8) depends on whether SoC RBB or LPSR RBB is set. This selection needs to be done before enabling well-bias.

17.3.1.1 Body Biasing (BB)

Forward Body Biasing (FBB) and Reverse Body Biasing (RBB) areis implemented to boost performance and reduce power. The implementation of BB in each power domain are listed below:

- FBB is implemented in Cortex-M7 Platform to achieve higher performance
- RBB is implemented in LPSR and SoC power domains to reduce power consumption in low-power modes

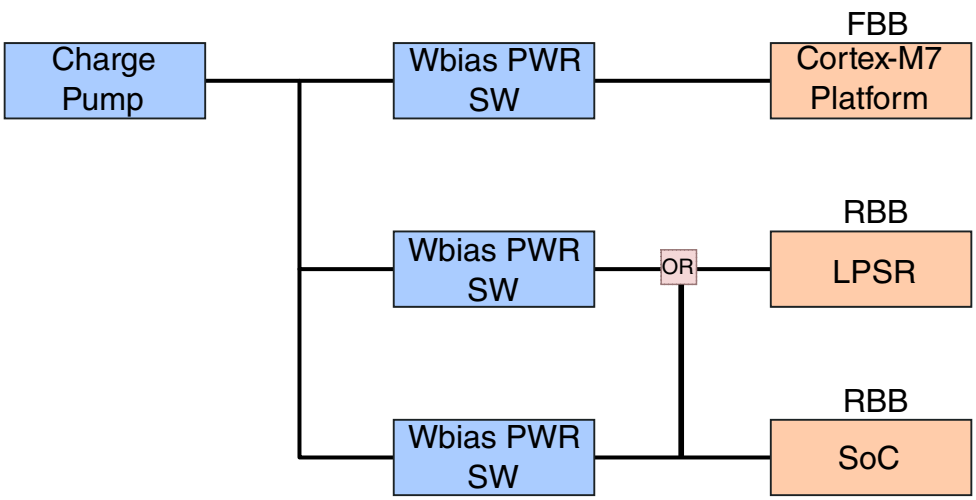


Figure 17-3. Body Biasing Diagram

NOTE

Software must ensure that FBB and RBB are not enabled simultaneously, only one can be used at a time.

17.3.2 Control Mode

The PMU supports two control modes that can be configured by x_CONTROL_MODE fields. These two modes are Sofware control mode and Hardware control mode.

- The Software control mode enables LDOs, switches, and bandgap to be fully configured by registers.
- The Hardware control mode enables the ON/OFF of the LDOs, LDO bypass switches, and bandgap to be configured by GPC Setpoints. The voltage of LPSR_DIG and operation modes of LDOs (lp_mode), are also configured by GPC Setpoints. The other settings remain unchanged.

The following table shows the functions controlled by GPC:

Table 17-2. Functions controlled by GPC

IP	ON/OFF Control	Operation Mode	Tracking Mode	Bypass Mode (Switch ON/ OFF)	Standby Mode (Low Power Mode)	Voltage Control
LDO_PLL	Y	-	-	-	Y	-
LPSR_ANA	-	HP/LP	Y	Y	HP/LP	-
LPSR_DIG	-	HP/LP	Y	Y	HP/LP	Y
REFTOP (Bandgap)	Y	-	-	-	Y	-
FBB_M7	Y	-	-	-	Y	-
RBB_SOC	Y	-	-	-	Y	-
RBB_LPSR	Y	-	-	-	Y	-

Each time there is a change in Setpoint, there will be two steps of requests - voltage down and voltage up. The new target is compared with the current setting to check if the voltage change should happen at each step. When there is a Setpoint change request, the 'done' signals will be generated until all the settings have been changed. When there is a standby in request, the acknowledge signal will be held high until the set modules are put into low-power mode. When there is standby out request, the acknowledge signal will held high until the modules are brought back from low power mode.

17.3.2.1 ON/OFF Control Mode

- The ON/OFF control for LDOs:
  - When LDO_x_CONTROL_MODE = 0 (Software control mode) LD0_x_ENABLE controls the ON/OFF
  - When LDO_x_CONTROL_MODE = 1 (Hardware / GPC control mode) system is in Setpoint n, and LDO_x_ENABLE_SP[ON_OFF_SETPOINTn] controls the ON/OFF
- Biasing ON/OFF:

- When yBB_x_CONTROL_MODE = 0 (Software control mode) yBB_x_ENABLE controls the ON/OFF of the Biasing Circuit
- When yBB_x_CONTROL_MODE = 1 (Hardware / GPC control mode) system is in Setpoint n, and yBB_x_ENABLE_SP[ON_OFF_SETPOINTn] controls the ON/OFF of the biasing circuit

- Bandgap ON/OFF Control:
  - When REF_CONTROL_MODE = 0 (Software control mode) REF_ENABLE controls the ON/OFF of the bandgap
  - When REF_CONTROL_MODE = 1 (Hardware / GPC control mode) system is in Setpoint n, and BANDGAP_ENABLE_SP[ON_OFF_SETPOINTn] controls the ON/OFF of the bandgap

NOTE

Software mode supports flexible control, but care and consideration should be followed in regards to analog dependecies and power structure sequences. In GPC mode, everything is handled precisely by hardware.

17.3.2.2 Low Power Control Mode

GPC will wait for an acknowledgement to be asserted before it continues to the next step. When standby sequence is initiated, GPC will wait for another acknowledgement to be asserted before it continues to the next step. When standby exit sequence is completed, LDO enters the low-power mode based on the following condition:

- When LDO_x_CONTROL_MODE = 0 (Software control mode), LDO_x_LP_MODE register is set
- When LDO_x_CONTROL_MODE = 1 (Hardware / GPC control mode) system is in Setpoint n, and LDO_x_LP_MODE_SP[LP_MODE_SETPOINTn] is set to 1

The LDOs can also enter low power mode based on following condition:

- When LDO_x_CONTROL_MODE = 0 (Software control mode) standby_request signal is asserted, and LDO_x_STBY_EN register is set to 1 (0 by default)
- When LDO_x_CONTROL_MODE = 1 (Hardware / GPC control mode) standby_request signal is asserted, and system in Setpoint n, and LDO_x_STBY_EN_SP[LP_MODE_SETPOINT] is set to 1

17.3.2.3 Tracking Mode

This mode is supported on LDO_LPSR_ANA and LDO_LPSR_DIG domains. When enabled on LDO_LPSR_DIG, it's output will track the VDD_SOC to confirm that the LPSR_DIG voltage is same as the VDD_SOC when bypass mode is enabled. When enabled on LDO_LPSR_ANA, its output will track the VDDA_1P8 to confirm the LPSR_ANA voltage is same as the VDDA_1P8 when bypass mode is enabled.

17.3.2.4 Bypass Mode

When bypass mode is enabled on LDO_LPSR_DIG, it's output will be shorted with VDD_SOC, and the LPSR_DIG domain will be supplied by DCDC_DIG. When bypass mode is enabled on LDO_LPSR_ANA, it's output will be shorted with VDDA_1P8, and the LPSR_ANA domain will be supplied by DCDC_ANA.

NOTE

When tracking is enabled, force low power mode to disable.
Tracking enable should happen before bypass enable and
Tracking disable should happen after bypass disable.

17.3.2.5 Setpoint Change Sequence

The following sequence should be followed to change the LDO_DIG voltage setpoint (Voltage Down):

- The Setpoint voltage down request is asserted by GPC
- The current Setpoint setting is captured, after detecting the rising edge offset point of voltage down request
- The current LDO voltage is compared with the new setting
- If the new setting is lower, the new settings are applied. If the new setting is higher, the request is ignored.
- The setpoint voltage down done signal is asserted

The following sequence should be followed to change the LDO_DIG voltage setpoint (Voltage Up):

- The setpoint voltage up request is asserted by GPC
- The current Setpoint setting after detecting the rising edge offset point of voltage up request
- The current LDO voltage is compared with the new setting

- If the new setting is higher, the new settings are applied. If the new setting is lower, the request is ignored.
- The setpoint voltage up done signal is asserted

17.3.2.6 Well Bias Enable Sequence

The following sequence should be followed to enable Well Bias:

- Set well bias setting(voltage, LVT/RVT)
  - PMU_BIAS_CTRL[1] is set to 1 (NWELL is configured to supply and LVT CORE,FBB)
  - PMU_BIAS_CTRL[12] is set to 1 to enable FBB correctly
  - All the other bits are set to 0
- Turn on CM7 FBB switch, and well bias (must be executed in the same step)
  - PMU_BIAS_CTRL2[WB_EN] is set to 1
  - PMU_BIAS_CTRL2[WB_PWR_SW_EN_1P8] is set to 1
  - All other bits set to 0
- Check the PMU_BIAS_CTRL2[WB_OK] bit to confirm Well Bias is stable

17.3.3 Clocks

The PMU doesn't have any relevant application clock sources.

17.4 External Signals

Please see the supply contact assignments in the datasheet for more information.

17.5 Memory Map and register definition

This section includes the PMU module memory map and detailed descriptions of all registers.

17.5.1 PMU register descriptions

17.5.1.1 PMU memory map

PMU base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
500	PMU_LDO_PLL_REGISTER (PMU_LDO_PLL)	32	RW	0000_0001
550	PMU_BIAS_CTRL_REGISTER (PMU_BIAS_CTRL)	32	RW	0000_0000
560	PMU_BIAS_CTRL2_REGISTER (PMU_BIAS_CTRL2)	32	RW	0000_0000
570	PMU_REF_CTRL_REGISTER (PMU_REF_CTRL)	32	RW	0000_0000
580	PMU_POWER_DETECT_CTRL_REGISTER (PMU_POWER_DETECT_CTRL)	32	RW	0000_0000
600	LDO_PLL_ENABLE_SP_REGISTER (LDO_PLL_ENABLE_SP)	32	RW	0000_0000
610	LDO_LPSR_ANA_ENABLE_SP_REGISTER (LDO_LPSR_ANA_ENABLE_SP)	32	RW	0000_0000
620	LDO_LPSR_ANA_LP_MODE_SP_REGISTER (LDO_LPSR_ANA_LP_MODE_SP)	32	RW	0000_0000
630	LDO_LPSR_ANA_TRACKING_EN_SP_REGISTER (LDO_LPSR_ANA_TRACKING_EN_SP)	32	RW	0000_0000
640	LDO_LPSR_ANA_BYPASS_EN_SP_REGISTER (LDO_LPSR_ANA_BYPASS_EN_SP)	32	RW	0000_0000
650	LDO_LPSR_ANA_STBY_EN_SP_REGISTER (LDO_LPSR_ANA_STBY_EN_SP)	32	RW	0000_0000
660	LDO_LPSR_DIG_ENABLE_SP_REGISTER (LDO_LPSR_DIG_ENABLE_SP)	32	RW	0000_0000
670	LDO_LPSR_DIG_TRG_SP0_REGISTER (LDO_LPSR_DIG_TRG_SP0)	32	RW	0000_0000
680	LDO_LPSR_DIG_TRG_SP1_REGISTER (LDO_LPSR_DIG_TRG_SP1)	32	RW	0000_0000
690	LDO_LPSR_DIG_TRG_SP2_REGISTER (LDO_LPSR_DIG_TRG_SP2)	32	RW	0000_0000
6A0	LDO_LPSR_DIG_TRG_SP3_REGISTER (LDO_LPSR_DIG_TRG_SP3)	32	RW	0000_0000
6B0	LDO_LPSR_DIG_LP_MODE_SP_REGISTER (LDO_LPSR_DIG_LP_MODE_SP)	32	RW	0000_0000
6C0	LDO_LPSR_DIG_TRACKING_EN_SP_REGISTER (LDO_LPSR_DIG_TRACKING_EN_SP)	32	RW	0000_0000
6D0	LDO_LPSR_DIG_BYPASS_EN_SP_REGISTER (LDO_LPSR_DIG_BYPASS_EN_SP)	32	RW	0000_0000
6E0	LDO_LPSR_DIG_STBY_EN_SP_REGISTER (LDO_LPSR_DIG_STBY_EN_SP)	32	RW	0000_0000
6F0	BANDGAP_ENABLE_SP_REGISTER (BANDGAP_ENABLE_SP)	32	RW	0000_0000
700	FBB_M7_ENABLE_SP_REGISTER (FBB_M7_ENABLE_SP)	32	RW	0000_0000
710	RBB_SOC_ENABLE_SP_REGISTER (RBB_SOC_ENABLE_SP)	32	RW	0000_0000
720	RBB_LPSR_ENABLE_SP_REGISTER (RBB_LPSR_ENABLE_SP)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
730	BANDGAP_STBY_EN_SP_REGISTER (BANDGAP_STBY_EN_SP)	32	RW	0000_0000
740	PLL_LDO_STBY_EN_SP_REGISTER (PLL_LDO_STBY_EN_SP)	32	RW	0000_0000
750	FBB_M7_STBY_EN_SP_REGISTER (FBB_M7_STBY_EN_SP)	32	RW	0000_0000
760	RBB_SOC_STBY_EN_SP_REGISTER (RBB_SOC_STBY_EN_SP)	32	RW	0000_0000
770	RBB_LPSR_STBY_EN_SP_REGISTER (RBB_LPSR_STBY_EN_SP)	32	RW	0000_0000
780	FBB_M7_CONFIGURE_REGISTER (FBB_M7_CONFIGURE)	32	RW	0000_2F11
790	RBB_LPSR_CONFIGURE_REGISTER (RBB_LPSR_CONFIGURE)	32	RW	0000_3022
7A0	RBB_SOC_CONFIGURE_REGISTER (RBB_SOC_CONFIGURE)	32	RW	0000_0044
7B0	REFTOP_OTP_TRIM_VALUE_REGISTER (REFTOP_OTP_TRIM_VALUE)	32	RO	0000_0000
7D0	LPSR_1P8_LDO_OTP_TRIM_VALUE_REGISTER (LPSR_1P8_LDO_OTP_TRIM_VALUE)	32	RO	0000_0000

17.5.1.2 PMU_LDO_PLL_REGISTER (PMU_LDO_PLL)

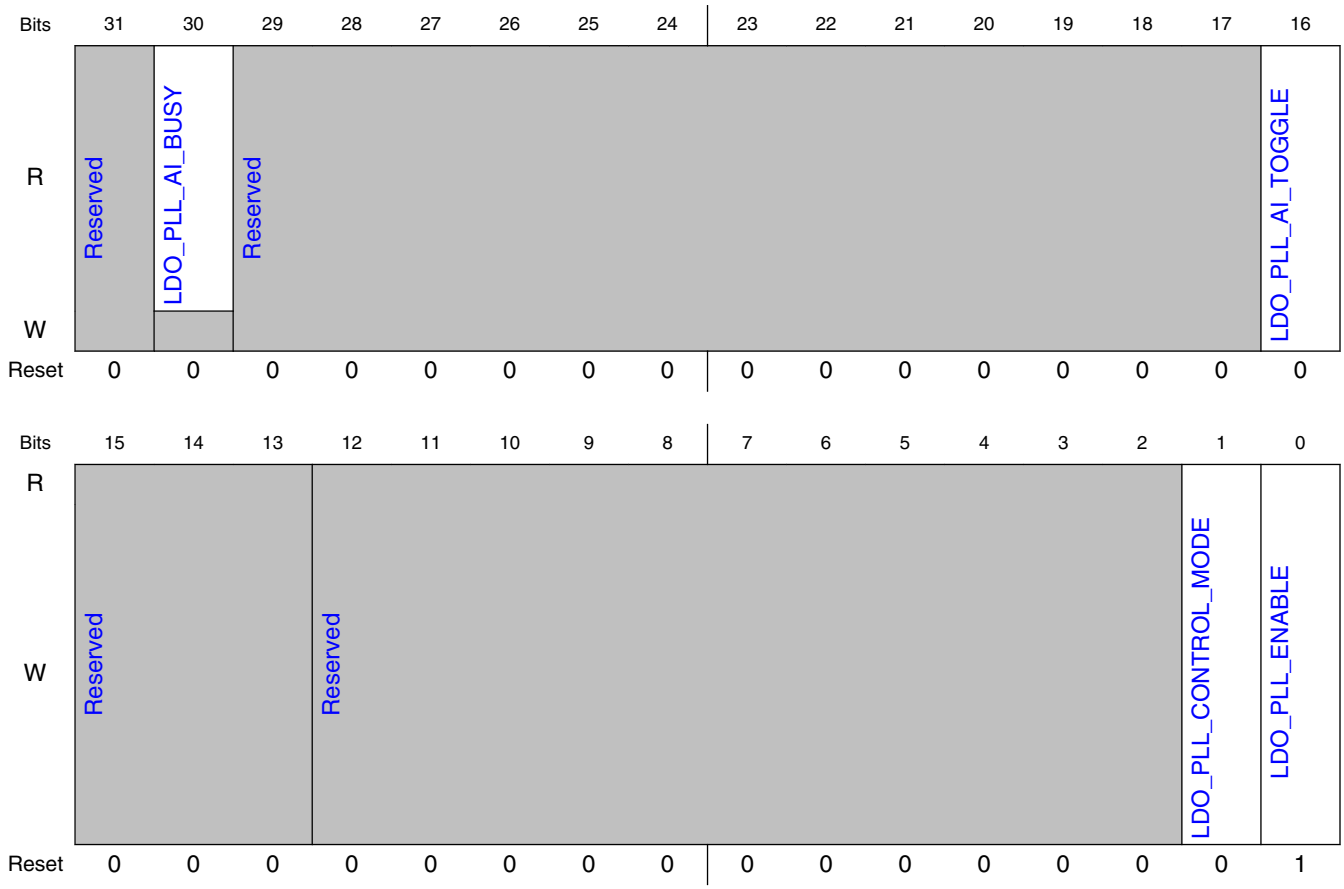
PMU_LDO_PLL_Regulator_Control_Register

This register defines the control and status bits for PLL regulator. This regulator is designed to power the digital portions of the analog cells

17.5.1.2.1 Offset

Register	Offset
PMU_LDO_PLL	500h

17.5.1.2.2 Diagram



17.5.1.2.3 Fields

Field	Description
31 —	Reserved
30 LDO_PLL_AI_BUSY	ldo_pll_busy For this AI bridge register field, there will be a detail description on how to use it.
29-17 —	Reserved Always set to zero (0).
16 LDO_PLL_AI_TOGGLE	ldo_pll_ai_toggle For this AI bridge register field, Please see 'Analog IP (AI) Interface' topic for details on how to use this bit.
15-13 —	Reserved Always set to zero (0).
12-2 —	Reserved

Table continues on the next page...

Field	Description
1 LDO_PLL_CONTROL_MODE	LDO_PLL_CONTROL_MODE LDO_PLL has two mode to for enable. One is software mode, the other is GPC mode(Setpoint). LDO_PLL_CONTROL_MODE select 0 - Software control mode 1 - Hardware / GPC control mode
0 LDO_PLL_ENABLE	LDO_PLL_ENABLE The usage of this bit is for software mode switching to gpc mode: If the Setpoint 0 is ldo_pll on : set this ldo_pll_enable to default state 1. Else the Setpoint 0 is ldo_pll off : set this ldo_pll_enable to state 0. After the mode switched to gpc mode, keep this bit as it is should be fine.

17.5.1.3 PMU_BIAS_CTRL_REGISTER (PMU_BIAS_CTRL)

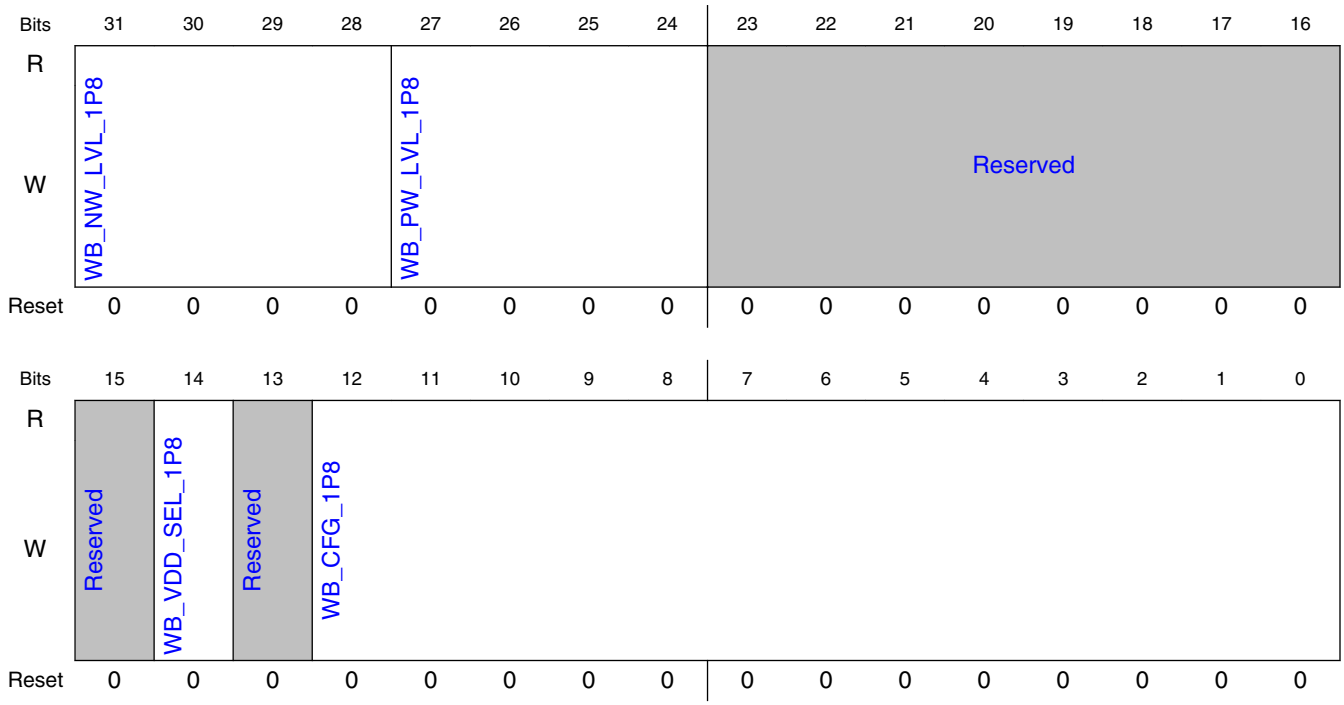
PMU_Well_BIAS_Control_Register

This register defines functions for wb bias.

17.5.1.3.1 Offset

Register	Offset
PMU_BIAS_CTRL	550h

17.5.1.3.2 Diagram



17.5.1.3.3 Fields

Field	Description
31-28 WB_NW_LVL_1P8	WB_NW_LVL_1P8 NW Output voltage range, RBB function enabled; valid range is 0000 ~ 1000. TRIM BUS.
27-24 WB_PW_LVL_1P8	WB_PW_LVL_1P8 PW Output voltage range selection. Valide range is 0000 ~ 1000 TRIM BUS.
23-15 —	Reserved
14 WB_VDD_SEL_1P8	wb_vdd_sel_1p8 Well bias VDD selector 1P8. There are two power sources (LPSR_DIG_LDO or DCDC). **NOTE:** This selection depends on which RBB is working, SOC RBB (DCDC) or LPSR RBB (LPSR_DIG_LDO). This select bit needs to be done before enabling wbias 0 - VDD_LV1 supplies the power stage and NWELL sampler (LPSR_DIG_LDO) 1 - VDD_LV2 supplies the power stage and NWELL sampler (DCDC)
13 —	Reserved
12-0 WB_CFG_1P8	wb_cfg_1p8 Well Bias Configuration 1P8.

Field	Description
	For bit 0 : • 0 PWELL and NWELL is turned on when wb_en_1p8 is set; • 1 PWELL regulator is turned on only when wb_en_1p8 is set. NWELL is kept disabled. For bit 1 : Body bias • 0 NWELL is configured to supply and RVT CORE - RBB. • 1 NWELL is configured to supply and LVT CORE - FBB. For bit 2-4 : Select size of bias area • 000 Imax = 180uA; Areamax-RVT = 6.00mm2 at 125C • 001 Imax = 150uA; Areamax-RVT = 5.00mm2 at 125C • 010 Imax = 120uA; Areamax-RVT = 4.00mm2 at 125C • 011 Imax = 90uA; Areamax-RVT = 3.00mm2 at 125C • 100 Imax = 60uA; Areamax-RVT = 2.00mm2 at 125C • 101 Imax = 45uA; Areamax-RVT = 0.15mm2 at 125C • 110 Imax = 30uA; Areamax-RVT = 1.0mm2 at 125C • 111 Imax = 15uA; Areamax-RVT = 0.50mm2 at 125C For bit 5 : Adaptive function • 1 Adaptive frequency disabled. Frequency determined by wb_cfg_1p8[8:6]. • 0 Frequency change after each half cycle Minimum frequency determined by wb_cfg_1p8[8:6] For bit 6-8 : Oscillator bits • 000 typical frequency = osc_freq/128 • 001 typical frequency = osc_freq/64 • 010 typical frequency = osc_freq/32 • 011 typical frequency =osc_freq/16 • 100 typical frequency = osc_freq/8 • 110 typical frequency = osc_freq/2 • 111 typical frequency = oscillator Frequency (osc_freq) For bit 9 : TRIM BUS. It is used to set an internal adaptive configuration. It configures the adaptive clock source. The options are: 1) the oscillator clock 2) clock synchronous with the Charge Pump clock. These 2 signals have the same frequency, but delayed one to other. It determines if the adaptive control signal changes the frequency in the same or in the next oscillator cycle. • 0 The option is the oscillator clock • 1 The option is clock synchronous with the Charge Pump clock For bit 10 - 11 : TRIM BUS • 00 no frequency reduction • 01 30% frequency reduction due to cap. increment. • 10 40% frequency reduction due to cap. increment. • 11 50% frequency reduction due to cap. increment. For bit 12 : • 1 Pull down option is enabled. • 0 Pull down option is disabled.

17.5.1.4 PMU_BIAS_CTRL2_REGISTER (PMU_BIAS_CTRL2)

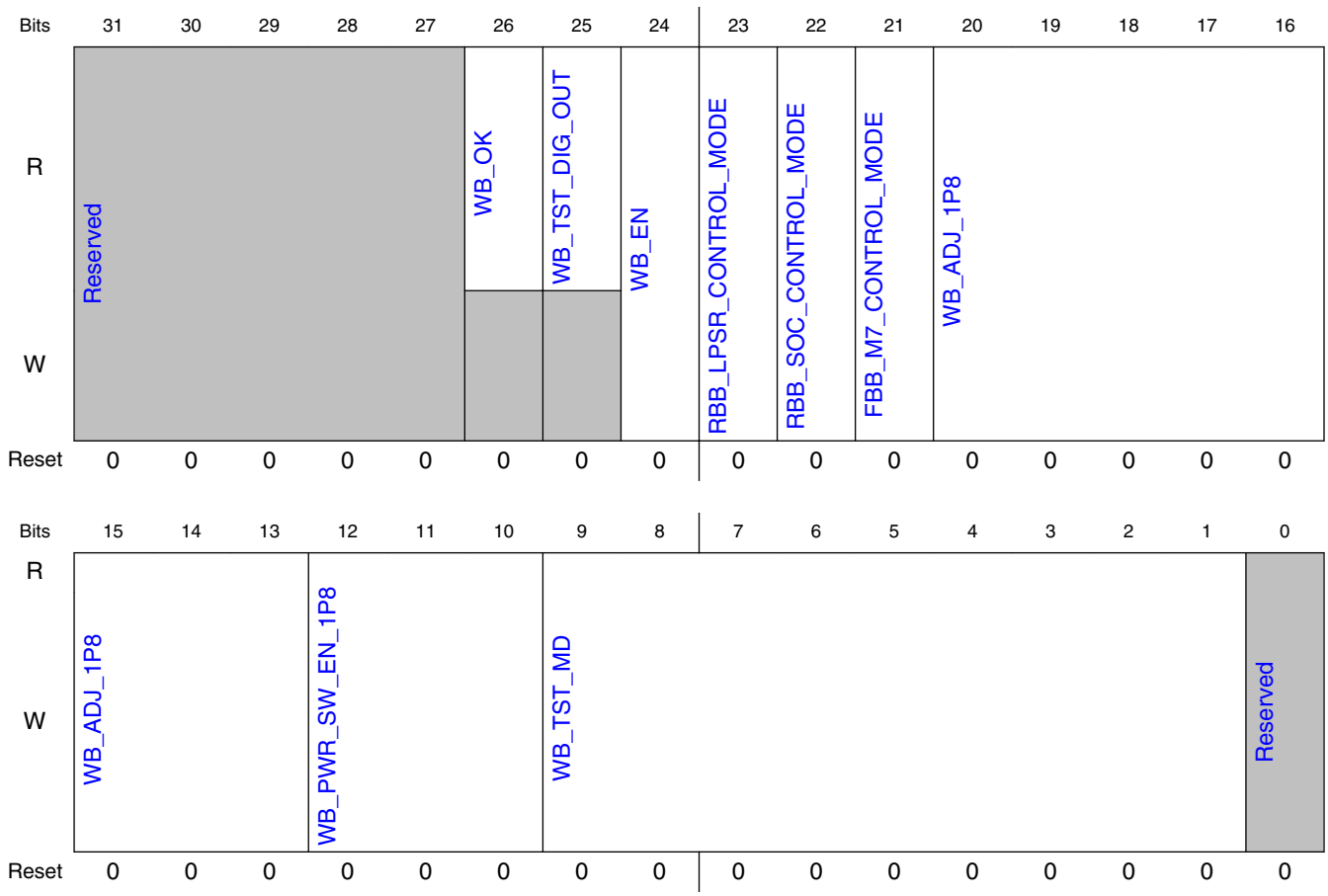
Well Bias IP control register

This register defines the test mode function for wb bias.

17.5.1.4.1 Offset

Register	Offset
PMU_BIAS_CTRL2	560h

17.5.1.4.2 Diagram



17.5.1.4.3 Fields

Field	Description
31-27	Reserved
—	Always set to zero (0).
26	Digital Output pin.
WB_OK	Turn on/off acknowledge bit from regulator

Table continues on the next page...

Field	Description
25 WB_TST_DIG_OUT	Digital output For test purposes
24 WB_EN	wb_en
23 RBB_LPSR_CONTROL_MODE	RBB_LPSR_CONTROL_MODE 0 - Software control mode 1 - Hardware / GPC control mode
22 RBB_SOC_CONTROL_MODE	RBB_SOC_CONTROL_MODE 0 - Software control mode 1 - Hardware / GPC control mode
21 FBB_M7_CONTROL_MODE	FBB_M7_CONTROL_MODE 0 - Software control mode 1 - Hardware / GPC control mode
20-13 WB_ADJ_1P8	wb_adj_1p8 Well Bias Adjustment 1P8. The bit values detailed below is a representation of bits 0 - 3 : TRIM BUS, and bits 4 - 7. 00000000 - Cref= 0fF Cspl= 0fF DeltaC= 0fF 00000001 - Cref= 0fF Cspl= 30fF DeltaC= -30fF 00000010 - Cref= 0fF Cspl= 43fF DeltaC= -43fF 00000011 - Cref= 0fF Cspl= 62fF DeltaC=-62fF 00000100 - Cref= 0fF Cspl=105fF DeltaC=-105fF 00000101 - Cref= 30fF Cspl= 0fF DeltaC= 30fF 00000110 - Cref= 30fF Cspl= 43fF DeltaC= -12fF 00000111 - Cref= 30fF Cspl=105fF DeltaC= -75fF 00001000 - Cref= 43fF Cspl= 0fF DeltaC= 43fF 00001001 - Cref= 43fF Cspl= 30fF DeltaC= 13fF 00001010 - Cref= 43fF Cspl= 62fF DeltaC= -19fF 00001011 - Cref= 62fF Cspl= 0fF DeltaC= 62fF 00001100 - Cref= 62fF Cspl= 43fF DeltaC= 19fF 00001101 - Cref=105fF Cspl= 0fF DeltaC= 105fF 00001110 - Cref=105fF Cspl=30fF DeltaC= 75fF 00001111 - Cref=0fF Cspl=0fF DeltaC= 0fF
12-10 WB_PWR_SW_EN_1P8	MODSEL_wb_tst_md_1p8 Power switch enable This is a PWR_SW_EN, that is a module selection. Setting these bits will connect the NWELL/PWELL to back-biasing power supplies (0 - not connected, 1 - connected). Each bit represents one configuration. bit [0] represents FBB M7 bit [1] represents RBB LPSR

Table continues on the next page...

Field	Description
	bit [2] RBB SOC + LPSR **NOTE:** Only one bit can be 1 at the same time. 001 - NWELL/PWELL FBB is connected to CM7 010 - NWELL/PWELL RBB is connected to LPSR 100 - NWELL/PWELL RBB is connected to SOC.
9-1 WB_TST_MD	TMOD_wb_tst_md_1p8 TMOD Test modes inside the module
0 —	Reserved Always set to zero (0).

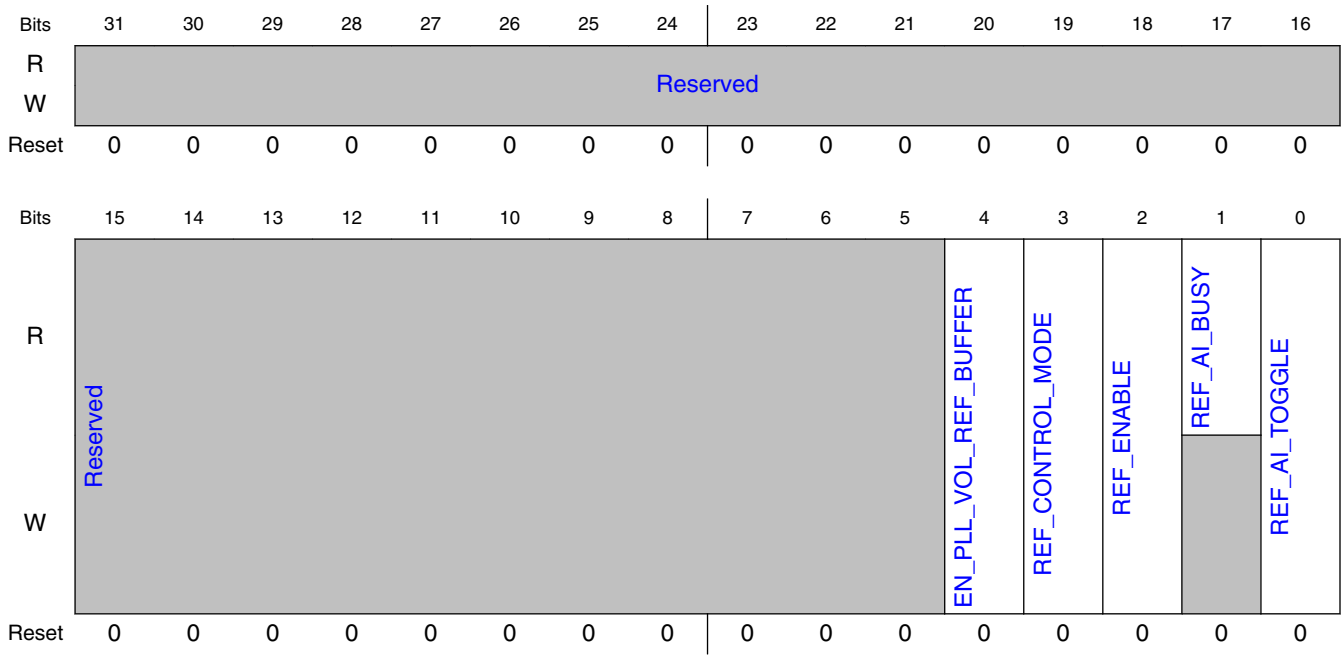
17.5.1.5 PMU_REF_CTRL_REGISTER (PMU_REF_CTRL)

Anadig_Reference_Analog_Control_and_Status_Control_Register

17.5.1.5.1 Offset

Register	Offset
PMU_REF_CTRL	570h

17.5.1.5.2 Diagram



17.5.1.5.3 Fields

Field	Description
31-5 —	Reserved Always set to zero (0).
4 EN_PLL_VOL_REF_BUFFER	en_pll_vol_ref_buffer This control bit enables or disables the reference voltage for the PLLs.
3 REF_CONTROL_MODE	REF_CONTROL_MODE SW/HW control mode for reftop 0 - Software control mode 1 - Hardware / GPC control mode
2 REF_ENABLE	REF_ENABLE The usage of this bit is for software mode switching to gpc mode: If the Setpoint 0 is bandgap on: set this ref_enable to default state 0. Else the Setpoint 0 is bandgap off: set this ref_enable to state 1. After the mode switched to gpc mode, keep this bit as it is should be fine. It is recommended to let software handle all the possible cases.
1 REF_AI_BUSY	ref_ai_busy ref_ai_busy
0 REF_AI_TOGGLE	ref_ai_toggle ref_ai_toggle

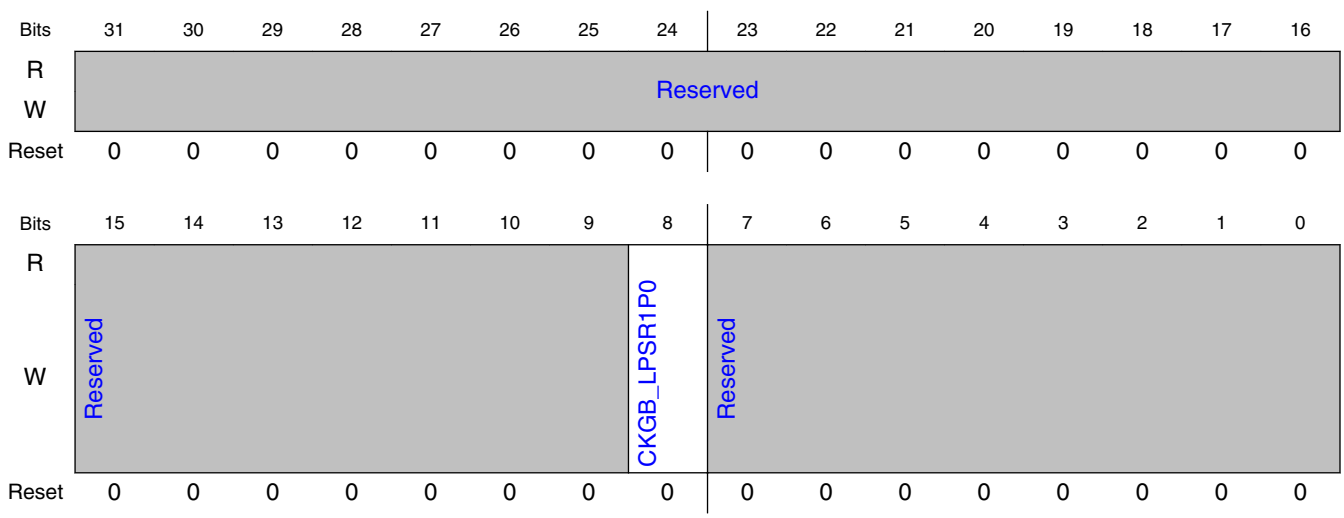
17.5.1.6 PMU_POWER_DETECT_CTRL_REGISTER (PMU_POWER_DETECT_CTRL)

This register will be deleted in next release except ckgb bit

17.5.1.6.1 Offset

Register	Offset
PMU_POWER_DETECT_CTRL	580h

17.5.1.6.2 Diagram



17.5.1.6.3 Fields

Field	Description
31-16 —	Reserved
15-9 —	Reserved
8 CKGB_LPSR1P0	ckgb_lpsr1p0 PHY LDO isolation control bit. This bit is used during PHY LDO ON/OFF sequence. This bit is for software usage. In GPC mode, it will automatically be handled by hardware.

Table continues on the next page...

Field	Description
7-0 —	Reserved

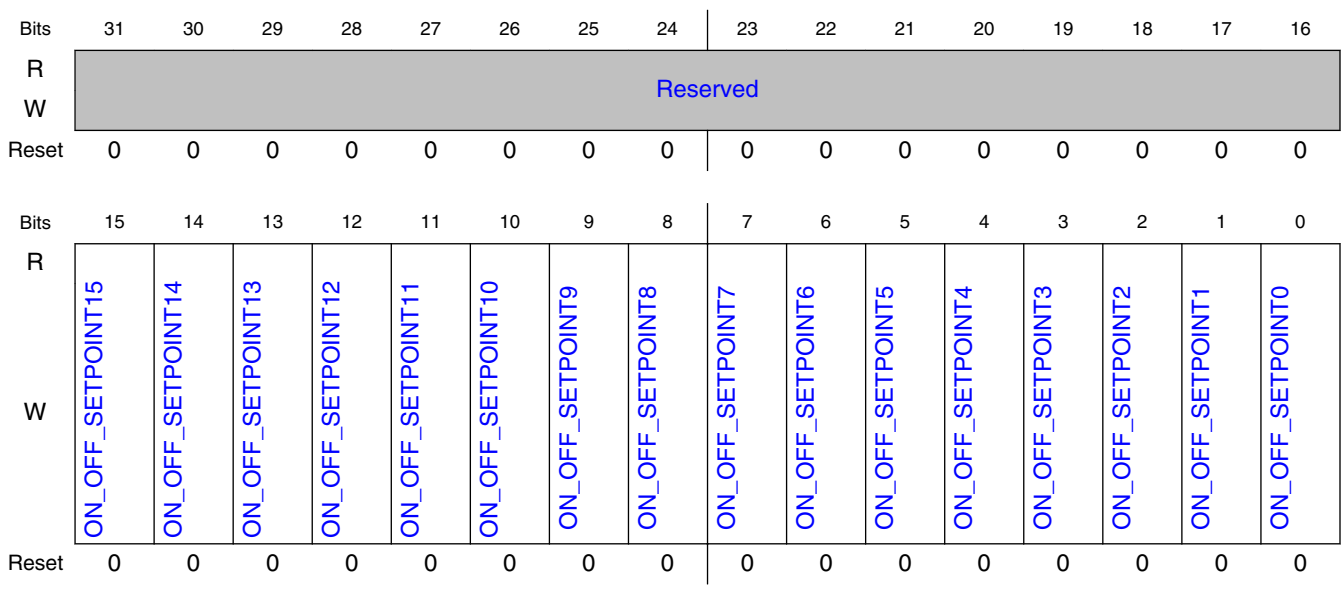
17.5.1.7 LDO_PLL_ENABLE_SP_REGISTER (LDO_PLL_ENABLE_SP)

LDO_PLL Setpoint enable and disable register

17.5.1.7.1 Offset

Register	Offset
LDO_PLL_ENABLE_SP	600h

17.5.1.7.2 Diagram



17.5.1.7.3 Fields

Field	Description
31-16 —	Reserved
15	ON_OFF_SETPOINT15

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT15	0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12 ON_OFF_SETPOINT12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2	ON_OFF_SETPOINT2

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT2	0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

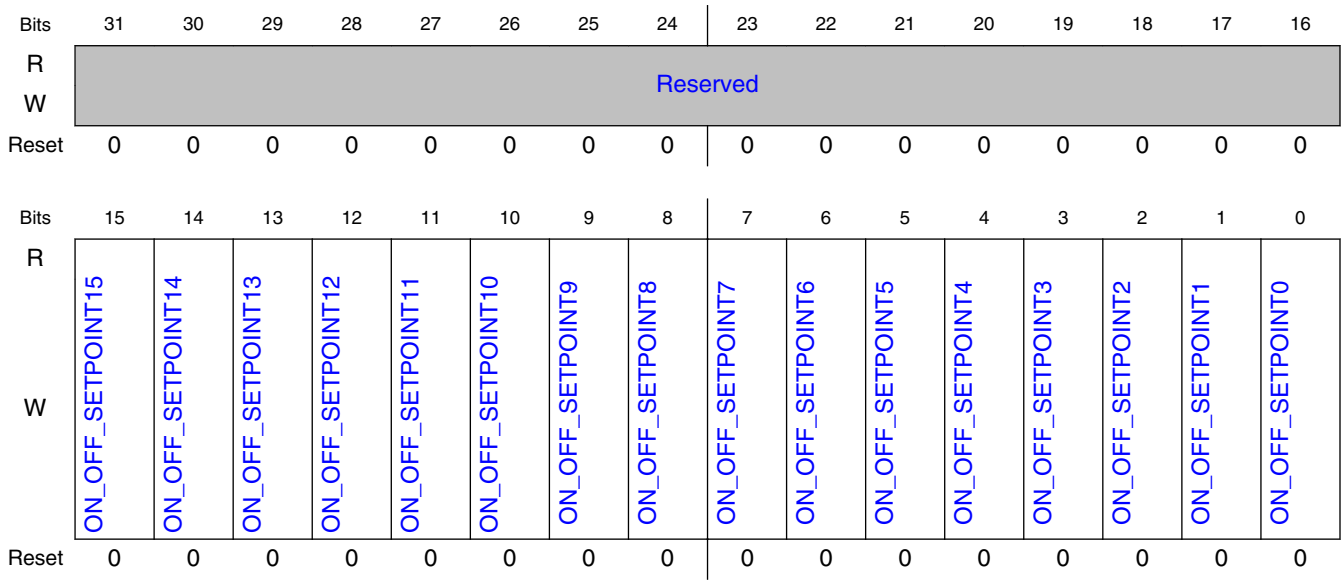
17.5.1.8 LDO_LPSR_ANA_ENABLE_SP_REGISTER (LDO_LPSR_ANA_ENABLE_SP)

LDO_LPSR_ANA LDO, the ON/OFF control did not show the responsibility of LDO ON/OFF, it is the combination usage for bypass function. The setting should be aligned with BYPASS. When LDO bypass enable , ***_ENABLE_SP - 1'b1, ***_TRACKING_EN_SP - 1'b1, ***_BYPASS_EN_SP - 1'b1.

17.5.1.8.1 Offset

Register	Offset
LDO_LPSR_ANA_ENABLE_SP	610h

17.5.1.8.2 Diagram



17.5.1.8.3 Fields

Field	Description
31-16 —	Reserved
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12 ON_OFF_SETPOINT12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

Table continues on the next page...

Field	Description
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

17.5.1.9 LDO_LPSR_ANA_LP_MODE_SP_REGISTER (LDO_LPSR_ANA_LP_MODE_SP)

LDO_LPSR_ANA high_power mode and low_power mode SP register

17.5.1.9.1 Offset

Register	Offset
LDO_LPSR_ANA_LP_MODE_SP	620h

17.5.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LP_MODE_SETPONIT15	LP_MODE_SETPONIT14	LP_MODE_SETPONIT13	LP_MODE_SETPONIT12	LP_MODE_SETPONIT11	LP_MODE_SETPONIT10	LP_MODE_SETPONIT9	LP_MODE_SETPONIT8	LP_MODE_SETPONIT7	LP_MODE_SETPONIT6	LP_MODE_SETPONIT5	LP_MODE_SETPONIT4	LP_MODE_SETPONIT3	LP_MODE_SETPONIT2	LP_MODE_SETPOINT1	LP_MODE_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.9.3 Fields

Field	Description
31-16 —	Reserved
15 LP_MODE_SETPONIT15	LP_MODE_SETPOINT15 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
14 LP_MODE_SETPONIT14	LP_MODE_SETPOINT14 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
13 LP_MODE_SETPONIT13	LP_MODE_SETPOINT13 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
12	LP_MODE_SETPOINT12

Table continues on the next page...

Field	Description
LP_MODE_SETPONIT12	0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
11 LP_MODE_SETPONIT11	LP_MODE_SETPOINT11 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
10 LP_MODE_SETPONIT10	LP_MODE_SETPOINT10 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
9 LP_MODE_SETPONIT9	LP_MODE_SETPOINT9 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
8 LP_MODE_SETPONIT8	LP_MODE_SETPOINT8 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
7 LP_MODE_SETPONIT7	LP_MODE_SETPOINT7 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
6 LP_MODE_SETPONIT6	LP_MODE_SETPOINT6 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
5 LP_MODE_SETPONIT5	LP_MODE_SETPOINT5 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
4 LP_MODE_SETPONIT4	LP_MODE_SETPOINT4 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
3 LP_MODE_SETPONIT3	LP_MODE_SETPOINT3 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
2 LP_MODE_SETPONIT2	LP_MODE_SETPOINT2 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
1 LP_MODE_SETPOINT1	LP_MODE_SETPOINT1 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
0 LP_MODE_SETPOINT0	LP_MODE_SETPOINT0 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint

17.5.1.10 LDO_LPSR_ANA_TRACKING_EN_SP_REGISTER (LDO_LPSR_ANA_TRACKING_EN_SP)

LDO_LPSR_ANA tracking mode sp enable register

17.5.1.10.1 Offset

Register	Offset
LDO_LPSR_ANA_TRACKING_EN_SP	630h

17.5.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRACKING_EN_SETPOINT15	TRACKING_EN_SETPOINT14	TRACKING_EN_SETPOINT13	TRACKING_EN_SETPOINT12	TRACKING_EN_SETPOINT11	TRACKING_EN_SETPOINT10	TRACKING_EN_SETPOINT9	TRACKING_EN_SETPOINT8	TRACKING_EN_SETPOINT7	TRACKING_EN_SETPOINT6	TRACKING_EN_SETPOINT5	TRACKING_EN_SETPOINT4	TRACKING_EN_SETPOINT3	TRACKING_EN_SETPOINT2	TRACKING_EN_SETPOINT1	TRACKING_EN_SETPOINT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.10.3 Fields

Field	Description
31-16 —	Reserved
15	TRACKING_EN_SETPOINT15 0 - Tracking mode is disabled in certain Setpoint

Table continues on the next page...

Field	Description
TRACKING_EN_SETPOINT15	1 - Tracking mode is enabled in certain Setpoint
14 TRACKING_EN_SETPOINT14	TRACKING_EN_SETPOINT14 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
13 TRACKING_EN_SETPOINT13	TRACKING_EN_SETPOINT13 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
12 TRACKING_EN_SETPOINT12	TRACKING_EN_SETPOINT12 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
11 TRACKING_EN_SETPOINT11	TRACKING_EN_SETPOINT11 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
10 TRACKING_EN_SETPOINT10	TRACKING_EN_SETPOINT10 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
9 TRACKING_EN_SETPOINT9	TRACKING_EN_SETPOINT9 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
8 TRACKING_EN_SETPOINT8	TRACKING_EN_SETPOINT8 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
7 TRACKING_EN_SETPOINT7	TRACKING_EN_SETPOINT7 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
6 TRACKING_EN_SETPOINT6	TRACKING_EN_SETPOINT6 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
5 TRACKING_EN_SETPOINT5	TRACKING_EN_SETPOINT5 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
4 TRACKING_EN_SETPOINT4	TRACKING_EN_SETPOINT4 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
3 TRACKING_EN_SETPOINT3	TRACKING_EN_SETPOINT3 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
2	TRACKING_EN_SETPOINT2 0 - Tracking mode is disabled in certain Setpoint

Table continues on the next page...

Field	Description
TRACKING_EN_SETPOINT2	1 - Tracking mode is enabled in certain Setpoint
1 TRACKING_EN_SETPOINT1	TRACKING_EN_SETPOINT1 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
0 TRACKING_EN_SETPOINT0	TRACKING_EN_SETPOINT0 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint

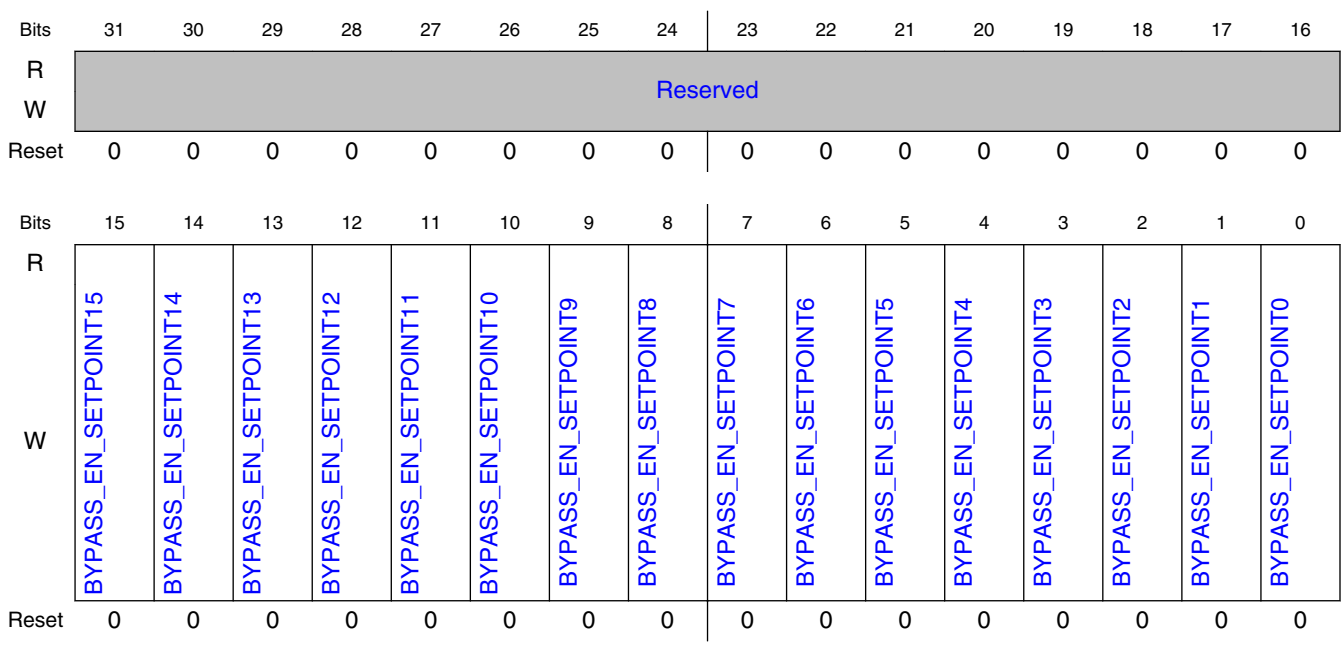
17.5.1.11 LDO_LPSR_ANA_BYPASS_EN_SP_REGISTER (LDO_LPSR_ANA_BYPASS_EN_SP)

LDO_LPSR_ANA bypass mode sp enable register

17.5.1.11.1 Offset

Register	Offset
LDO_LPSR_ANA_BYPASS_EN_SP	640h

17.5.1.11.2 Diagram



17.5.1.11.3 Fields

Field	Description
31-16 —	Reserved
15 BYPASS_EN_SETPOINT15	BYPASS_EN_SETPOINT15 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
14 BYPASS_EN_SETPOINT14	BYPASS_EN_SETPOINT14 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
13 BYPASS_EN_SETPOINT13	BYPASS_EN_SETPOINT13 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
12 BYPASS_EN_SETPOINT12	BYPASS_EN_SETPOINT12 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
11 BYPASS_EN_SETPOINT11	BYPASS_EN_SETPOINT11 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
10 BYPASS_EN_SETPOINT10	BYPASS_EN_SETPOINT10 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
9 BYPASS_EN_SETPOINT9	BYPASS_EN_SETPOINT9 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
8 BYPASS_EN_SETPOINT8	BYPASS_EN_SETPOINT 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
7 BYPASS_EN_SETPOINT7	BYPASS_EN_SETPOINT7 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
6 BYPASS_EN_SETPOINT6	BYPASS_EN_SETPOINT6 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
5 BYPASS_EN_SETPOINT5	BYPASS_EN_SETPOINT5 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
4	BYPASS_EN_SETPOINT4

Table continues on the next page...

Field	Description
BYPASS_EN_SETPOINT4	0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
3 BYPASS_EN_SETPOINT3	BYPASS_EN_SETPOINT3 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
2 BYPASS_EN_SETPOINT2	BYPASS_EN_SETPOINT2 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
1 BYPASS_EN_SETPOINT1	BYPASS_EN_SETPOINT1 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
0 BYPASS_EN_SETPOINT0	BYPASS_EN_SETPOINT0 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint

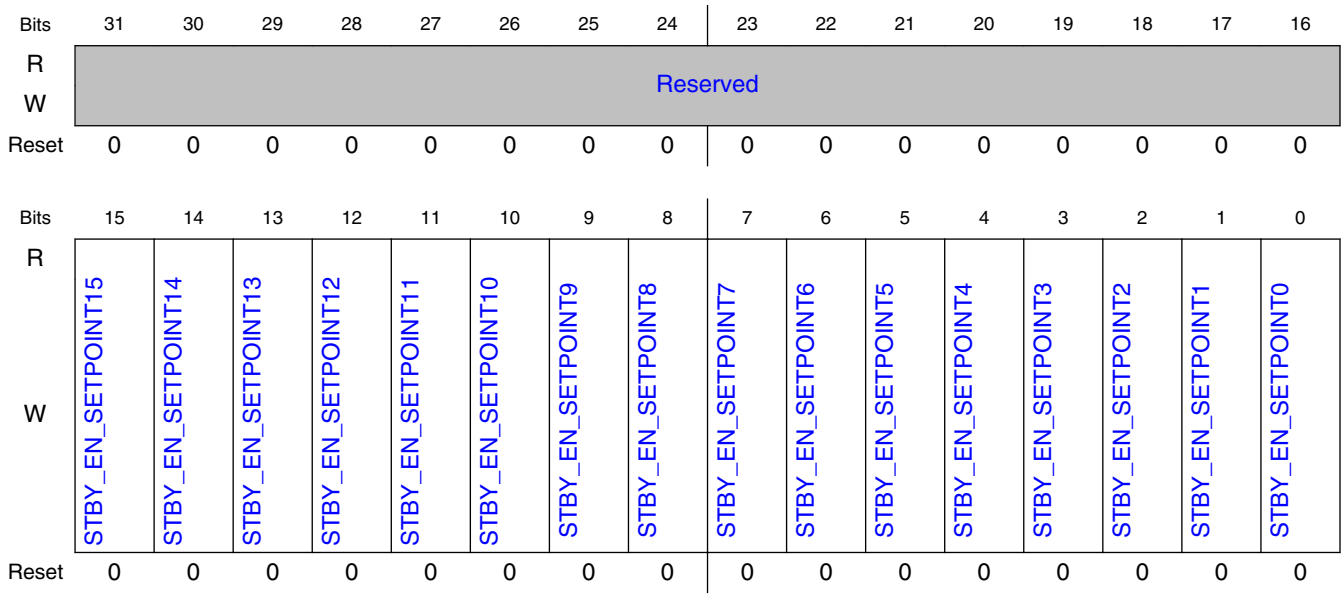
17.5.1.12 LDO_LPSR_ANA_STBY_EN_SP_REGISTER (LDO_LPSR_ANA_STBY_EN_SP)

LDO_LPSR_ANA STBY MODE enable sp register stby mode represent HP/LP mode transist.

17.5.1.12.1 Offset

Register	Offset
LDO_LPSR_ANA_STBY_EN_SP	650h

17.5.1.12.2 Diagram



17.5.1.12.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	STBY_EN_SETPOINT15 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
14 STBY_EN_SETPOINT14	STBY_EN_SETPOINT14 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	STBY_EN_SETPOINT13 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12 STBY_EN_SETPOINT12	STBY_EN_SETPOINT12 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	STBY_EN_SETPOINT11 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	STBY_EN_SETPOINT10 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
9 STBY_EN_SETPOINT9	STBY_EN_SETPOINT9 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	STBY_EN_SETPOINT8 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	STBY_EN_SETPOINT7 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	STBY_EN_SETPOINT6 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	STBY_EN_SETPOINT5 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SETPOINT4	STBY_EN_SETPOINT4 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SETPOINT3	STBY_EN_SETPOINT3 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SETPOINT2	STBY_EN_SETPOINT2 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
1 STBY_EN_SETPOINT1	STBY_EN_SETPOINT1 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SETPOINT0	STBY_EN_SETPOINT0 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

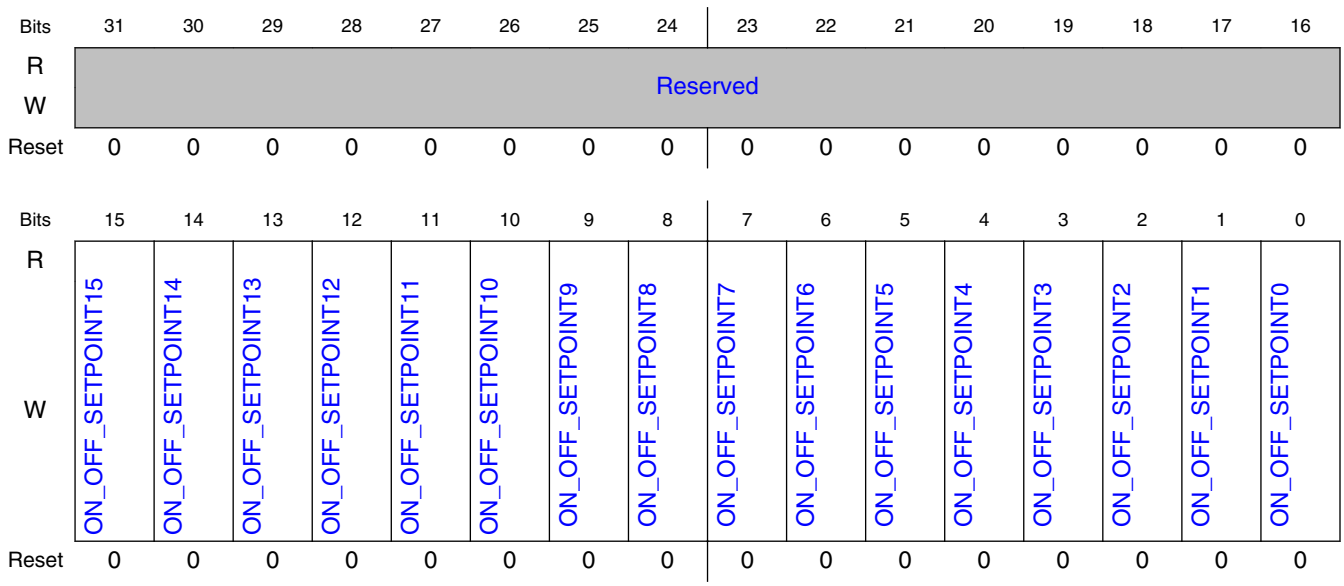
17.5.1.13 LDO_LPSR_DIG_ENABLE_SP_REGISTER (LDO_LPSR_DIG_ENABLE_SP)

LDO_LPSR_ANA LDO, the ON/OFF control did not show the responsibility of LDO ON/OFF, it is the combination usage for bypass function. The setting should be aligned with BYPASS. When LDO bypass enable , ***_ENABLE_SP - 1'b1, ***_TRACKING_EN_SP - 1'b1, ***_BYPASS_EN_SP - 1'b1.

17.5.1.13.1 Offset

Register	Offset
LDO_LPSR_DIG_ENABLE_SP	660h

17.5.1.13.2 Diagram



17.5.1.13.3 Fields

Field	Description
31-16 —	Reserved
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12	ON_OFF_SETPOINT12

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT12	0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

17.5.1.14 LDO_LPSR_DIG_TRG_SP0_REGISTER (LDO_LPSR_DIG_TRG_SP0)

This register store the lpsr_dig voltage step which will be performed during GPC mode.

17.5.1.14.1 Offset

Register	Offset
LDO_LPSR_DIG_TRG_SP0	670h

17.5.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VOLTAGE_SETPOINT3								VOLTAGE_SETPOINT2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VOLTAGE_SETPOINT1								VOLTAGE_SETPOINT0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.14.3 Fields

Field	Description
31-24 VOLTAGE_SETPOINT3	VOLTAGE_SETPOINT3 This field store the voltage step will be performed during identical Setpoint
23-16 VOLTAGE_SETPOINT2	VOLTAGE_SETPOINT2 This field store the voltage step will be performed during identical Setpoint
15-8 VOLTAGE_SETPOINT1	VOLTAGE_SETPOINT1 This field store the voltage step will be performed during identical Setpoint
7-0 VOLTAGE_SETPOINT0	VOLTAGE_SETPOINT0 This field store the voltage step will be performed during identical Setpoint

17.5.1.15 LDO_LPSR_DIG_TRG_SP1_REGISTER (LDO_LPSR_DIG_TRG_SP1)

This register store the lpsr_dig voltage step which will be performed during GPC mode.

17.5.1.15.1 Offset

Register	Offset
LDO_LPSR_DIG_TRG_SP1	680h

17.5.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VOLTAGE_SETPOINT7								VOLTAGE_SETPOINT6							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VOLTAGE_SETPOINT5								VOLTAGE_SETPOINT4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.15.3 Fields

Field	Description
31-24 VOLTAGE_SETPOINT7	VOLTAGE_SETPOINT7 This field store the voltage step will be performed during identical Setpoint
23-16 VOLTAGE_SETPOINT6	VOLTAGE_SETPOINT6 This field store the voltage step will be performed during identical Setpoint
15-8 VOLTAGE_SETPOINT5	VOLTAGE_SETPOINT5 This field store the voltage step will be performed during identical Setpoint
7-0 VOLTAGE_SETPOINT4	VOLTAGE_SETPOINT4 This field store the voltage step will be performed during identical Setpoint

17.5.1.16 LDO_LPSR_DIG_TRG_SP2_REGISTER (LDO_LPSR_DIG_TRG_SP2)

This register store the lpsr_dig voltage step which will be performed during GPC mode.

17.5.1.16.1 Offset

Register	Offset
LDO_LPSR_DIG_TRG_SP2	690h

17.5.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VOLTAGE_SETPOINT11								VOLTAGE_SETPOINT10							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VOLTAGE_SETPOINT9								VOLTAGE_SETPOINT8							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.16.3 Fields

Field	Description
31-24 VOLTAGE_SETPOINT11	VOLTAGE_SETPOINT11 This field store the voltage step will be performed during identical Setpoint
23-16 VOLTAGE_SETPOINT10	VOLTAGE_SETPOINT10 This field store the voltage step will be performed during identical Setpoint
15-8 VOLTAGE_SETPOINT9	VOLTAGE_SETPOINT9 This field store the voltage step will be performed during identical Setpoint
7-0 VOLTAGE_SETPOINT8	VOLTAGE_SETPOINT8 This field store the voltage step will be performed during identical Setpoint

17.5.1.17 LDO_LPSR_DIG_TRG_SP3_REGISTER (LDO_LPSR_DIG_TRG_SP3)

This register store the lpsr_dig voltage step which will be performed during GPC mode.

17.5.1.17.1 Offset

Register	Offset
LDO_LPSR_DIG_TRG_SP3	6A0h

17.5.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VOLTAGE_SETPOINT15								VOLTAGE_SETPOINT14							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VOLTAGE_SETPOINT13								VOLTAGE_SETPOINT12							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.17.3 Fields

Field	Description
31-24 VOLTAGE_SETPOINT15	VOLTAGE_SETPOINT15 This field store the voltage step will be performed during identical Setpoint
23-16 VOLTAGE_SETPOINT14	VOLTAGE_SETPOINT14 This field store the voltage step will be performed during identical Setpoint
15-8 VOLTAGE_SETPOINT13	VOLTAGE_SETPOINT13 This field store the voltage step will be performed during identical Setpoint
7-0 VOLTAGE_SETPOINT12	VOLTAGE_SETPOINT12 This field store the voltage step will be performed during identical Setpoint

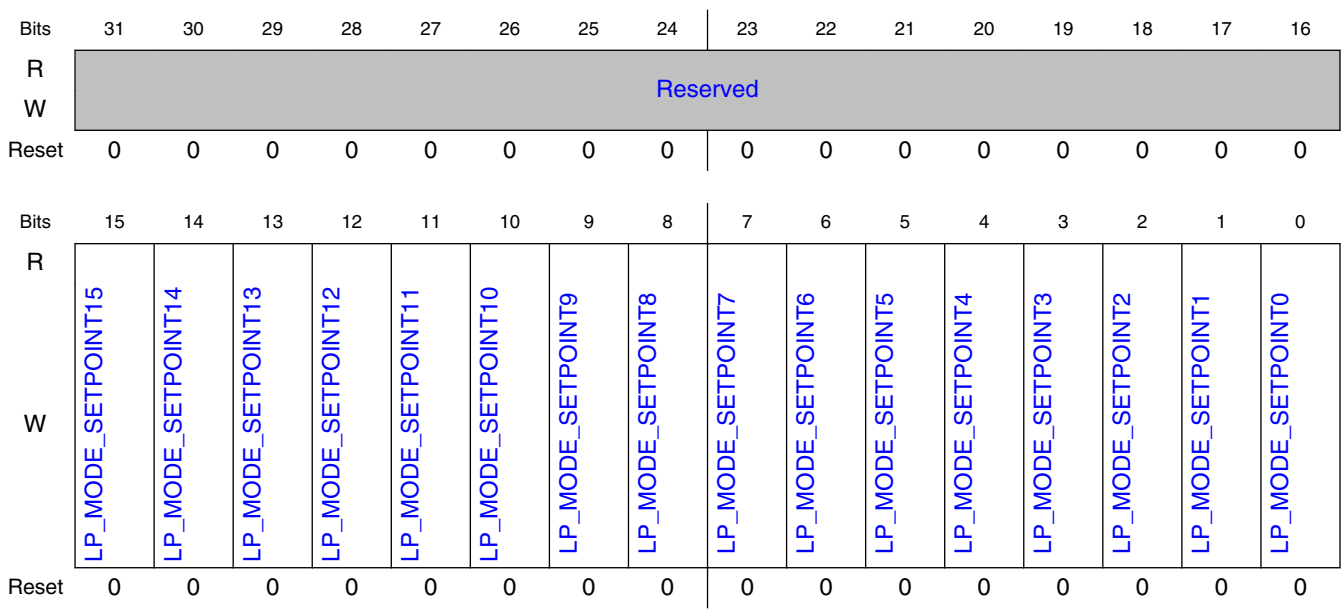
17.5.1.18 LDO_LPSR_DIG_LP_MODE_SP_REGISTER (LDO_LPSR_DIG_LP_MODE_SP)

This register provide the entry to the LP mode configure in GPC mode

17.5.1.18.1 Offset

Register	Offset
LDO_LPSR_DIG_LP_MODE_SP	6B0h

17.5.1.18.2 Diagram



17.5.1.18.3 Fields

Field	Description
31-16 —	Reserved
15 LP_MODE_SETPOINT15	LP_MODE_SETPOINT15 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
14	LP_MODE_SETPOINT14

Table continues on the next page...

Field	Description
LP_MODE_SETPOINT14	0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
13 LP_MODE_SETPOINT13	LP_MODE_SETPOINT13 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
12 LP_MODE_SETPOINT12	LP_MODE_SETPOINT12 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
11 LP_MODE_SETPOINT11	LP_MODE_SETPOINT11 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
10 LP_MODE_SETPOINT10	LP_MODE_SETPOINT10 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
9 LP_MODE_SETPOINT9	LP_MODE_SETPOINT9 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
8 LP_MODE_SETPOINT8	LP_MODE_SETPOINT8 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
7 LP_MODE_SETPOINT7	LP_MODE_SETPOINT7 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
6 LP_MODE_SETPOINT6	LP_MODE_SETPOINT6 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
5 LP_MODE_SETPOINT5	LP_MODE_SETPOINT5 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
4 LP_MODE_SETPOINT4	LP_MODE_SETPOINT4 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
3 LP_MODE_SETPOINT3	LP_MODE_SETPOINT3 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
2 LP_MODE_SETPOINT2	LP_MODE_SETPOINT2 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
1	LP_MODE_SETPOINT1

Table continues on the next page...

Field	Description
LP_MODE_SETPOINT1	0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint
0 LP_MODE_SETPOINT0	LP_MODE_SETPOINT0 0 - LP mode in certain Setpoint 1 - HP mode in certain Setpoint

17.5.1.19 LDO_LPSR_DIG_TRACKING_EN_SP_REGISTER (LDO_LPSR_DIG_TRACKING_EN_SP)

This register provide the entry to the tracking mode configure in GPC mode

17.5.1.19.1 Offset

Register	Offset
LDO_LPSR_DIG_TRACKING_EN_SP	6C0h

17.5.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TRACKING_EN_SETPOINT15	TRACKING_EN_SETPOINT14	TRACKING_EN_SETPOINT13	TRACKING_EN_SETPOINT12	TRACKING_EN_SETPOINT11	TRACKING_EN_SETPOINT10	TRACKING_EN_SETPOINT9	TRACKING_EN_SETPOINT8	TRACKING_EN_SETPOINT7	TRACKING_EN_SETPOINT6	TRACKING_EN_SETPOINT5	TRACKING_EN_SETPOINT4	TRACKING_EN_SETPOINT3	TRACKING_EN_SETPOINT2	TRACKING_EN_SETPOINT1	TRACKING_EN_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.19.3 Fields

Field	Description
31-16 —	Reserved
15 TRACKING_EN_SETPOINT15	TRACKING_EN_SETPOINT15 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
14 TRACKING_EN_SETPOINT14	TRACKING_EN_SETPOINT14 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
13 TRACKING_EN_SETPOINT13	TRACKING_EN_SETPOINT13 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
12 TRACKING_EN_SETPOINT12	TRACKING_EN_SETPOINT12 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
11 TRACKING_EN_SETPOINT11	TRACKING_EN_SETPOINT11 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
10 TRACKING_EN_SETPOINT10	TRACKING_EN_SETPOINT10 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
9 TRACKING_EN_SETPOINT9	TRACKING_EN_SETPOINT9 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
8 TRACKING_EN_SETPOINT8	TRACKING_EN_SETPOINT8 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
7 TRACKING_EN_SETPOINT7	TRACKING_EN_SETPOINT7 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
6 TRACKING_EN_SETPOINT6	TRACKING_EN_SETPOINT6 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
5 TRACKING_EN_SETPOINT5	TRACKING_EN_SETPOINT5 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
4	TRACKING_EN_SETPOINT4 0 - Tracking mode is disabled in certain Setpoint

Table continues on the next page...

Field	Description
TRACKING_EN_SETPOINT4	1 - Tracking mode is enabled in certain Setpoint
3 TRACKING_EN_SETPOINT3	TRACKING_EN_SETPOINT3 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
2 TRACKING_EN_SETPOINT2	TRACKING_EN_SETPOINT2 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
1 TRACKING_EN_SETPOINT1	TRACKING_EN_SETPOINT1 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint
0 TRACKING_EN_SETPOINT0	TRACKING_EN_SETPOINT0 0 - Tracking mode is disabled in certain Setpoint 1 - Tracking mode is enabled in certain Setpoint

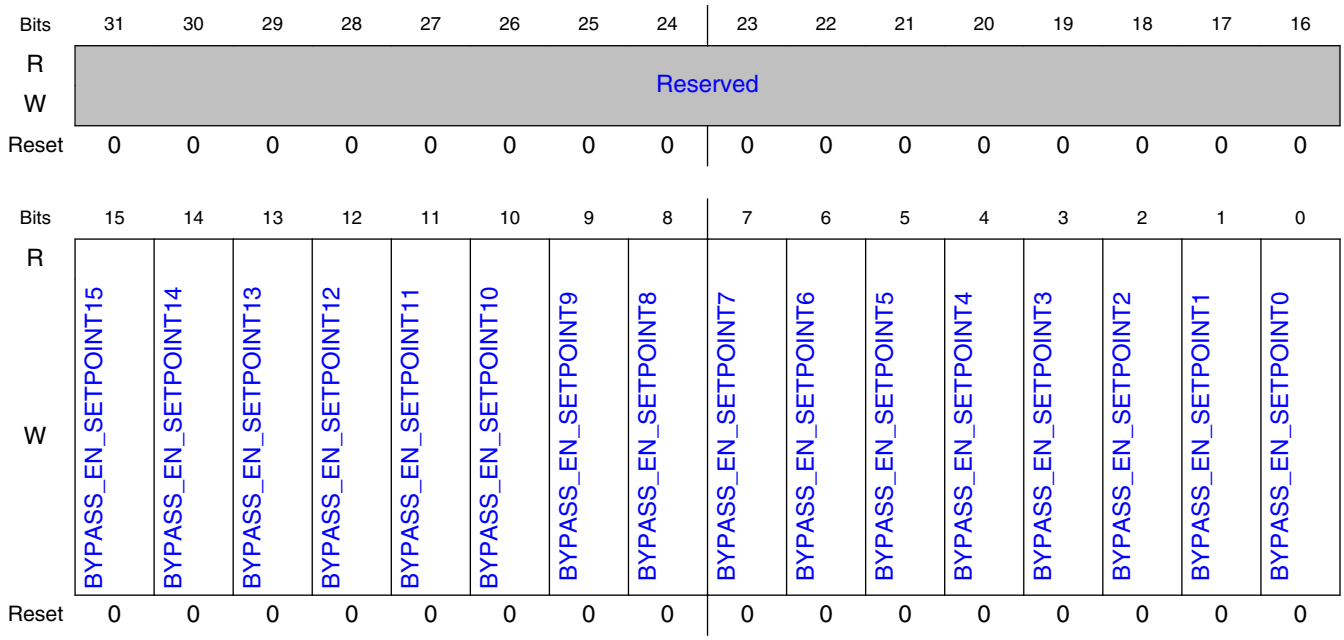
17.5.1.20 LDO_LPSR_DIG_BYPASS_EN_SP_REGISTER (LDO_LPSR_DIG_BYPASS_EN_SP)

This register provide the entry to the bypass_en configure in GPC mode

17.5.1.20.1 Offset

Register	Offset
LDO_LPSR_DIG_BYPASS_EN_SP	6D0h

17.5.1.20.2 Diagram



17.5.1.20.3 Fields

Field	Description
31-16 —	Reserved
15 BYPASS_EN_SETPOINT15	BYPASS_EN_SETPOINT15 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
14 BYPASS_EN_SETPOINT14	BYPASS_EN_SETPOINT14 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
13 BYPASS_EN_SETPOINT13	BYPASS_EN_SETPOINT13 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
12 BYPASS_EN_SETPOINT12	BYPASS_EN_SETPOINT12 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
11 BYPASS_EN_SETPOINT11	BYPASS_EN_SETPOINT11 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
10	BYPASS_EN_SETPOINT10 0 - Bypass mode is disabled in certain Setpoint

Table continues on the next page...

Field	Description
BYPASS_EN_SETPOINT10	1 - Bypass mode is enabled in certain Setpoint
9 BYPASS_EN_SETPOINT9	BYPASS_EN_SETPOINT9 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
8 BYPASS_EN_SETPOINT8	BYPASS_EN_SETPOINT8 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
7 BYPASS_EN_SETPOINT7	BYPASS_EN_SETPOINT7 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
6 BYPASS_EN_SETPOINT6	BYPASS_EN_SETPOINT6 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
5 BYPASS_EN_SETPOINT5	BYPASS_EN_SETPOINT5 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
4 BYPASS_EN_SETPOINT4	BYPASS_EN_SETPOINT4 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
3 BYPASS_EN_SETPOINT3	BYPASS_EN_SETPOINT3 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
2 BYPASS_EN_SETPOINT2	BYPASS_EN_SETPOINT2 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
1 BYPASS_EN_SETPOINT1	BYPASS_EN_SETPOINT1 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint
0 BYPASS_EN_SETPOINT0	BYPASS_EN_SETPOINT0 0 - Bypass mode is disabled in certain Setpoint 1 - Bypass mode is enabled in certain Setpoint

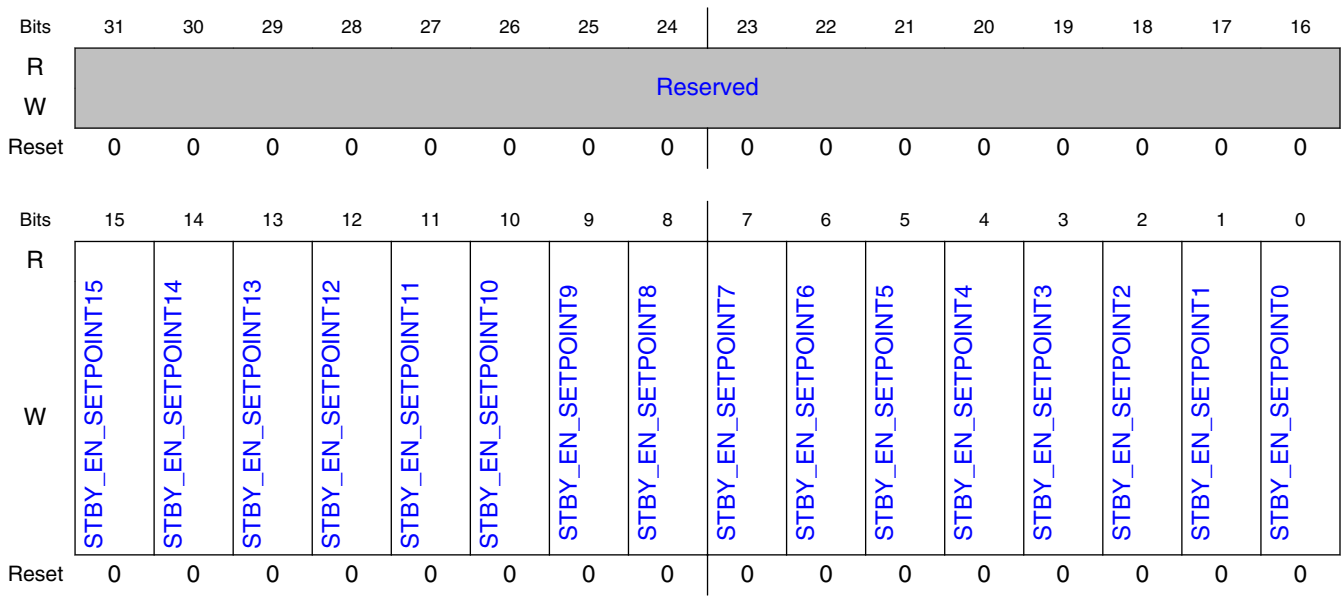
17.5.1.21 LDO_LPSR_DIG_STBY_EN_SP_REGISTER (LDO_LPSR_DIG_STBY_EN_SP)

This register provide the entry to the standby configure in GPC mode

17.5.1.21.1 Offset

Register	Offset
LDO_LPSR_DIG_STBY_EN_SP	6E0h

17.5.1.21.2 Diagram



17.5.1.21.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	STBY_EN_SETPOINT15 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
14 STBY_EN_SETPOINT14	STBY_EN_SETPOINT14 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	STBY_EN_SETPOINT13 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12	STBY_EN_SETPOINT12 0 - STANDBY mode is disabled

Table continues on the next page...

Field	Description
STBY_EN_SETPOINT12	1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	STBY_EN_SETPOINT11 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	STBY_EN_SETPOINT10 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
9 STBY_EN_SETPOINT9	STBY_EN_SETPOINT9 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	STBY_EN_SETPOINT8 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	STBY_EN_SETPOINT7 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	STBY_EN_SETPOINT6 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	STBY_EN_SETPOINT5 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SETPOINT4	STBY_EN_SETPOINT4 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SETPOINT3	STBY_EN_SETPOINT3 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SETPOINT2	STBY_EN_SETPOINT2 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
1 STBY_EN_SETPOINT1	STBY_EN_SETPOINT1 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SETPOINT0	STBY_EN_SETPOINT0 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

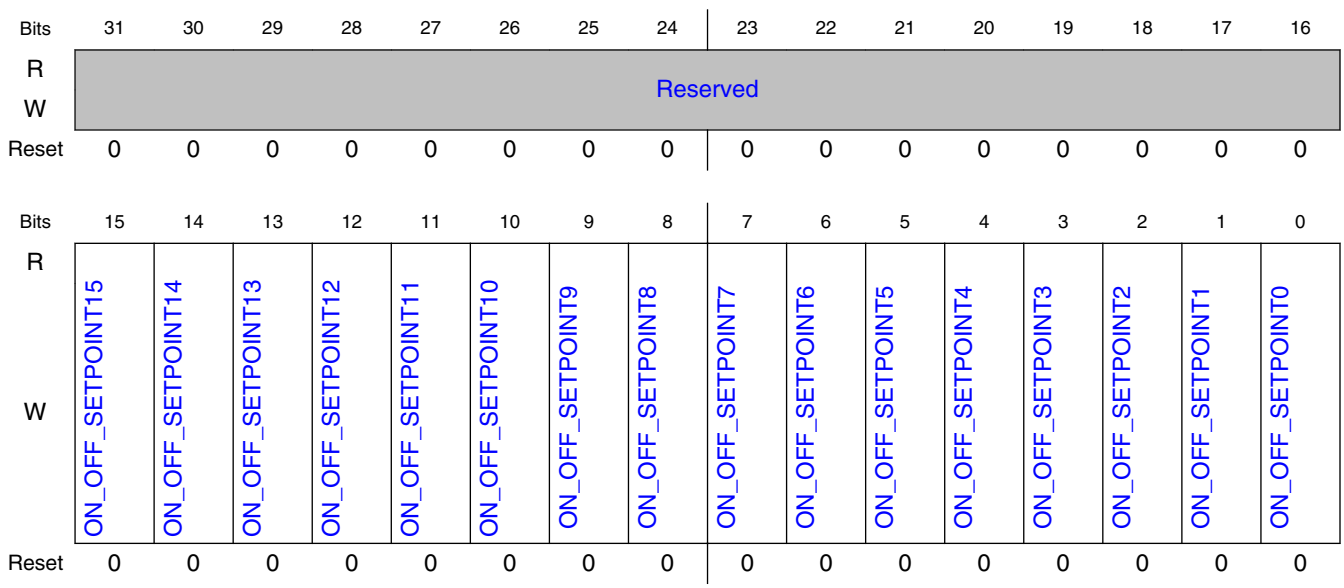
17.5.1.22 BANDGAP_ENABLE_SP_REGISTER (BANDGAP_ENABLE_SP)

BANDGAP, ON/OFF control register

17.5.1.22.1 Offset

Register	Offset
BANDGAP_ENABLE_SP	6F0h

17.5.1.22.2 Diagram



17.5.1.22.3 Fields

Field	Description
31-16 —	Reserved
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT14	1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12 ON_OFF_SETPOINT12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT1	1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

17.5.1.23 FBB_M7_ENABLE_SP_REGISTER (FBB_M7_ENABLE_SP)

FBB M7, ON/OFF control register

17.5.1.23.1 Offset

Register	Offset
FBB_M7_ENABLE_SP	700h

17.5.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ON_OFF_SETPOINT15	ON_OFF_SETPOINT14	ON_OFF_SETPOINT13	ON_OFF_SETPOINT12	ON_OFF_SETPOINT11	ON_OFF_SETPOINT10	ON_OFF_SETPOINT9	ON_OFF_SETPOINT8	ON_OFF_SETPOINT7	ON_OFF_SETPOINT6	ON_OFF_SETPOINT5	ON_OFF_SETPOINT4	ON_OFF_SETPOINT3	ON_OFF_SETPOINT2	ON_OFF_SETPOINT1	ON_OFF_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.23.3 Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Field	Description
—	
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12 ON_OFF_SETPOINT12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint

Table continues on the next page...

Field	Description
	1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

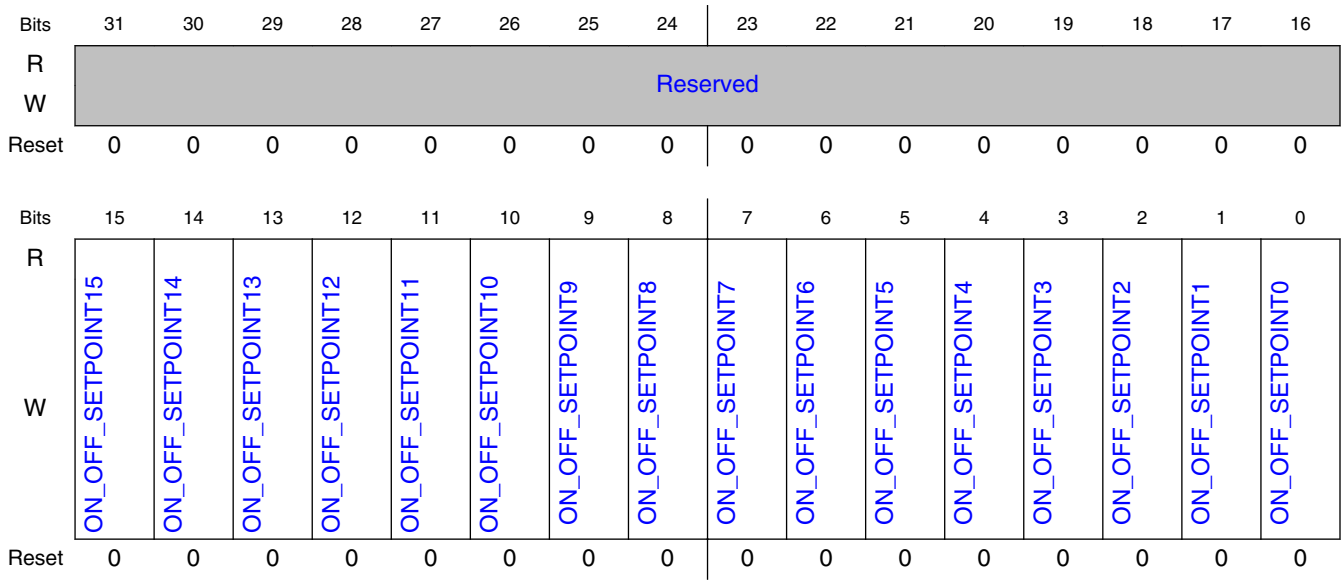
17.5.1.24 RBB_SOC_ENABLE_SP_REGISTER (RBB_SOC_ENABLE_SP)

Wellbias control SOC represent SOC + LPSR in the Setpoint definition table. No matter it is SW or GPC mode. SOC domain + LPSR domain , ON/OFF control register

17.5.1.24.1 Offset

Register	Offset
RBB_SOC_ENABLE_SP	710h

17.5.1.24.2 Diagram



17.5.1.24.3 Fields

Field	Description
31-16 —	Reserved
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12 ON_OFF_SETPOINT12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

Table continues on the next page...

Field	Description
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

17.5.1.25 RBB_LPSR_ENABLE_SP_REGISTER (RBB_LPSR_ENABLE_SP)

RBB LPSR, ON/OFF control register

17.5.1.25.1 Offset

Register	Offset
RBB_LPSR_ENABLE_SP	720h

17.5.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ON_OFF_SETPOINT15	ON_OFF_SETPOINT14	ON_OFF_SETPOINT13	ON_OFF_SETPOINT12	ON_OFF_SETPOINT11	ON_OFF_SETPOINT10	ON_OFF_SETPOINT9	ON_OFF_SETPOINT8	ON_OFF_SETPOINT7	ON_OFF_SETPOINT6	ON_OFF_SETPOINT5	ON_OFF_SETPOINT4	ON_OFF_SETPOINT3	ON_OFF_SETPOINT2	ON_OFF_SETPOINT1	ON_OFF_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.25.3 Fields

Field	Description
31-16 —	Reserved
15 ON_OFF_SETPOINT15	ON_OFF_SETPOINT15 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
14 ON_OFF_SETPOINT14	ON_OFF_SETPOINT14 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
13 ON_OFF_SETPOINT13	ON_OFF_SETPOINT13 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
12	ON_OFF_SETPOINT12 0 - ON in certain Setpoint

Table continues on the next page...

Field	Description
ON_OFF_SETPOINT12	1 - OFF in certain Setpoint
11 ON_OFF_SETPOINT11	ON_OFF_SETPOINT11 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
10 ON_OFF_SETPOINT10	ON_OFF_SETPOINT10 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
9 ON_OFF_SETPOINT9	ON_OFF_SETPOINT9 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
8 ON_OFF_SETPOINT8	ON_OFF_SETPOINT8 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
7 ON_OFF_SETPOINT7	ON_OFF_SETPOINT7 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
6 ON_OFF_SETPOINT6	ON_OFF_SETPOINT6 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
5 ON_OFF_SETPOINT5	ON_OFF_SETPOINT5 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
4 ON_OFF_SETPOINT4	ON_OFF_SETPOINT4 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
3 ON_OFF_SETPOINT3	ON_OFF_SETPOINT3 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
2 ON_OFF_SETPOINT2	ON_OFF_SETPOINT2 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
1 ON_OFF_SETPOINT1	ON_OFF_SETPOINT1 0 - ON in certain Setpoint 1 - OFF in certain Setpoint
0 ON_OFF_SETPOINT0	ON_OFF_SETPOINT0 0 - ON in certain Setpoint 1 - OFF in certain Setpoint

17.5.1.26 BANDGAP_STBY_EN_SP_REGISTER (BANDGAP_STBY_EN_SP)

BANDGAP STBY MODE enable sp register stby mode represent ON/OFF mode transist.

17.5.1.26.1 Offset

Register	Offset
BANDGAP_STBY_EN_SP	730h

17.5.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_EN_SETPOINT15	STBY_EN_SETPOINT14	STBY_EN_SETPOINT13	STBY_EN_SETPOINT12	STBY_EN_SETPOINT11	STBY_EN_SETPOINT10	STBY_EN_SETPOINT9	STBY_EN_SETPOINT8	STBY_EN_SETPOINT7	STBY_EN_SETPOINT6	STBY_EN_SETPOINT5	STBY_EN_SETPOINT4	STBY_EN_SETPOINT3	STBY_EN_SETPOINT2	STBY_EN_SETPOINT1	STBY_EN_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.26.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
14 STBY_EN_SETPOINT14	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12 STBY_EN_SETPOINT12	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
9 STBY_EN_SETPOINT9	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SETPOINT4	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SETPOINT3	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SETPOINT2	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
1 STBY_EN_SETPOINT1	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SETPOINT0	STBY_EN_SETPOINT 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

17.5.1.27 PLL_LDO_STBY_EN_SP_REGISTER (PLL_LDO_STBY_EN_SP)

PLL_LDO STBY MODE enable sp register stby mode represent ON/OFF mode transist.

17.5.1.27.1 Offset

Register	Offset
PLL_LDO_STBY_EN_SP	740h

17.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_EN_SETPOINT15	STBY_EN_SETPOINT14	STBY_EN_SETPOINT13	STBY_EN_SETPOINT12	STBY_EN_SETPOINT11	STBY_EN_SETPOINT10	STBY_EN_SETPOINT9	STBY_EN_SETPOINT8	STBY_EN_SETPOINT7	STBY_EN_SETPOINT6	STBY_EN_SETPOINT5	STBY_EN_SETPOINT4	STBY_EN_SETPOINT3	STBY_EN_SETPOINT2	STBY_EN_SETPOINT1	STBY_EN_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.27.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
14 STBY_EN_SETPOINT14	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12 STBY_EN_SETPOINT12	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
9 STBY_EN_SETPOINT9	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4	Standy mode 0 - STANDBY mode is disabled

Table continues on the next page...

Field	Description
STBY_EN_SET POINT4	1 - STANDBY mode is enabled
3 STBY_EN_SET POINT3	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SET POINT2	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
1 STBY_EN_SET POINT1	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SET POINT0	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

17.5.1.28 FBB_M7_STBY_EN_SP_REGISTER (FBB_M7_STBY_EN_SP)

FBB_M7 STBY MODE enable sp register stby mode represent ON/OFF mode transist.

17.5.1.28.1 Offset

Register	Offset
FBB_M7_STBY_EN_SP	750h

17.5.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_EN_SETPOINT15	STBY_EN_SETPOINT14	STBY_EN_SETPOINT13	STBY_EN_SETPOINT12	STBY_EN_SETPOINT11	STBY_EN_SETPOINT10	STBY_EN_SETPOINT9	STBY_EN_SETPOINT8	STBY_EN_SETPOINT7	STBY_EN_SETPOINT6	STBY_EN_SETPOINT5	STBY_EN_SETPOINT4	STBY_EN_SETPOINT3	STBY_EN_SETPOINT2	STBY_EN_SETPOINT1	STBY_EN_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.28.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
14 STBY_EN_SETPOINT14	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12 STBY_EN_SETPOINT12	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
9 STBY_EN_SET POINT9	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SET POINT8	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SET POINT7	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SET POINT6	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SET POINT5	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SET POINT4	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SET POINT3	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SET POINT2	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
1 STBY_EN_SET POINT1	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SET POINT0	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

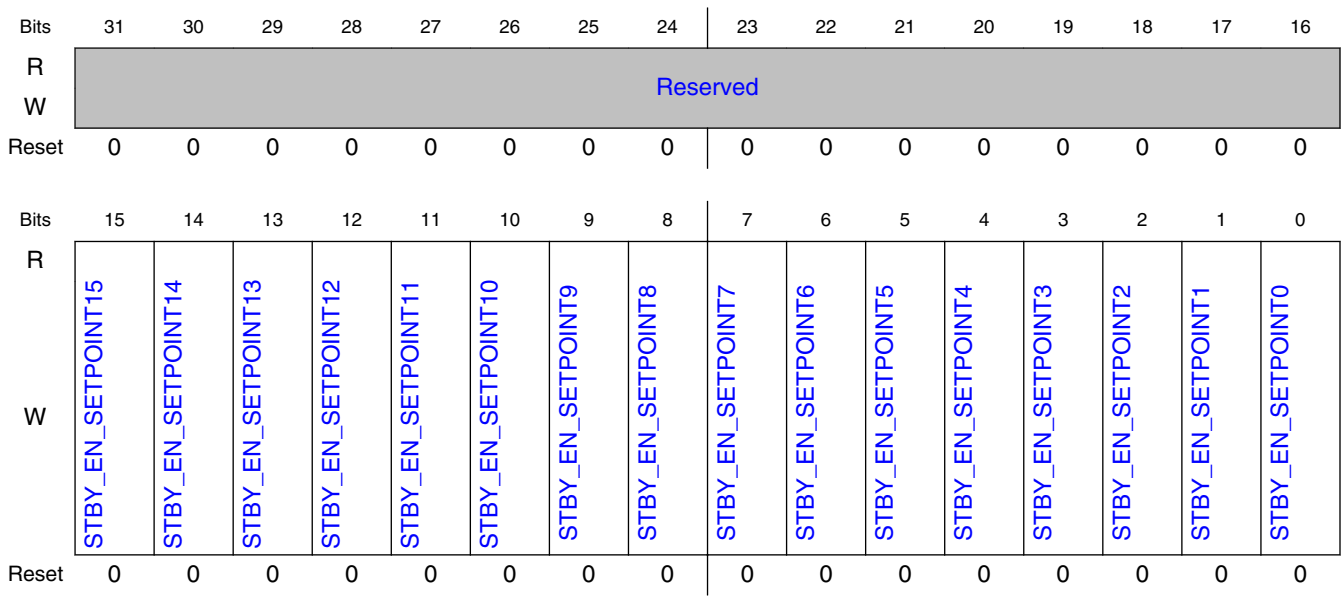
17.5.1.29 RBB_SOC_STBY_EN_SP_REGISTER (RBB_SOC_STBY_EN_SP)

SOC domain + LPSR domain RBB_SOC STBY MODE enable sp register stby mode represent ON/OFF mode transist.

17.5.1.29.1 Offset

Register	Offset
RBB_SOC_STBY_EN_SP	760h

17.5.1.29.2 Diagram



17.5.1.29.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
14 STBY_EN_SETPOINT14	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12	Standy mode 0 - STANDBY mode is disabled

Table continues on the next page...

Field	Description
STBY_EN_SETPOINT12	1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
9 STBY_EN_SETPOINT9	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SETPOINT4	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SETPOINT3	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SETPOINT2	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
1 STBY_EN_SETPOINT1	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SETPOINT0	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

17.5.1.30 RBB_LPSR_STBY_EN_SP_REGISTER (RBB_LPSR_STBY_EN_SP)

RBB_LPSR STBY MODE enable sp register stby mode represent ON/OFF mode transist.

17.5.1.30.1 Offset

Register	Offset
RBB_LPSR_STBY_EN_SP	770h

17.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_EN_SETPOINT15	STBY_EN_SETPOINT14	STBY_EN_SETPOINT13	STBY_EN_SETPOINT12	STBY_EN_SETPOINT11	STBY_EN_SETPOINT10	STBY_EN_SETPOINT9	STBY_EN_SETPOINT8	STBY_EN_SETPOINT7	STBY_EN_SETPOINT6	STBY_EN_SETPOINT5	STBY_EN_SETPOINT4	STBY_EN_SETPOINT3	STBY_EN_SETPOINT2	STBY_EN_SETPOINT1	STBY_EN_SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.5.1.30.3 Fields

Field	Description
31-16 —	Reserved
15 STBY_EN_SETPOINT15	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
14 STBY_EN_SETPOINT14	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
13 STBY_EN_SETPOINT13	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
12 STBY_EN_SETPOINT12	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
11 STBY_EN_SETPOINT11	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
10 STBY_EN_SETPOINT10	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
9 STBY_EN_SETPOINT9	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
8 STBY_EN_SETPOINT8	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
7 STBY_EN_SETPOINT7	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
6 STBY_EN_SETPOINT6	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
5 STBY_EN_SETPOINT5	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
4 STBY_EN_SETPOINT4	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
3 STBY_EN_SETPOINT3	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
2 STBY_EN_SETPOINT2	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

Table continues on the next page...

Field	Description
1 STBY_EN_SET POINT1	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled
0 STBY_EN_SET POINT0	Standy mode 0 - STANDBY mode is disabled 1 - STANDBY mode is enabled

17.5.1.31 FBB_M7_CONFIGURE_REGISTER (FBB_M7_CONFIGURE)

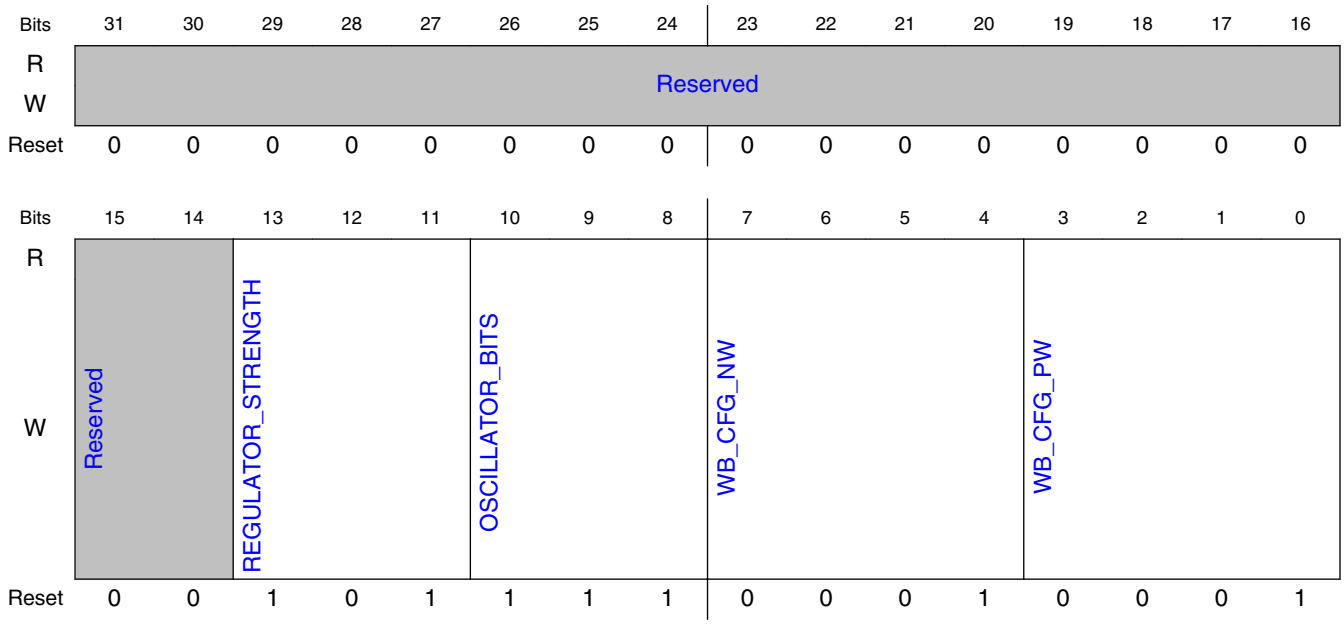
This register is used in GPC mode to make selection of BIAS configure function.
Software and Hardware (GPC) mode

- WB_EN : Wellbias Enable
- WB_CFG_1P8[1] -
  - 0 : RBB
  - 1 : FBB
- WB_PWR_SW_EN_1P8 -
  - [0] FBB M7
  - [1] RBB LPSR
  - [2] RBB SOG
- WBB_VDD_SEL - LPSR_DIG_LDO or DCDC
- WB_CFG_1P8[4:2] - Select size of of bias area.
- WB_CFG_1P8[5] - Adaptive function
- WB_CFG_1P8[8:6] - Oscillator bits

17.5.1.31.1 Offset

Register	Offset
FBB_M7_CONFIGURE	780h

17.5.1.31.2 Diagram



17.5.1.31.3 Fields

Field	Description
31-14 —	Reserved
13-11 REGULATOR_STRENGTH	regulator_strength This field is used in GPC mode to select regulator_strength. The hardware and software mode settings are the same. See WB_CFG_1P8[4:2] for details.
10-8 OSCILLATOR_BITS	oscillator_bits This field is used in GPC mode to select size of oscillator bits. The hardware and software mode settings are the same. See WB_CFG_1P8[8:6] for details.
7-4 WB_CFG_NW	wb_cfg_nw This field is used in GPC mode and it takes control of Software mode NWELL Output Voltage Range Selection defined by WB_NW_LVL_1P8. RBB function enabled; valid range is 0000 ~ 1000.
3-0 WB_CFG_PW	wb_cfg_pw This field is used in GPC mode and it takes control of Software mode PWELL Output Voltage Range Selection defined by WB_PW_LVL_1P8. Valid range is 0000 ~ 1000.

17.5.1.32 RBB_LPSR_CONFIGURE_REGISTER (RBB_LPSR_CONFIGURE)

This register is used in GPC mode to make selection of BIAS configure function.

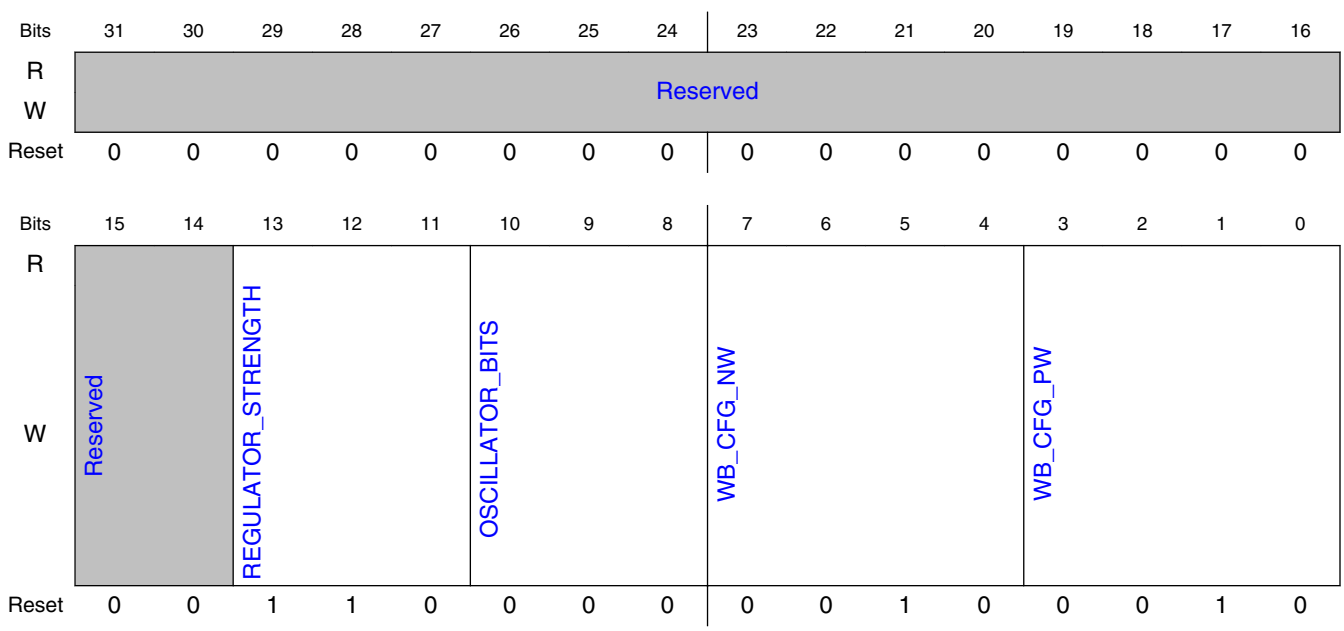
Software and Hardware (GPC) mode

- WB_EN : Wellbias Enable
- WB_CFG_1P8[1] -
  - 0 : RBB
  - 1 : FBB
- WB_PWR_SW_EN_1P8 -
  - [0] FBB M7
  - [1] RBB LPSR
  - [2] RBB SOG
- WBB_VDD_SEL - LPSR_DIG_LDO or DCDC
- WB_CFG_1P8[4:2] - Select size of of bias area.
- WB_CFG_1P8[5] - Adaptive function
- WB_CFG_1P8[8:6] - Oscillator bits

17.5.1.32.1 Offset

Register	Offset
RBB_LPSR_CONFIGURE	790h

17.5.1.32.2 Diagram



17.5.1.32.3 Fields

Field	Description
31-14 —	Reserved
13-11 REGULATOR_STRENGTH	regulator_strength This field is used in GPC mode to select regulator_strength. The hardware and software mode settings are the same. See WB_CFG_1P8[4:2] for details.
10-8 OSCILLATOR_BITS	oscillator_bits This field is used in GPC mode to select size of oscillator bits. The hardware and software mode settings are the same. See WB_CFG_1P8[8:6] for details.
7-4 WB_CFG_NW	wb_cfg_nw This field is used in GPC mode and it takes control of Software mode NWELL Output Voltage Range Selection defined by WB_NW_LVL_1P8. RBB function enabled; valid range is 0000 ~ 1000.
3-0 WB_CFG_PW	wb_cfg_pw This field is used in GPC mode and it takes control of Software mode PWELL Output Voltage Range Selection defined by WB_PW_LVL_1P8. Valid range is 0000 ~ 1000.

17.5.1.33 RBB_SOC_CONFIGURE_REGISTER (RBB_SOC_CONFIGURE)

This register is used in GPC mode to make selection of BIAS configure function.

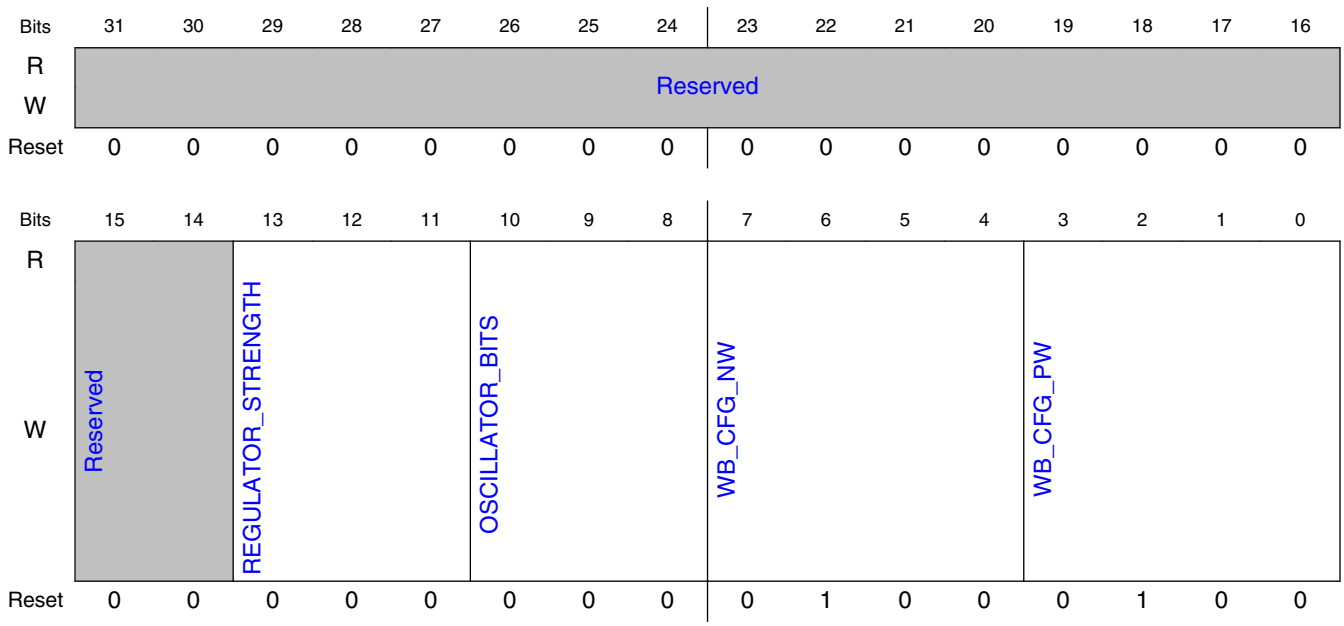
Software and Hardware (GPC) mode

- WB_EN : Wellbias Enable
- WB_CFG_1P8[1] -
  - 0 : RBB
  - 1 : FBB
- WB_PWR_SW_EN_1P8 -
  - [0] FBB M7
  - [1] RBB LPSR
  - [2] RBB SOG
- WBB_VDD_SEL - LPSR_DIG_LDO or DCDC
- WB_CFG_1P8[4:2] - Select size of of bias area.
- WB_CFG_1P8[5] - Adaptive function
- WB_CFG_1P8[8:6] - Oscillator bits

17.5.1.33.1 Offset

Register	Offset
RBB_SOC_CONFIGURE	7A0h

17.5.1.33.2 Diagram



17.5.1.33.3 Fields

Field	Description
31-14 —	Reserved
13-11 REGULATOR_STRENGTH	regulator_strength This field is used in GPC mode to select regulator_strength. The hardware and software mode settings are the same. See WB_CFG_1P8[4:2] for details.
10-8 OSCILLATOR_BITS	oscillator_bits This field is used in GPC mode to select size of oscillator bits. The hardware and software mode settings are the same. See WB_CFG_1P8[8:6] for details.
7-4 WB_CFG_NW	wb_cfg_nw This field is used in GPC mode and it takes control of Software mode NWELL Output Voltage Range Selection defined by WB_NW_LVL_1P8. RBB function enabled; valid range is 0000 ~ 1000.
3-0 WB_CFG_PW	wb_cfg_pw This field is used in GPC mode and it takes control of Software mode PWELL Output Voltage Range Selection defined by WB_PW_LVL_1P8. Valid range is 0000 ~ 1000.

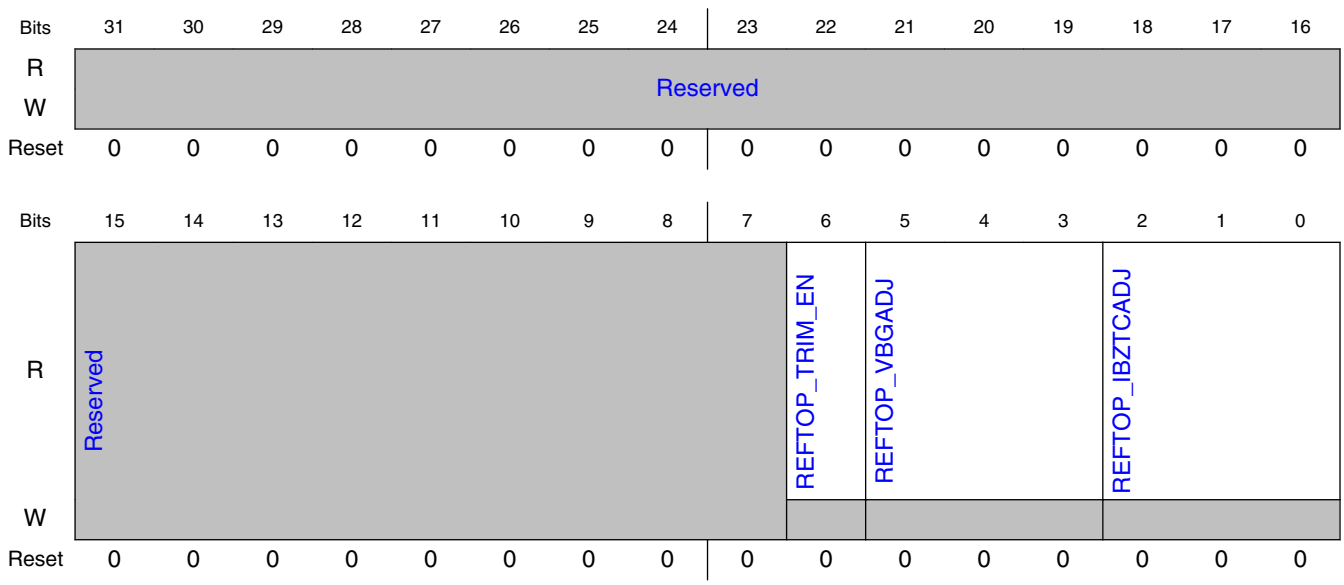
17.5.1.34 REFTOP_OTP_TRIM_VALUE_REGISTER (REFTOP_OTP_TRIM_VALUE)

This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.

17.5.1.34.1 Offset

Register	Offset
REFTOP_OTP_TRIM_VALUE	7B0h

17.5.1.34.2 Diagram



17.5.1.34.3 Fields

Field	Description
31-7 —	Reserved
6 REFTOP_TRIM_EN	REFTOP_TRIM_EN This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.
5-3	REFTOP_VBGADJ

Table continues on the next page...

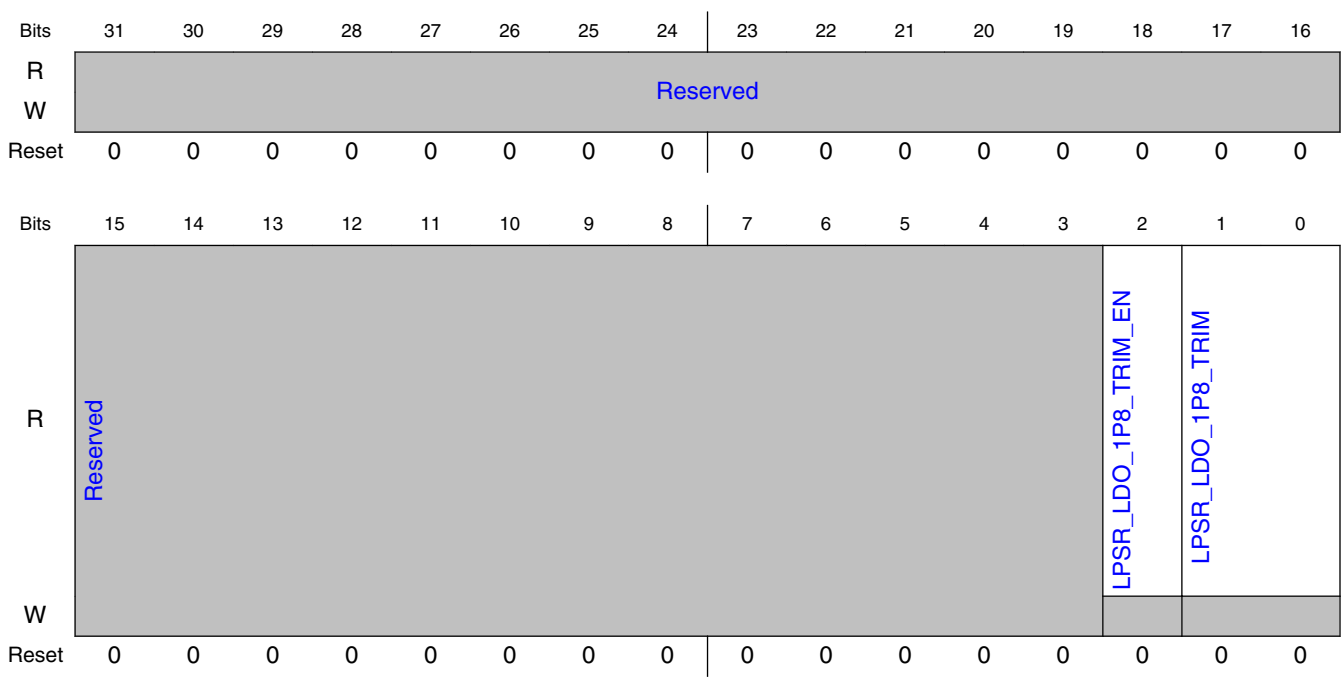
Field	Description
REFTOP_VBGADJ	This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.
2-0 REFTOP_IBZTCADJ	REFTOP_IBZTCADJ This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.

17.5.1.35 LPSR_1P8_LDO_OTP_TRIM_VALUE_REGISTER (LPSR_1P8_LDO_OTP_TRIM_VALUE)

17.5.1.35.1 Offset

Register	Offset
LPSR_1P8_LDO_OTP_TRIM_VALUE	7D0h

17.5.1.35.2 Diagram



17.5.1.35.3 Fields

Field	Description
31-3 —	Reserved
2 LPSR_LDO_1P8_TRIM_EN	LPSR_LDO_1P8_TRIM_EN This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.
1-0 LPSR_LDO_1P8_TRIM	LPSR_LDO_1P8_TRIM This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.

17.5.2 PMU_LDO_SNVS register descriptions

17.5.2.1 LDO_SNVS memory map

ANADIG base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
510	PMU_LDO_LPSR_ANA_REGISTER (PMU_LDO_LPSR_ANA)	32	RW	0000_0108
520	PMU_LDO_LPSR_DIG_2_REGISTER (PMU_LDO_LPSR_DIG_2)	32	RW	0000_0002
530	PMU_LDO_LPSR_DIG_REGISTER (PMU_LDO_LPSR_DIG)	32	RW	0130_1C05

17.5.2.2 PMU_LDO_LPSR_ANA_REGISTER (PMU_LDO_LPSR_ANA)

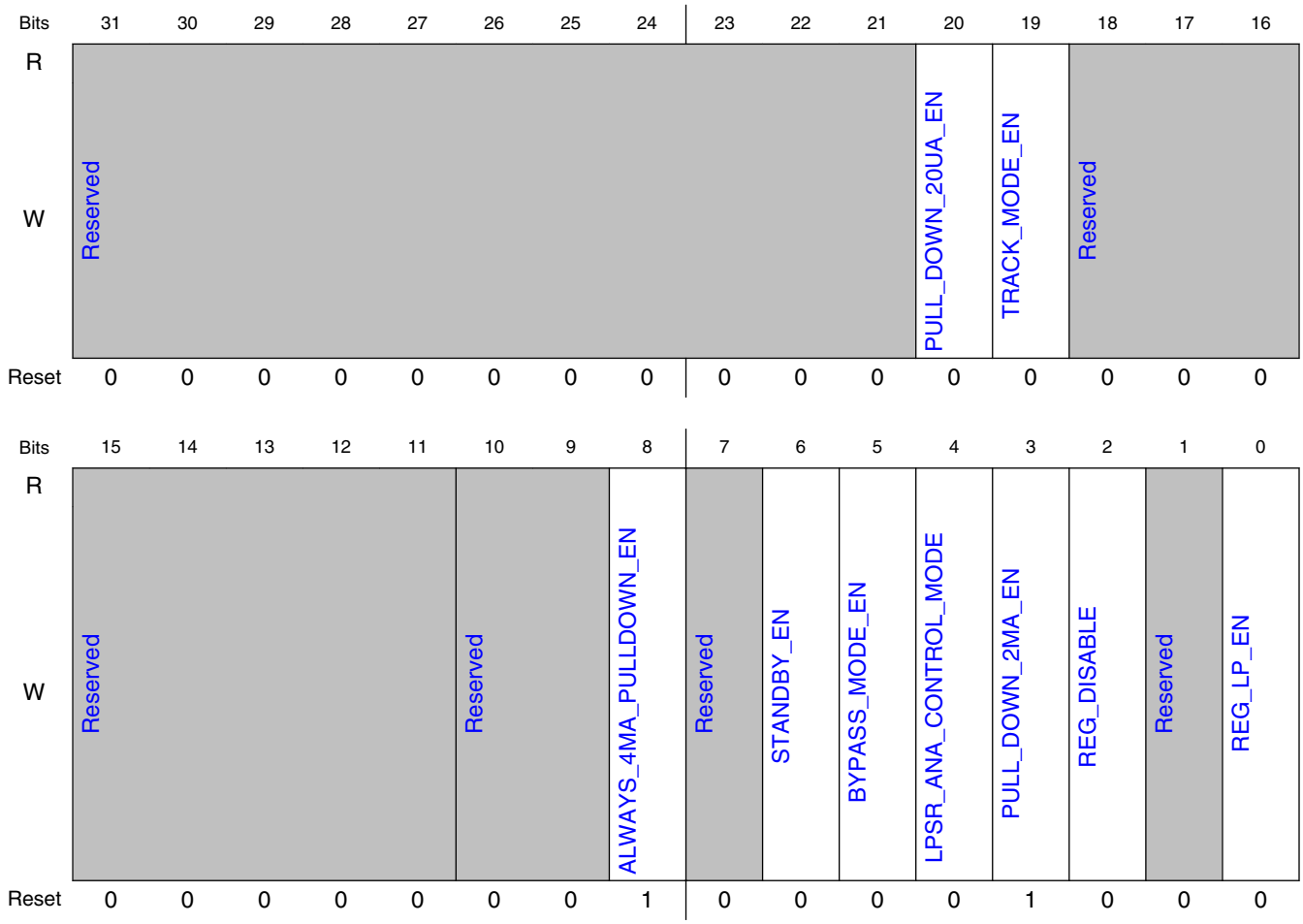
PMU_LDO_LPSR_ANA_Regulator_Control_Register

This register defines the control and status bits for LDO_LPSR_ANA regulator. This regulator is designed to power the digital portions of the analog cells

17.5.2.2.1 Offset

Register	Offset
PMU_LDO_LPSR_ANA	510h

17.5.2.2.2 Diagram



17.5.2.2.3 Fields

Field	Description
31-21 —	Reserved Always set to zero (0).
20 PULL_DOWN_20UA_EN	pull_down_20ua_en High: enable 20uA loading to prevent the overshoot.
19 TRACK_MODE_EN	Track Mode Enable This bit lets the power switch enter a preparation stage that allows power supply switching between LDO and DCDC. This bit should be set before the power supply switch and cleared after power switch is complete. This bit is used during software mode. 0 - Normal use 1 - Switch preparation

Table continues on the next page...

Field	Description
18-11 —	Reserved Always set to zero (0).
10-9 —	Reserved Always set to zero (0).
8 ALWAYS_4MA_PULLDOWN_EN	always_4ma_pulldown_en High: enable 4mA loading to prevent the big voltage drop when a sharp loading coming.It's recommend to set this bit to high under the reset, and set it to low under the normal use case.
7 —	Reserved Always set to zero (0).
6 STANDBY_EN	standby_en This is the standby mode for LDO_LPSR_ANA
5 BYPASS_MODE_EN	bypass_mode_en Work together with track_mode_enable to ensure the LDO_1P8 can be bypassed by DCDC_1P8 well. This the bypass mode for lpsr_ana
4 LPSR_ANA_CONTROL_MODE	LPSR_ANA_CONTROL_MODE LPSR_ANA_CONTROL_MODE This bit is the mode select between software mode and GPC mode. 0 - SW Control. Software control mode 1 - HW Control. Hardware / GPC control mode
3 PULL_DOWN_2MA_EN	pull_down_2ma_en High: enable 2mA loading to prevent the overshoot.
2 REG_DISABLE	reg_disable high: disable the output of "vreg_1p8"
1 —	Reserved
0 REG_LP_EN	reg_lp_en High: enable the low-power mode of the ldo

17.5.2.3 PMU_LDO_LPSR_DIG_2_REGISTER (PMU_LDO_LPSR_DIG_2)

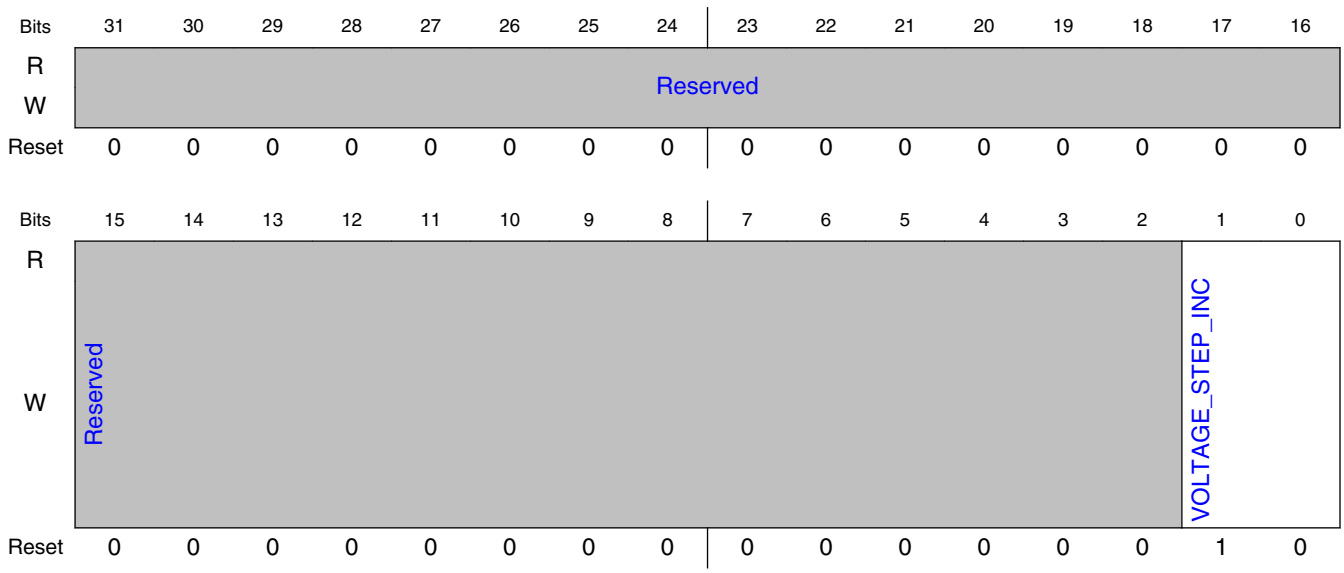
PMU_LDO_LPSR_DIG_2_Regulator_Control Register

This register defines the control and status bits for LDO_LPSR_DIG regulator. This regulator is designed to power the digital portions of the analog cells

17.5.2.3.1 Offset

Register	Offset
PMU_LDO_LPSR_DIG_2	520h

17.5.2.3.2 Diagram



17.5.2.3.3 Fields

Field	Description
31-2 —	Reserved Always set to zero (0).
1-0 VOLTAGE_STEP_INC	voltage_step_inc voltage_step_time for lpsr_dig 0x00:15us 0x01:25us 0x10:50us 0x11:100us

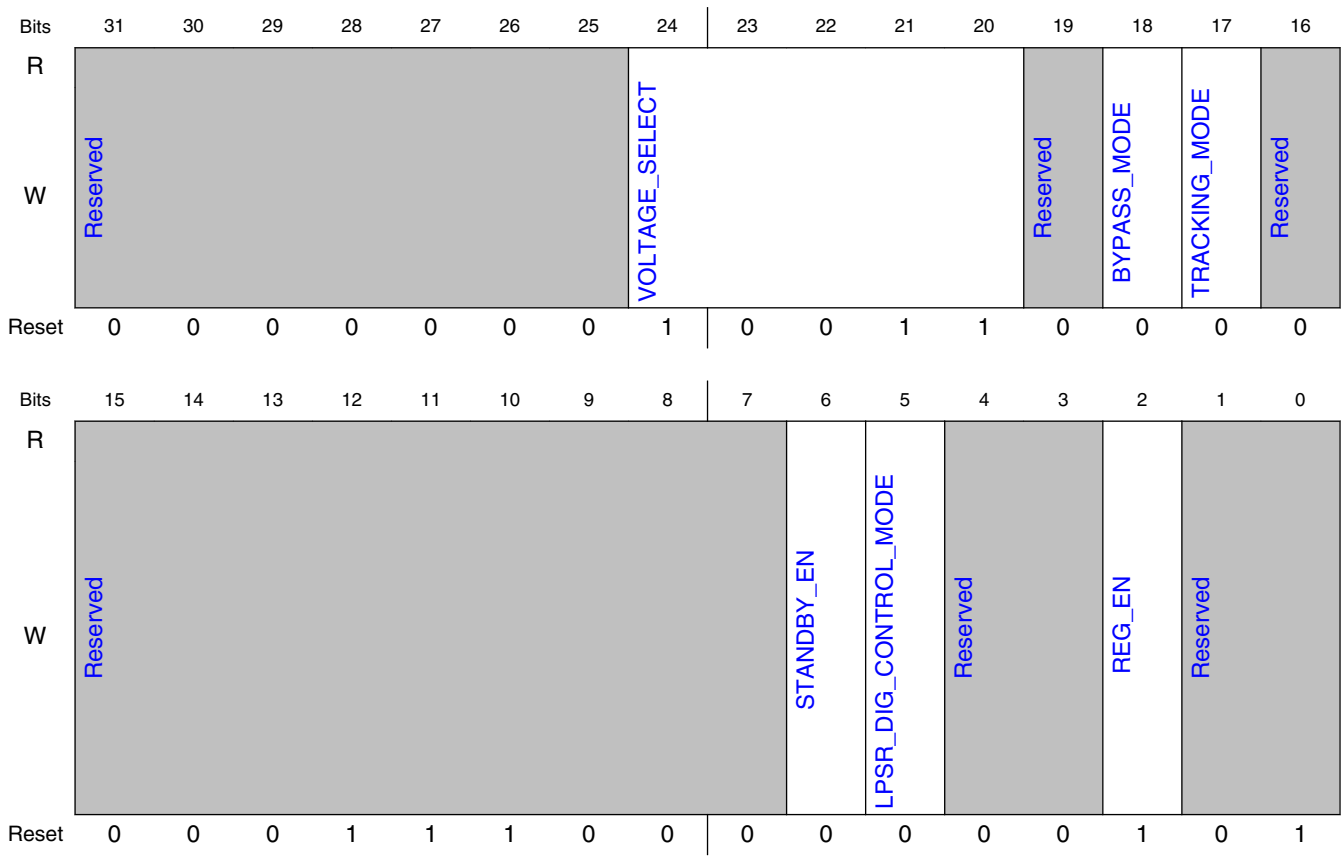
17.5.2.4 PMU_LDO_LPSR_DIG_REGISTER (PMU_LDO_LPSR_DIG)

PMU_LDO_LPSR_DIG_Regulator_Control_Register

17.5.2.4.1 Offset

Register	Offset
PMU_LDO_LPSR_DIG	530h

17.5.2.4.2 Diagram



17.5.2.4.3 Fields

Field	Description
31-25 —	Reserved Always set to zero (0).
24-20 VOLTAGE_SELECT	VOLTAGE_SELECT VOLTAGE SELECT for LPSR_DIG For lpsr_dig voltage switch to different voltage In SW mode, software is responsible to take care of the stepping time. In HW mode, hardware is responsible to take care of the stepping time.

Table continues on the next page...

Field	Description
	During the mode transistion, in case the lpsr_dig voltage is out of table, software is responsible to take care of the stepping time (i.e. mode transist), software calculates the steps difference multiply the stepping requirement, after the voltage is stable. GPC mode transistion is allowed. The values below are the stable voltage with the range in parentheses (e.g. Stable Voltage (Range)V). 00000 - Stable Voltage (range). 0.631 (+0.039)V 00001 - Stable Voltage (range). 0.65 (+0.041)V 00010 - Stable Voltage (range). 0.67 (+0.041)V 00011 - Stable Voltage (range). 0.689 (+0.043)V 00100 - Stable Voltage (range). 0.709 (+0.044)V 00101 - Stable Voltage (range). 0.728 (+0.045)V 00110 - Stable Voltage (range). 0.748 (+0.046)V 00111 - Stable Voltage (range). 0.767 (+0.047)V 01000 - Stable Voltage (range). 0.786 (+0.049)V 01001 - Stable Voltage (range). 0.806 (+0.05)V 01010 - Stable Voltage (range). 0.825 (+0.051)V 01011 - Stable Voltage (range). 0.845 (+0.052)V 01100 - Stable Voltage (range). 0.864 (+0.054)V 01101 - Stable Voltage (range). 0.883 (+0.055)V 01110 - Stable Voltage (range). 0.903 (+0.056)V 01111 - Stable Voltage (range). 0.922 (+0.057)V 10000 - Stable Voltage (range). 0.942 (+0.058)V 10001 - Stable Voltage (range). 0.961 (+0.06)V 10010 - Stable Voltage (range). 0.981 (+0.06)V 10011 - Stable Voltage (range). 1 (+0.062)V 10100 - Stable Voltage (range). 1.019 (+0.063)V 10101 - Stable Voltage (range). 1.039 (+0.064)V 10110 - Stable Voltage (range). 1.058 (+0.066)V 10111 - Stable Voltage (range). 1.078 (+0.066)V 11000 - Stable Voltage (range). 1.097 (+0.068)V 11001 - Stable Voltage (range). 1.117 (+0.069)V 11010 - Stable Voltage (range). 1.136 (+0.07)V 11011 - Stable Voltage (range). 1.155 (+0.072)V 11100 - Stable Voltage (range). 1.175 (+0.072)V 11101 - Stable Voltage (range). 1.194 (+0.074)V 11110 - Stable Voltage (range). 1.214 (+0.075)V 11111 - Stable Voltage (range). 1.233 (+0.076)V
19	Reserved
—	Always set to zero (0).
18	bypass_mode

Table continues on the next page...

Field	Description
BYPASS_MODE	bypass_mode This mode is defined by ANALOG IP, and this bit is software mode control bit to enter this bypass mode or not.
17 TRACKING_MODE	tracking_mode tracking_mode. This mode is defined by ANALOG IP, and this bit is software mode control bit to enter this tracking mode or not.
16-7 —	Reserved Always set to zero (0).
6 STANDBY_EN	standby_en standby_en There is Standby mode defined in SOC, this bit is about whether to enter standby mode for lpsr_dig
5 LPSR_DIG_CONTROL_MODE	LPSR_DIG_CONTROL_MODE lpsr_dig_control_mode For LPSR DIG there is two mode for operation, one is software mode, the other is GPC mode. 0 - SW Control. Software control mode 1 - HW Control. Hardware / GPC control mode
4-3 —	Reserved Always set to zero (0).
2 REG_EN	ENABLE_ILIMIT Control bit to enable the current-limit circuitry in the regulator.
1-0 —	Reserved Always set to zero (0).

17.5.3 PMU_LDO_SNVS_DIG register descriptions

17.5.3.1 LDO_SNVS_DIG memory map

PMU_LDO_SNVS_DIG base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
540	PMU_LDO_SNVS_DIG_REGISTER (PMU_LDO_SNVS_DIG)	32	RW	0000_0001

17.5.3.2 PMU_LDO_SNVS_DIG_REGISTER (PMU_LDO_SNVS_DIG)

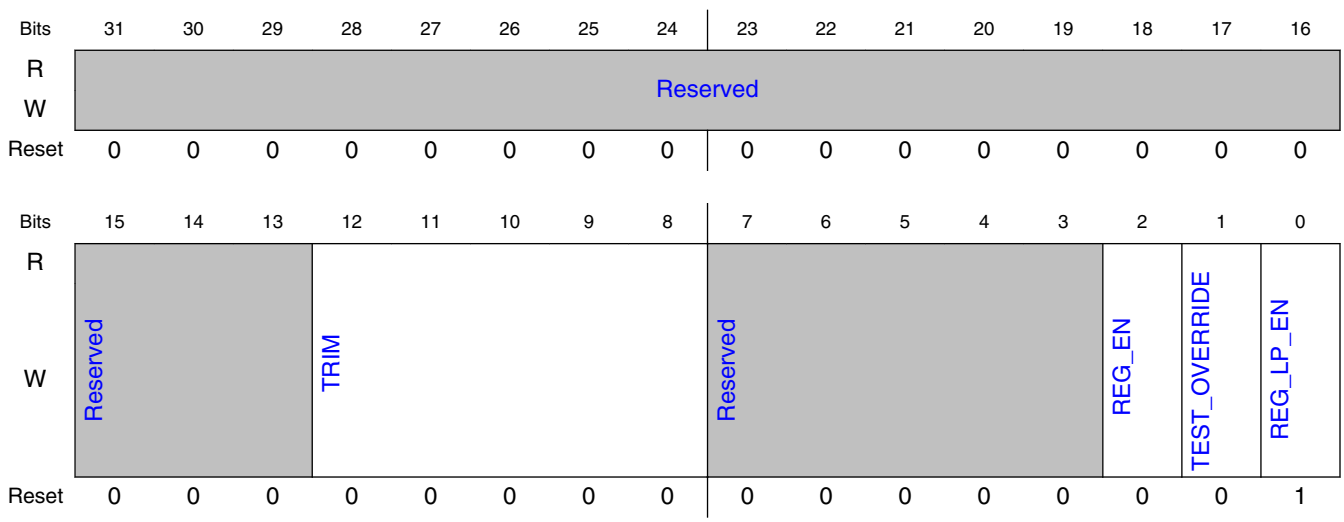
PMU_LDO_SNVS_DIG

This register defines the control and status bits for LDO_SNVS regulator. This regulator is designed to power the digital portions of the analog cells

17.5.3.2.1 Offset

Register	Offset
PMU_LDO_SNVS_DIG	540h

17.5.3.2.2 Diagram



17.5.3.2.3 Fields

Field	Description
31-13 —	Reserved Always set to zero (0).
12-8 TRIM	TRIM Trim bus for LDO_SNVS_DIG.
7-3 —	Reserved Always set to zero (0).
2 REG_EN	REG_EN Control bit to enable the LDO or not.
1 TEST_OVERRIDE	test_override test_override

Table continues on the next page...

Field	Description
0 REG_LP_EN	REG_LP_EN reg_lp_en, LDO power mode control. LP mode ; HP mode. Defined by identical LDO.

17.5.4 IPS Domain register descriptions

17.5.4.1 IPS_DOMAIN memory map

IPS Domain base address: 40C8_7C00h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Slot Control Register (SLOT0_CTRL)	32	RW	0000_000F
10	Slot Control Register (SLOT1_CTRL)	32	RW	0000_000F
20	Slot Control Register (SLOT2_CTRL)	32	RW	0000_000F
30	Slot Control Register (SLOT3_CTRL)	32	RW	0000_000F
40	Slot Control Register (SLOT4_CTRL)	32	RW	0000_000F
50	Slot Control Register (SLOT5_CTRL)	32	RW	0000_000F
60	Slot Control Register (SLOT6_CTRL)	32	RW	0000_000F
70	Slot Control Register (SLOT7_CTRL)	32	RW	0000_000F
80	Slot Control Register (SLOT8_CTRL)	32	RW	0000_000F
90	Slot Control Register (SLOT9_CTRL)	32	RW	0000_000F
A0	Slot Control Register (SLOT10_CTRL)	32	RW	0000_000F
B0	Slot Control Register (SLOT11_CTRL)	32	RW	0000_000F
C0	Slot Control Register (SLOT12_CTRL)	32	RW	0000_000F
D0	Slot Control Register (SLOT13_CTRL)	32	RW	0000_000F
E0	Slot Control Register (SLOT14_CTRL)	32	RW	0000_000F
F0	Slot Control Register (SLOT15_CTRL)	32	RW	0000_000F
100	Slot Control Register (SLOT16_CTRL)	32	RW	0000_000F
110	Slot Control Register (SLOT17_CTRL)	32	RW	0000_000F
120	Slot Control Register (SLOT18_CTRL)	32	RW	0000_000F
130	Slot Control Register (SLOT19_CTRL)	32	RW	0000_000F
140	Slot Control Register (SLOT20_CTRL)	32	RW	0000_000F
150	Slot Control Register (SLOT21_CTRL)	32	RW	0000_000F
160	Slot Control Register (SLOT22_CTRL)	32	RW	0000_000F
170	Slot Control Register (SLOT23_CTRL)	32	RW	0000_000F
180	Slot Control Register (SLOT24_CTRL)	32	RW	0000_000F

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
190	Slot Control Register (SLOT25_CTRL)	32	RW	0000_000F
1A0	Slot Control Register (SLOT26_CTRL)	32	RW	0000_000F
1B0	Slot Control Register (SLOT27_CTRL)	32	RW	0000_000F
1C0	Slot Control Register (SLOT28_CTRL)	32	RW	0000_000F
1D0	Slot Control Register (SLOT29_CTRL)	32	RW	0000_000F
1E0	Slot Control Register (SLOT30_CTRL)	32	RW	0000_000F
1F0	Slot Control Register (SLOT31_CTRL)	32	RW	0000_000F
200	Slot Control Register (SLOT32_CTRL)	32	RW	0000_000F
210	Slot Control Register (SLOT33_CTRL)	32	RW	0000_000F
220	Slot Control Register (SLOT34_CTRL)	32	RW	0000_000F
230	Slot Control Register (SLOT35_CTRL)	32	RW	0000_000F
240	Slot Control Register (SLOT36_CTRL)	32	RW	0000_000F
250	Slot Control Register (SLOT37_CTRL)	32	RW	0000_000F

17.5.4.2 Slot Control Register (SLOT0_CTRL - SLOT37_CTRL)

Table 17-3. Slot Assignments

Slot Number	Assignment
0	ANATOP_CTRL
1	48M
2	24M
3	400M
4	16M
5	OSC TRIM
6	PLL_ARM
7	SYS_PLL3 (480 PLL)
8	SYS_PLL2 (528 PLL)
9	SYS_PLL1 (1G PLL)
10	PLL_AUDIO
11	PLL_VIDEO
12	TEMPSENSOR
13	ANAMUX
14	PWRDET
15	TEMPSENSOR_OTP
16	PWR_OTP
17	PLL_LDO

Table continues on the next page...

Table 17-3. Slot Assignments (continued)

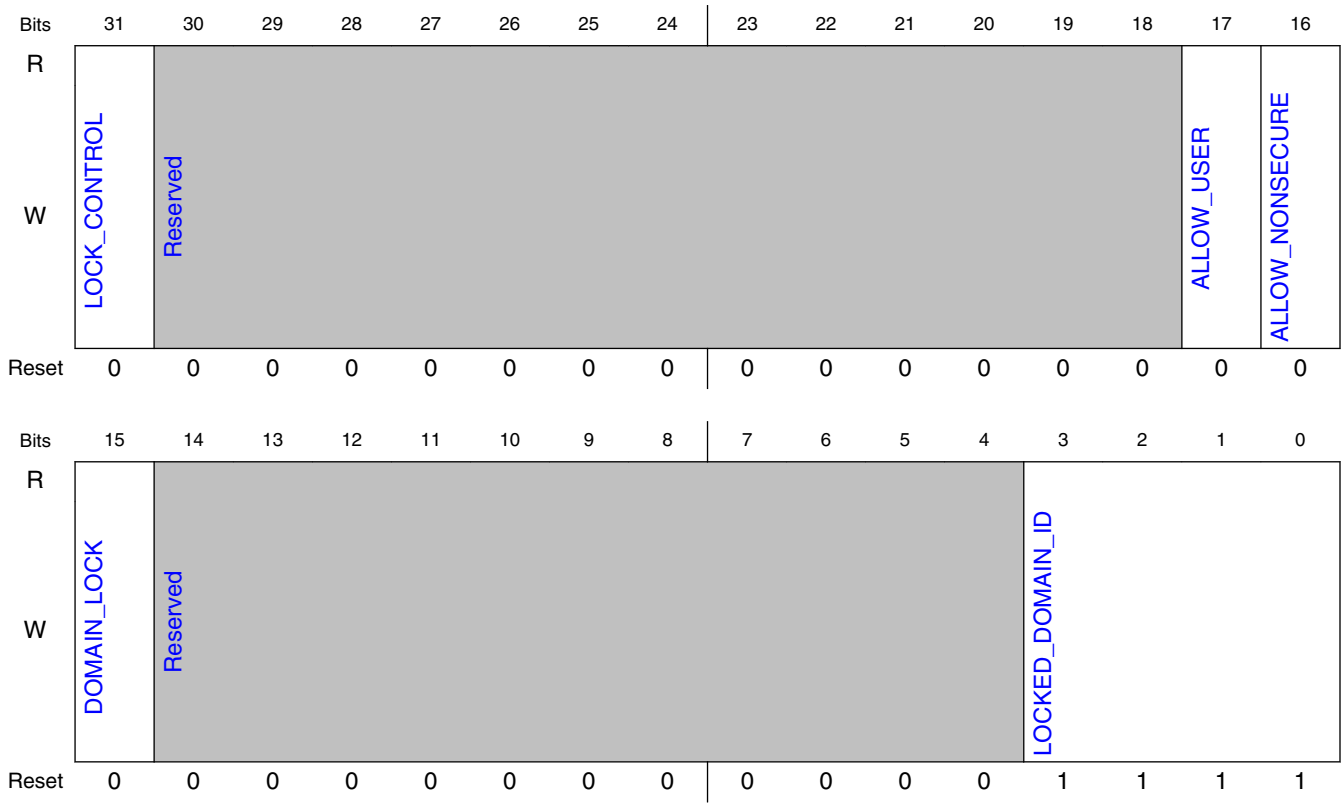
Slot Number	Assignment
18	LPSR_ANA
19	LPSR_DIG
20	SNVS_DIG
21	BIAS
22	BANDGAP
23	CKGB
24	SET_POINT
25	BANDGAP_OTP
26	WB_OTP
27	LPSR_ANA_OTP
28	MISC
29	LVDS
30	AI_SOC
31	AI_1G
32	AI_AUDIO
33	AI_VIDEO
34	AI_LPSR
35	AI_TMPSNS
36	AI_400M
37	AI_400M

17.5.4.2.1 Offset

For n = 0 to 37:

Register	Offset
SLOTn_CTRL	0h + (n × 10h)

17.5.4.2.2 Diagram



17.5.4.2.3 Fields

Field	Description
31 LOCK_CONTROL	Lock control of this slot 0 - Do not lock the control register of this slot 1 - Lock the control register of this slot
30-18 —	Reserved
17 ALLOW_USER	Allow user write access to this domain control register or domain register 0 - Do not allow user write access 1 - Allow user write access
16 ALLOW_NONSECURE	Allow non-secure write access to this domain control register or domain register 0 - Do not allow non-secure write access 1 - Allow non-secure write access
15 DOMAIN_LOCK	Lock domain ID of this slot 0 - Do not lock the domain ID 1 - Lock the domain ID
14-4	Reserved

Table continues on the next page...

Field	Description
—	
3-0 LOCKED_DOMAIN_ID	Domain ID of the slot to be locked Domain id3~0 maps to bit3~0. It indicates whether related domain can write to domain register or not. 1'b1 indicates domain write access is allowed. while domain id is locked, only related domain can write to bit31 and bit17~16

Chapter 18
PHY LDO (PHY_LDO)

18.1 Chip-specific PHY LDO information

Table 18-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

PHY_LDO is also LDO_PLL.

18.2 Overview

The PHY LDO creates a regulated 1.0V output voltage for digital PHYs from an external 1.8V source. The PHY LDO is controlled through the AI interface.

18.2.1 Features

The PHY LDO has the following featurest:

- Analog 200 mA linear regulator
- 1.8V +/- 10% VDD external input
- 1.0V output

- Programmable adjustment in 0.25mV steps (0.6V to 1.375V)
- AI Interface

18.3 Memory Map and register definition

This section includes the PHY LDO module memory map and detailed descriptions of all registers.

NOTE

For related IPS Domain Slot control, please see IPS Domain Registers in Power Management Unit (PMU) chapter.

18.3.1 register descriptions

18.3.1.1 PHY LDO Memory Map

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Analog Control Register CTRL0 (CTRL0)	32	RW	0000_0000
4	Analog Control Register CTRL0 (CTRL0_SET)	32	RW	0000_0000
8	Analog Control Register CTRL0 (CTRL0_CLR)	32	RW	0000_0000
C	Analog Control Register CTRL0 (CTRL0_TOG)	32	RW	0000_0000
50	Analog Status Register STAT0 (STAT0)	32	RO	0000_0000
54	Analog Status Register STAT0 (STAT0_SET)	32	RO	0000_0000
58	Analog Status Register STAT0 (STAT0_CLR)	32	RO	0000_0000
5C	Analog Status Register STAT0 (STAT0_TOG)	32	RO	0000_0000

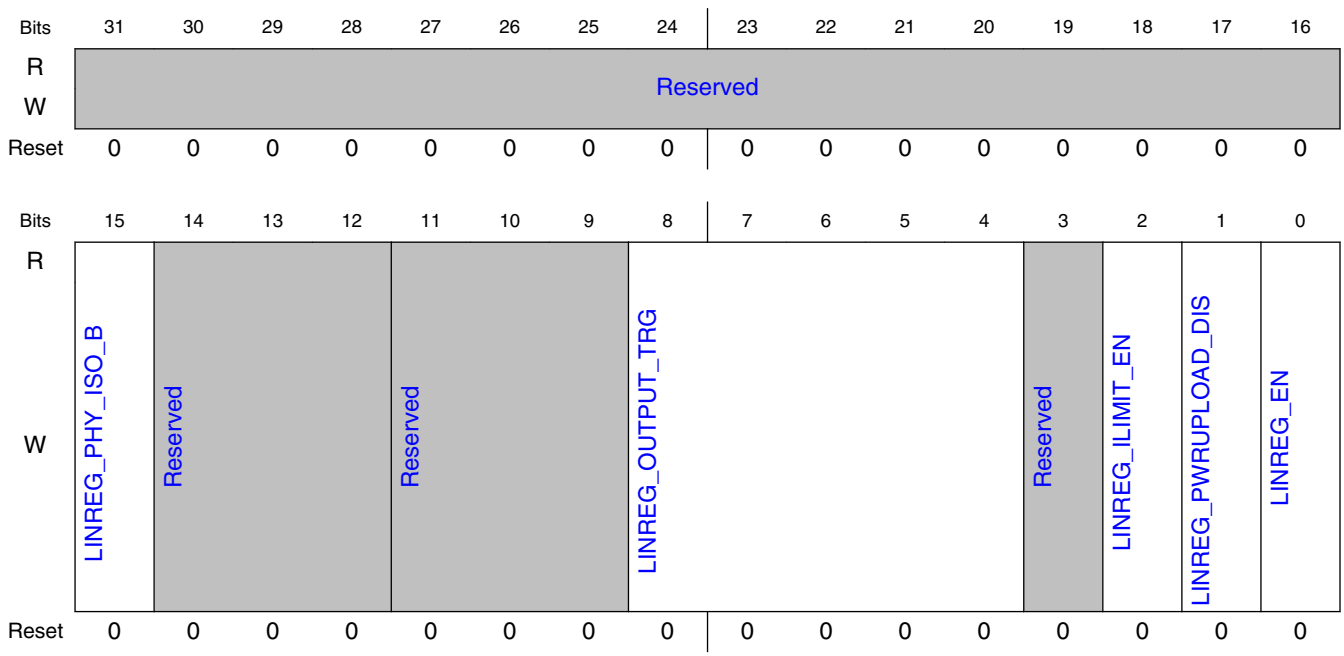
18.3.1.2 Analog Control Register CTRL0 (CTRL0)

This register contains analog control bits.

18.3.1.2.1 Offset

Register	Offset	Description
CTRL0	0h	Analog Control Register CTRL0
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

18.3.1.2.2 Diagram



18.3.1.2.3 Fields

Field	Description
31-16	Reserved
—	This read-only field is reserved and always has the value 0.
15	Isolation control for attached PHY load
LINREG_PHY_ISO_B	This control bit is to be used by the system controller to isolate the attached PHY load when the LinReg is powered down. During a power-up event of the regulator it is expected that this control signal is set high at least 100us after the main regulator is enabled. During a power-down event of the regulator it is expected that this control signal is set low before the main regulator is disabled/power-down.
14-12	Reserved

Table continues on the next page...

Field	Description
—	This read-only field is reserved and always has the value 0.
11-9	Reserved
—	This read-only field is reserved and always has the value 0.
8-4	LinReg output voltage target setting
LINREG_OUTPUT_TRG	LinReg output voltage target setting. The nominal voltage step per code is 25mV. Setting the output voltage beyond the technology reliability limit is not recommended. 00000 - Set output voltage to x.xV 10000 - Sets output voltage to 1.0V 11111 - Set output voltage to x.xV
3	Reserved
—	This read-only field is reserved and always has the value 0.
2	LinReg current-limit enable
LINREG_ILIMIT_EN	LinReg current-limit enable. Setting this bit will enable the current-limiter in the regulator.
1	LinReg power-up load disable
LINREG_PWRUPLOAD_DIS	LinReg power-up load disable control bit. 0 - Internal pull-down enabled 1 - Internal pull-down disabled
0	LinrReg master enable
LINREG_EN	LinReg master enable. Setting this bit will enable the regulator.

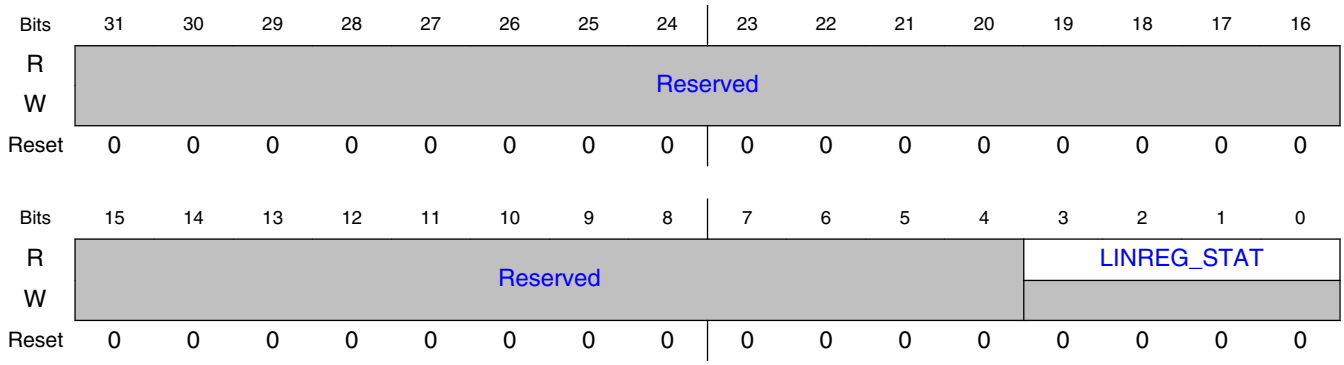
18.3.1.3 Analog Status Register STAT0 (STAT0)

This register contains analog status bits.

18.3.1.3.1 Offset

Register	Offset	Description
STAT0	50h	Analog Status Register STAT0
STAT0_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in STAT0
STAT0_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in STAT0
STAT0_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in STAT0

18.3.1.3.2 Diagram



18.3.1.3.3 Fields

Field	Description
31-4 —	Reserved This read-only field is reserved and always has the value 0.
3-0 LINREG_STAT	LinReg Status Bits LinReg status bits

Chapter 19
General Power Controller (GPC)

19.1 Chip-specific General Power Controller (GPC) information

Table 19-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

19.2 Overview

The General Power Controller (GPC) is the centralized power controller, which controls the power mode of the processor(s). The GPC takes the Wait For Interrupt (WFI) signal from CPU platforms, and wakeup events from peripherals, to determine the power mode based on the GPC power management policy. The GPC can also control the power mode transition.

19.2.1 Block diagram

The GPC block consists of the following sub-modules:

- CPU Mode Control (CMC): It contains CPU mode controllers, one for each CPU platform. They control CPU mode of CPU platforms and their private resources.
- UPI CM Mapping (UPI): The connection between CMC and CPU platform is directly hard-wired, but CPU platform can be assigned to any domain in the system. So this block is used to map CPU mode control signals into the correct domain.
- Setpoint Control (SPC): This controls Setpoint status and transition of system resources.
- Standby Control (SBC): It indicates which resource controllers go in and out from standby mode

The figure below shows the block diagram of the General Power Controller.

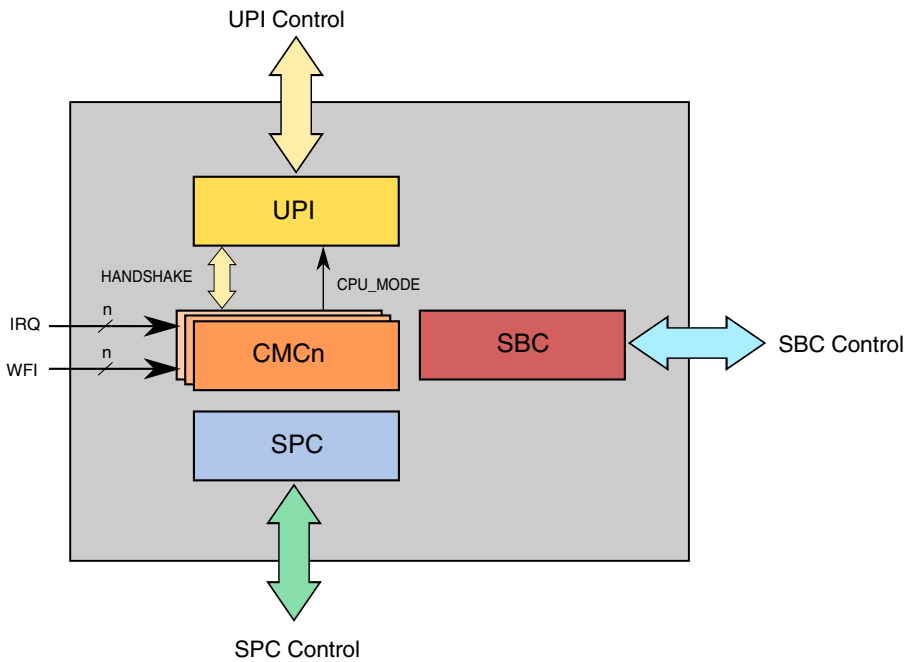


Figure 19-1. GPC block diagram

19.2.2 Features

The GPC includes the following features:

- Standby management
- Domain access control
- Controllable steps of sleep and wakeup sequence

19.3 Functional description

The following sections describe functional details of the GPC module.

19.3.1 CPU Mode Control (CMC)

The CPU mode is a power mode of the CPU platform. The following are the four CPU modes:

Table 19-2. CPU Modes

CPU Mode	Description
RUN	The CPU core is active and running under normal operation. All the blocks inside the CPU platform can be accessed when needed. The state of private resources are fully controlled by software configuration.
WAIT	The CPU is in WFI/WFE state, but can to get back to RUN mode with very short latency. In typical applications, the CPU will enter WAIT mode whenever there is no active thread running. In WAIT mode, the clock to the CPU core is gated off, the cache is clock gated, and the TCM is still active since there are other modules, such as DMA, that still needs access to it.
STOP	The CPU is in WFI/WFE state, and does not require an extremely short exit time. In STOP mode, the clocks to the CPU core, Cache, and TCM are all gated off. The clocks to the bus and peripherals are also gated off. This is the lowest power consumption mode without losing the state of the peripheral. When exiting from STOP mode, there is no need to re-initialize the peripherals.
SUSPEND	Entering SUSPEND mode is for lowest power consumption, and exit time is not critical. In SUSPEND mode, CPU, Cache, and peripherals are all power gated. The biggest difference between SUSPEND and STOP mode is the peripherals will be power gated. Because the peripherals are power gated, entering SUSPEND mode requires the CPU to save the state of the peripherals. When exiting from SUSPEND mode, the CPU needs to restore these states.

CPU Mode transitions happen when the following events occur:

- Sleep Event: CPU enters sleep state with WFI/WFE instruction
- Wakeup Event: Any unmasked IRQ wakeup

On a Sleep Event, the CPU has the option to enter WAIT/STOP/SUSPEND mode, or stay in RUN mode. The transition to WAIT/STOP/SUSPEND is called the sleep sequence, and the next state is configured by CM_MODE_CTRL[CPU_MODE_TARGET].

When CPU is in WAIT/STOP/SUSPEND mode, a wakeup request can wake the CPU, and bring it back to RUN mode. This is transition is called the wakeup sequence. The CM_IRQ_WAKEUP_MASK_n register selects which IRQs can wakeup the CPU platform. There are also non-IRQ requests, which can wakeup the CPU. These non-IRQ requests are configured by the CM_NON_IRQ_WAKEUP_MASK register.

During a CPU mode transition, the GPC communicates with the system resource controllers by sending the target CPU mode and transition step. Please see the UPI interface for more detials.

If a resource is set as private to a single CPU platform, the resource's power state will change with the CPU platform's mode. If a resource is set as shared by multiple CPU platforms, the resource controller needs to look at all the current modes of the CPU platforms to determine the power state of the resource.

19.3.2 Setpoint Control (SPC)

The GPC supports 16 Setpoints. Setpoints are implemented to configure the power state of Public Resources. Public Resources are system-level resources that aren't owned or controlled by any of the CPU platforms.

During Setpoint transition, GPC communicates with system resource controllers through the Unified Power Management Interface (UPI). Setpoint transition is triggered by a request from any of the CPU platforms. The triggers include the following events:

- CPU Sleep Event
  - A sleep event triggers a CPU mode transition, and CPU platform also requests a Setpoint transition after CPU mode transition completes. The field CM_SP_CTRL[CPU_SP_SLEEP_EN] controls whether Setpoint transition will be triggered, and the field CM_SP_CTRL[CPU_SP_SLEEP] controls which Setpoint the CMC want to transfer to.
- CPU Wakeup Event
  - An unmasked wakeup event triggers a CPU mode transition, and CPU platform requests for a Setpoint transition before CPU mode transition starts. The field CM_SP_CTRL[CPU_SP_WAKEUP_EN] controls whether Setpoint transition will be triggered, and the fields CM_SP_CTRL[CPU_SP_WAKEUP] and CM_SP_CTRL[CPU_SP_WAKEUP_SEL] control which set point the CMC want to transfer to.
- Software
  - CPU platform is in RUN mode and the SW running on the CPU core makes a write operation to CMC register CPU_SP_RUN_EN to request for a Setpoint transition. Register CPU_SP_RUN controls which Setpoint the CMC want to transfer to.

With CPU Mode and Setpoint defined properly, the Power Mode of a system is defined as a combination of the CPU Mode and Setpoint.

19.3.3 System Setpoint management

The SPC receives a Setpoint change request from CMCs, it then determines the Setpoint of the system, and manages the transition sequence. When a CMC Setpoint request is accepted, the Setpoint value can then be read from SPC SP_ACCEPTED_CPUn register.

If multiple CPU platforms want to enter different Setpoints, the SPC needs to calculate the final target Setpoint. There is a mapping table between each Setpoint of a CPU platform and the Setpoint, which it can accept. For example, CPU1 requests the system to transit to Setpoint 7 after it transits into STOP mode. CPU1 must provide a Setpoint list to show which Setpoints are also allowed besides SP-7. This is configured by the CMC CM_SPx_MAPPING Register. Below are examples of Setpoint list configurations:

- If CPU1_SP7_MAPPING register is set to 16'b00000000_10011111, CPU1 requests system to transition to SP-7 (bit 7), but also allows transitions to SP-0,1,2,3,4 (bits 0 to 4).
- If CPU0_SP4_MAPPING register is set to 16'b00000000_00010000, CPU0 has requested the system to transit to SP-4 after STOP.

In the SPC, the SP_SYS_STAT[SYS_SP_ALLOWED] field will be the ANDed result of allowed Setpoints from all CMCs. If all bits in SP_SYS_STAT[SYS_SP_ALLOWED] is 0, it means there are no Setpoints allowed by both platforms. This would be an invalid state, and there should be no Setpoint transition in this condition. GPC should generate an error IRQ. Below is an example of allowed Setpoint configuration:

- If SP_SYS_STAT[SYS_SP_ALLOWED] is 16'b00000000_00010000, it means only SP-4 is allowed. The system should transit to this Setpoint.

The calculated Setpoint number can be read at SPC CM_SP_STAT[CPU_SP_TARGET] register.

If more than one bit in SP_SYS_STAT[SYS_SP_ALLOWED] is set to 1, the Setpoint will be picked based on the following rules:

- If the allowed Setpoints match only one SP_CPU_REQ[SP_ACCEPTED_CPUn], this Setpoint will be used as SPC CM_SP_STAT[CPU_SP_TARGET]
- If the allowed Setpoints match more than one SP_CPU_REQ[SP_ACCEPTED_CPUn], the matched Setpoint with highest priority will be used as SPC CM_SP_STAT[CPU_SP_TARGET]
- If the allowed Setpoints do not match any of the SP_CPU_REQ[SP_ACCEPTED_CPUn], the CM_SP_STAT[CPU_SP_TARGET] will be the allowed Setpoint with highest priority

NOTE

The priority settings are set in the SPC SP_PRIORITY_x[SYS_SPn_PRIORITY] register configuration.

19.3.4 Standby Control (SBC)

Standby mode is a low-power mode that has distinguishing settings outside of CPU mode and Setpoint mode. Standby mode is related to state of all CPU platforms and has a much shorter transition time than Setpoint.

The CMC can send a standby request to SBC. The fields CM_STBY_CTRL[STBY_WAIT], CM_STBY_CTRL[STBY_STOP], and CM_STBY_CTRL[STBY_SUSPEND] will determine if the chip enters into Standby mode when the CPU enters WAIT mode, STOP mode, or SUSPEND mode respectively.

The SBC maintains the system standby status, and only when all CPU platforms send standby request, can the system enter into Standby mode. If a CPU is disabled by FUSE, GPC considers it in Standby mode by default.

19.3.5 Unified power management interface (UPI)

GPC does not control the state of the resource directly. During power mode transition, it communicates with the resource controllers through the UPI (Unified Power management Interface) to accomplish the power mode transition.

Handshake Method:

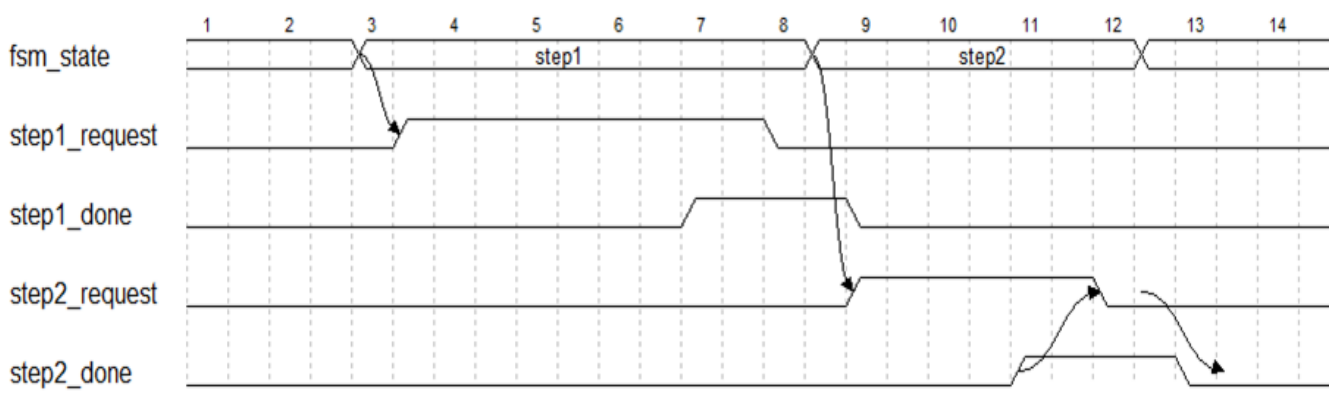


Figure 19-2. Handshake

19.3.6 Low Power Sequence

19.3.6.1 CPU mode transition flow

Each CMC has a state machine to control the sequence of CPU mode transition.

When CMC receives a sleep request from CPU platform, it starts the sleep sequence, it cannot be broken by wakeup request until finishing SLEEP_POWER request. In SLEEP_SP step, the SPC runs set point transition flow. In SLEEP_STBY step, the SBC runs standby transition flow. After the sleep sequence finishes, the CMC stays at IDLE_SLEEP and waits for a wakeup event to start wakeup sequence, which is showed in the left part of the diagram.

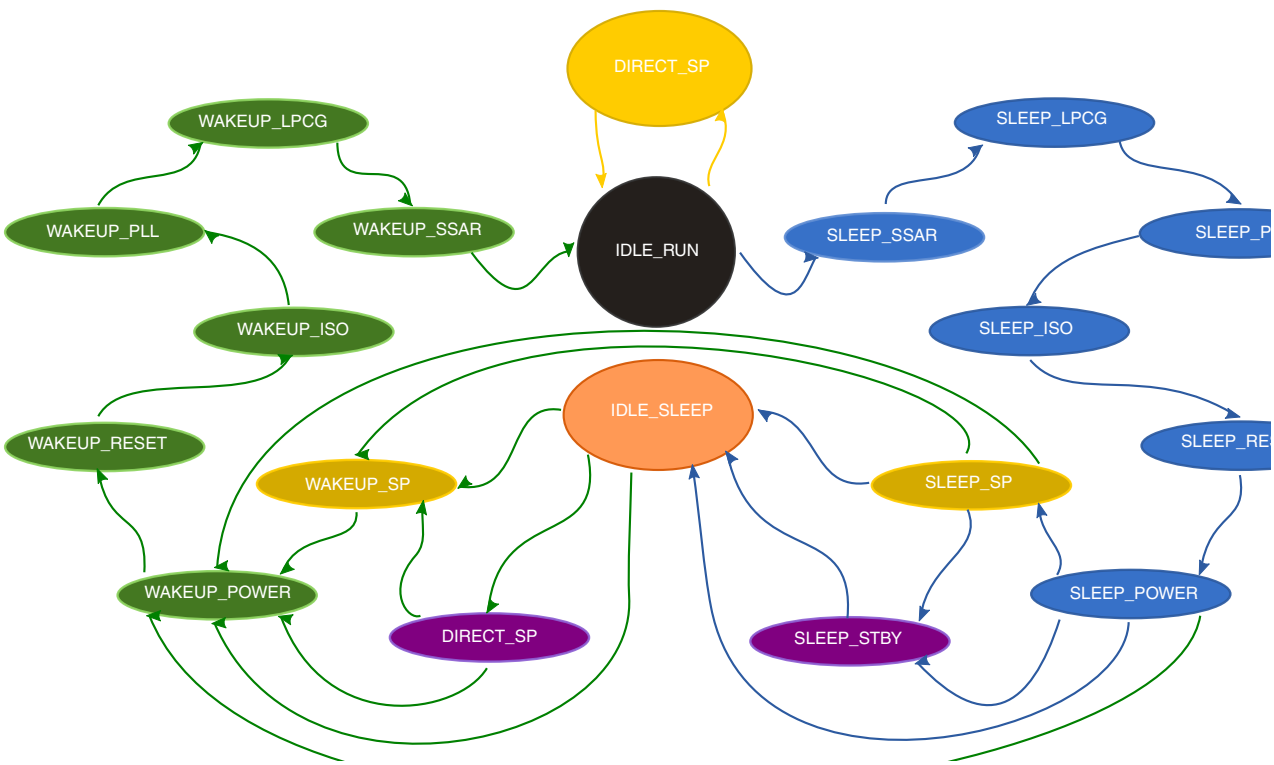


Figure 19-3. CPU Mode Transition Flow

19.3.6.2 Setpoint transition flow

The SPC has a state machine to control the sequence of Setpoint transition.

The SPC state machine starts after it receives a Setpoint change request from CMC. SPC uses one state to calculate whether or not it needs to change Setpoint based on all CMC status. If the calculated result is YES, the Setpoint transition sequence starts. This sequence cannot be broken by a wakeup request.

The Setpoint transition sequence contains both a sleep sequence and a wakeup sequence, because GPC doesn't know whether a Setpoint transition is a wakeup or sleep sequence. GPC sends all step requests by sequence, and the system resource controller determines which request need be processed.

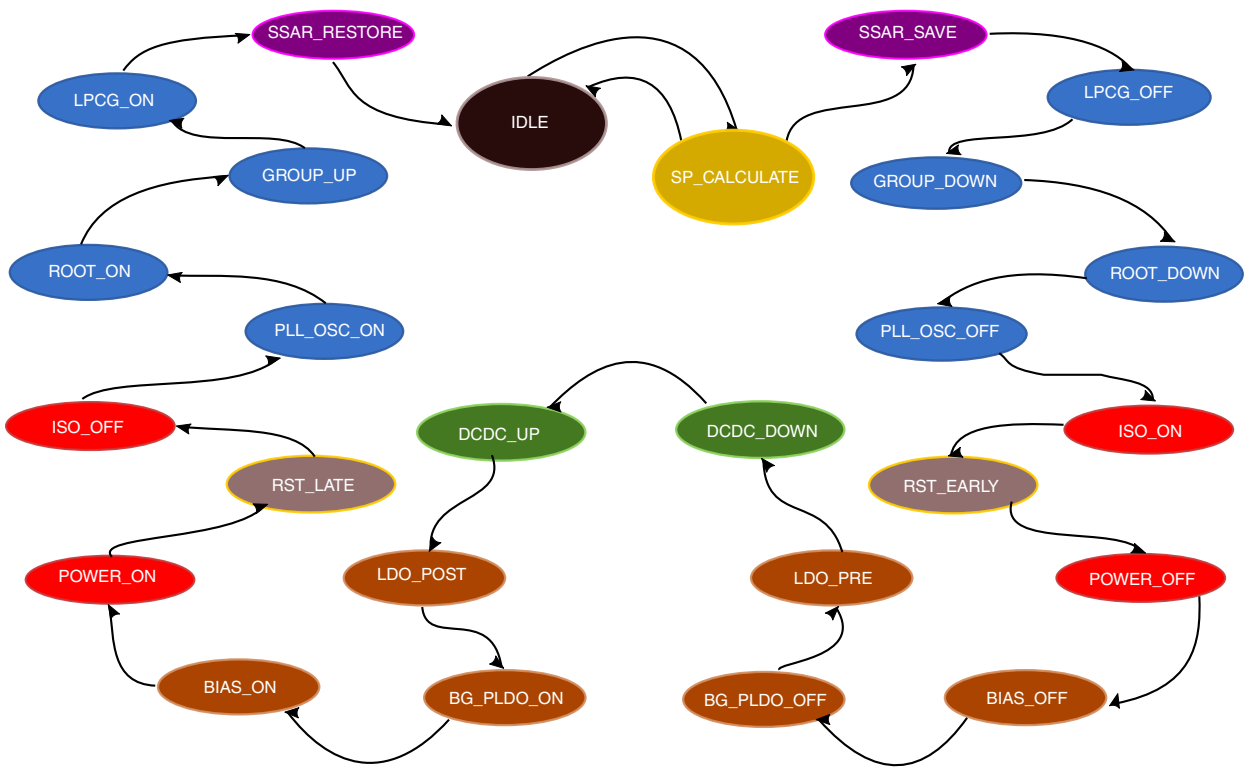


Figure 19-4. Setpoint Transition Flow

19.3.6.3 Standby transition flow

The SBC state machine starts when CMC sends a standby request, it looks at all CMC standby status, if any CPU platform is not in sleep mode or not allow standby, this standby request will be refused. The standby_in sequence can be broken after any step complete because of wakeup time requirement.

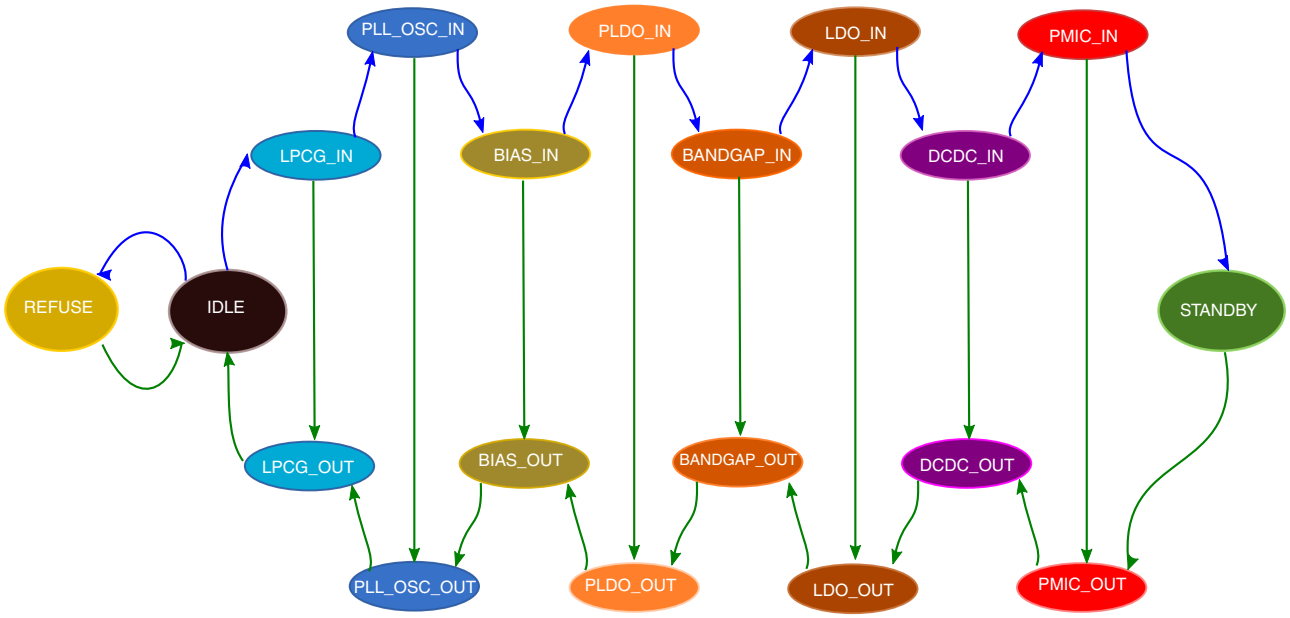


Figure 19-5. Standby transition flow

19.3.6.4 Step Control

For each step of transition, there is a control register to adjust the step duration.

The STEP_CNT[DISABLE] field to used to disable the step so that no request is sent out in this transition step. There is also a read only counter to record the time when step response is received.

19.3.7 System Domain

19.3.7.1 Domain assignment

Each CPU platform and the CMC connected to it can be assigned to the same domain.

Based on the domain assignment, a mapping mechanism between CPU platform and system domain is implemented inside GPC.

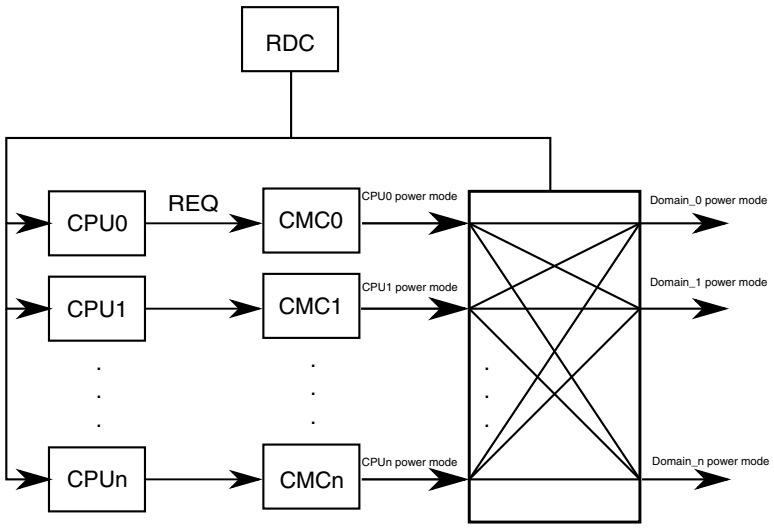


Figure 19-6. Domain Assignement

When more than one CMC is assigned to the same domain, a two stage priority selects which CMC can control the domain's UPI output. The first stage checks for the master_cpu flag from CMC. The master_cpu flag is enabled by register CM_MISC[MASTER_CPU]. The CMC with the master_cpu flag owns the Domain UPI output. If all or none of the CPUs own the master_cpu flag, the second stage priority assigns Domain UPI output to the lower indexed CMC.

19.3.7.2 Register Access Permissions

The GPC register access is controlled by the following registers and bitfields:

- CM_AUTHEN_CTRL register for CMC access permission
- SP_AUTHEN_CTRL register for SPC and SBC access permission.
- CM_AUTHEN_CTRL[WHITE_LIST] field configures which Domain IDs can write to the registers.
- x_AUTHEN_CTRL[USER] and x_AUTHEN_CTRL[NONSECURE] fields are used to set the privilege and security permission.
- CM_AUTHEN_CTRL[LOCK_CFG] locks configuration fields and should only be set at initialization.

19.3.8 Clocks

This section describes the clocks for the GPC module.

Table 19-3. Clocks

Clock	Description
ipg_clk	Main GPC clock. This clock originates from RCOSC and can be turned off by setting SP_ROSC_CTRL[SP_ALLOW_ROSC_OFF] when the system is in a Setpoint, which allows shutting of the RCOSC. After transitioning into Standby mode, the GPC will shut off the RCOSC.
32k_clk	32 kHz clock used to wakeup GPC and RCOSC. The 32 kHz clock isn't needed by ipg_clk since they are fully asynchronous

19.3.9 Interrupts

The interrupt inputs of CMC0 and CMC1 are independent and the same as the CPU platforms they are connected to. Each CMC has up to 256 IRQ input to wake up the core or system.

The following are the types of interrupts the GPC can generate:

- CM_INT_CTRL[SP_REQ_NOT_ALLOWED_SLEEP_INT]: Setpoint target is not allowed by MODE_MAPPING
- CM_INT_CTRL[SP_REQ_NOT_ALLOWED_WAKEUP_INT]: The wakeup Setpoint target is not allowed by RUN_MODE_MAPPING
- CM_INT_CTRL[SP_REQ_NOT_ALLOWED_SOFT_INT]: The software Setpoint target is not allowed by RUN_MODE_MAPPING
- SP_INT_CTRL[NO_ALLOWED_SP_INT]: No allowed Setpoint target for all CMCs

NOTE

All the interrupts share one pin, users need to check above registers to find out which violation occurs.

19.4 External Signals

There are no external signals pinned out for the GPC.

19.5 Initialization

See Low Power Sequence for low power entry and exit flow and considerations.

19.5.1 CPU Sleep Hold

There is a certain delay between the time the core falls asleep and when the core clock stops. If a wakeup IRQ appears inside this delay window, the core will process the IRQ and awake, but the GPC sleep sequence is still shutting off the core clock. Since the sleep sequence is not interruptable, this scenario may lead to unpredictable behavior.

To avoid this scenario, a CPU Sleep Hold is implemented to hold the core in the sleep state until the sleep sequence is finished. The GPC will then send a signal to release the core. The CM_MISC[SLEEP_HOLD_EN] field is used to enable this function.

The core clock can also be gated by a WFI event without any GPC sequence. The GPC uses a signal to clear this gate when a wakeup trigger occurs.

19.6 Memory Map and register definition

This section includes the GPC module memory map and detailed descriptions of all registers.

19.6.1 GPC_CPU register descriptions

19.6.1.1 gpc_cpu_ctrl memory map

GPC_CPU_MODE_CTRL_0 base address: 40C0_0000h

GPC_CPU_MODE_CTRL_1 base address: 40C0_0800h

Offset	Register	Width (In bits)	Access	Reset value
4h	CM Authentication Control (CM_AUTHEN_CTRL)	32	RW	0000_0F00h
8h	CM Interrupt Control (CM_INT_CTRL)	32	RW	0000_0007h
Ch	Miscellaneous (CM_MISC)	32	RW	0000_0006h
10h	CPU mode control (CM_MODE_CTRL)	32	RW	0000_0000h
14h	CM CPU mode Status (CM_MODE_STAT)	32	R	0000_0000h
100h	CM IRQ0~31 wakeup mask (CM_IRQ_WAKEUP_MASK_0)	32	RW	0000_0000h
104h	CM IRQ32~63 wakeup mask (CM_IRQ_WAKEUP_MASK_1)	32	RW	0000_0000h
108h	CM IRQ64~95 wakeup mask (CM_IRQ_WAKEUP_MASK_2)	32	RW	0000_0000h
10Ch	CM IRQ96~127 wakeup mask (CM_IRQ_WAKEUP_MASK_3)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
110h	CM IRQ128~159 wakeup mask (CM_IRQ_WAKEUP_MASK_4)	32	RW	0000_0000h
114h	CM IRQ160~191 wakeup mask (CM_IRQ_WAKEUP_MASK_5)	32	RW	0000_0000h
118h	CM IRQ192~223 wakeup mask (CM_IRQ_WAKEUP_MASK_6)	32	RW	0000_0000h
11Ch	CM IRQ224~255 wakeup mask (CM_IRQ_WAKEUP_MASK_7)	32	RW	0000_0000h
140h	CM non-irq wakeup mask (CM_NON_IRQ_WAKEUP_MASK)	32	RW	0000_0001h
150h	CM IRQ0~31 wakeup status (CM_IRQ_WAKEUP_STAT_0)	32	R	0000_0000h
154h	CM IRQ32~63 wakeup status (CM_IRQ_WAKEUP_STAT_1)	32	R	0000_0000h
158h	CM IRQ64~95 wakeup status (CM_IRQ_WAKEUP_STAT_2)	32	R	0000_0000h
15Ch	CM IRQ96~127 wakeup status (CM_IRQ_WAKEUP_STAT_3)	32	R	0000_0000h
160h	CM IRQ128~159 wakeup status (CM_IRQ_WAKEUP_STAT_4)	32	R	0000_0000h
164h	CM IRQ160~191 wakeup status (CM_IRQ_WAKEUP_STAT_5)	32	R	0000_0000h
168h	CM IRQ192~223 wakeup status (CM_IRQ_WAKEUP_STAT_6)	32	R	0000_0000h
16Ch	CM IRQ224~255 wakeup status (CM_IRQ_WAKEUP_STAT_7)	32	R	0000_0000h
190h	CM non-irq wakeup status (CM_NON_IRQ_WAKEUP_STAT)	32	R	0000_0000h
200h	CM sleep SSAR control (CM_SLEEP_SSAR_CTRL)	32	RW	0000_0000h
208h	CM sleep LPCG control (CM_SLEEP_LPCG_CTRL)	32	RW	0000_0000h
210h	CM sleep PLL control (CM_SLEEP_PLL_CTRL)	32	RW	0000_0000h
218h	CM sleep isolation control (CM_SLEEP_ISO_CTRL)	32	RW	0000_0000h
220h	CM sleep reset control (CM_SLEEP_RESET_CTRL)	32	RW	0000_0000h
228h	CM sleep power control (CM_SLEEP_POWER_CTRL)	32	RW	0000_0000h
290h	CM wakeup power control (CM_WAKEUP_POWER_CTRL)	32	RW	0000_0000h
298h	CM wakeup reset control (CM_WAKEUP_RESET_CTRL)	32	RW	0000_0000h
2A0h	CM wakeup isolation control (CM_WAKEUP_ISO_CTRL)	32	RW	0000_0000h
2A8h	CM wakeup PLL control (CM_WAKEUP_PLL_CTRL)	32	RW	0000_0000h
2B0h	CM wakeup LPCG control (CM_WAKEUP_LPCG_CTRL)	32	RW	0000_0000h
2B8h	CM wakeup SSAR control (CM_WAKEUP_SSAR_CTRL)	32	RW	0000_0000h
300h	CM Setpoint Control (CM_SP_CTRL)	32	RW	0000_0000h
304h	CM Setpoint Status (CM_SP_STAT)	32	R	0000_0000h
310h	CM Run Mode Setpoint Allowed (CM_RUN_MODE_MAPPING)	32	RW	0000_FFFFh
314h	CM Wait Mode Setpoint Allowed (CM_WAIT_MODE_MAPPING)	32	RW	0000_FFFFh
318h	CM Stop Mode Setpoint Allowed (CM_STOP_MODE_MAPPING)	32	RW	0000_FFFFh
31Ch	CM Suspend Mode Setpoint Allowed (CM_SUSPEND_MODE_MAPPING)	32	RW	0000_FFFFh
320h	CM Setpoint 0 Mapping (CM_SP0_MAPPING)	32	RW	0000_FFFFh
324h	CM Setpoint 1 Mapping (CM_SP1_MAPPING)	32	RW	0000_FFFFh
328h	CM Setpoint 2 Mapping (CM_SP2_MAPPING)	32	RW	0000_FFFFh
32Ch	CM Setpoint 3 Mapping (CM_SP3_MAPPING)	32	RW	0000_FFFFh
330h	CM Setpoint 4 Mapping (CM_SP4_MAPPING)	32	RW	0000_FFFFh
334h	CM Setpoint 5 Mapping (CM_SP5_MAPPING)	32	RW	0000_FFFFh
338h	CM Setpoint 6 Mapping (CM_SP6_MAPPING)	32	RW	0000_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
33Ch	CM Setpoint 7 Mapping (CM_SP7_MAPPING)	32	RW	0000_FFFFh
340h	CM Setpoint 8 Mapping (CM_SP8_MAPPING)	32	RW	0000_FFFFh
344h	CM Setpoint 9 Mapping (CM_SP9_MAPPING)	32	RW	0000_FFFFh
348h	CM Setpoint 10 Mapping (CM_SP10_MAPPING)	32	RW	0000_FFFFh
34Ch	CM Setpoint 11 Mapping (CM_SP11_MAPPING)	32	RW	0000_FFFFh
350h	CM Setpoint 12 Mapping (CM_SP12_MAPPING)	32	RW	0000_FFFFh
354h	CM Setpoint 13 Mapping (CM_SP13_MAPPING)	32	RW	0000_FFFFh
358h	CM Setpoint 14 Mapping (CM_SP14_MAPPING)	32	RW	0000_FFFFh
35Ch	CM Setpoint 15 Mapping (CM_SP15_MAPPING)	32	RW	0000_FFFFh
380h	CM standby control (CM_STBY_CTRL)	32	RW	0000_0000h

19.6.1.2 CM Authentication Control (CM_AUTHEN_CTRL)

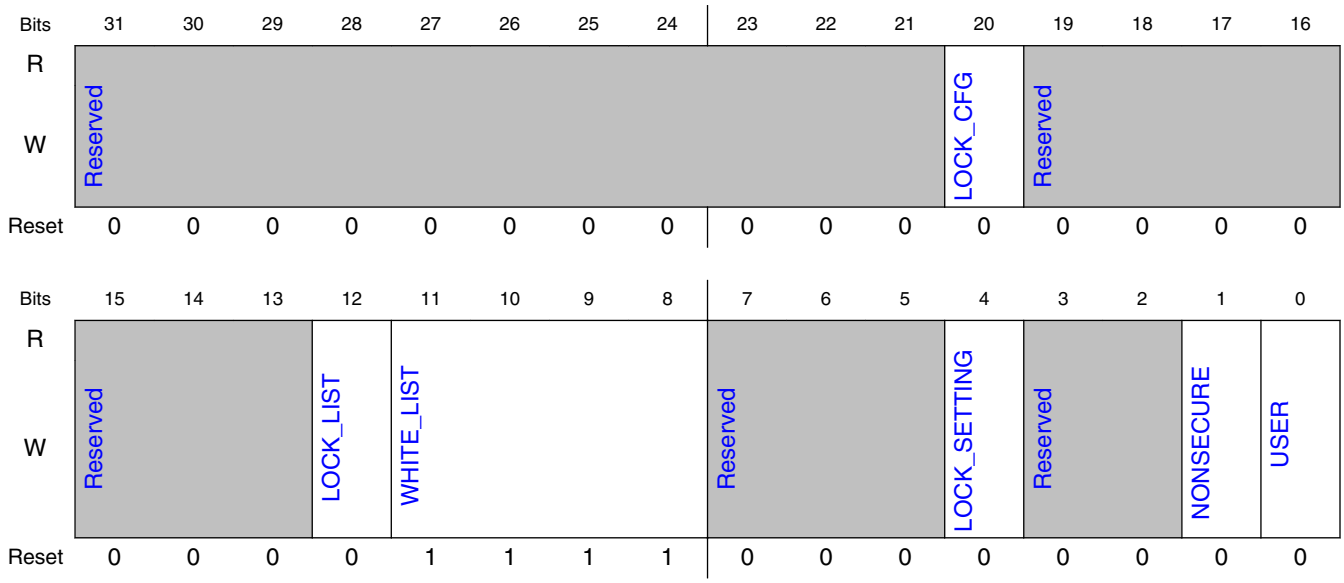
19.6.1.2.1 Offset

Register	Offset
CM_AUTHEN_CTRL	4h

19.6.1.2.2 Function

Authentication control for CMC

19.6.1.2.3 Diagram



19.6.1.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields. Unlocked by System Reset.
19-13 —	Reserved
12 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field
11-8 WHITE_LIST	Domain ID white list when bit setting to 1, the corresponding domain ID can access CPU mode control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Muliple bits with a value of 1 is permitted.
7-5 —	Reserved
4 LOCK_SETTING	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
3-2 —	Reserved
1 NONSECURE	Allow non-secure mode access 0b - Allow only secure mode to access CPU mode control registers 1b - Allow both secure and non-secure mode to access CPU mode control registers

Table continues on the next page...

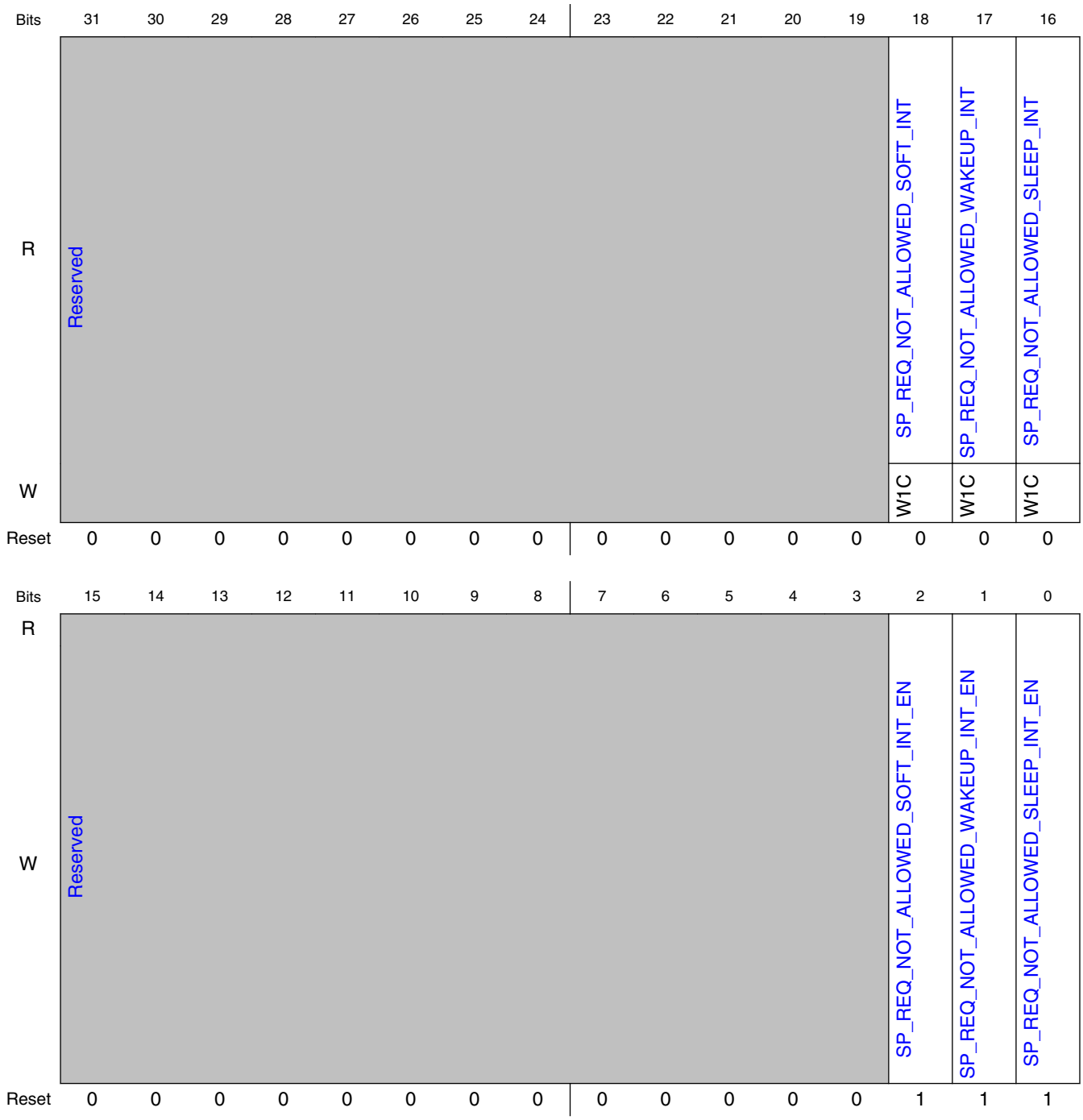
Field	Function
0 USER	Allow user mode access 0b - Allow only privilege mode to access CPU mode control registers 1b - Allow both privilege and user mode to access CPU mode control registers

19.6.1.3 CM Interrupt Control (CM_INT_CTRL)

19.6.1.3.1 Offset

Register	Offset
CM_INT_CTRL	8h

19.6.1.3.2 Diagram



19.6.1.3.3 Fields

Field	Function
31-19	Reserved

Table continues on the next page...

Field	Function
—	
18 SP_REQ_NOT_ALLOWED_SOFT_INT	sp_req_not_allowed_for_soft interrupt status and clear register when writing a "1" to CPU_SP_RUN_EN to trigger a software Setpoint transition request, if the target Setpoint in CPU_SP_RUN is not allowed by CM_RUN_MODE_MAPPING, this interrupt asserts.
17 SP_REQ_NOT_ALLOWED_WAKEUP_INT	sp_req_not_allowed_for_wakeup interrupt status and clear register for wakeup sequence, if the target Setpoint is not allowed by CM_RUN_MODE_MAPPING, this interrupt asserts. The target Setpoint is determined by CPU_SP_WAKEUP_EN, CPU_SP_WAKEUP_SEL, CPU_SP_WAKEUP and CPU_SP_CURRENT
16 SP_REQ_NOT_ALLOWED_SLEEP_INT	sp_req_not_allowed_for_sleep interrupt status and clear register during sleep sequence, if the target Setpoint is not allowed by CM_X_MODE_MAPPING, this interrupt asserts. The "X" is inside WAIT, STOP and SUSPEND, which is determined by CPU_MODE_TARGET. The target Setpoint is determined by CPU_SP_SLEEP if CPU_SP_SLEEP_EN set, and is determined by CPU_SP_CURRENT if CPU_SP_SLEEP_EN unset
15-3 —	Reserved
2 SP_REQ_NOT_ALLOWED_SOFT_INT_EN	sp_req_not_allowed_for_soft interrupt enable See SP_REQ_NOT_ALLOWED_SOFT_INT for more information. 0b - Interrupt disable 1b - Interrupt enable
1 SP_REQ_NOT_ALLOWED_WAKEUP_INT_EN	sp_req_not_allowed_for_wakeup interrupt enable See SP_REQ_NOT_ALLOWED_WAKEUP_INT for more information. 0b - Interrupt disable 1b - Interrupt enable
0 SP_REQ_NOT_ALLOWED_SLEEP_INT_EN	sp_req_not_allowed_for_sleep interrupt enable See SP_REQ_NOT_ALLOWED_SLEEP_INT for more information. 0b - Interrupt disable 1b - Interrupt enable

19.6.1.4 Miscellaneous (CM_MISC)

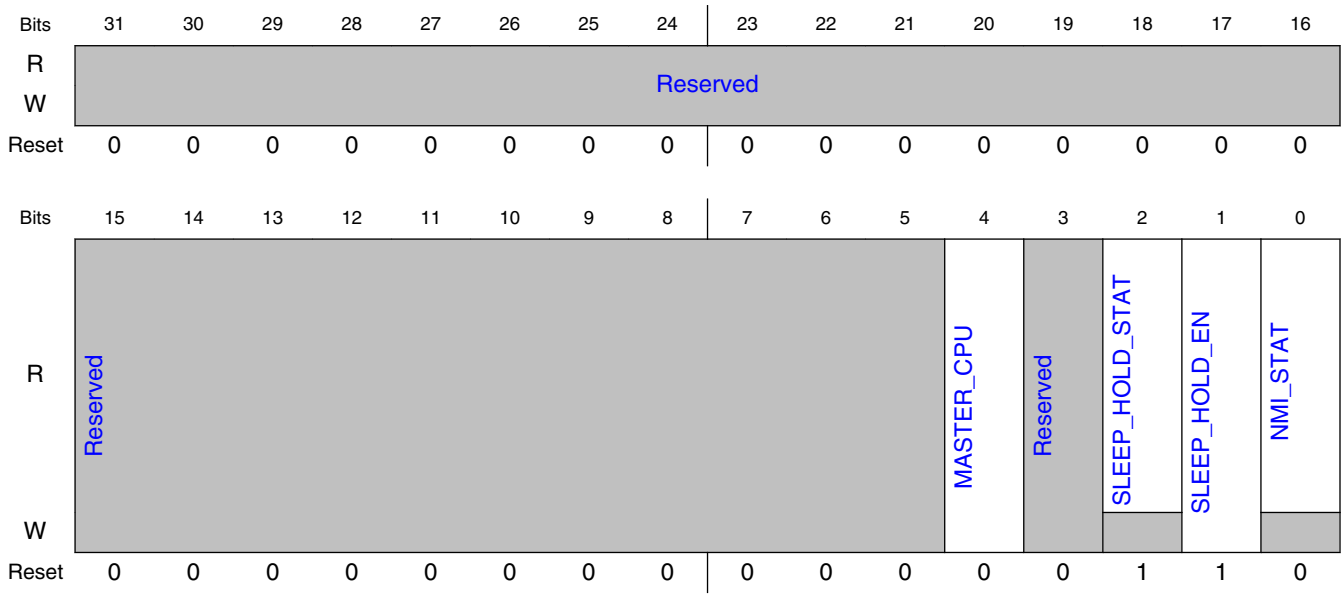
19.6.1.4.1 Offset

Register	Offset
CM_MISC	Ch

19.6.1.4.2 Function

Miscellaneous register

19.6.1.4.3 Diagram



19.6.1.4.4 Fields

Field	Function
31-5 —	Reserved
4 MASTER_CPU	Master CPU Indicate this CPU is a master CPU inside the domain, when mux the UPI domain output, the master CPU is selected if more than one CPU is assigned to same domain. Locked by LOCK_CFG field
3 —	Reserved
2 SLEEP_HOLD_STAT	Status of cpu_sleep_hold_ack_b
1 SLEEP_HOLD_EN	Allow cpu_sleep_hold_req assert during CPU low power status 0b - Disable cpu_sleep_hold_req 1b - Allow cpu_sleep_hold_req assert during CPU low power status
0 NMI_STAT	Non-masked interrupt status 0b - NMI is not asserting 1b - NMI is asserting

19.6.1.5 CPU mode control (CM_MODE_CTRL)

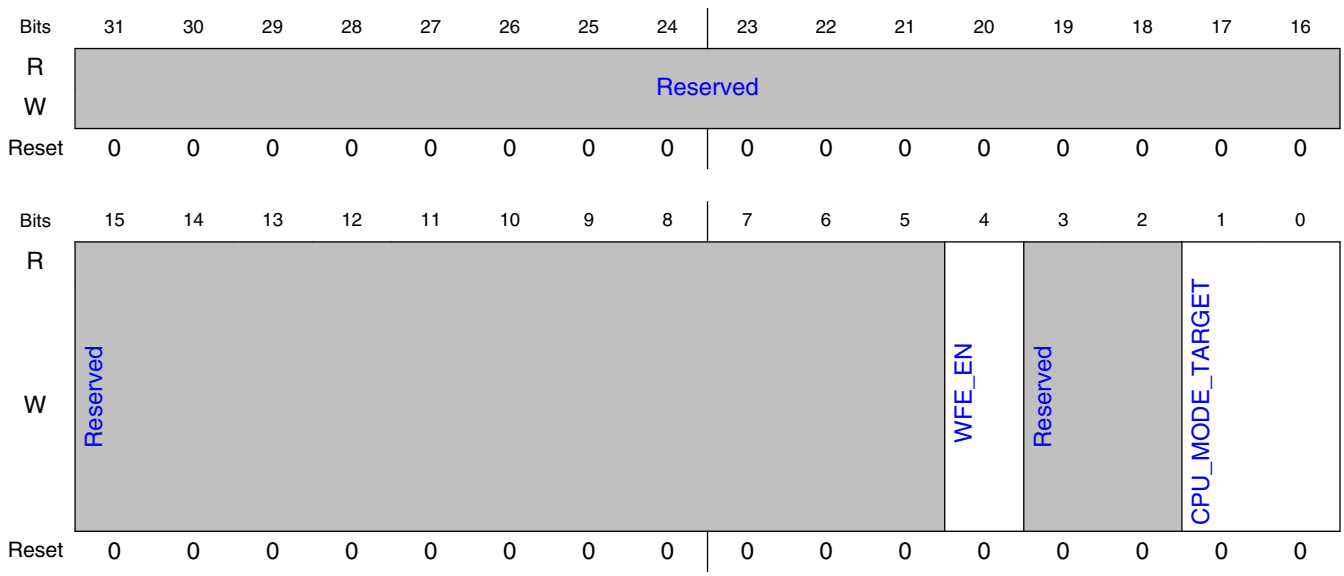
19.6.1.5.1 Offset

Register	Offset
CM_MODE_CTRL	10h

19.6.1.5.2 Function

CPU mode control register

19.6.1.5.3 Diagram



19.6.1.5.4 Fields

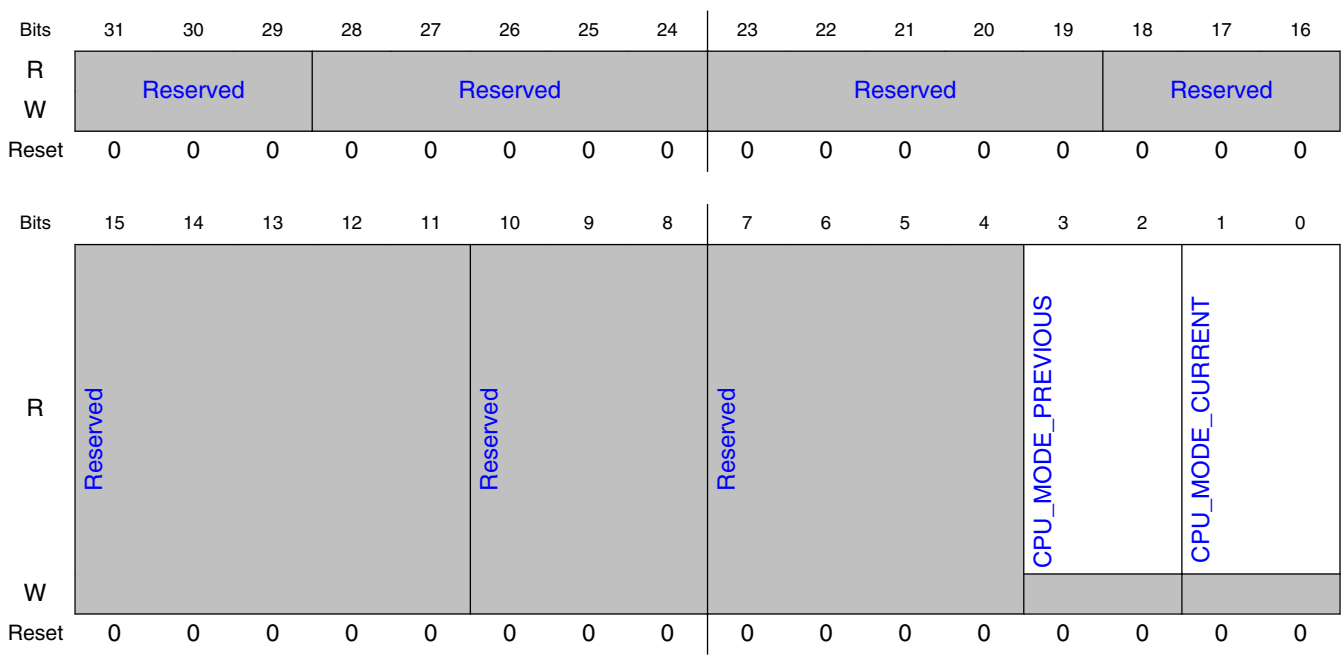
Field	Function
31-5 —	Reserved
4 WFE_EN	WFE assertion can be sleep event 0b - WFE assertion can not trigger low power 1b - WFE assertion can trigger low power
3-2 —	Reserved
1-0 CPU_MODE_TARGET	The CPU mode the CPU platform should transit to on next sleep event Important: Core MUST re-configure this field EACH TIME before entering sleep 00b - Stay in RUN mode 01b - Transit to WAIT mode 10b - Transit to STOP mode 11b - Transit to SUSPEND mode

19.6.1.6 CM CPU mode Status (CM_MODE_STAT)

19.6.1.6.1 Offset

Register	Offset
CM_MODE_STAT	14h

19.6.1.6.2 Diagram



19.6.1.6.3 Fields

Field	Function
31-29 —	Reserved
28-24 —	Reserved
23-19 —	Reserved
18-16	Reserved

Table continues on the next page...

Field	Function
—	
15-11 —	Reserved
10-8 —	Reserved
7-4 —	Reserved
3-2 CPU_MODE_PREVIOUS	Previous CPU mode 00b - CPU was previously in RUN mode 01b - CPU was previously in WAIT mode 10b - CPU was previously in STOP mode 11b - CPU was previously in SUSPEND mode
1-0 CPU_MODE_CURRENT	Current CPU mode 00b - CPU is currently in RUN mode 01b - CPU is currently in WAIT mode 10b - CPU is currently in STOP mode 11b - CPU is currently in SUSPEND mode

19.6.1.7 CM IRQ0~31 wakeup mask (CM_IRQ_WAKEUP_MASK_0)

19.6.1.7.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_0	100h

19.6.1.7.2 Function

IRQ0~31 mask register

19.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_0_31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_0_31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.7.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_0_31	"1" means the IRQ cannot wakeup CPU platform

19.6.1.8 CM IRQ32~63 wakeup mask (CM_IRQ_WAKEUP_MASK_1)

19.6.1.8.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_1	104h

19.6.1.8.2 Function

IRQ32~63 mask register

19.6.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_32_63															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_32_63															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.8.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_32_63	"1" means the IRQ cannot wakeup CPU platform

19.6.1.9 CM IRQ64~95 wakeup mask (CM_IRQ_WAKEUP_MASK_2)

19.6.1.9.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_2	108h

19.6.1.9.2 Function

IRQ64~95 mask register

19.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_64_95															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_64_95															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.9.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_64_95	"1" means the IRQ cannot wakeup CPU platform

19.6.1.10 CM IRQ96~127 wakeup mask (CM_IRQ_WAKEUP_MASK_3)

19.6.1.10.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_3	10Ch

19.6.1.10.2 Function

IRQ96~127 mask register

19.6.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_96_127															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_96_127															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.10.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_96_127	"1" means the IRQ cannot wakeup CPU platform

19.6.1.11 CM IRQ128~159 wakeup mask (CM_IRQ_WAKEUP_MASK_4)

19.6.1.11.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_4	110h

19.6.1.11.2 Function

IRQ128~159 mask register

19.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_128_159															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_128_159															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.11.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_128_159	"1" means the IRQ cannot wakeup CPU platform

19.6.1.12 CM IRQ160~191 wakeup mask (CM_IRQ_WAKEUP_MASK_5)

19.6.1.12.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_5	114h

19.6.1.12.2 Function

IRQ160~191 mask register

19.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_160_191															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_160_191															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.12.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_160_191	"1" means the IRQ cannot wakeup CPU platform

19.6.1.13 CM IRQ192~223 wakeup mask (CM_IRQ_WAKEUP_MASK_6)

19.6.1.13.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_6	118h

19.6.1.13.2 Function

IRQ192~223 mask register

19.6.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_192_223															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_192_223															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.13.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_192_223	"1" means the IRQ cannot wakeup CPU platform

19.6.1.14 CM IRQ224~255 wakeup mask (CM_IRQ_WAKEUP_MASK_7)

19.6.1.14.1 Offset

Register	Offset
CM_IRQ_WAKEUP_MASK_7	11Ch

19.6.1.14.2 Function

IRQ224~255 mask register

19.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQ_WAKEUP_MASK_224_255															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQ_WAKEUP_MASK_224_255															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.14.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_224_255	"1" means the IRQ cannot wakeup CPU platform

19.6.1.15 CM non-irq wakeup mask (CM_NON_IRQ_WAKEUP_MASK)

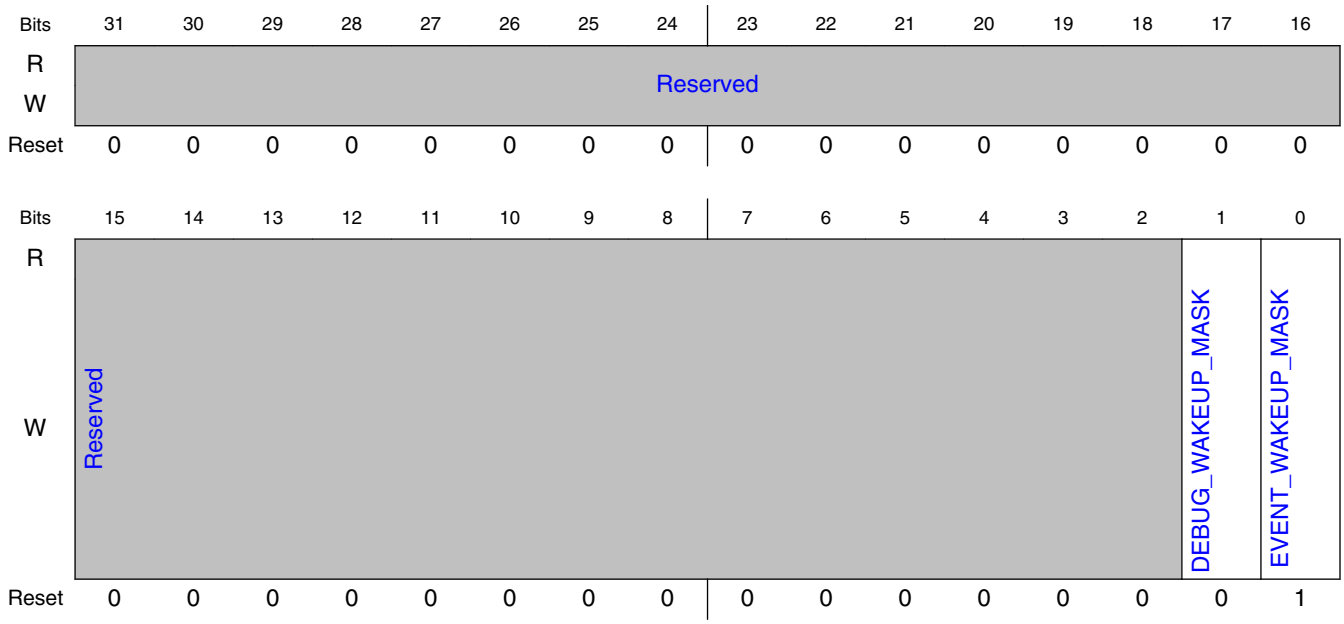
19.6.1.15.1 Offset

Register	Offset
CM_NON_IRQ_WAKEUP_MASK	140h

19.6.1.15.2 Function

This register is used to mask wakeup events which are not interrupt

19.6.1.15.3 Diagram



19.6.1.15.4 Fields

Field	Function
31-2 —	Reserved
1 DEBUG_WAKEUP_MASK	"1" means the debug_wakeup_request cannot wakeup CPU platform
0 EVENT_WAKEUP_MASK	There are 256 interrupts and 1 event as a wakeup source for GPC. This field masks the 1 event wakeup source. 1b - The event cannot wakeup CPU platform

19.6.1.16 CM IRQ0~31 wakeup status (CM_IRQ_WAKEUP_STAT_0)

19.6.1.16.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_0	150h

19.6.1.16.2 Function

Interrupt status

19.6.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_0_31															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_0_31															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.16.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_0_31	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.17 CM IRQ32~63 wakeup status (CM_IRQ_WAKEUP_STAT_1)

19.6.1.17.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_1	154h

19.6.1.17.2 Function

Interrupt status

19.6.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_32_63															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_32_63															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.17.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_32_63	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.18 CM IRQ64~95 wakeup status (CM_IRQ_WAKEUP_STAT_2)

19.6.1.18.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_2	158h

19.6.1.18.2 Function

Interrupt status

19.6.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_64_95															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_64_95															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.18.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_64_95	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.19 CM IRQ96~127 wakeup status (CM_IRQ_WAKEUP_STAT_3)

19.6.1.19.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_3	15Ch

19.6.1.19.2 Function

Interrupt status

19.6.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_96_127															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_96_127															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.19.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_96_127	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.20 CM IRQ128~159 wakeup status (CM_IRQ_WAKEUP_STAT_4)

19.6.1.20.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_4	160h

19.6.1.20.2 Function

Interrupt status

19.6.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_128_159															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_128_159															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.20.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_128_159	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.21 CM IRQ160~191 wakeup status (CM_IRQ_WAKEUP_STAT_5)

19.6.1.21.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_5	164h

19.6.1.21.2 Function

Interrupt status

19.6.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_160_191															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_160_191															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.21.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_160_191	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.22 CM IRQ192~223 wakeup status (CM_IRQ_WAKEUP_STAT_6)

19.6.1.22.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_6	168h

19.6.1.22.2 Function

Interrupt status

19.6.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_STAT_192_223															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_STAT_192_223															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.22.4 Fields

Field	Function
31-0 IRQ_WAKEUP_STAT_192_223	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.23 CM IRQ224~255 wakeup status (CM_IRQ_WAKEUP_STAT_7)

19.6.1.23.1 Offset

Register	Offset
CM_IRQ_WAKEUP_STAT_7	16Ch

19.6.1.23.2 Function

Interrupt status

19.6.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQ_WAKEUP_MASK_224_255															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ_WAKEUP_MASK_224_255															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.23.4 Fields

Field	Function
31-0 IRQ_WAKEUP_MASK_224_255	IRQ status 0000_0000_0000_0000_0000_0000_0000_0000b - None. No pending interrupts 0000_0000_0000_0000_0000_0000_0000_0001b - Valid. A valid interrupt is pending

19.6.1.24 CM non-irq wakeup status (CM_NON_IRQ_WAKEUP_STAT)

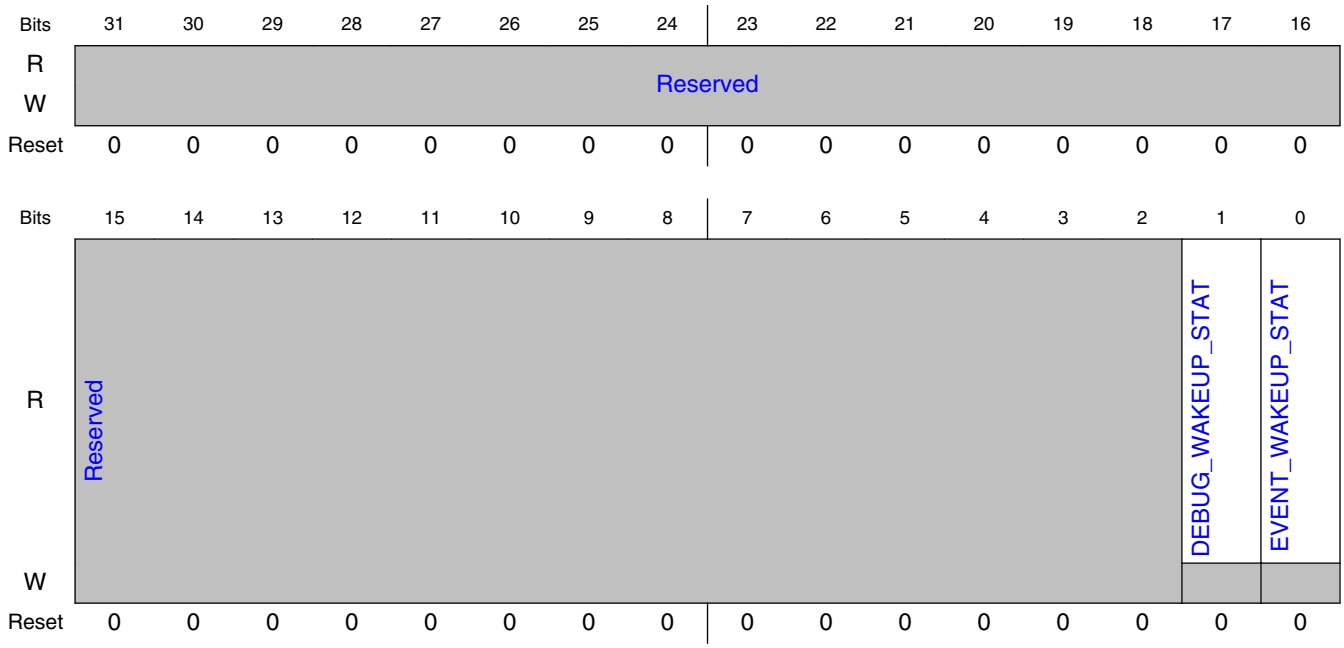
19.6.1.24.1 Offset

Register	Offset
CM_NON_IRQ_WAKEUP_STAT	190h

19.6.1.24.2 Function

Wakeup event status

19.6.1.24.3 Diagram



19.6.1.24.4 Fields

Field	Function
31-2 —	Reserved
1 DEBUG_WAKEUP_STAT	Debug wakeup status
0 EVENT_WAKEUP_STAT	Event wakeup status 1b - Interrupt is asserting (pending)

19.6.1.25 CM sleep SSAR control (CM_SLEEP_SSAR_CTRL)

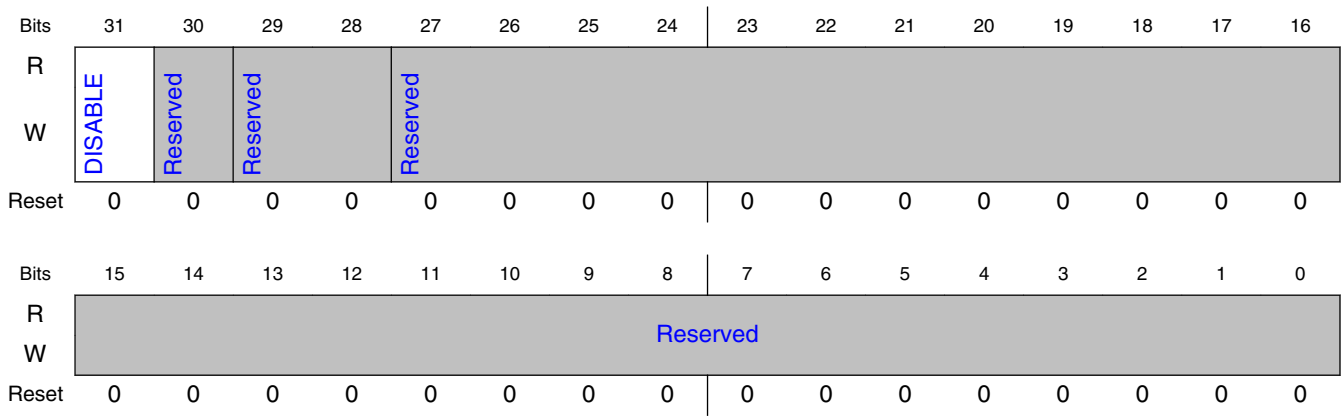
19.6.1.25.1 Offset

Register	Offset
CM_SLEEP_SSAR_CTRL	200h

19.6.1.25.2 Function

Locked by LOCK_CFG field.

19.6.1.25.3 Diagram



19.6.1.25.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.26 CM sleep LPCG control (CM_SLEEP_LPCG_CTRL)

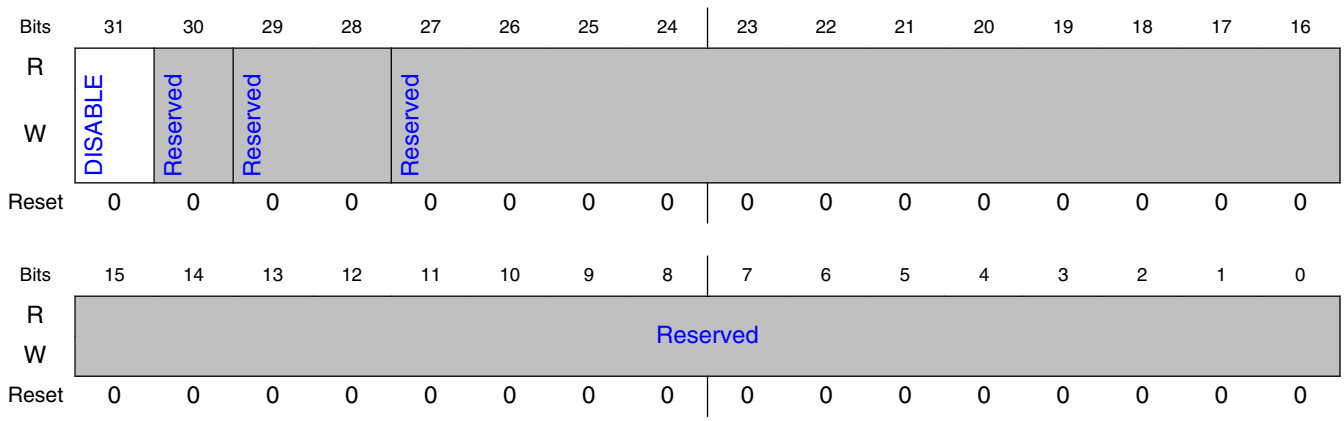
19.6.1.26.1 Offset

Register	Offset
CM_SLEEP_LPCG_CTRL	208h

19.6.1.26.2 Function

Locked by LOCK_CFG field.

19.6.1.26.3 Diagram



19.6.1.26.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.27 CM sleep PLL control (CM_SLEEP_PLL_CTRL)

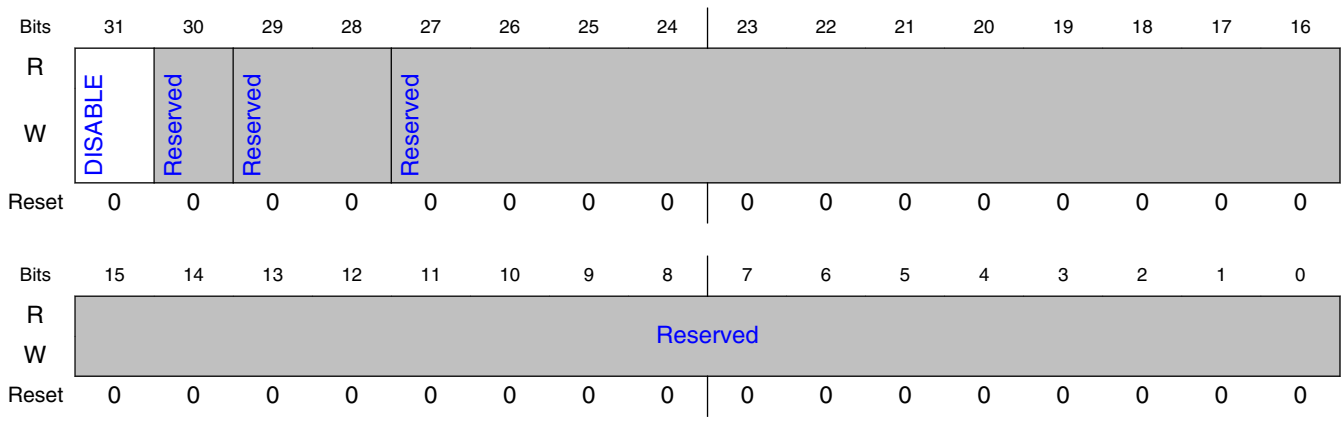
19.6.1.27.1 Offset

Register	Offset
CM_SLEEP_PLL_CTRL	210h

19.6.1.27.2 Function

Locked by LOCK_CFG field.

19.6.1.27.3 Diagram



19.6.1.27.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.28 CM sleep isolation control (CM_SLEEP_ISO_CTRL)

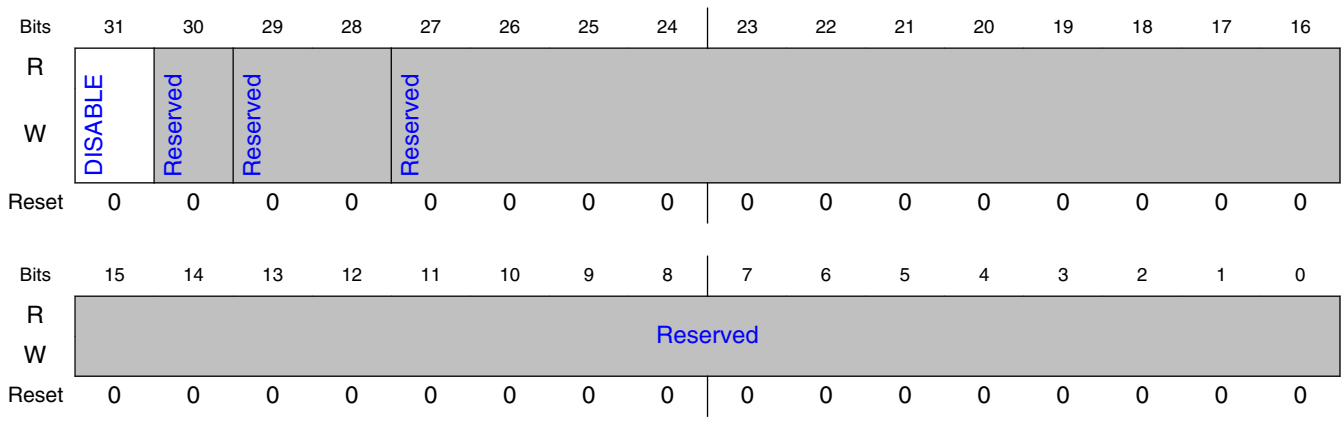
19.6.1.28.1 Offset

Register	Offset
CM_SLEEP_ISO_CTRL	218h

19.6.1.28.2 Function

Locked by LOCK_CFG field.

19.6.1.28.3 Diagram



19.6.1.28.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.29 CM sleep reset control (CM_SLEEP_RESET_CTRL)

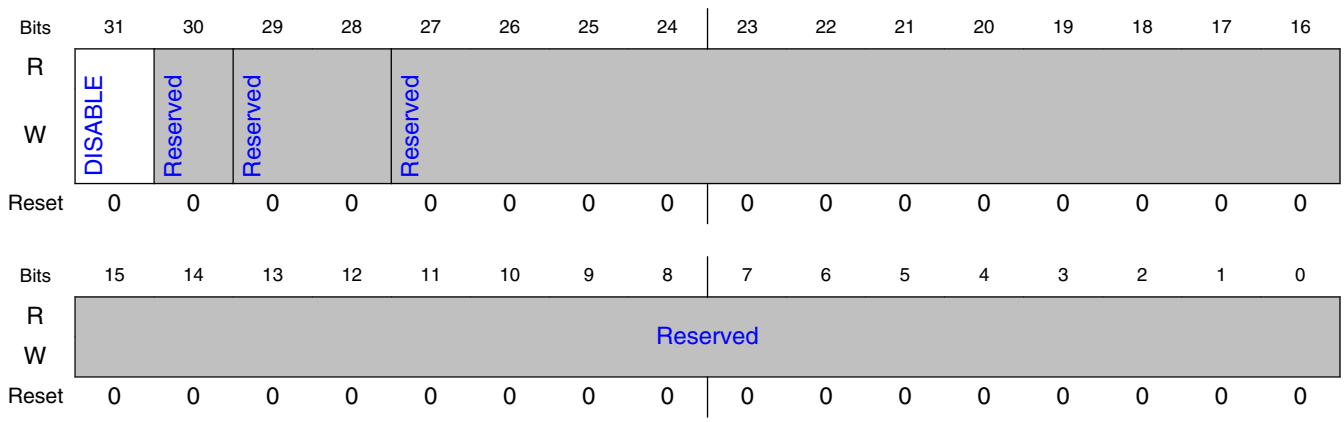
19.6.1.29.1 Offset

Register	Offset
CM_SLEEP_RESET_CTRL	220h

19.6.1.29.2 Function

Locked by LOCK_CFG field.

19.6.1.29.3 Diagram



19.6.1.29.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.30 CM sleep power control (CM_SLEEP_POWER_CTRL)

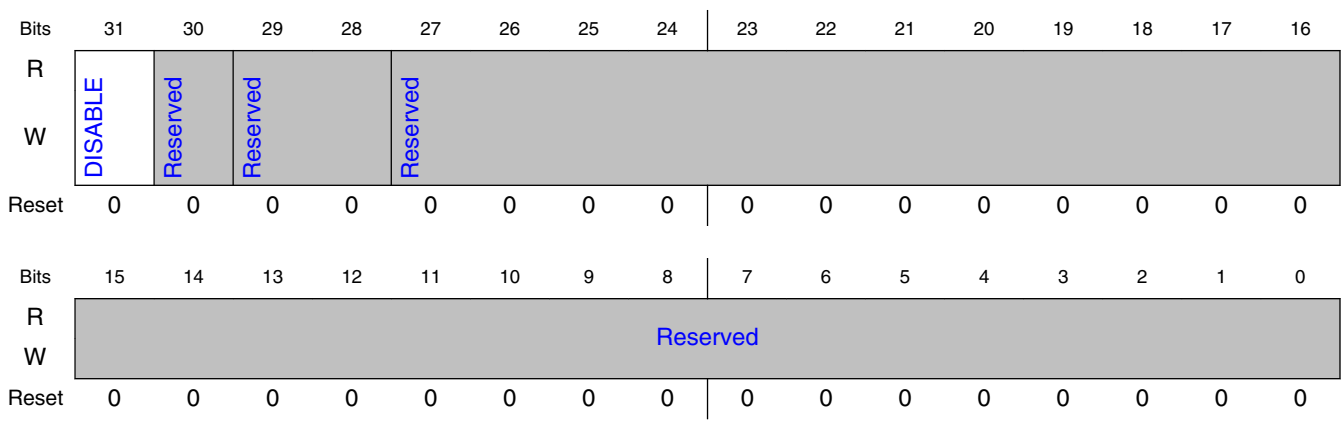
19.6.1.30.1 Offset

Register	Offset
CM_SLEEP_POWER_CTRL	228h

19.6.1.30.2 Function

Locked by LOCK_CFG field.

19.6.1.30.3 Diagram



19.6.1.30.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.31 CM wakeup power control (CM_WAKEUP_POWER_CTRL)

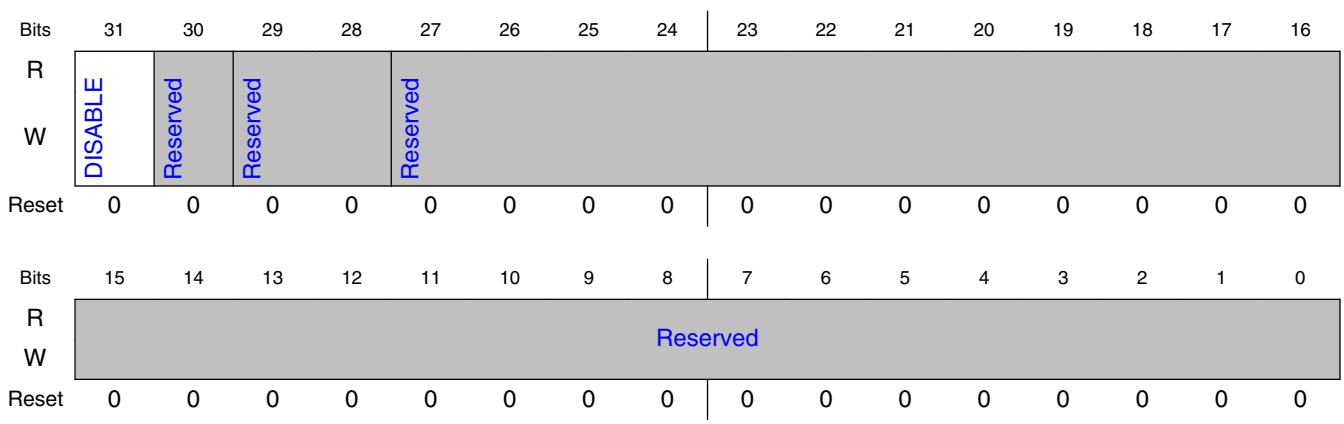
19.6.1.31.1 Offset

Register	Offset
CM_WAKEUP_POWER_CTRL	290h

19.6.1.31.2 Function

Locked by LOCK_CFG field.

19.6.1.31.3 Diagram



19.6.1.31.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.32 CM wakeup reset control (CM_WAKEUP_RESET_CTRL)

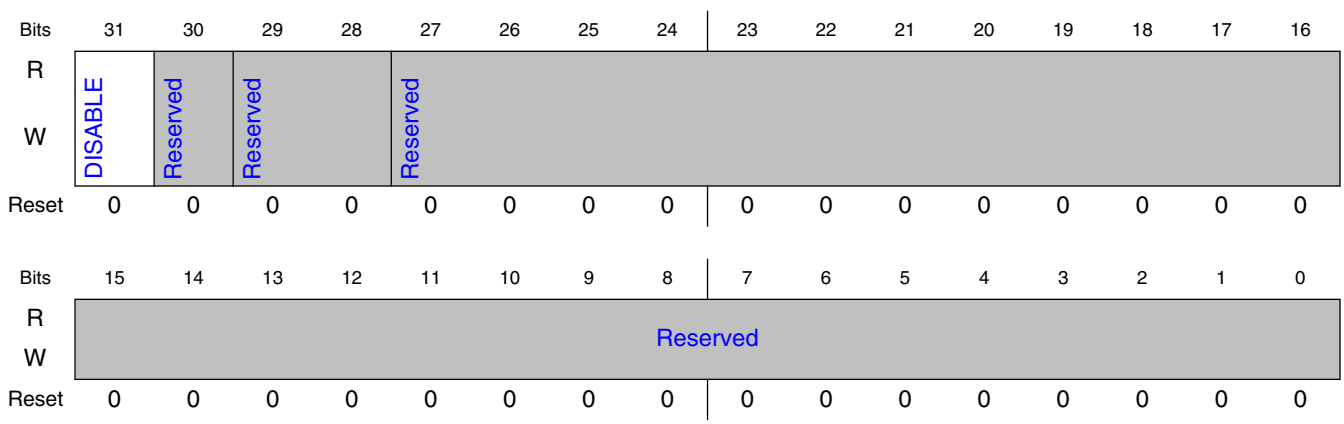
19.6.1.32.1 Offset

Register	Offset
CM_WAKEUP_RESET_CTRL	298h

19.6.1.32.2 Function

Locked by LOCK_CFG field.

19.6.1.32.3 Diagram



19.6.1.32.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved

Table continues on the next page...

Field	Function
15-0 —	Reserved

19.6.1.33 CM wakeup isolation control (CM_WAKEUP_ISO_CTRL)

19.6.1.33.1 Offset

Register	Offset
CM_WAKEUP_ISO_CTRL	2A0h

19.6.1.33.2 Function

Locked by LOCK_CFG field.

19.6.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	Reserved		Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.33.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28	Reserved

Table continues on the next page...

Field	Function
—	
27-16 —	Reserved
15-0 —	Reserved

19.6.1.34 CM wakeup PLL control (CM_WAKEUP_PLL_CTRL)

19.6.1.34.1 Offset

Register	Offset
CM_WAKEUP_PLL_CTRL	2A8h

19.6.1.34.2 Function

Locked by LOCK_CFG field.

19.6.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	Reserved		Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.34.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request

Table continues on the next page...

Field	Function
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.35 CM wakeup LPCG control (CM_WAKEUP_LPCG_CTRL)

19.6.1.35.1 Offset

Register	Offset
CM_WAKEUP_LPCG_CTRL	2B0h

19.6.1.35.2 Function

Locked by LOCK_CFG field.

19.6.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	Reserved		Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.1.35.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.36 CM wakeup SSAR control (CM_WAKEUP_SSAR_CTRL)

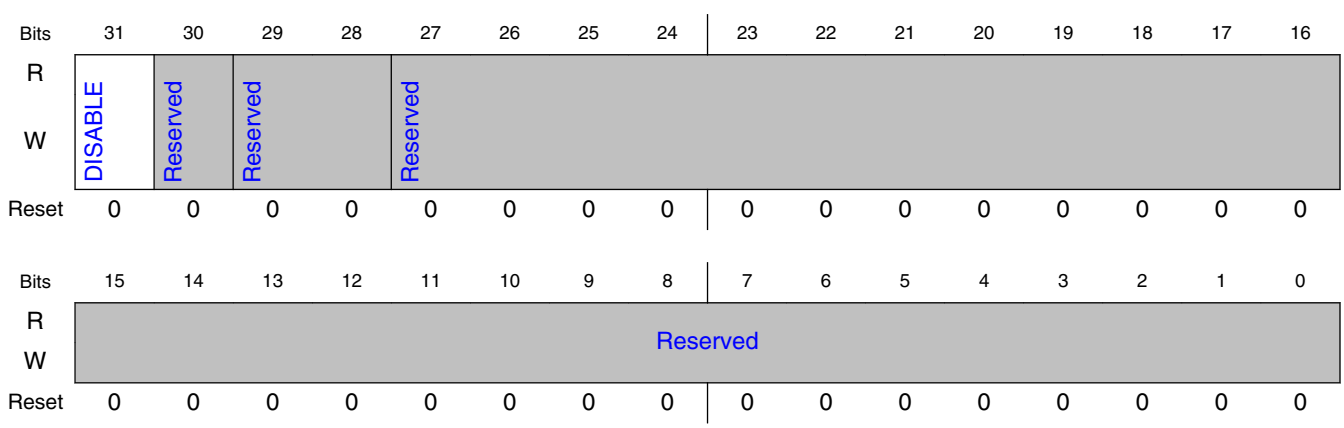
19.6.1.36.1 Offset

Register	Offset
CM_WAKEUP_SSAR_CTRL	2B8h

19.6.1.36.2 Function

Locked by LOCK_CFG field.

19.6.1.36.3 Diagram



19.6.1.36.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.1.37 CM Setpoint Control (CM_SP_CTRL)

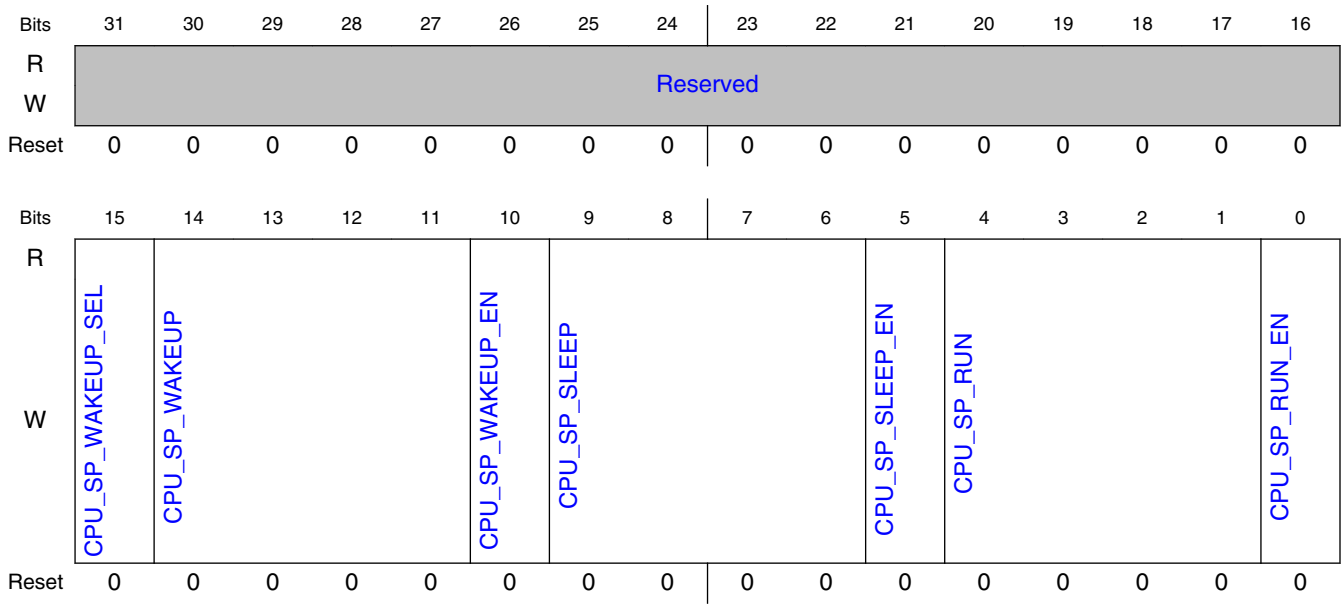
19.6.1.37.1 Offset

Register	Offset
CM_SP_CTRL	300h

19.6.1.37.2 Function

Setpoint control register.

19.6.1.37.3 Diagram



19.6.1.37.4 Fields

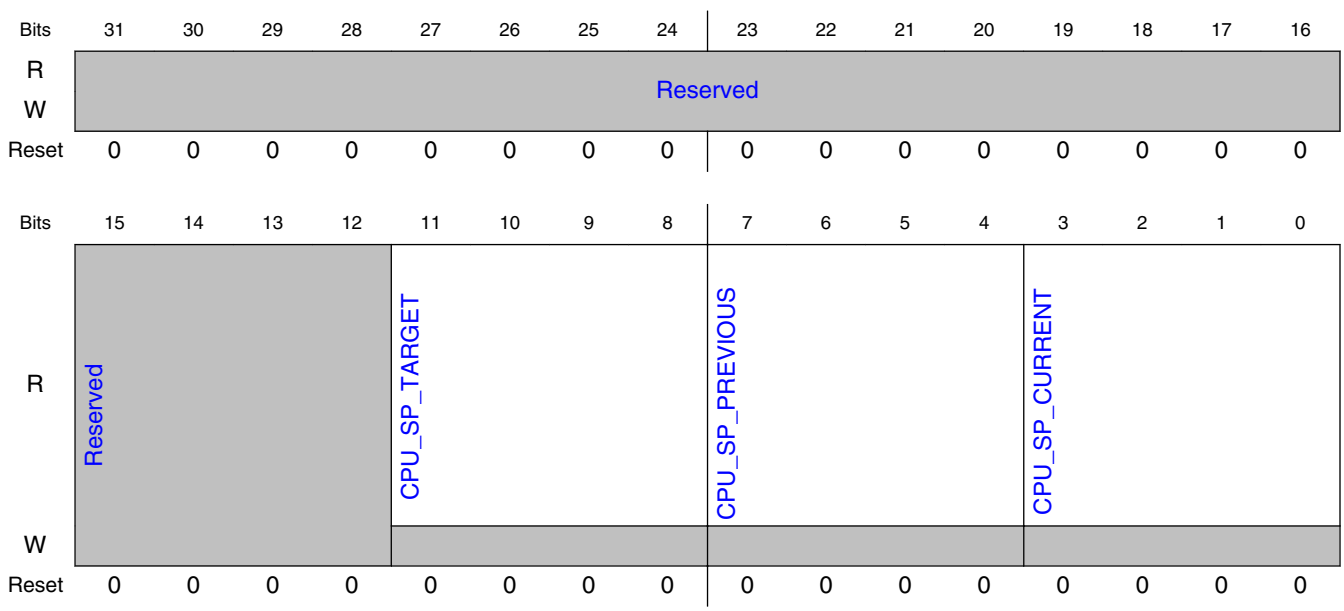
Field	Function
31-16 —	Reserved
15 CPU_SP_WAKEUP_SEL	Select the Setpoint transiton on the next CPU platform wakeup sequence 0b - Request SP transition to CPU_SP_WAKEUP 1b - Request SP transition to the Setpoint when the sleep event happens, which is captured in CPU_SP_PREVIOUS
14-11 CPU_SP_WAKEUP	The Setpoint that CPU want the system to transit to on next CPU platform wakeup sequence
10 CPU_SP_WAKEUP_EN	1 means enable Setpoint transition on next CPU platform wakeup sequence
9-6 CPU_SP_SLEEP	The Setpoint that CPU want the system to transit to on next CPU platform sleep sequence
5 CPU_SP_SLEEP_EN	1 means enable Setpoint transition on next CPU platform sleep sequence
4-1 CPU_SP_RUN	The Setpoint that CPU want the system to transit to when CPU_SP_RUN_EN is set
0 CPU_SP_RUN_EN	Request a Setpoint transition when this bit is set Write 1 will trigger Setpoint transition. The field is read as 1 until transition complete.

19.6.1.38 CM Setpoint Status (CM_SP_STAT)

19.6.1.38.1 Offset

Register	Offset
CM_SP_STAT	304h

19.6.1.38.2 Diagram



19.6.1.38.3 Fields

Field	Function
31-16 —	Reserved
15-12 —	Reserved
11-8 CPU_SP_TARGET	The requested Setpoint from the CPU platform 0000 - Setpoint 0 0001 - Setpoint 1 ...

Table continues on the next page...

Field	Function
	1111 - Setpoint 15
7-4 CPU_SP_PREVIOUS	The previous Setpoint of the system 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
3-0 CPU_SP_CURRENT	The current Setpoint of the system 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15

19.6.1.39 CM Run Mode Setpoint Allowed (CM_RUN_MODE_MAPPING)

19.6.1.39.1 Offset

Register	Offset
CM_RUN_MODE_MAPPING	310h

19.6.1.39.2 Function

Allowed setpoint list for CPU RUN mode

19.6.1.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_RUN_MODE_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.39.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_RUN_MODE_MAPPING	Defines which Setpoint is allowed when CPU enters RUN mode. Each bit stands for 1 Setpoint, locked by LOCK_CFG field

19.6.1.40 CM Wait Mode Setpoint Allowed (CM_WAIT_MODE_MAPPING)

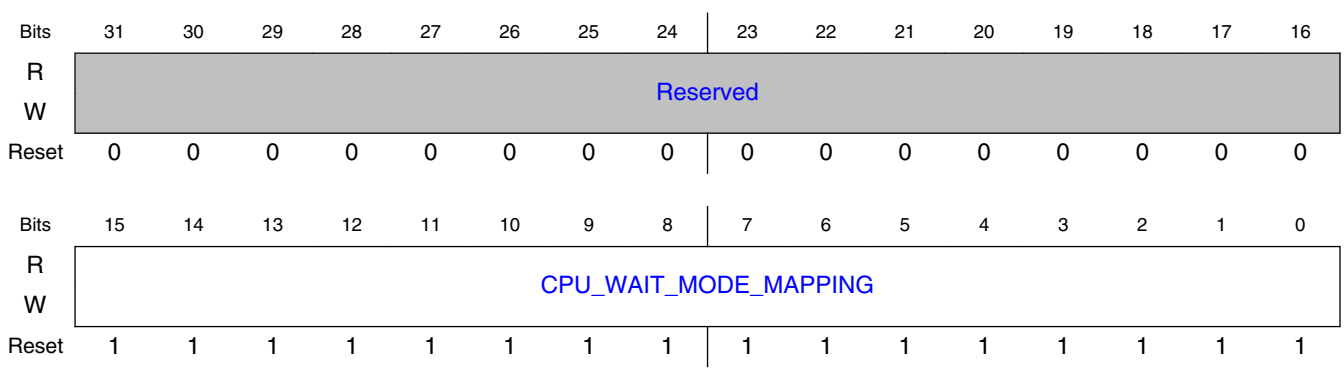
19.6.1.40.1 Offset

Register	Offset
CM_WAIT_MODE_MAPPING	314h

19.6.1.40.2 Function

Allowed setpoint list for CPU WAIT mode

19.6.1.40.3 Diagram



19.6.1.40.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_WAIT_MODE_MAPPING	Defines which Setpoint is allowed when CPU enters WAIT mode. Each bit stands for 1 Setpoint, locked by LOCK_CFG

19.6.1.41 CM Stop Mode Setpoint Allowed (CM_STOP_MODE_MAPPING)

19.6.1.41.1 Offset

Register	Offset
CM_STOP_MODE_MAPPING	318h

19.6.1.41.2 Function

Allowed setpoint list for CPU STOP mode

19.6.1.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_STOP_MODE_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.41.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 CPU_STOP_MODE_MAPPING	Defines which Setpoint is allowed when CPU enters STOP mode. Each bit stands for 1 Setpoint, locked by LOCK_CFG

19.6.1.42 CM Suspend Mode Setpoint Allowed (CM_SUSPEND_MODE_MAPPING)

19.6.1.42.1 Offset

Register	Offset
CM_SUSPEND_MODE_MAPPING	31Ch

19.6.1.42.2 Function

Allowed setpoint list for CPU SUSPEND mode

19.6.1.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SUSPEND_MODE_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.42.4 Fields

Field	Function
31-16 —	Reserved
15-0	Defines which Setpoint is allowed when CPU enters SUSPEND mode. Each bit stands for 1 Setpoint, locked by LOCK_CFG

Field	Function
CPU_SUSPEND_MODE_MAPPING	

19.6.1.43 CM Setpoint 0 Mapping (CM_SP0_MAPPING)

19.6.1.43.1 Offset

Register	Offset
CM_SP0_MAPPING	320h

19.6.1.43.2 Function

Allowed setpoint list for setpoint0 target

19.6.1.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP0_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.43.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP0_MAPPING	Defines when SP0 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.44 CM Setpoint 1 Mapping (CM_SP1_MAPPING)

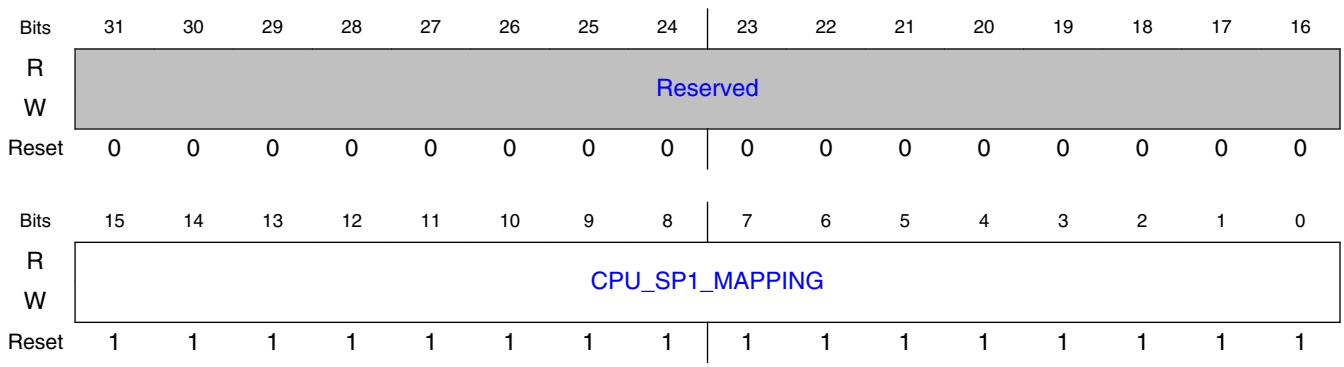
19.6.1.44.1 Offset

Register	Offset
CM_SP1_MAPPING	324h

19.6.1.44.2 Function

Allowed setpoint list for setpoint1 target

19.6.1.44.3 Diagram



19.6.1.44.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP1_MAPPING	Defines when SP1 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.45 CM Setpoint 2 Mapping (CM_SP2_MAPPING)

19.6.1.45.1 Offset

Register	Offset
CM_SP2_MAPPING	328h

19.6.1.45.2 Function

Allowed setpoint list for setpoint2 target

19.6.1.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP2_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.45.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP2_MAPPING	Defines when SP2 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.46 CM Setpoint 3 Mapping (CM_SP3_MAPPING)

19.6.1.46.1 Offset

Register	Offset
CM_SP3_MAPPING	32Ch

19.6.1.46.2 Function

Allowed setpoint list for setpoint3 target

19.6.1.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP3_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.46.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP3_MAPPING	Defines when SP3 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.47 CM Setpoint 4 Mapping (CM_SP4_MAPPING)

19.6.1.47.1 Offset

Register	Offset
CM_SP4_MAPPING	330h

19.6.1.47.2 Function

Allowed setpoint list for setpoint4 target

19.6.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP4_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.47.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP4_MAPPING	Defines when SP4 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.48 CM Setpoint 5 Mapping (CM_SP5_MAPPING)

19.6.1.48.1 Offset

Register	Offset
CM_SP5_MAPPING	334h

19.6.1.48.2 Function

Allowed setpoint list for setpoint5 target

19.6.1.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP5_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.48.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP5_MAPPING	Defines when SP5 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.49 CM Setpoint 6 Mapping (CM_SP6_MAPPING)

19.6.1.49.1 Offset

Register	Offset
CM_SP6_MAPPING	338h

19.6.1.49.2 Function

Allowed setpoint list for setpoint6 target

19.6.1.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP6_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.49.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP6_MAPPING	Defines when SP6 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.50 CM Setpoint 7 Mapping (CM_SP7_MAPPING)

19.6.1.50.1 Offset

Register	Offset
CM_SP7_MAPPING	33Ch

19.6.1.50.2 Function

Allowed setpoint list for setpoint7 target

19.6.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP7_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.50.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP7_MAPPING	Defines when SP7 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.51 CM Setpoint 8 Mapping (CM_SP8_MAPPING)

19.6.1.51.1 Offset

Register	Offset
CM_SP8_MAPPING	340h

19.6.1.51.2 Function

Allowed setpoint list for setpoint8 target

19.6.1.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP8_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.51.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP8_MAPPING	Defines when SP8 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.52 CM Setpoint 9 Mapping (CM_SP9_MAPPING)

19.6.1.52.1 Offset

Register	Offset
CM_SP9_MAPPING	344h

19.6.1.52.2 Function

Allowed setpoint list for setpoint9 target

19.6.1.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP9_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.52.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP9_MAPPING	Defines when SP9 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.53 CM Setpoint 10 Mapping (CM_SP10_MAPPING)

19.6.1.53.1 Offset

Register	Offset
CM_SP10_MAPPING	348h

19.6.1.53.2 Function

Allowed setpoint list for setpoint10 target

19.6.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP10_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.53.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP10_MAPPING	Defines when SP10 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.54 CM Setpoint 11 Mapping (CM_SP11_MAPPING)

19.6.1.54.1 Offset

Register	Offset
CM_SP11_MAPPING	34Ch

19.6.1.54.2 Function

Allowed setpoint list for setpoint11 target

19.6.1.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP11_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.54.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP11_MAPPING	Defines when SP11 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.55 CM Setpoint 12 Mapping (CM_SP12_MAPPING)

19.6.1.55.1 Offset

Register	Offset
CM_SP12_MAPPING	350h

19.6.1.55.2 Function

Allowed setpoint list for setpoint12 target

19.6.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP12_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.55.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP12_MAPPING	Defines when SP12 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.56 CM Setpoint 13 Mapping (CM_SP13_MAPPING)

19.6.1.56.1 Offset

Register	Offset
CM_SP13_MAPPING	354h

19.6.1.56.2 Function

Allowed setpoint list for setpoint13 target

19.6.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP13_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.56.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP13_MAPPING	Defines when SP13 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.57 CM Setpoint 14 Mapping (CM_SP14_MAPPING)

19.6.1.57.1 Offset

Register	Offset
CM_SP14_MAPPING	358h

19.6.1.57.2 Function

Allowed setpoint list for setpoint14 target

19.6.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP14_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.57.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP14_MAPPING	Defines when SP14 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.58 CM Setpoint 15 Mapping (CM_SP15_MAPPING)

19.6.1.58.1 Offset

Register	Offset
CM_SP15_MAPPING	35Ch

19.6.1.58.2 Function

Allowed setpoint list for setpoint15 target

19.6.1.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CPU_SP15_MAPPING															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.6.1.58.4 Fields

Field	Function
31-16 —	Reserved
15-0 CPU_SP15_MAPPING	Defines when SP15 is set as the CPU_SP_TARGET, which SP is allowed, locked by LOCK_CFG field

19.6.1.59 CM standby control (CM_STBY_CTRL)

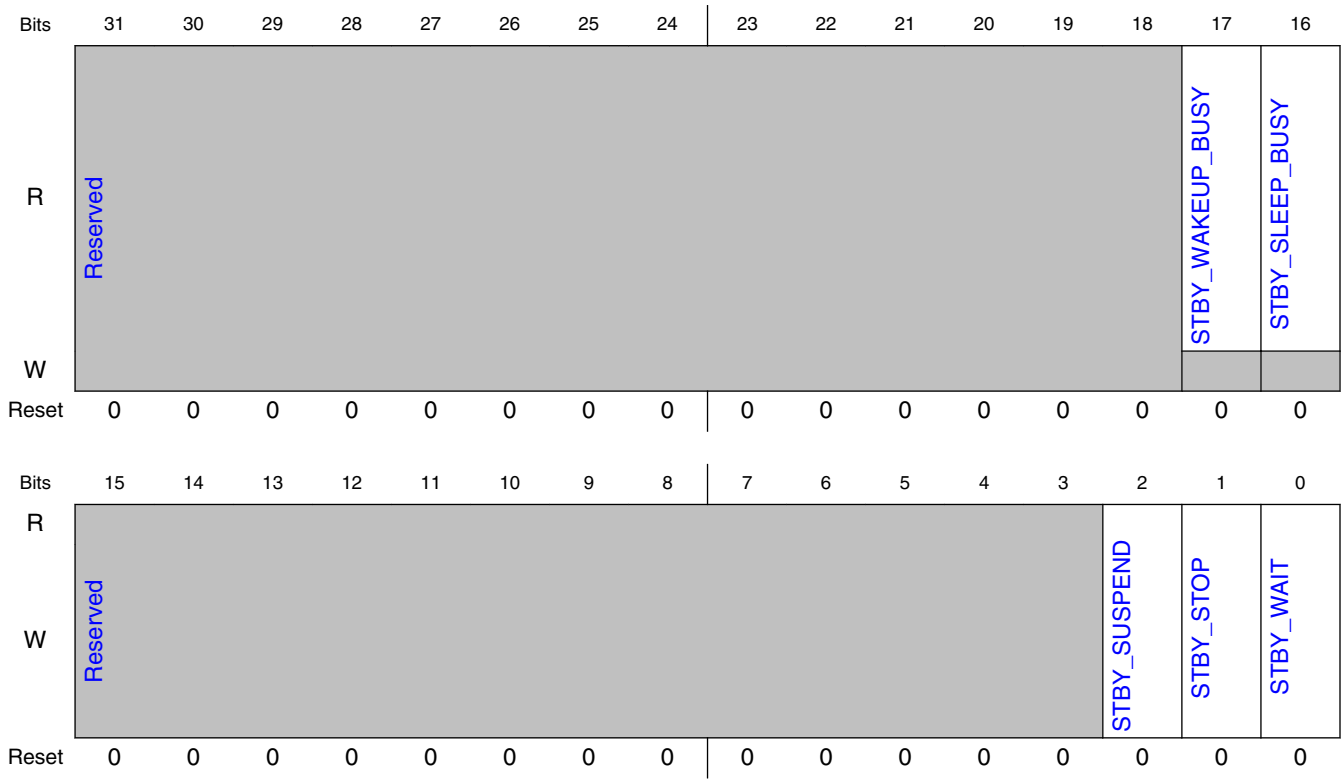
19.6.1.59.1 Offset

Register	Offset
CM_STBY_CTRL	380h

19.6.1.59.2 Function

Standby control register

19.6.1.59.3 Diagram



19.6.1.59.4 Fields

Field	Function
31-18 —	Reserved
17 STBY_WAKEUP_BUSY	Indicate the CPU is busy exiting standby mode.
16 STBY_SLEEP_BUSY	Indicate the CPU is busy entering standby mode.
15-3 —	Reserved
2 STBY_SUSPEND	0x1: Request the chip into standby mode when CPU entering SUSPEND mode, locked by LOCK_CFG field.
1 STBY_STOP	0x1: Request the chip into standby mode when CPU entering STOP mode, locked by LOCK_CFG field.
0 STBY_WAIT	0x1: Request the chip into standby mode when CPU entering WAIT mode, locked by LOCK_CFG field.

19.6.2 GPC_SP register descriptions

19.6.2.1 gpc_sp_ctrl memory map

GPC_SET_POINT_CTRL base address: 40C0_2000h

Offset	Register	Width (In bits)	Access	Reset value
4h	SP Authentication Control (SP_AUTHEN_CTRL)	32	RW	0000_0F00h
8h	SP Interrupt Control (SP_INT_CTRL)	32	RW	0000_0001h
10h	CPU SP Request (SP_CPU_REQ)	32	R	0000_0000h
14h	SP System Status (SP_SYS_STAT)	32	R	0000_FFFFh
1Ch	SP ROSC Control (SP_ROSC_CTRL)	32	RW	0000_0000h
40h	SP0~7 Priority (SP_PRIORITY_0_7)	32	RW	7654_3210h
44h	SP8~15 Priority (SP_PRIORITY_8_15)	32	RW	FEDC_BA98h
100h	SP SSAR save control (SP_SSAR_SAVE_CTRL)	32	RW	0000_0000h
110h	SP LPCG off control (SP_LPCG_OFF_CTRL)	32	RW	0000_0000h
120h	SP group down control (SP_GROUP_DOWN_CTRL)	32	RW	0000_0000h
130h	SP root down control (SP_ROOT_DOWN_CTRL)	32	RW	0000_0000h
140h	SP PLL off control (SP_PLL_OFF_CTRL)	32	RW	0000_0000h
150h	SP ISO on control (SP_ISO_ON_CTRL)	32	RW	0000_0000h
160h	SP reset early control (SP_RESET_EARLY_CTRL)	32	RW	0000_0000h
170h	SP power off control (SP_POWER_OFF_CTRL)	32	RW	0000_0000h
180h	SP bias off control (SP_BIAS_OFF_CTRL)	32	RW	0000_0000h
190h	SP bandgap and PLL_LDO off control (SP_BG_PLDO_OFF_CTRL)	32	RW	0000_0000h
1A0h	SP LDO pre control (SP_LDO_PRE_CTRL)	32	RW	0000_0000h
1B0h	SP DCDC down control (SP_DCDC_DOWN_CTRL)	32	RW	0000_0000h
200h	SP DCDC up control (SP_DCDC_UP_CTRL)	32	RW	0000_0000h
210h	SP LDO post control (SP_LDO_POST_CTRL)	32	RW	0000_0000h
220h	SP bandgap and PLL_LDO on control (SP_BG_PLDO_ON_CTRL)	32	RW	0000_0000h
230h	SP bias on control (SP_BIAS_ON_CTRL)	32	RW	0000_0000h
240h	SP power on control (SP_POWER_ON_CTRL)	32	RW	0000_0000h
250h	SP reset late control (SP_RESET_LATE_CTRL)	32	RW	0000_0000h
260h	SP ISO off control (SP_ISO_OFF_CTRL)	32	RW	0000_0000h
270h	SP PLL on control (SP_PLL_ON_CTRL)	32	RW	0000_0000h
280h	SP root up control (SP_ROOT_UP_CTRL)	32	RW	0000_0000h
290h	SP group up control (SP_GROUP_UP_CTRL)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2A0h	SP LPCG on control (SP_LPCG_ON_CTRL)	32	RW	0000_0000h
2B0h	SP SSAR restore control (SP_SSAR_RESTORE_CTRL)	32	RW	0000_0000h

19.6.2.2 SP Authentication Control (SP_AUTHEN_CTRL)

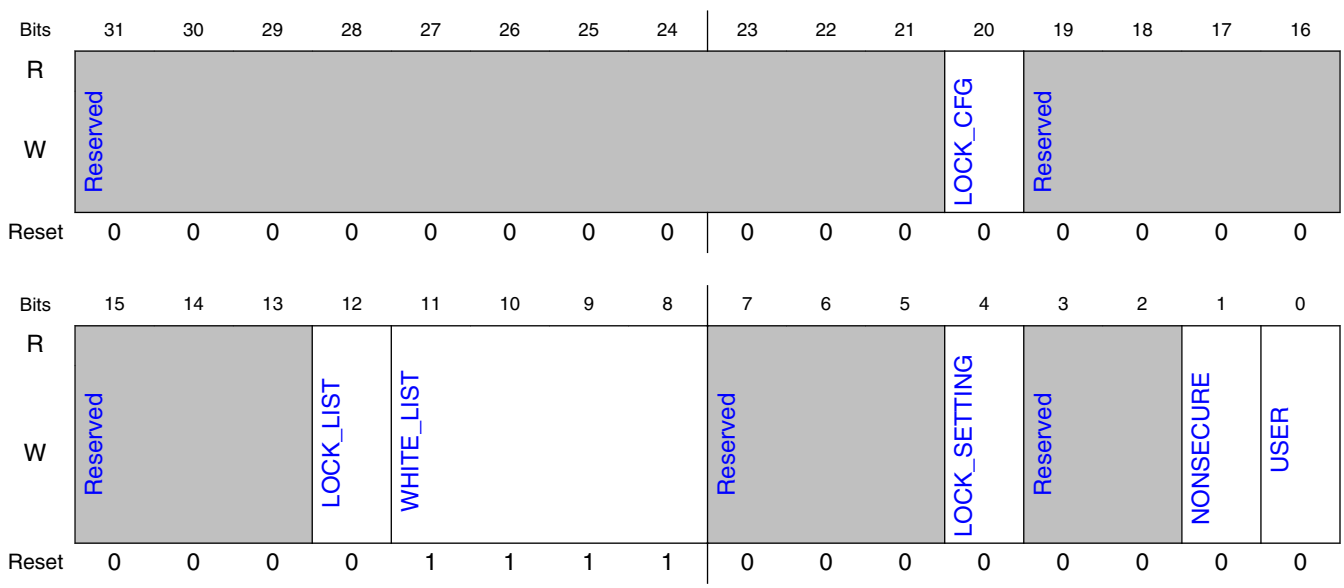
19.6.2.2.1 Offset

Register	Offset
SP_AUTHEN_CTRL	4h

19.6.2.2.2 Function

Authentication control of setpoint controller

19.6.2.2.3 Diagram



19.6.2.2.4 Fields

Field	Function
31-21	Reserved

Table continues on the next page...

Field	Function
—	
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields
19-13 —	Reserved
12 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field
11-8 WHITE_LIST	Domain ID white list when bit setting to 1, the corresponding domain ID can access CPU mode control registers
7-5 —	Reserved
4 LOCK_SETTING	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
3-2 —	Reserved
1 NONSECURE	Allow non-secure mode access 0b - Allow only secure mode to access setpoint control registers 1b - Allow both secure and non-secure mode to access setpoint control registers
0 USER	Allow user mode access 0b - Allow only privilege mode to access setpoint control registers 1b - Allow both privilege and user mode to access setpoint control registers

19.6.2.3 SP Interrupt Control (SP_INT_CTRL)

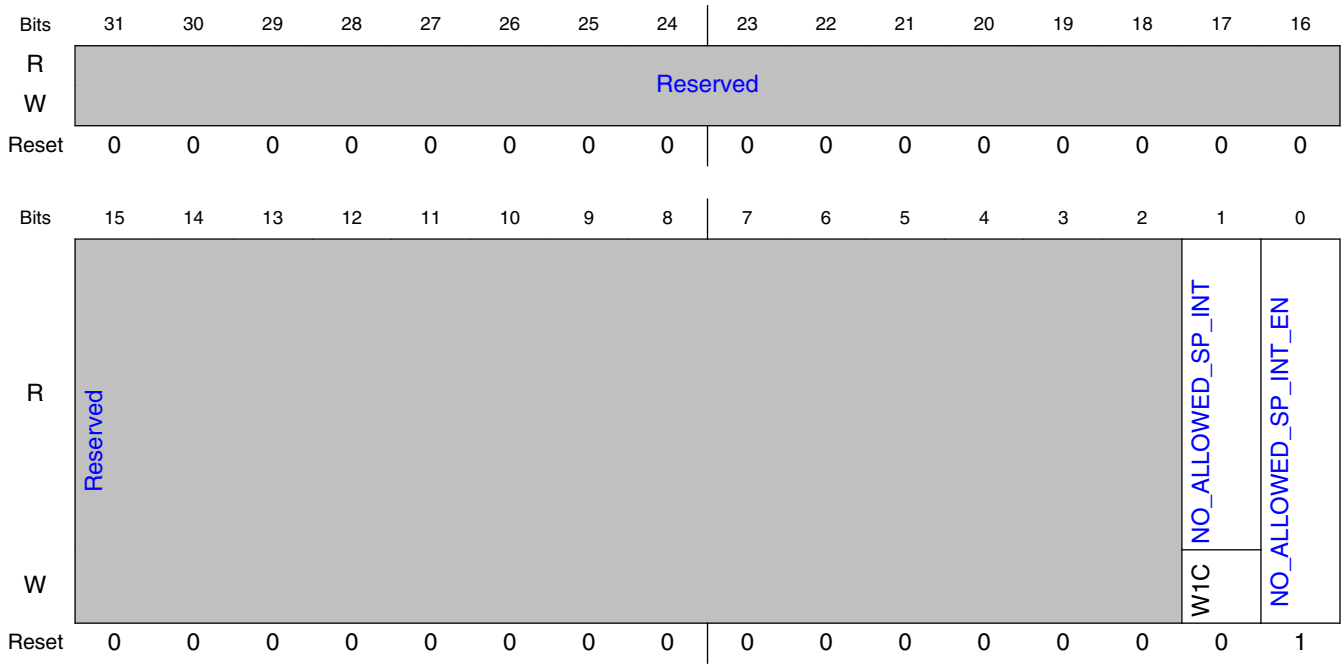
19.6.2.3.1 Offset

Register	Offset
SP_INT_CTRL	8h

19.6.2.3.2 Function

Setpoint controller interrupt control register

19.6.2.3.3 Diagram



19.6.2.3.4 Fields

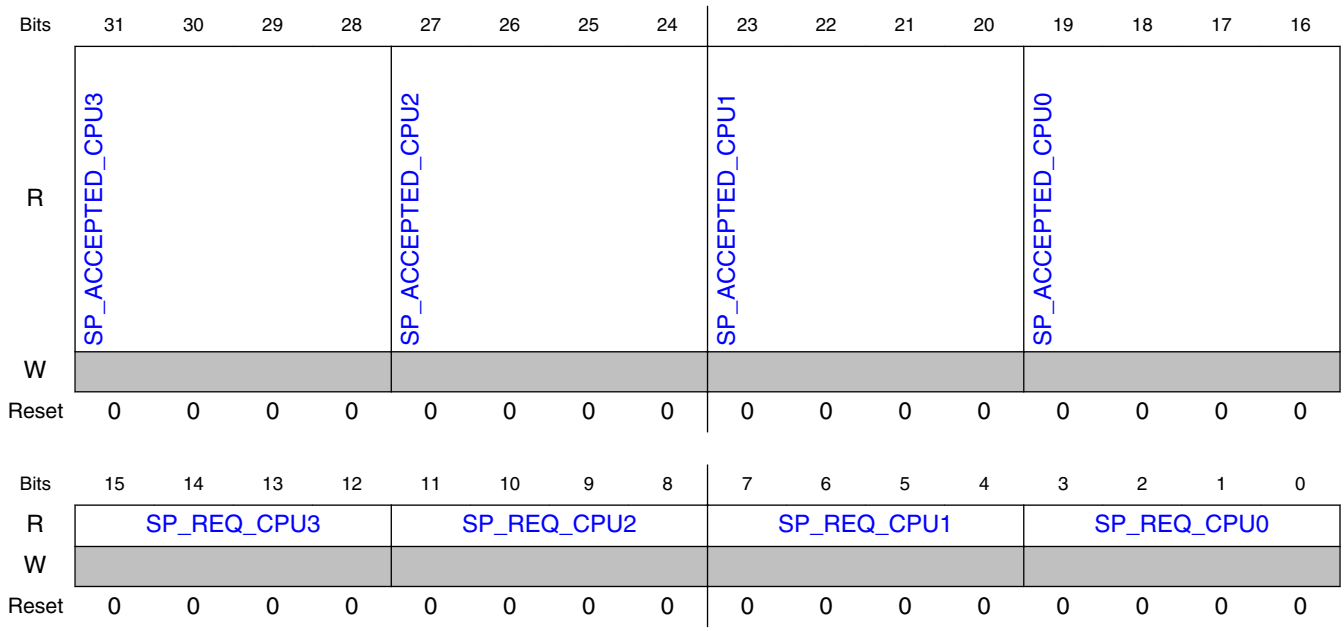
Field	Function
31-2 —	Reserved
1 NO_ALLOWED_SP_INT	no_allowed_set_point interrupt if the calculation result shows there is not any setpoint value can be allowed by all the CPU mode controller, this interrupt happens
0 NO_ALLOWED_SP_INT_EN	no_allowed_set_point interrupt enable 1 means enable to assert interrupt when no_allowed_set_point happens

19.6.2.4 CPU SP Request (SP_CPU_REQ)

19.6.2.4.1 Offset

Register	Offset
SP_CPU_REQ	10h

19.6.2.4.2 Diagram



19.6.2.4.3 Fields

Field	Function
31-28 SP_ACCEPTED_CPU3	CPU3 Setpoint accepted by SP controller 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
27-24 SP_ACCEPTED_CPU2	CPU2 Setpoint accepted by SP controller 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
23-20 SP_ACCEPTED_CPU1	CPU1 Setpoint accepted by SP controller 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
19-16 SP_ACCEPTED_CPU0	CPU0 Setpoint accepted by SP controller 0000 - Setpoint 0 0001 - Setpoint 1

Table continues on the next page...

Field	Function
	... 1111 - Setpoint 15
15-12 SP_REQ_CPU3	Setpoint requested by CPU3 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
11-8 SP_REQ_CPU2	Setpoint requested by CPU2 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
7-4 SP_REQ_CPU1	Setpoint requested by CPU1 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
3-0 SP_REQ_CPU0	Setpoint requested by CPU0 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15

19.6.2.5 SP System Status (SP_SYS_STAT)

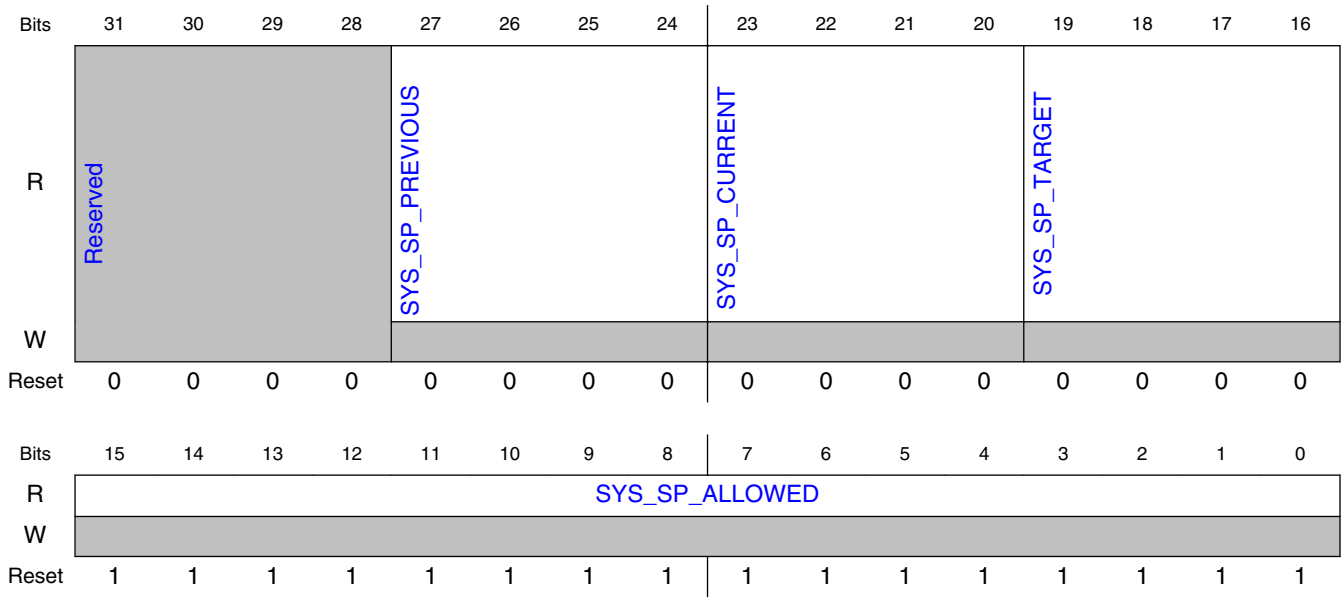
19.6.2.5.1 Offset

Register	Offset
SP_SYS_STAT	14h

19.6.2.5.2 Function

.

19.6.2.5.3 Diagram



19.6.2.5.4 Fields

Field	Function
31-28 —	Reserved
27-24 SYS_SP_PREVIOUS	Previous Setpoint, only valid when not SP trans busy 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
23-20 SYS_SP_CURRENT	Current Setpoint, only valid when not SP trans busy 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
19-16 SYS_SP_TARGET	The Setpoint chosen as the target setpoint 0000 - Setpoint 0 0001 - Setpoint 1 ... 1111 - Setpoint 15
15-0 SYS_SP_ALLOWED	Allowed Setpoints by all current CPU Setpoint requests This field is the ANDed result of allowed Setpoints from all CMCs. If all bits are 0, there are no Setpoints, and no Setpoint transitions are allowed. See System Setpoint Management topic for more information.

Field	Function
	Bit 0: Setpoint 0 is allowed Bit 1: Setpoint 1 is allowed ... Bit 15: Setpoint 15 is allowed

19.6.2.6 SP ROSC Control (SP_ROSC_CTRL)

19.6.2.6.1 Offset

Register	Offset
SP_ROSC_CTRL	1Ch

19.6.2.6.2 Function

On-chip oscillator control register

19.6.2.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SP_ALLOW_ROSC_OFF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.2.6.4 Fields

Field	Function
31-16 —	Reserved
15-0 SP_ALLOW_ROSC_OFF	Allow shutting off the ROSC ROSC is the main clock source of GPC when system is in a selected Setpoint. Locked by LOCK_CFG field. Each bit represents a Setpoint index.

Field	Function
	1b1 - Allows Shutting off RCOSC at the Setpoint index. 1b0 - Does not allow shutting off RCOSC at the Setpoint index.

19.6.2.7 SP0~7 Priority (SP_PRIORITY_0_7)

19.6.2.7.1 Offset

Register	Offset
SP_PRIORITY_0_7	40h

19.6.2.7.2 Function

Larger number means higher priority, locked by LOCK_CFG field

19.6.2.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	SYS_SP7_PRIORITY				SYS_SP6_PRIORITY				SYS_SP5_PRIORITY				SYS_SP4_PRIORITY			
Reset	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	SYS_SP3_PRIORITY				SYS_SP2_PRIORITY				SYS_SP1_PRIORITY				SYS_SP0_PRIORITY			
Reset	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

19.6.2.7.4 Fields

Field	Function
31-28 SYS_SP7_PRIORITY	priority of Setpoint 7
27-24 SYS_SP6_PRIORITY	priority of Setpoint 6
23-20 SYS_SP5_PRIORITY	priority of Setpoint 5
19-16 SYS_SP4_PRIORITY	priority of Setpoint 4
15-12 SYS_SP3_PRIORITY	priority of Setpoint 3
11-8 SYS_SP2_PRIORITY	priority of Setpoint 2
7-4 SYS_SP1_PRIORITY	priority of Setpoint 1
3-0 SYS_SP0_PRIORITY	priority of Setpoint 0

19.6.2.8 SP8~15 Priority (SP_PRIORITY_8_15)

19.6.2.8.1 Offset

Register	Offset
SP_PRIORITY_8_15	44h

19.6.2.8.2 Function

Larger number means higher priority, locked by LOCK_CFG field

19.6.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYS_SP15_PRIORITY				SYS_SP14_PRIORITY				SYS_SP13_PRIORITY				SYS_SP12_PRIORITY			
W																
Reset	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYS_SP11_PRIORITY				SYS_SP10_PRIORITY				SYS_SP9_PRIORITY				SYS_SP8_PRIORITY			
W																
Reset	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0

19.6.2.8.4 Fields

Field	Function
31-28 SYS_SP15_PRIORITY	priority of Setpoint 15
27-24 SYS_SP14_PRIORITY	priority of Setpoint 14
23-20 SYS_SP13_PRIORITY	priority of Setpoint 13
19-16 SYS_SP12_PRIORITY	priority of Setpoint 12
15-12 SYS_SP11_PRIORITY	priority of Setpoint 11
11-8	priority of Setpoint 10

Table continues on the next page...

Field	Function
SYS_SP10_PRIORITY	
7-4 SYS_SP9_PRIORITY	priority of Setpoint 9
3-0 SYS_SP8_PRIORITY	priority of Setpoint 8

19.6.2.9 SP SSAR save control (SP_SSAR_SAVE_CTRL)

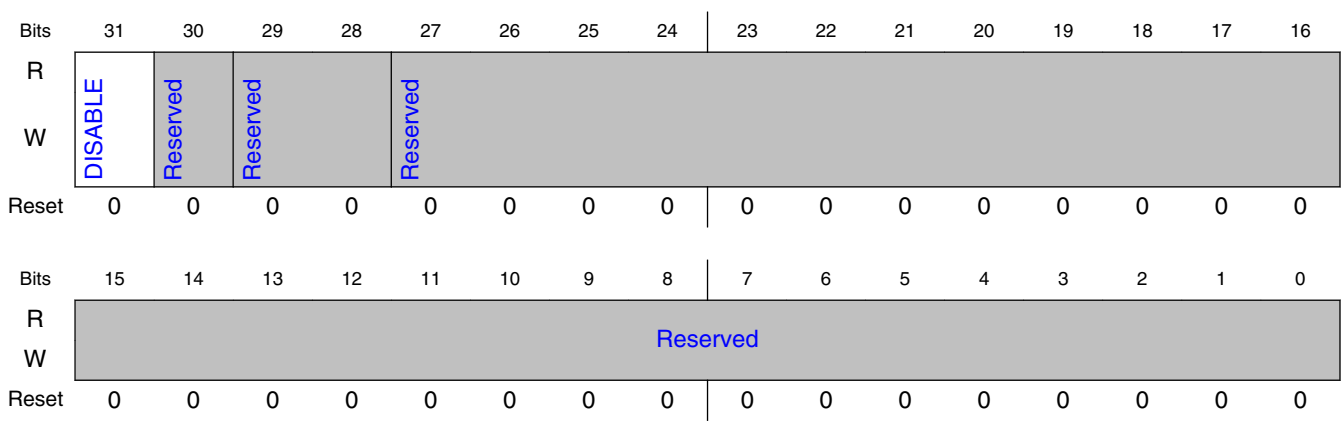
19.6.2.9.1 Offset

Register	Offset
SP_SSAR_SAVE_CTRL	100h

19.6.2.9.2 Function

Locked by LOCK_CFG field.

19.6.2.9.3 Diagram



19.6.2.9.4 Fields

Field	Function
31	Disable this step

Table continues on the next page...

Field	Function
DISABLE	When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.10 SP LPCG off control (SP_LPCG_OFF_CTRL)

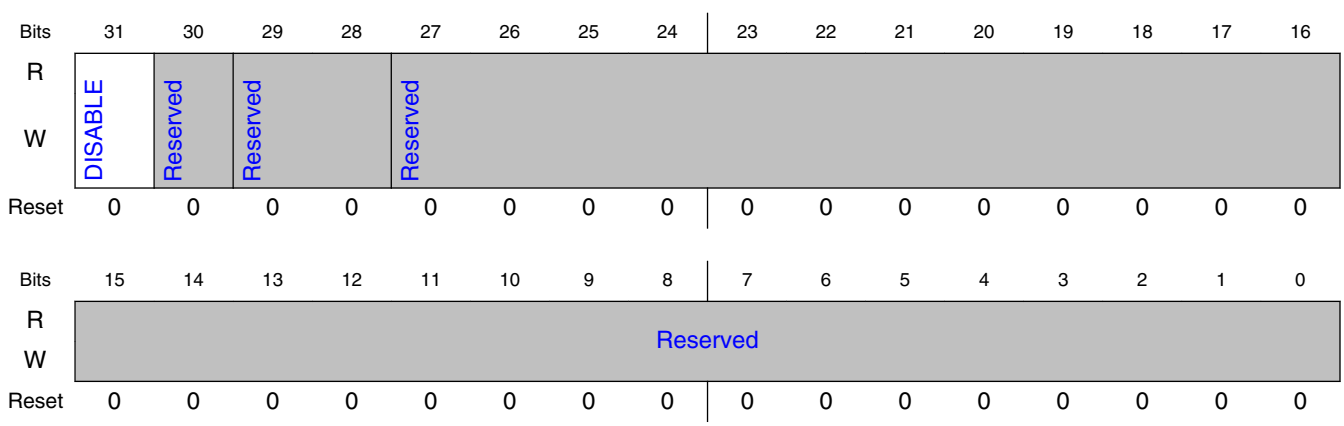
19.6.2.10.1 Offset

Register	Offset
SP_LPCG_OFF_CTRL	110h

19.6.2.10.2 Function

Locked by LOCK_CFG field.

19.6.2.10.3 Diagram



19.6.2.10.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.11 SP group down control (SP_GROUP_DOWN_CTRL)

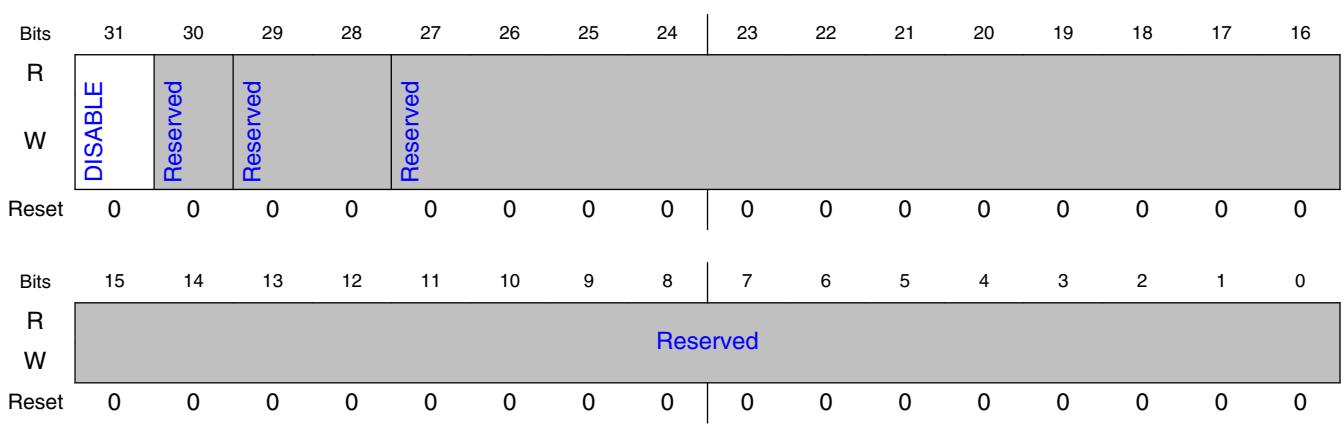
19.6.2.11.1 Offset

Register	Offset
SP_GROUP_DOWN_CTRL	120h

19.6.2.11.2 Function

Locked by LOCK_CFG field.

19.6.2.11.3 Diagram



19.6.2.11.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.12 SP root down control (SP_ROOT_DOWN_CTRL)

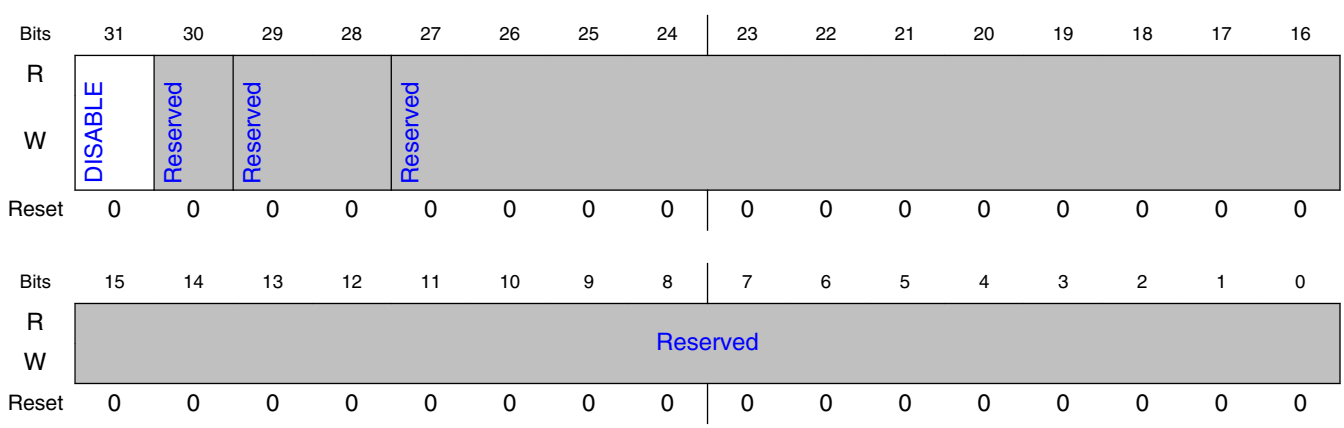
19.6.2.12.1 Offset

Register	Offset
SP_ROOT_DOWN_CTRL	130h

19.6.2.12.2 Function

Locked by LOCK_CFG field.

19.6.2.12.3 Diagram



19.6.2.12.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.13 SP PLL off control (SP_PLL_OFF_CTRL)

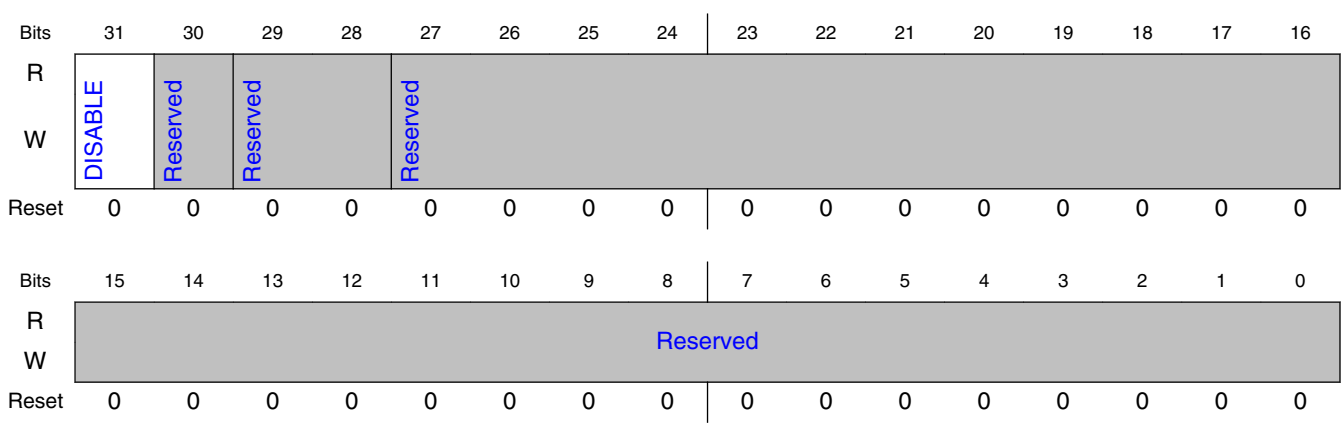
19.6.2.13.1 Offset

Register	Offset
SP_PLL_OFF_CTRL	140h

19.6.2.13.2 Function

Locked by LOCK_CFG field.

19.6.2.13.3 Diagram



19.6.2.13.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.14 SP ISO on control (SP_ISO_ON_CTRL)

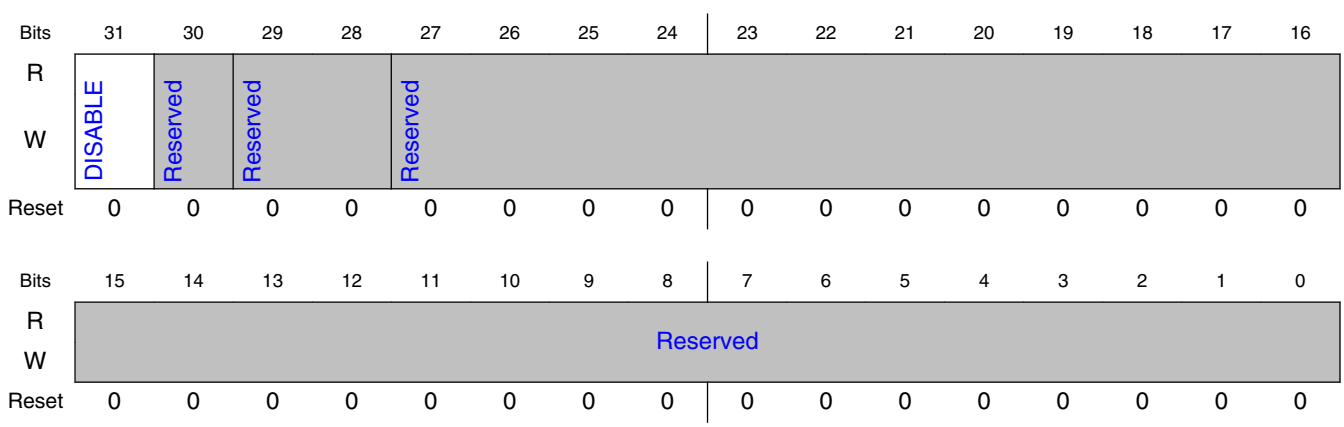
19.6.2.14.1 Offset

Register	Offset
SP_ISO_ON_CTRL	150h

19.6.2.14.2 Function

Locked by LOCK_CFG field.

19.6.2.14.3 Diagram



19.6.2.14.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.15 SP reset early control (SP_RESET_EARLY_CTRL)

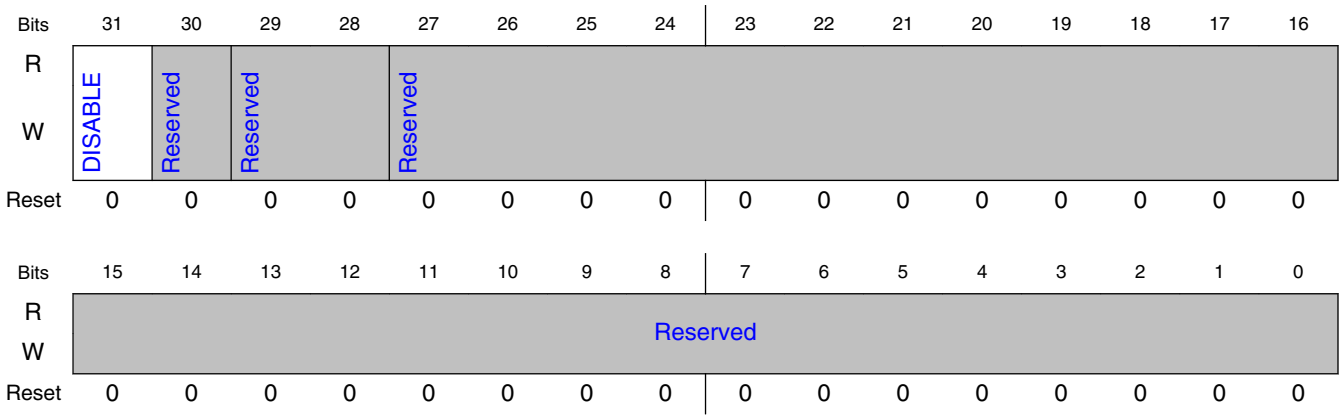
19.6.2.15.1 Offset

Register	Offset
SP_RESET_EARLY_CTRL	160h

19.6.2.15.2 Function

Locked by LOCK_CFG field.

19.6.2.15.3 Diagram



19.6.2.15.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.16 SP power off control (SP_POWER_OFF_CTRL)

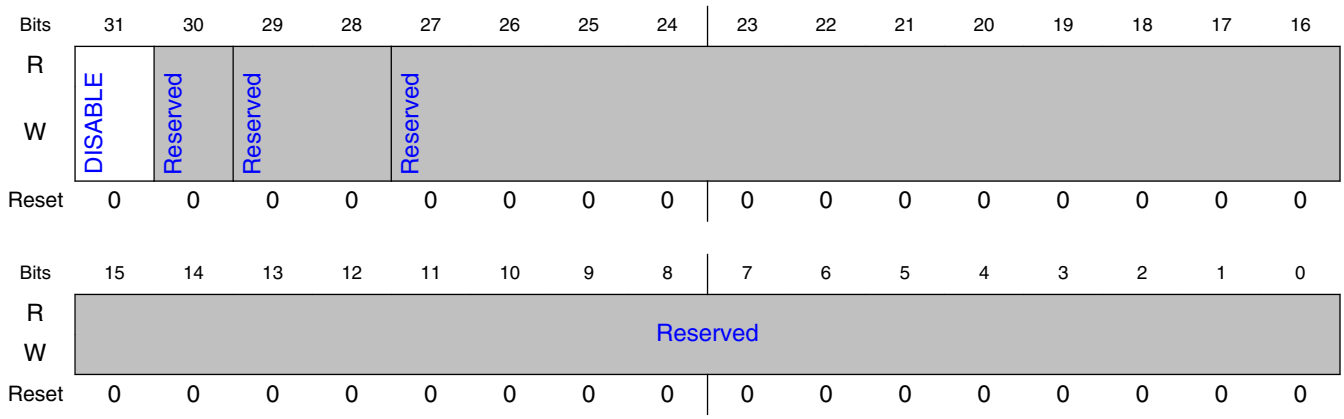
19.6.2.16.1 Offset

Register	Offset
SP_POWER_OFF_CTRL	170h

19.6.2.16.2 Function

Locked by LOCK_CFG field.

19.6.2.16.3 Diagram



19.6.2.16.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.17 SP bias off control (SP_BIAS_OFF_CTRL)

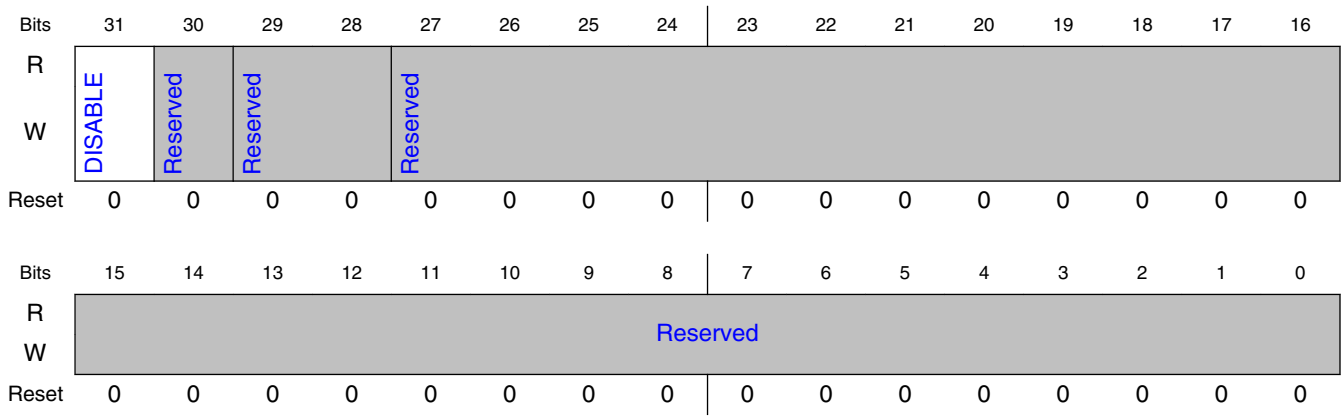
19.6.2.17.1 Offset

Register	Offset
SP_BIAS_OFF_CTRL	180h

19.6.2.17.2 Function

Locked by LOCK_CFG field.

19.6.2.17.3 Diagram



19.6.2.17.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.18 SP bandgap and PLL_LDO off control (SP_BG_PLDO_OFF_CTRL)

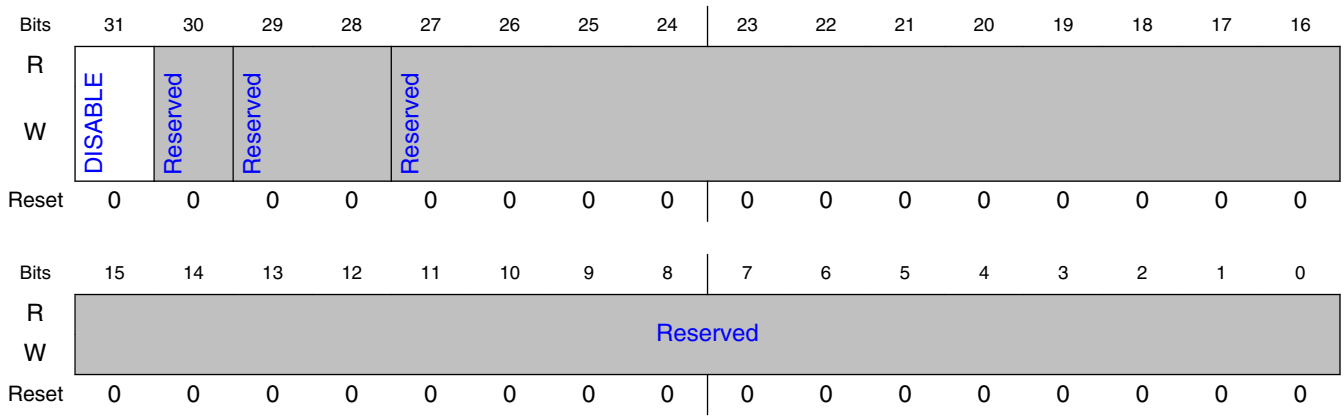
19.6.2.18.1 Offset

Register	Offset
SP_BG_PLDO_OFF_CTRL	190h

19.6.2.18.2 Function

Locked by LOCK_CFG field.

19.6.2.18.3 Diagram



19.6.2.18.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.19 SP LDO pre control (SP_LDO_PRE_CTRL)

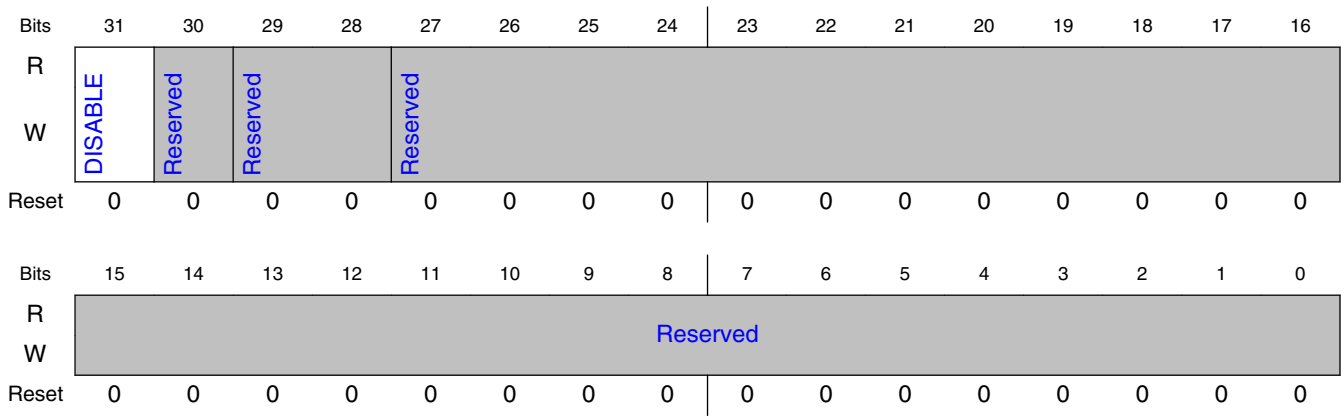
19.6.2.19.1 Offset

Register	Offset
SP_LDO_PRE_CTRL	1A0h

19.6.2.19.2 Function

Locked by LOCK_CFG field.

19.6.2.19.3 Diagram



19.6.2.19.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.20 SP DCDC down control (SP_DCDC_DOWN_CTRL)

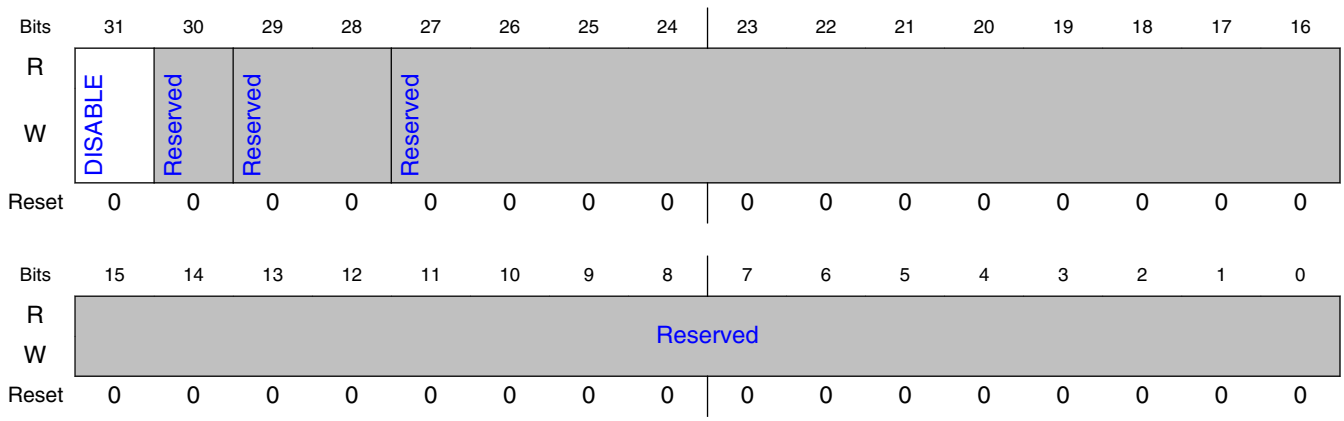
19.6.2.20.1 Offset

Register	Offset
SP_DCDC_DOWN_CTRL	1B0h

19.6.2.20.2 Function

Locked by LOCK_CFG field.

19.6.2.20.3 Diagram



19.6.2.20.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.21 SP DCDC up control (SP_DCDC_UP_CTRL)

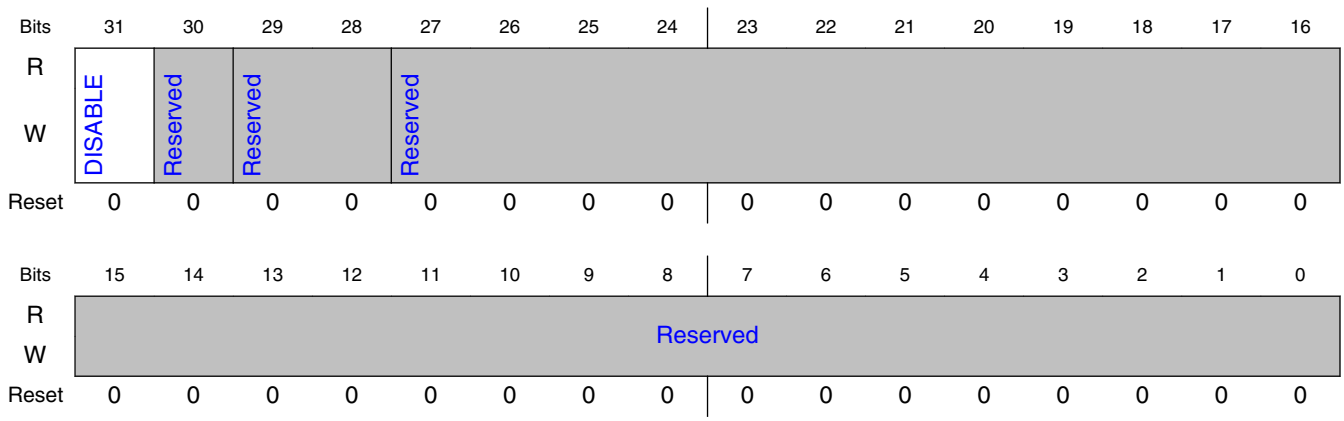
19.6.2.21.1 Offset

Register	Offset
SP_DCDC_UP_CTRL	200h

19.6.2.21.2 Function

Locked by LOCK_CFG field.

19.6.2.21.3 Diagram



19.6.2.21.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.22 SP LDO post control (SP_LDO_POST_CTRL)

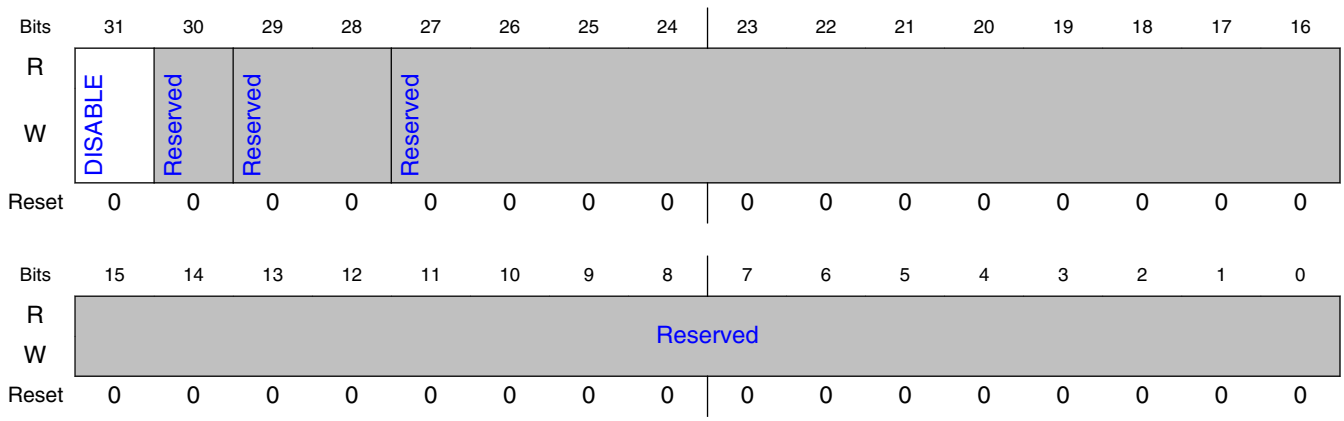
19.6.2.22.1 Offset

Register	Offset
SP_LDO_POST_CTRL	210h

19.6.2.22.2 Function

Locked by LOCK_CFG field.

19.6.2.22.3 Diagram



19.6.2.22.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.23 SP bandgap and PLL_LDO on control (SP_BG_PLDO_ON_CTRL)

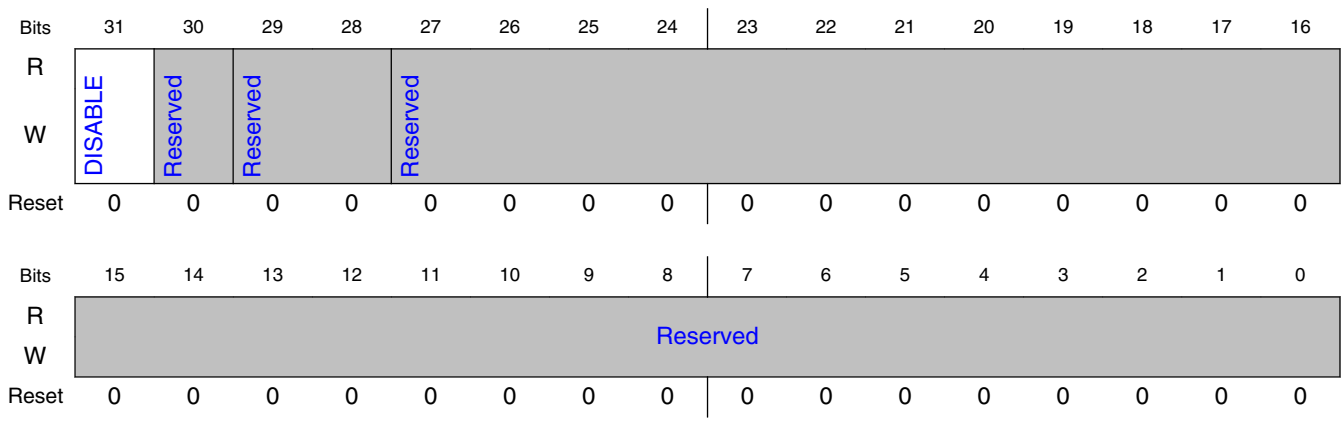
19.6.2.23.1 Offset

Register	Offset
SP_BG_PLDO_ON_CTRL	220h

19.6.2.23.2 Function

Locked by LOCK_CFG field.

19.6.2.23.3 Diagram



19.6.2.23.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.24 SP bias on control (SP_BIAS_ON_CTRL)

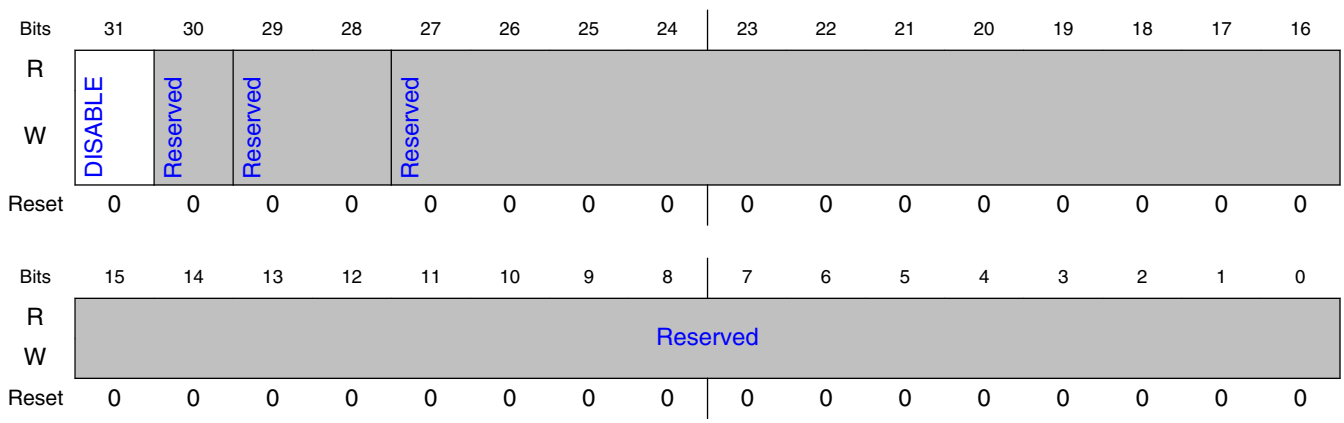
19.6.2.24.1 Offset

Register	Offset
SP_BIAS_ON_CTRL	230h

19.6.2.24.2 Function

Locked by LOCK_CFG field.

19.6.2.24.3 Diagram



19.6.2.24.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.25 SP power on control (SP_POWER_ON_CTRL)

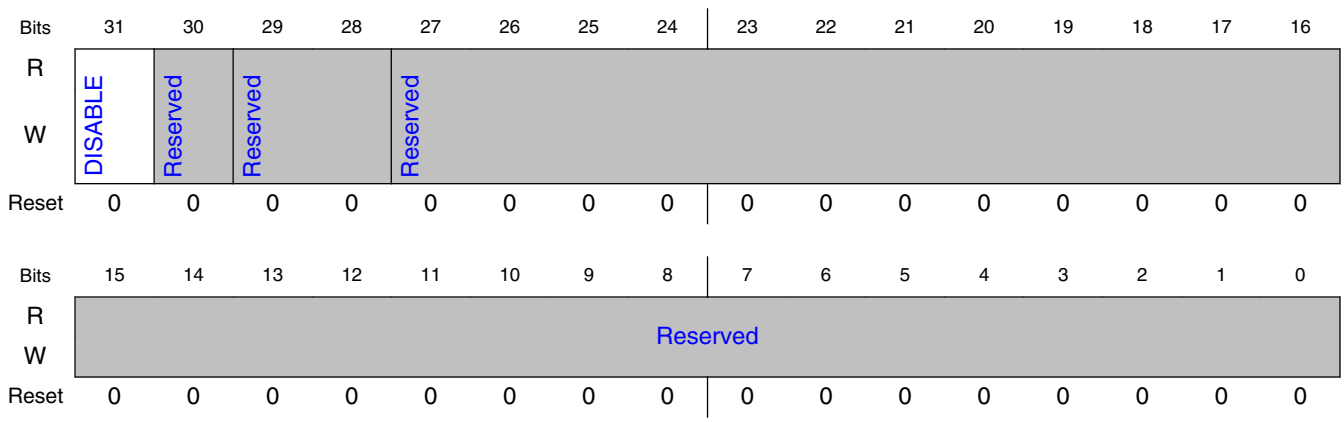
19.6.2.25.1 Offset

Register	Offset
SP_POWER_ON_CTRL	240h

19.6.2.25.2 Function

Locked by LOCK_CFG field.

19.6.2.25.3 Diagram



19.6.2.25.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.26 SP reset late control (SP_RESET_LATE_CTRL)

19.6.2.26.1 Offset

Register	Offset
SP_RESET_LATE_CTRL	250h

19.6.2.26.2 Function

Locked by LOCK_CFG field.

19.6.2.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	Reserved		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.2.26.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.27 SP ISO off control (SP_ISO_OFF_CTRL)

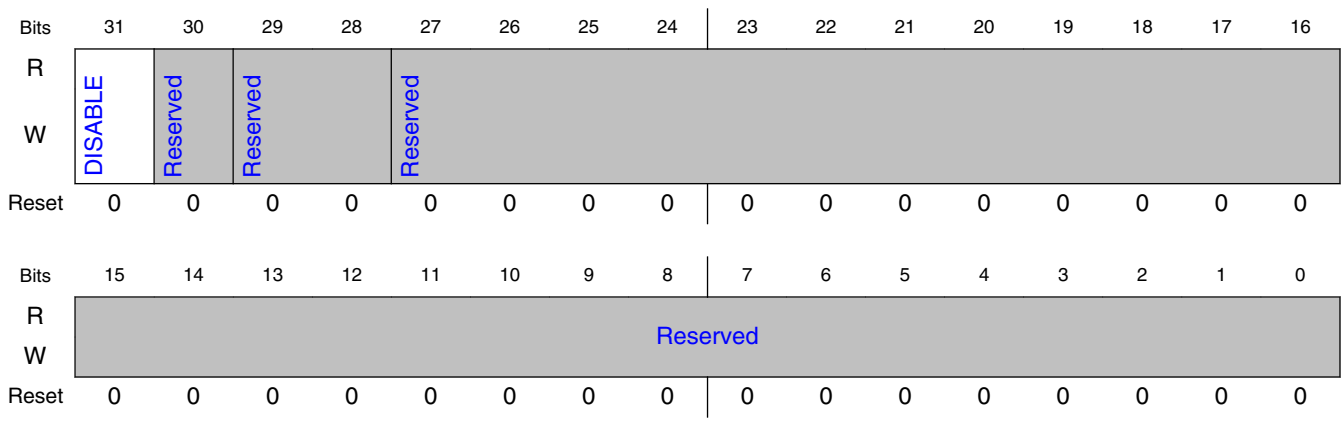
19.6.2.27.1 Offset

Register	Offset
SP_ISO_OFF_CTRL	260h

19.6.2.27.2 Function

Locked by LOCK_CFG field.

19.6.2.27.3 Diagram



19.6.2.27.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.28 SP PLL on control (SP_PLL_ON_CTRL)

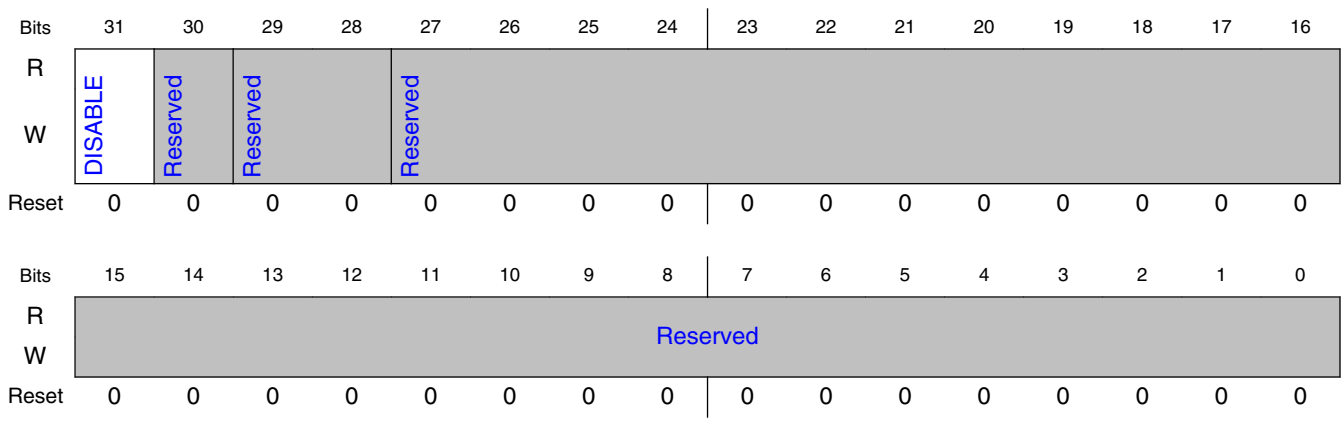
19.6.2.28.1 Offset

Register	Offset
SP_PLL_ON_CTRL	270h

19.6.2.28.2 Function

Locked by LOCK_CFG field.

19.6.2.28.3 Diagram



19.6.2.28.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.29 SP root up control (SP_ROOT_UP_CTRL)

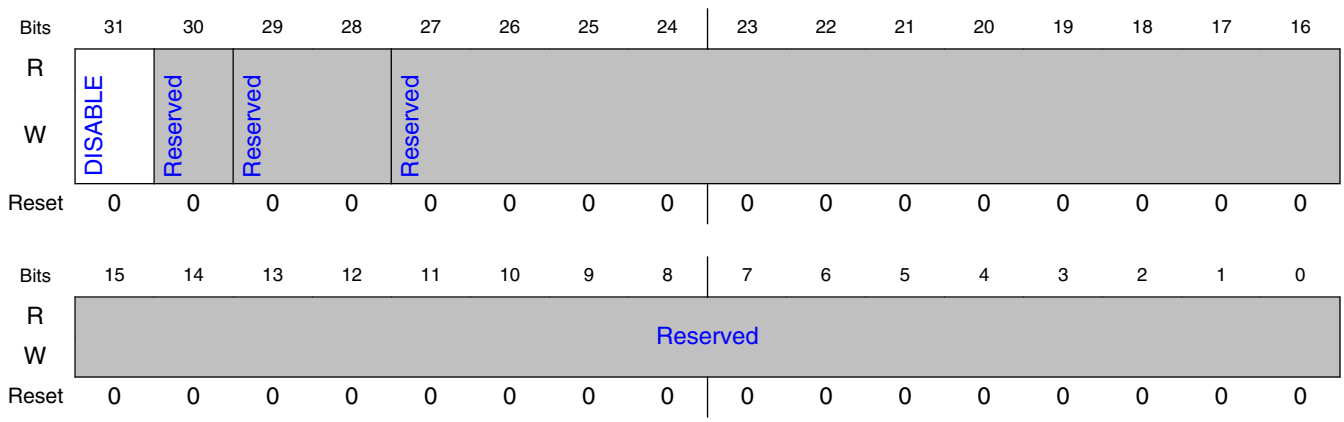
19.6.2.29.1 Offset

Register	Offset
SP_ROOT_UP_CTRL	280h

19.6.2.29.2 Function

Locked by LOCK_CFG field.

19.6.2.29.3 Diagram



19.6.2.29.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.30 SP group up control (SP_GROUP_UP_CTRL)

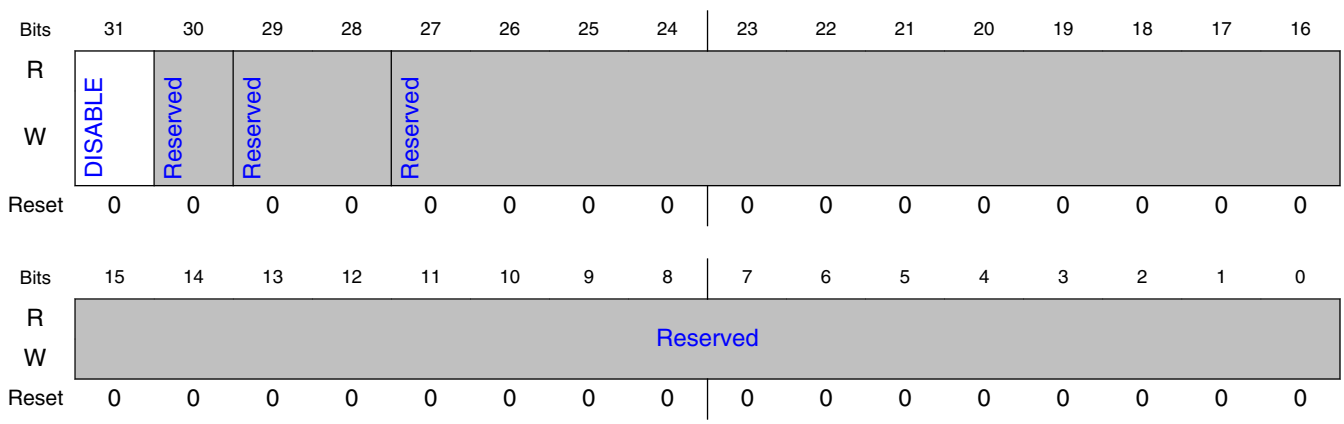
19.6.2.30.1 Offset

Register	Offset
SP_GROUP_UP_CTRL	290h

19.6.2.30.2 Function

Locked by LOCK_CFG field.

19.6.2.30.3 Diagram



19.6.2.30.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.31 SP LPCG on control (SP_LPCG_ON_CTRL)

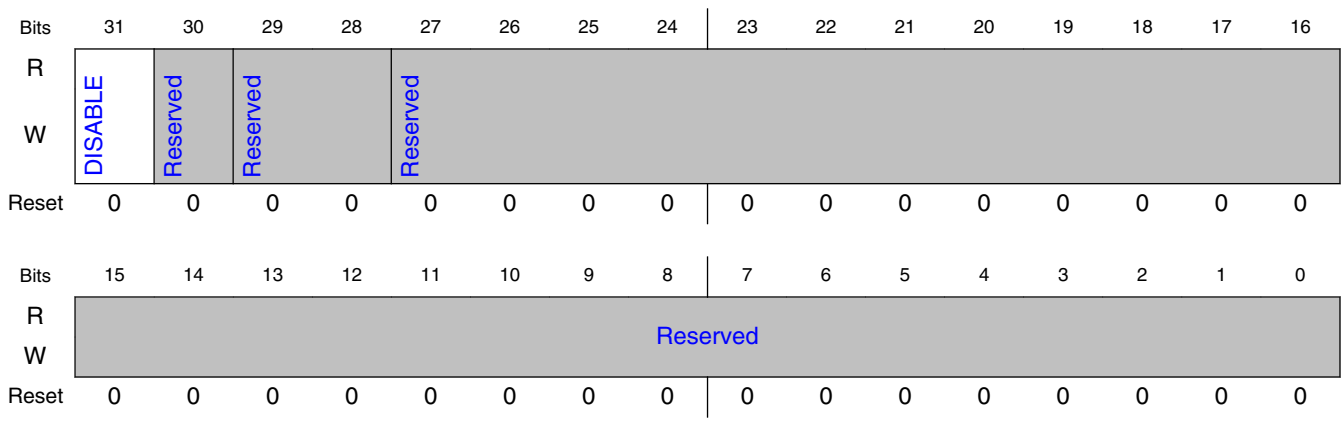
19.6.2.31.1 Offset

Register	Offset
SP_LPCG_ON_CTRL	2A0h

19.6.2.31.2 Function

Locked by LOCK_CFG field.

19.6.2.31.3 Diagram



19.6.2.31.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.2.32 SP SSAR restore control (SP_SSAR_RESTORE_CTRL)

19.6.2.32.1 Offset

Register	Offset
SP_SSAR_RESTORE_CTRL	2B0h

19.6.2.32.2 Function

Locked by LOCK_CFG field.

19.6.2.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	Reserved		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.2.32.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3 GPC_STBY register descriptions

19.6.3.1 gpc_stby_ctrl memory map

GPC_STBY_CTRL base address: 40C0_2800h

Offset	Register	Width (In bits)	Access	Reset value
4h	Standby Authentication Control (STBY_AUTHEN_CTRL)	32	RW	0000_0000h
Ch	STBY Misc (STBY_MISC)	32	RW	0000_0000h
F0h	STBY lpcg_in control (STBY_LPCG_IN_CTRL)	32	RW	0000_0000h
100h	STBY pll_in control (STBY_PLL_IN_CTRL)	32	RW	0000_0000h
110h	STBY bias_in control (STBY_BIAS_IN_CTRL)	32	RW	0000_0000h
120h	STBY pldo_in control (STBY_PLDO_IN_CTRL)	32	RW	0000_0000h
128h	STBY bandgap_in control (STBY_BANDGAP_IN_CTRL)	32	RW	0000_0000h
130h	STBY ldo_in control (STBY_LDO_IN_CTRL)	32	RW	0000_0000h
140h	STBY dcdc_in control (STBY_DCDC_IN_CTRL)	32	RW	0000_0000h
150h	STBY PMIC in control (STBY_PMIC_IN_CTRL)	32	RW	0000_0000h
200h	STBY PMIC out control (STBY_PMIC_OUT_CTRL)	32	RW	0000_0004h
210h	STBY DCDC out control (STBY_DCDC_OUT_CTRL)	32	RW	0000_0000h
220h	STBY LDO out control (STBY_LDO_OUT_CTRL)	32	RW	0000_0000h
230h	STBY bandgap out control (STBY_BANDGAP_OUT_CTRL)	32	RW	0000_0000h
238h	STBY pldo out control (STBY_PLDO_OUT_CTRL)	32	RW	0000_0000h
240h	STBY bias out control (STBY_BIAS_OUT_CTRL)	32	RW	0000_0000h
250h	STBY PLL out control (STBY_PLL_OUT_CTRL)	32	RW	0000_0000h
260h	STBY LPCG out control (STBY_LPCG_OUT_CTRL)	32	RW	0000_0000h

19.6.3.2 Standby Authentication Control (STBY_AUTHEN_CTRL)

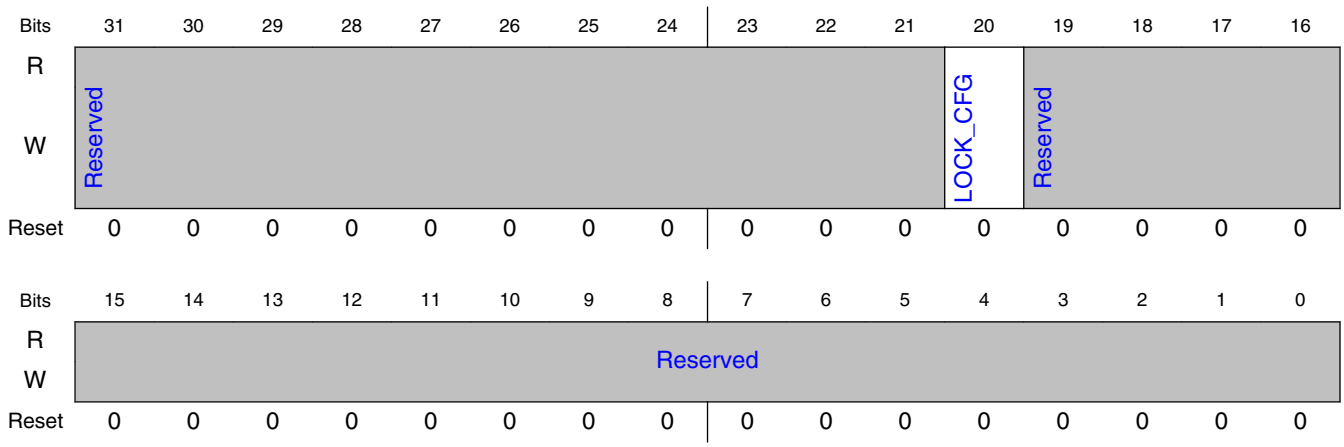
19.6.3.2.1 Offset

Register	Offset
STBY_AUTHEN_CTRL	4h

19.6.3.2.2 Function

Authentication control for standby controller.

19.6.3.2.3 Diagram



19.6.3.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields
19-0 —	Reserved

19.6.3.3 STBY Misc (STBY_MISC)

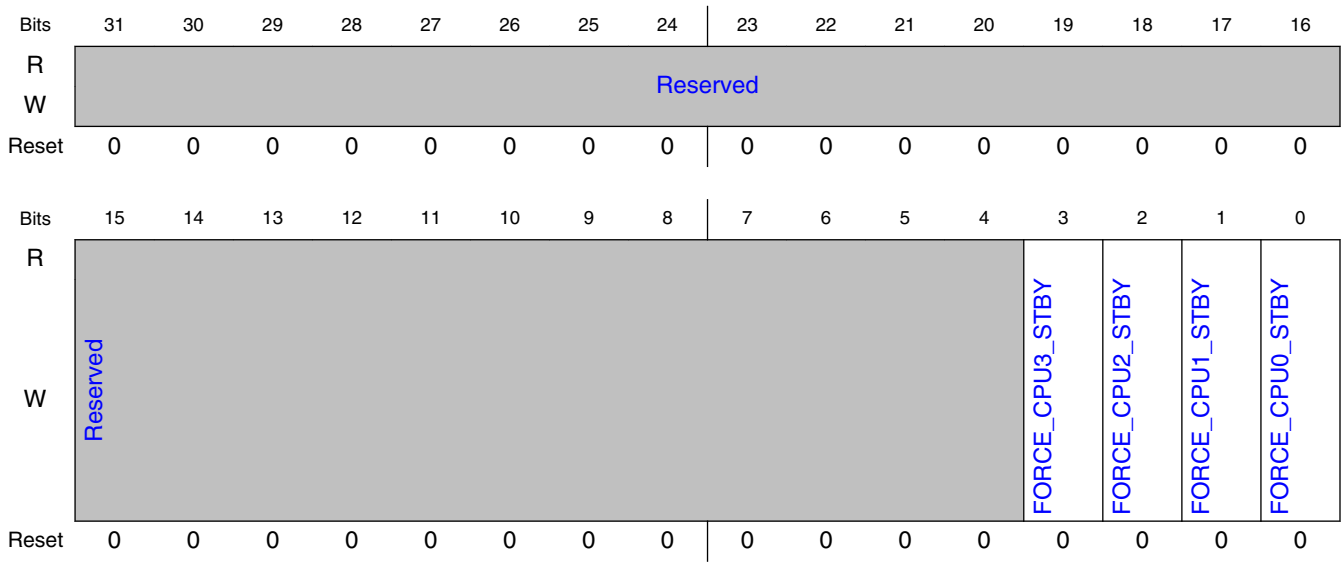
19.6.3.3.1 Offset

Register	Offset
STBY_MISC	Ch

19.6.3.3.2 Function

Miscellaneous register.

19.6.3.3.3 Diagram



19.6.3.3.4 Fields

Field	Function
31-4 —	Reserved
3 FORCE_CPU3_STBY	Force CPU3 requesting standby mode If CPU3 exists but is not used and is always in reset status, users can set this bit to force CPU3 requesting standby mode
2 FORCE_CPU2_STBY	Force CPU2 requesting standby mode If CPU2 exists but is not used and is always in reset status, users can set this bit to force CPU2 requesting standby mode
1 FORCE_CPU1_STBY	Force CPU0 requesting standby mode If CPU1 exists but is not used and is always in reset status, users can set this bit to force CPU1 requesting standby mode
0 FORCE_CPU0_STBY	Force CPU0 requesting standby mode If CPU0 exists but is not used and is always in reset status, users can set this bit to force CPU0 requesting standby mode

19.6.3.4 STBY lpcg_in control (STBY_LPCG_IN_CTRL)

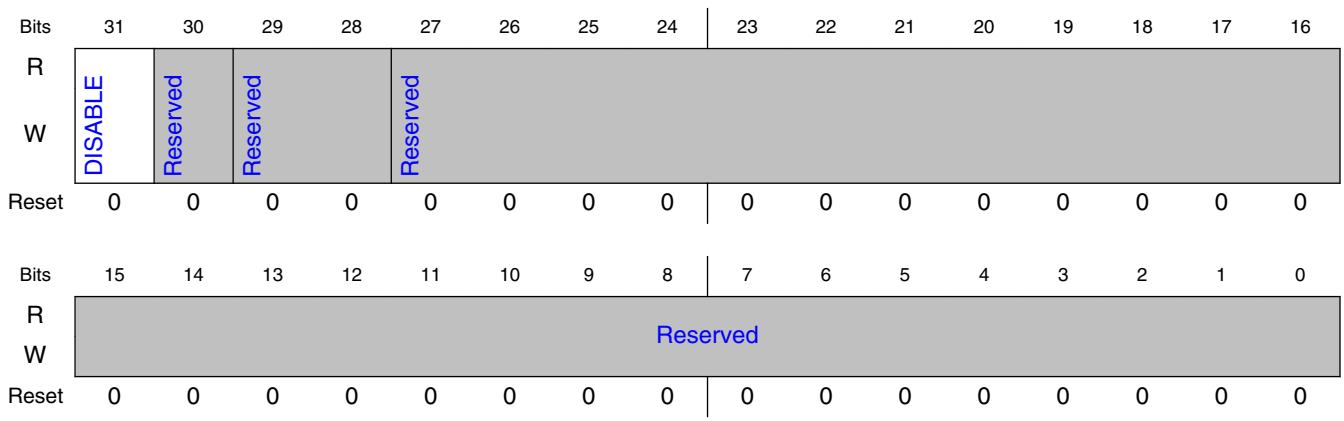
19.6.3.4.1 Offset

Register	Offset
STBY_LPCG_IN_CTRL	F0h

19.6.3.4.2 Function

LPCG_IN step control.

19.6.3.4.3 Diagram



19.6.3.4.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.5 STBY pll_in control (STBY_PLL_IN_CTRL)

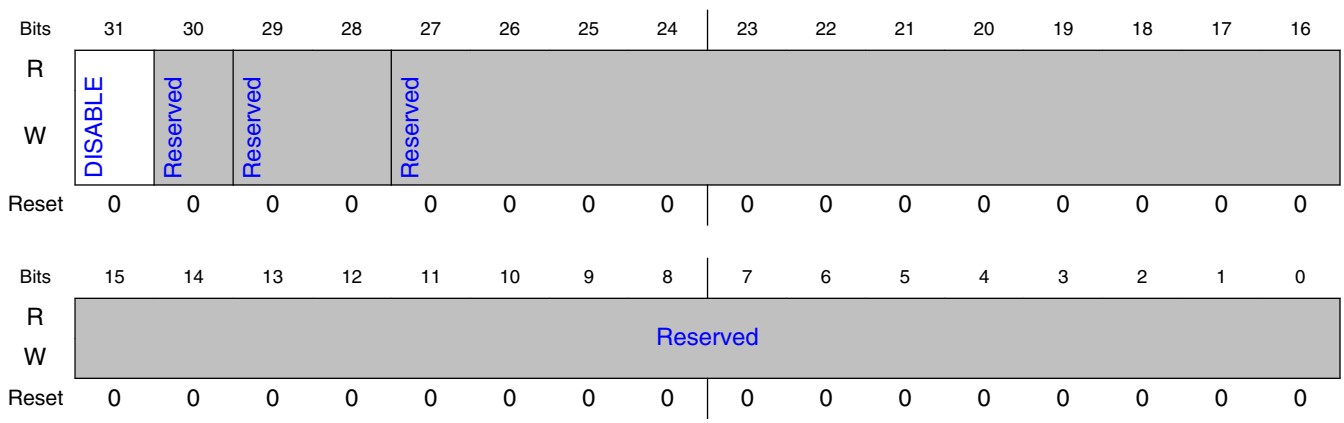
19.6.3.5.1 Offset

Register	Offset
STBY_PLL_IN_CTRL	100h

19.6.3.5.2 Function

PLL_IN step control.

19.6.3.5.3 Diagram



19.6.3.5.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.6 STBY bias_in control (STBY_BIAS_IN_CTRL)

19.6.3.6.1 Offset

Register	Offset
STBY_BIAS_IN_CTRL	110h

19.6.3.6.2 Function

BIAS_IN step control.

19.6.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	Reserved		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.3.6.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.7 STBY pldo_in control (STBY_PLDO_IN_CTRL)

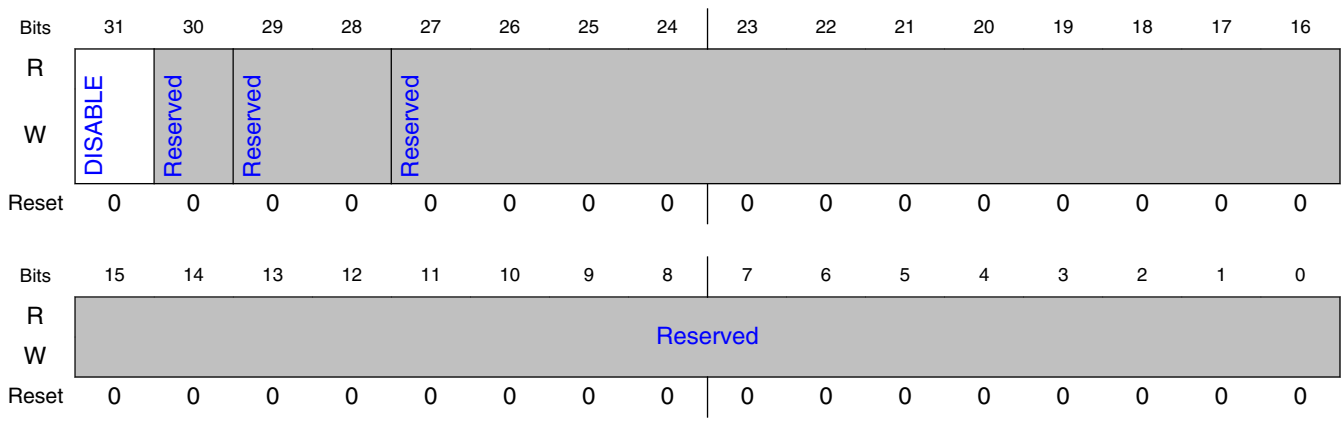
19.6.3.7.1 Offset

Register	Offset
STBY_PLDO_IN_CTRL	120h

19.6.3.7.2 Function

PLDO_IN step control.

19.6.3.7.3 Diagram



19.6.3.7.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.8 STBY bandgap_in control (STBY_BANDGAP_IN_CTRL)

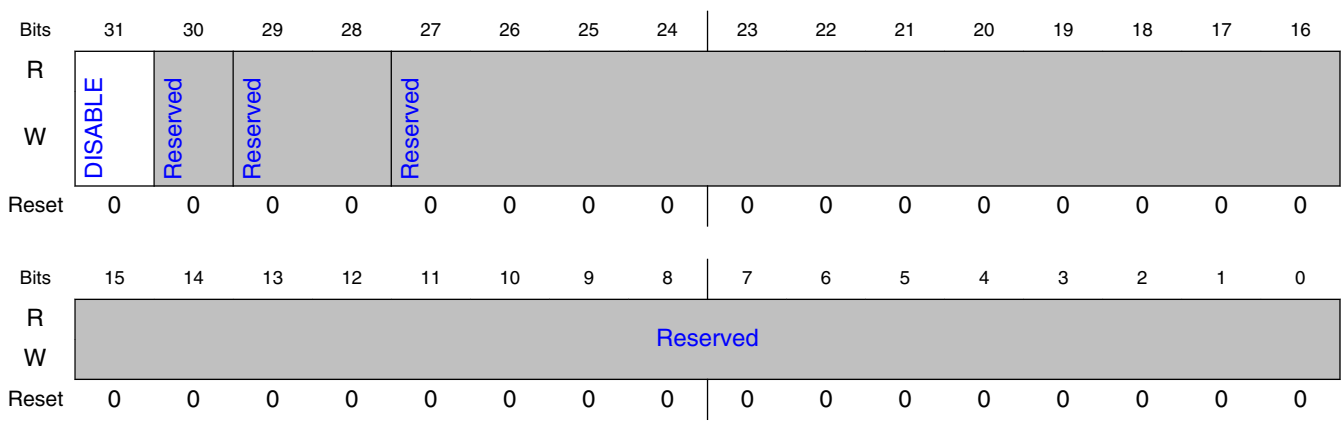
19.6.3.8.1 Offset

Register	Offset
STBY_BANDGAP_IN_CTRL	128h

19.6.3.8.2 Function

BANDGAP_IN step control.

19.6.3.8.3 Diagram



19.6.3.8.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.9 STBY ldo_in control (STBY_LDO_IN_CTRL)

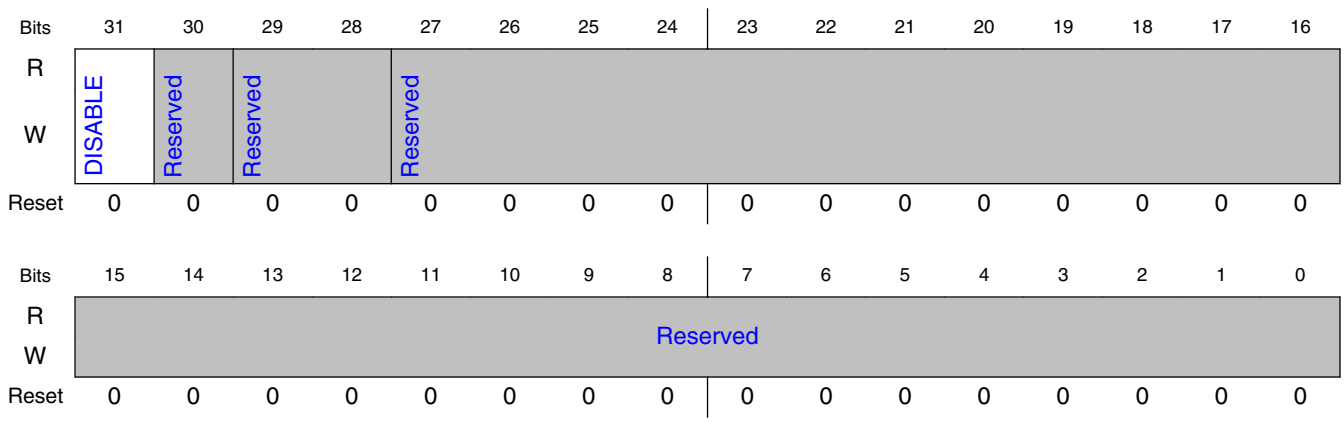
19.6.3.9.1 Offset

Register	Offset
STBY_LDO_IN_CTRL	130h

19.6.3.9.2 Function

LDO_IN step control.

19.6.3.9.3 Diagram



19.6.3.9.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.10 STBY dcdc_in control (STBY_DCDC_IN_CTRL)

19.6.3.10.1 Offset

Register	Offset
STBY_DCDC_IN_CTRL	140h

19.6.3.10.2 Function

DCDC_IN step control.

19.6.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	Reserved		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.3.10.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.11 STBY PMIC in control (STBY_PMIC_IN_CTRL)

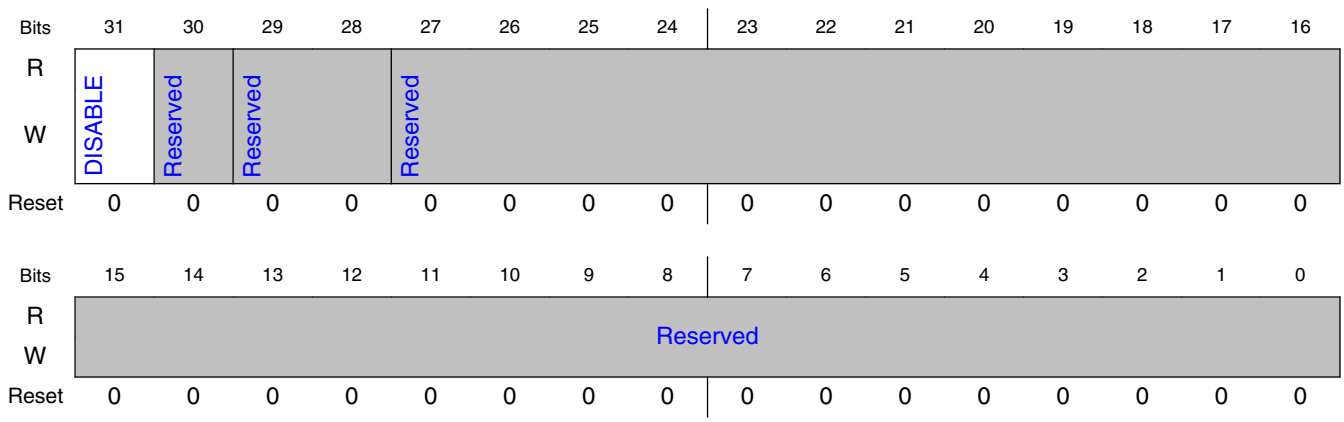
19.6.3.11.1 Offset

Register	Offset
STBY_PMIC_IN_CTRL	150h

19.6.3.11.2 Function

PMIC_IN step control.

19.6.3.11.3 Diagram



19.6.3.11.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.12 STBY PMIC out control (STBY_PMIC_OUT_CTRL)

19.6.3.12.1 Offset

Register	Offset
STBY_PMIC_OUT_CTRL	200h

19.6.3.12.2 Function

PMIC_OUT step control.

19.6.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DISABLE	Reserved	CNT_MODE		Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

19.6.3.12.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 CNT_MODE	Count mode Configure the step counter working mode 00b - Counter disable mode: not use step counter, step completes once receiving step_done 10b - Ignore step_done response, the counter starts to count once step begins, when counter reaches STEP_CNT value, the step completes

Table continues on the next page...

Field	Function
27-16 —	Reserved
15-0 STEP_CNT	Step count, useage is depending on CNT_MODE Must be greater than 2 under CNT_MODE 2 and 3

19.6.3.13 STBY DCDC out control (STBY_DCDC_OUT_CTRL)

19.6.3.13.1 Offset

Register	Offset
STBY_DCDC_OUT_CTRL	210h

19.6.3.13.2 Function

DCDC_OUT step control.

19.6.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DISABLE	Reserved	Reserved		Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.6.3.13.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30	Reserved

Table continues on the next page...

Field	Function
—	
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.14 STBY LDO out control (STBY_LDO_OUT_CTRL)

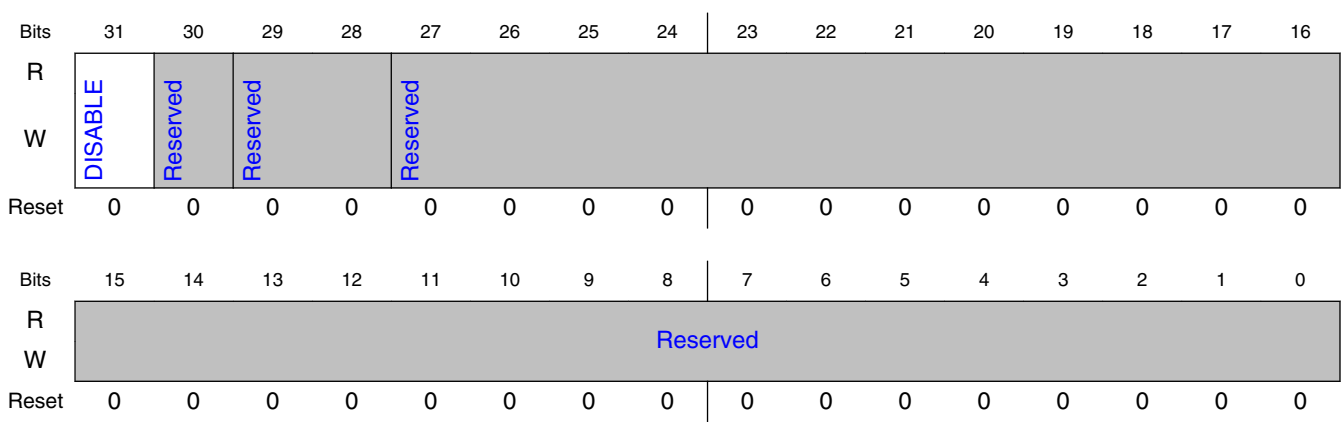
19.6.3.14.1 Offset

Register	Offset
STBY_LDO_OUT_CTRL	220h

19.6.3.14.2 Function

LDO_OUT step control.

19.6.3.14.3 Diagram



19.6.3.14.4 Fields

Field	Function
31	Disable this step

Table continues on the next page...

Field	Function
DISABLE	When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.15 STBY bandgap out control (STBY_BANDGAP_OUT_CTRL)

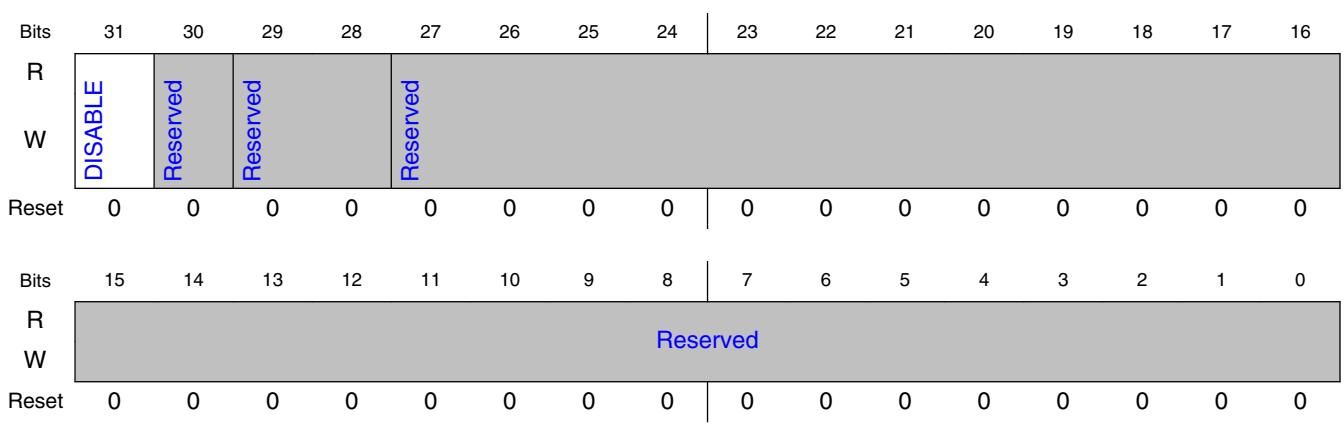
19.6.3.15.1 Offset

Register	Offset
STBY_BANDGAP_OUT_CTRL	230h

19.6.3.15.2 Function

BANDGAP_OUT step control.

19.6.3.15.3 Diagram



19.6.3.15.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.16 STBY pldo out control (STBY_PLDO_OUT_CTRL)

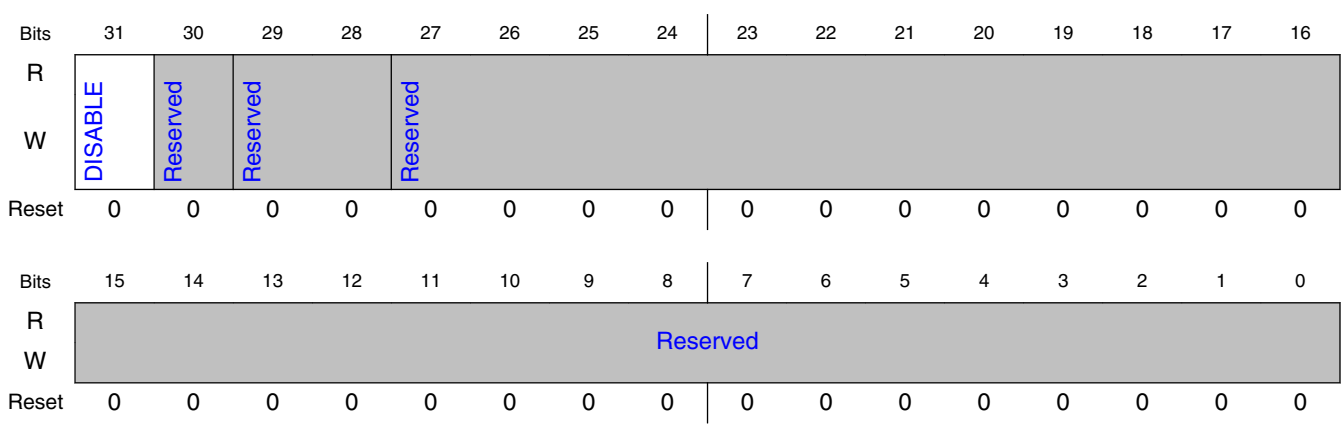
19.6.3.16.1 Offset

Register	Offset
STBY_PLDO_OUT_CTRL	238h

19.6.3.16.2 Function

PLDO_OUT step control.

19.6.3.16.3 Diagram



19.6.3.16.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.17 STBY bias out control (STBY_BIAS_OUT_CTRL)

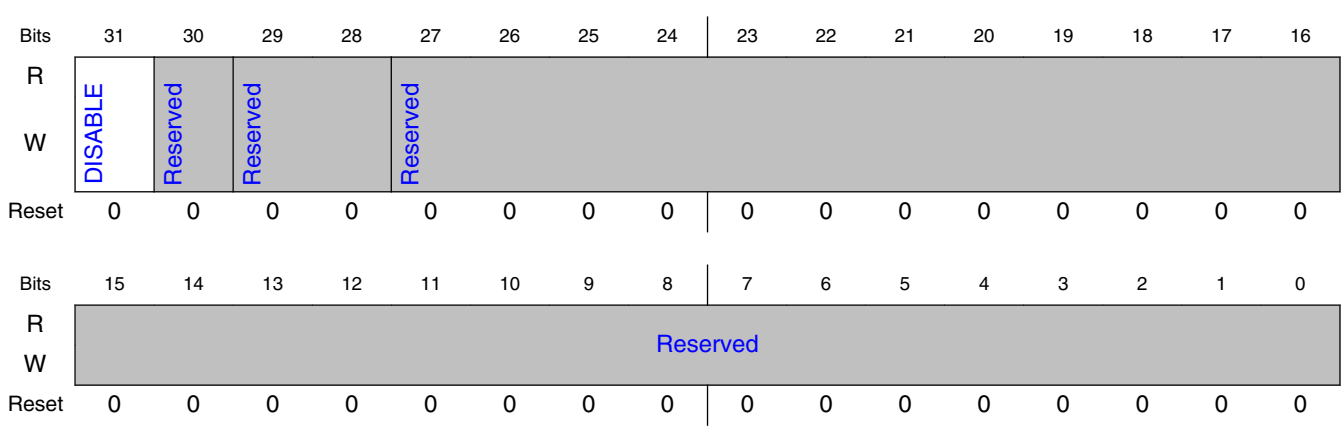
19.6.3.17.1 Offset

Register	Offset
STBY_BIAS_OUT_CTRL	240h

19.6.3.17.2 Function

BIAS_OUT step control.

19.6.3.17.3 Diagram



19.6.3.17.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.18 STBY PLL out control (STBY_PLL_OUT_CTRL)

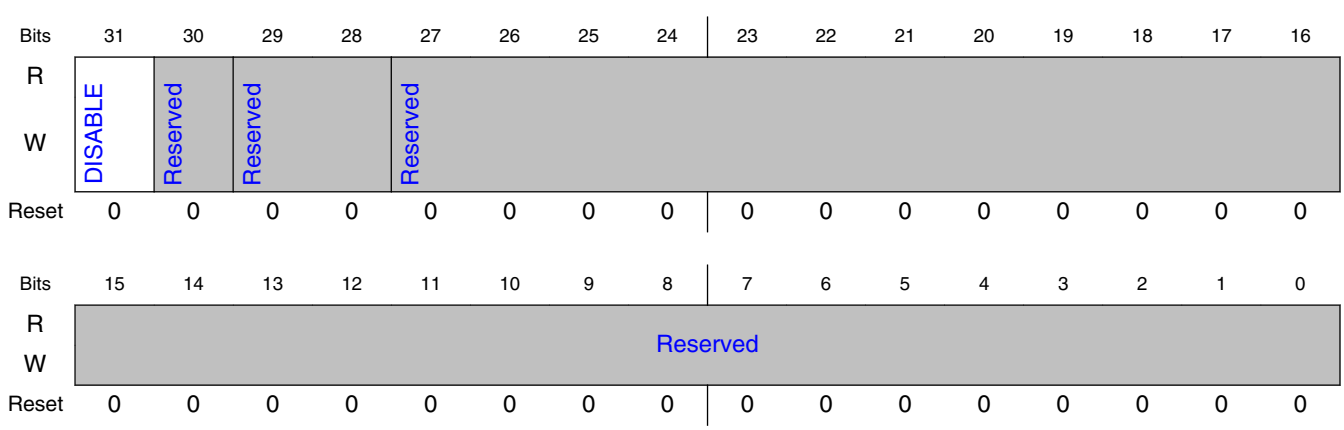
19.6.3.18.1 Offset

Register	Offset
STBY_PLL_OUT_CTRL	250h

19.6.3.18.2 Function

PLL_OUT step control.

19.6.3.18.3 Diagram



19.6.3.18.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

19.6.3.19 STBY LPCG out control (STBY_LPCG_OUT_CTRL)

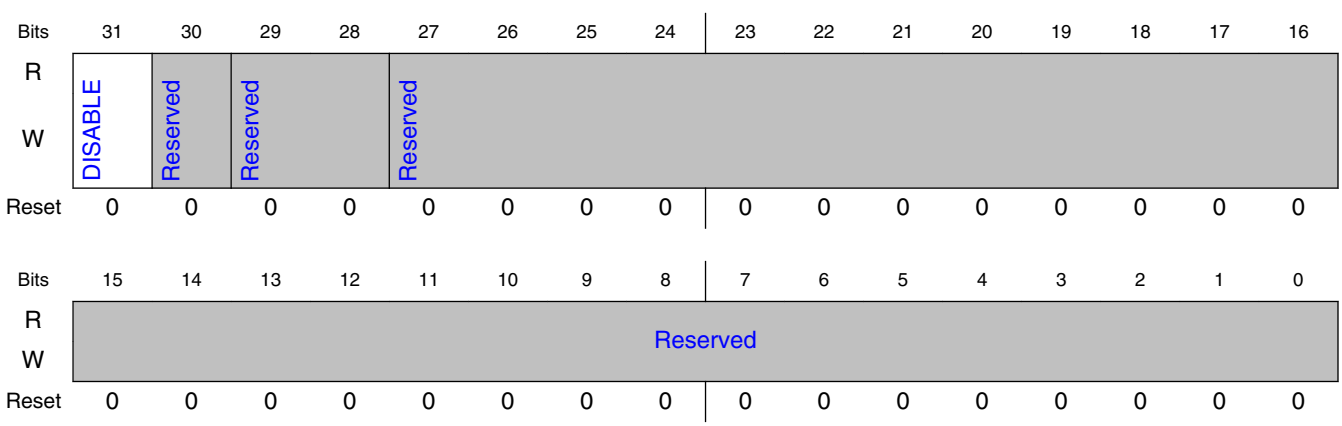
19.6.3.19.1 Offset

Register	Offset
STBY_LPCG_OUT_CTRL	260h

19.6.3.19.2 Function

LPCG_OUT step control.

19.6.3.19.3 Diagram



19.6.3.19.4 Fields

Field	Function
31 DISABLE	Disable this step When setting to 1, GPC will skip this step and not send any request
30 —	Reserved
29-28 —	Reserved
27-16 —	Reserved
15-0 —	Reserved

Chapter 20
Power Gating and Memory Controller (PGMC)

20.1 Chip-specific Power Gating and Memory Controller (PGMC) information

Table 20-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

20.2 Overview

The Power Gating and Memory Controller (PGMC) is a power management component that controls power gating of power domains and memory low power mode. This module is similar to a resource controller based on Unified Power Management Architecture (UPMA). The PGMC uses the Unified Power Management Interface (UPI) to control power switch, state save/restore, PMIC standby, and memory low power mode according to CPU mode, Setpoints, and standby status.

20.2.1 Block Diagram

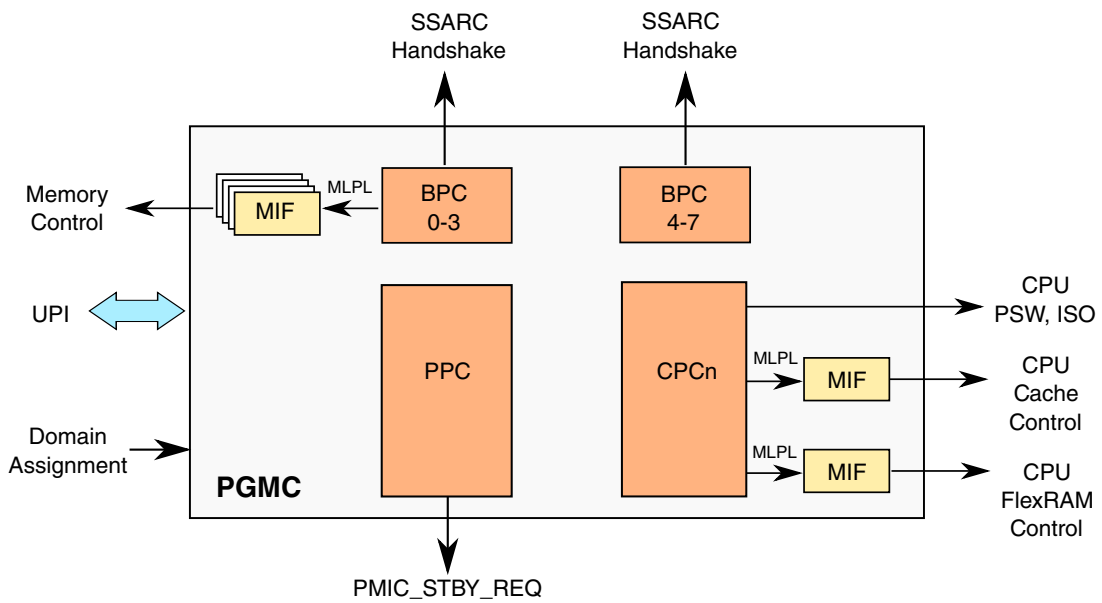


Figure 20-1. PGMC block diagram

There are three submodules for the PGMC block. They are:

- Basic Power Controller (BPC)
  - Controls power domains with simple isolation and power switch
- CPU Power Controller (CPC)
  - Controls CPU platform with complex power domain and sequence requirement
- PMIC Power Controller (PPC)
  - Controls PMIC standby mode outside of the chip

NOTE

The Memory Interface (MIF) block is designed for dedicated technology of memory.

20.2.2 Features

The PGMC includes the following features:

- Automatic power domain control based on UPI sequence
- Software power domain control
- Programmable registers to adjust low power sequence timing
- System domain assignment and restriction
- Programmable memory low power behavior

20.3 Functional description

The following sections describe functional details of the PGMC block.

20.3.1 Power Controller Assignment

The following table shows the power domains of the chip and the BPC it is controlled by. Please see the domain assignment in the clock and power management chapter for more details.

Table 20-2. Power Controller Assignment

Power Domain	Assignment
BPC0	MEGAMIX
BPC1	DISPLAYMIX
BPC2	WAKEUPMIX
BPC3	LPSRMIX
BPC4	MIPIPHY
BPC5	Virtual
BPC6	Virtual
BPC7	Virtual

NOTE

- BPC4 is used for MIPIPHY, which has only power switch and isolation. There is no memory to be controlled.
- Virtual power domains do not physically exist, but are used to trigger SSARC to save and restore, status and data.
- BPC can also control memories inside the target power domain, so each power domain will have a corresponding MIF.

The following table shows the cores controlled by CPCs.

CPC	Assignment
CPC0	CM7
CPC1	CM4

The following table shows the control areas of the MIFs.

MIF	Assignment
MIF0	Cache
MIF1	Local Memory

PPC and it's MIF is used for PMIC.

20.3.2 Control Mode

Resource domain is assigned by RDC or register configuration. For example, one CPU platform and its private clock and private peripherals can be assigned to a same resource domain.

Power domain is the region with same power or same non-interchangeable power supply.

The controllers can be configured to three kinds of control mode via x_MODE[CTRL_MODE] field, where x could be PPC, CPC_CORE, or BPC.

- 2’b00 means disable the low power control
- 2’b01 means the Power Controller belongs to a resource domain and is controlled by the domain’s CPU power mode
- 2’b10 means the Power Controller is a system resource and controlled by Setpoint mode.
- 2'b11 is Reserved

When x_MODE[CTRL_MODE] is 2'b01, the resource domain assignment is set in x_MODE[DOMAIN_ASSIGN] field.

- BPC can be set to be controlled by either CPU mode or Setpoint mode. The domain assignment is register programmable.
- PPC can only be set to Setpoint mode, and is controlled by standby request.
- CPC core can only be controlled by CPU mode. The CPC cache and local memory can be controlled either by CPU mode or Setpoint mode. The domain assignment of all CPC parts are hardware controlled by RDC, and cannot be programmed inside PGMC.

20.3.3 Power on and Power off

The BPC and CPC core can turn on or shut off the power domain. The behavior contains power switch (PSW) and isolation (ISO). The power switch consists of high fan-out PSW, low fan-out PSW, and their acknowledge signal.

- If x_MODE[CTRL_MODE] == 2'b01 (where x could be BPC or CPC_CORE), then x_POWER_CTRL[PWR_OFF_AT_WAIT], x_POWER_CTRL[PWR_OFF_AT_STOP], and x_POWER_CTRL[PWR_OFF_SUSPEND] fields set which CPU mode the power domain should be powered off.
- If x_MODE[CTRL_MODE] == 2'b10, then BPC_POWER_CTRL[PWR_OFF_AT_SP] sets the Setpoint the power domain should be powered off.

The registers x_PDN_CTRL and x_PUP_CTRL, control the delay time before PSW and ISO toggle. They are used to achieve timing sequence requirement. The requirement can be between PSW and ISO, or between different power domains. Please note that the default delay should be sufficient.

Software can use x_POWER_CTRL[ISO_OFF_SOFT], x_POWER_CTRL[ISO_ON_SOFT], x_POWER_CTRL[PSW_OFF_SOFT], and x_POWER_CTRL[PSW_ON_SOFT] fields to directly control the PSW and ISO.

20.3.4 State Save and Restore

When a power domain is powered off, its status is lost. If it happens, the State Save and Restore Controller (SSARC) will automatically save the data before power off and restore it after power on. The BPC sends save and restore request to SSAR and completes the handshake.

BPC controls SSAR requests for their corresponding power domains. The virtual power domains are purposed so other system data can be saved and restored in any CPU mode or Setpoint mode, even if the data is in non-switchable power domains.

The registers BPC_SSAR_SAVE_CTRL and BPC_SSAR_RESTORE_CTRL can control the SSAR request. The register BPC_SSAR_ACK_CTRL can be used to control the delay and wait response behaviour.

20.3.5 Memory Low Power Control

The memory low power control determines the Memory Low Power Level (MLPL) under CPU Mode or Setpoint. Part of the memory low power control is the Memory Interface (MIF), which sends control signal details to memory, according to the MLPL

request. The memory behavior for MLPL is programmable inside MIF, to simplify software design, the MIF register default value can be directly used without any programming.

Table 20-3. MIF Default Configuration

MLPL	description	HS	LS	IG	STDBY	arr off	per off	SLEEP	sw1 off	sw2 off	iso_en	comment
0	reserved											
1	high speed	H	L	L	L	L	L	L	L	H	L	over drive
2	reserved											
3	normal	L	L	L	L	L	L	L	L	H	L	
4	low speed	L	H	L	L	L	L	L	L	H	L	under drive
5	input gating	L	L	H	L	L	L	L	L	H	L	
6	standby	L	L	H	H	L	L	L	L	H	L	low leakage
7	reserved											
8	sleep	L	L	L	L	L	L	H	L	H	L	retention
9	arr on per off	L	L	L	L	L	H	H	L	H	H	retention
10	arr off per on	L	L	L	L	H	L	H	L	H	L	
11	arr off per off	L	L	L	L	H	H	H	L	H	H	data lost
12												
13	sw2	L	L	L	L	L	L	H	H	L	L	low voltage
14	sw2 per off	L	L	L	L	L	H	H	H	L	H	low voltage
15	sw1sw2 off	L	L	L	L	L	L	H	H	H	H	data lost

The above table shows the MIF default configuration. These values can be changed by the user.

NOTE

The default MLPL is normal.

There are special requirements regarding the SLEEP and INITN pin, on power ramp-up. The memory SLEEP and INITN pin should be enabled by the VDD-not-ready flag. The PGMC also provides an option, by enabling the BYPASS_VDD_OK field, the SLEEP and INITN pin is not affected by the VDD-not-ready flag.

The power switch architecture supported by MIF is shown below:

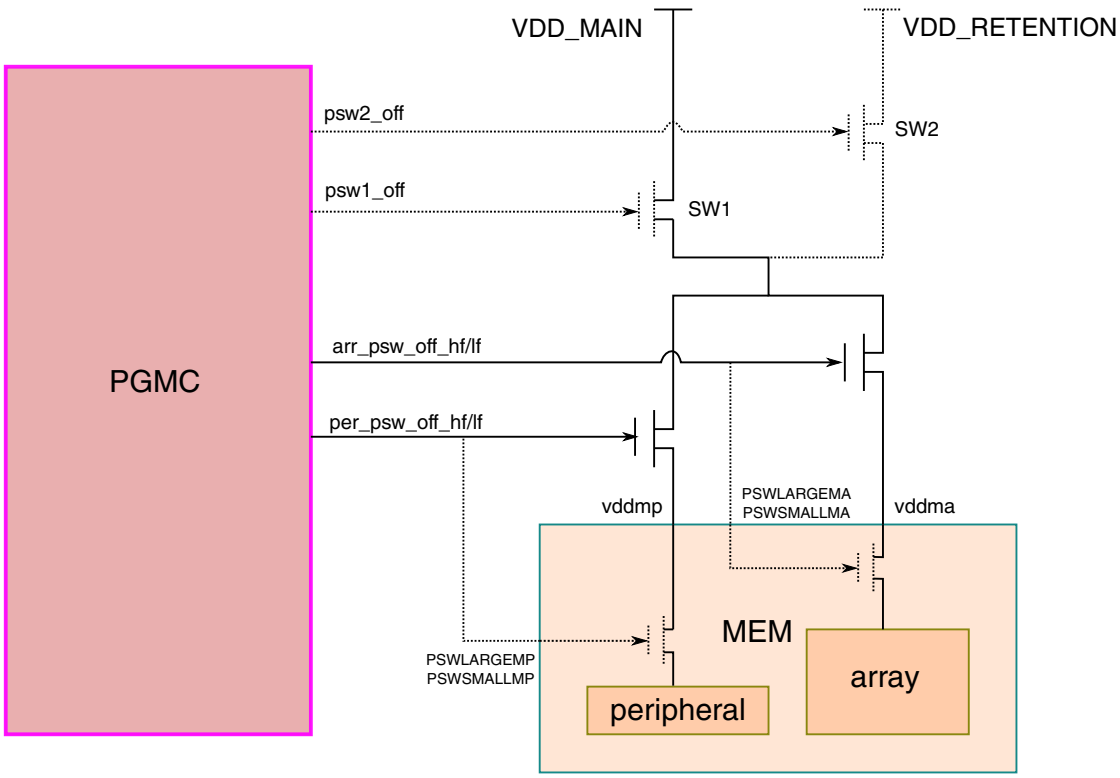


Figure 20-2. Power Switch Architecture Diagram

20.3.6 PMIC Standby

When the system is in standby mode, PPC can send PMIC_STBY_REQ to the external PMIC outside, thus enabling PMIC to provide a standby voltage.

The PPC_STBY_SP_CTRL[STBY_ON_AT_SP_SLEEP] field defines the Setpoint, which the PPC should send PMIC_STBY_REQ when the system is in standby mode.

20.3.7 Clocks

The table found here describes the clock sources for SSARC. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Table 20-4. Clocks

Clock Name	Description
ipg_clk	Main functional clock
ipg_clk_s	Bus clock, same as ipg_clk source
clk_32k	Low speed clock

20.3.8 Reset

The PGMC module has one asynchronous active-low reset:

- System Reset (reset_b)

Please see the System Reset Controller (SRC) chapter for more details on system resets.

20.4 External Signals

Table 20-5. External Signals

External Signals	Description	Direction
PMIC_STBY_REQ	Signal notifies external power management IC (PMIC) to move from functional voltage to standby voltage.	O

20.5 Initialization

20.5.1 UPI Handshake

The following diagram shows an example in which a Setpoint transition is seen.

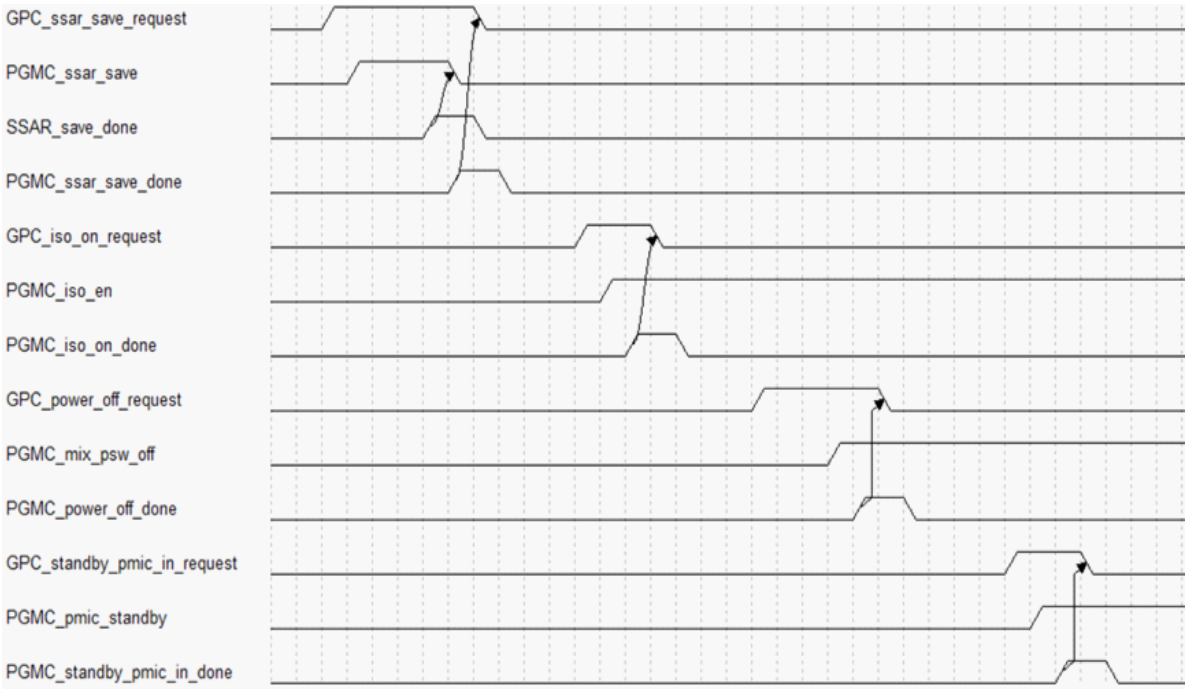


Figure 20-3. UPI Handshake

20.5.2 Register Access Permission

For PGMC register access, the register x_AUTHEN_CTRL is used to control access permission. The CPC and MPC shares AUTHEN_CTRL with their MIFs.

20.6 Memory Map and register definition

This section includes the PGMC module memory map and detailed descriptions of all registers.

20.6.1 PGMC_BPC register descriptions

20.6.1.1 pgmc_bpc memory map

PGMC_BPC0 base address: 40C8_8000h

PGMC_BPC1 base address: 40C8_8200h

PGMC_BPC2 base address: 40C8_8400h

PGMC_BPC3 base address: 40C8_8600h

PGMC_BPC4 base address: 40C8_8800h

PGMC_BPC5 base address: 40C8_8A00h

PGMC_BPC6 base address: 40C8_8C00h

PGMC_BPC7 base address: 40C8_8E00h

Offset	Register	Width (In bits)	Access	Reset value
4h	BPC Authentication Control (BPC_AUTHEN_CTRL)	32	Table 20-5	0000_0F00h
10h	BPC Mode (BPC_MODE)	32	Table 20-5	0000_0000h
14h	BPC power control (BPC_POWER_CTRL)	32	Table 20-5	0000_0000h
2Ch	BPC flag (BPC_FLAG)	32	Table 20-5	0000_0000h
40h	BPC SSAR save control (BPC_SSAR_SAVE_CTRL)	32	Table 20-5	0000_0000h
44h	BPC SSAR restore control (BPC_SSAR_RESTORE_CTRL)	32	Table 20-5	0000_0000h

20.6.1.2 BPC Authentication Control (BPC_AUTHEN_CTRL)

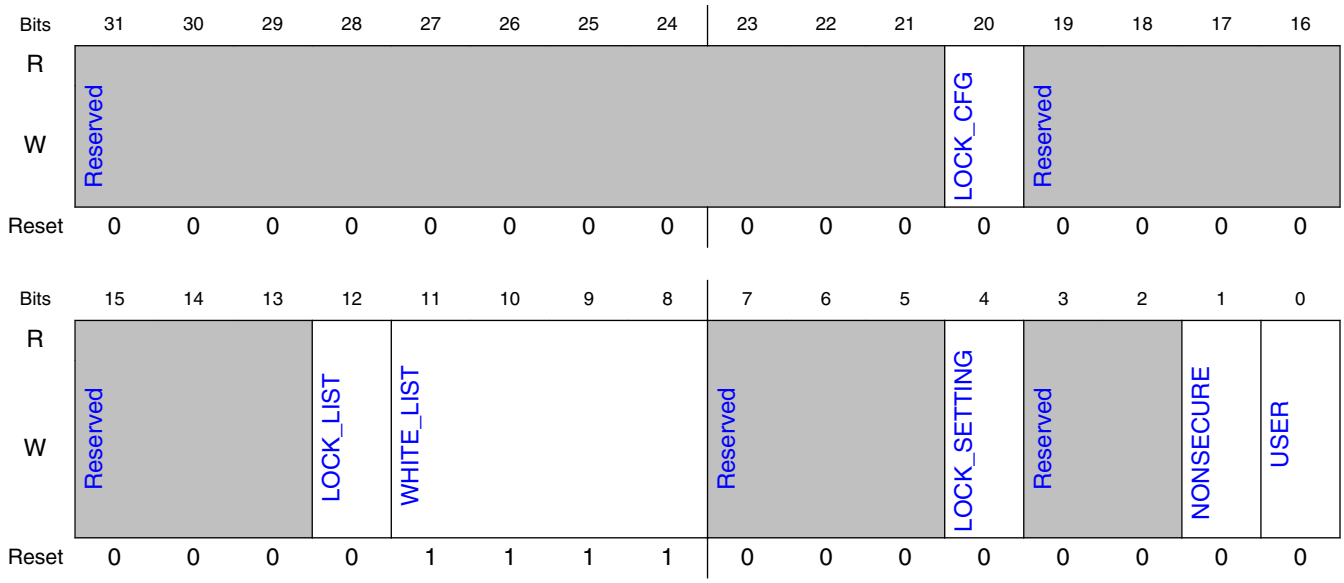
20.6.1.2.1 Offset

Register	Offset
BPC_AUTHEN_CTRL	4h

20.6.1.2.2 Function

Authentication control of basic power controller.

20.6.1.2.3 Diagram



20.6.1.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock This field is write once and can lock value of low power configuration fields. It is unlocked by system reset.
19-13 —	Reserved
12 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field
11-8 WHITE_LIST	Domain ID white list when bit setting to 1, the corresponding domain ID can access CPU mode control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Muliple bits with a value of 1 is permitted.
7-5 —	Reserved
4 LOCK_SETTING	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
3-2 —	Reserved
1	Allow non-secure mode access 0b - Allow only secure mode to access basic power control registers

Table continues on the next page...

Field	Function
NONSECURE	1b - Allow both secure and non-secure mode to access basic power control registers
0 USER	Allow user mode access 0b - Allow only privilege mode to access basic power control registers 1b - Allow both privilege and user mode to access basic power control registers

20.6.1.3 BPC Mode (BPC_MODE)

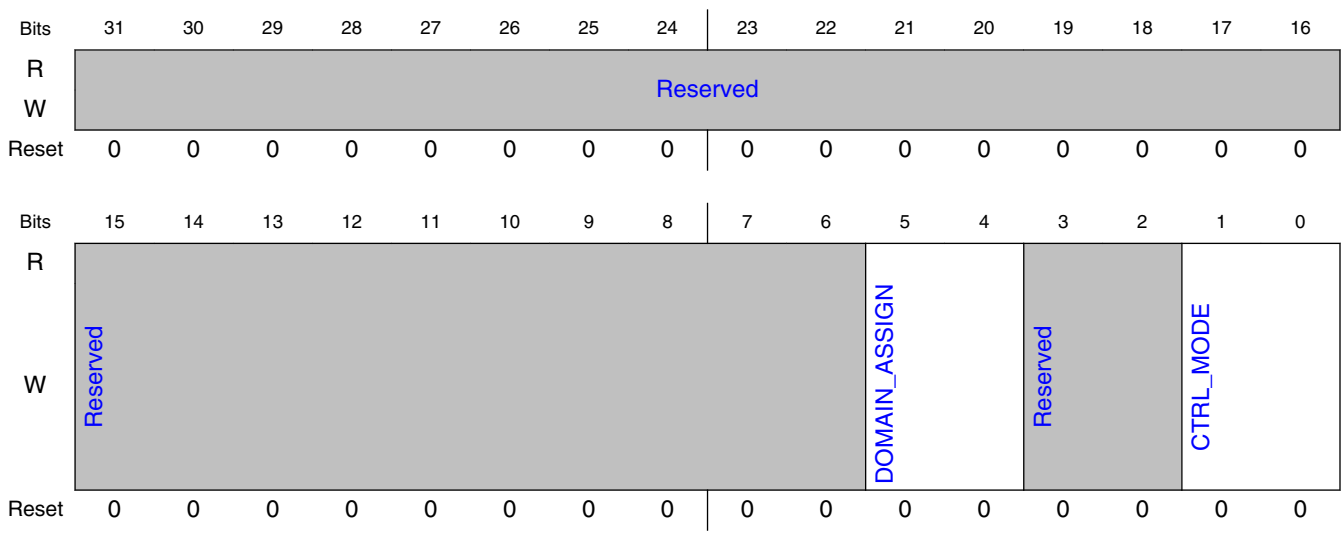
20.6.1.3.1 Offset

Register	Offset
BPC_MODE	10h

20.6.1.3.2 Function

.

20.6.1.3.3 Diagram



20.6.1.3.4 Fields

Field	Function
31-6	Reserved

Table continues on the next page...

Field	Function
—	
5-4 DOMAIN_ASSIGN	Domain assignment of the BPC Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Domain 0 01b - Domain 1 10b - Domain 2 11b - Domain 3
3-2 —	Reserved
1-0 CTRL_MODE	Control mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Not affected by any low power mode 01b - Controlled by CPU power mode of the domain 10b - Controlled by Setpoint 11b - Reserved

20.6.1.4 BPC power control (BPC_POWER_CTRL)

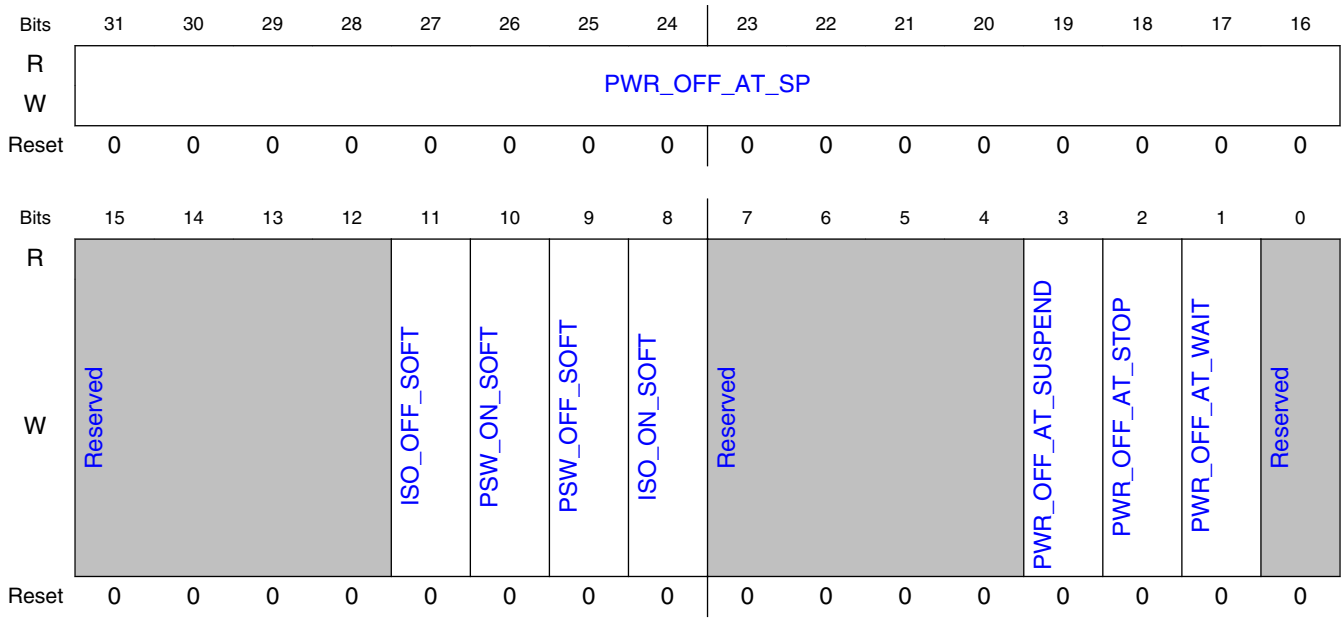
20.6.1.4.1 Offset

Register	Offset
BPC_POWER_CTRL	14h

20.6.1.4.2 Function

.

20.6.1.4.3 Diagram



20.6.1.4.4 Fields

Field	Function
31-16 PWR_OFF_AT_SP	Power off when system enters Setpoint number Setting a bit as 0x1 means power off. Valid when CTRL_MODE is 0x2. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-12 —	Reserved
11 ISO_OFF_SOFT	Software isolation off trigger Writing 0x1 to turn off the isolation of the power domain
10 PSW_ON_SOFT	Software power on trigger Writing 0x1 to turn on the power switch of the power domain
9 PSW_OFF_SOFT	Software power off trigger Writing 0x1 to turn off the power switch of the power domain
8 ISO_ON_SOFT	Software isolation on trigger Writing 0x1 to turn on isolation of the power domain
7-4 —	Reserved
3 PWR_OFF_AT_SUSPEND	0x1: Power off when domain enters SUSPEND mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
2 PWR_OFF_AT_STOP	0x1: Power off when domain enters STOP mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
1 PWR_OFF_AT_WAIT	0x1: Power off when domain enters WAIT mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
0 —	Reserved

20.6.1.5 BPC flag (BPC_FLAG)

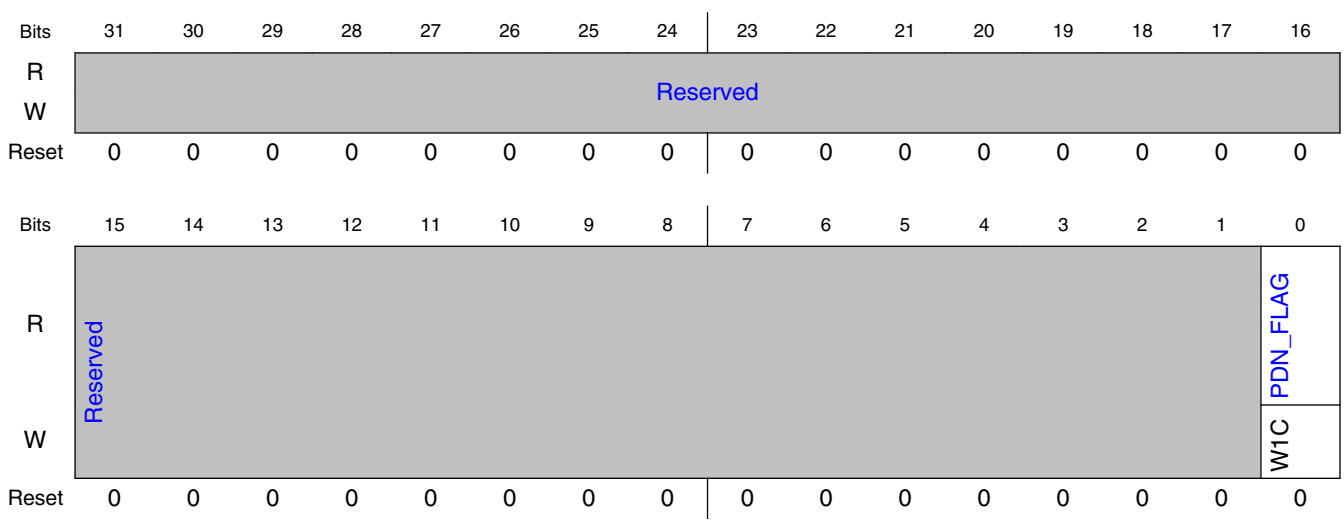
20.6.1.5.1 Offset

Register	Offset
BPC_FLAG	2Ch

20.6.1.5.2 Function

.

20.6.1.5.3 Diagram



20.6.1.5.4 Fields

Field	Function
31-1 —	Reserved
0 PDN_FLAG	set to 1 after power switch off, cleared by writing 1

20.6.1.6 BPC SSAR save control (BPC_SSAR_SAVE_CTRL)

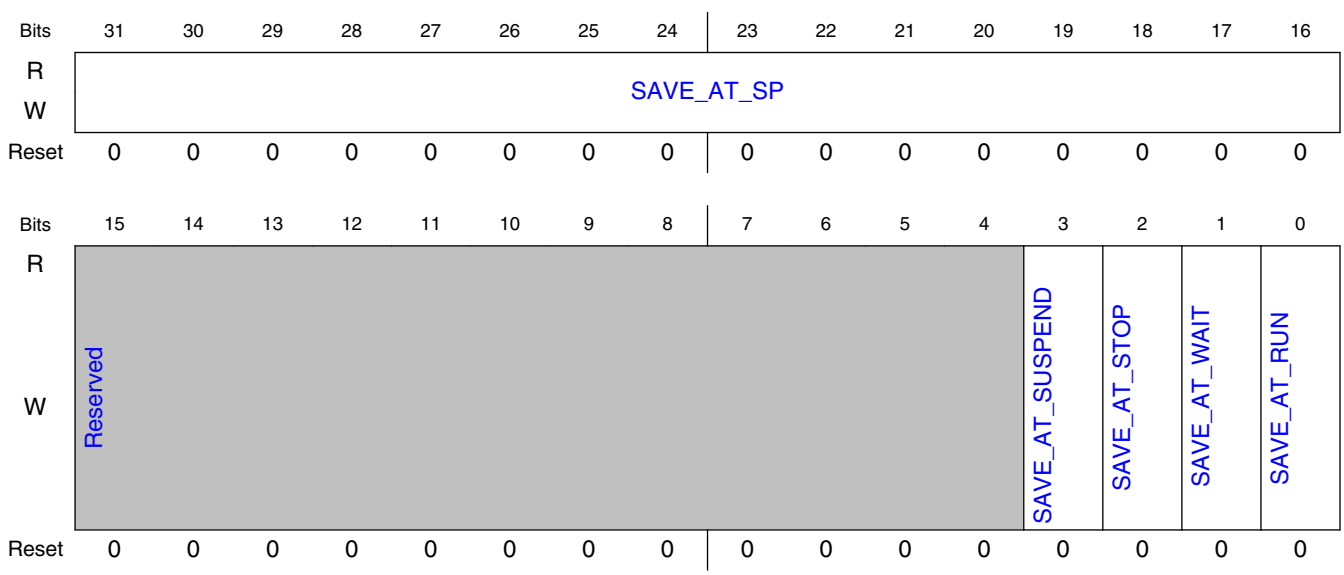
20.6.1.6.1 Offset

Register	Offset
BPC_SSAR_SAVE_CTRL	40h

20.6.1.6.2 Function

State save control for basic power controller.

20.6.1.6.3 Diagram



20.6.1.6.4 Fields

Field	Function
31-16 SAVE_AT_SP	Save data when system enters a Setpoint. Valid when CTRL_MODE is 0x2. This field is locked by AUTHEN_CTRL[LOCK_CFG] field (unlocked by system reset).
15-4 —	Reserved
3 SAVE_AT_SUSPEND	Save data when domain enters SUSPEND mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
2 SAVE_AT_STOP	Save data when domain enters STOP mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
1 SAVE_AT_WAIT	Save data when domain enters WAIT mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
0 SAVE_AT_RUN	Save data at RUN mode, software writting 0x1 to trigger SSARC to execute save process

20.6.1.7 BPC SSAR restore control (BPC_SSAR_RESTORE_CTRL)

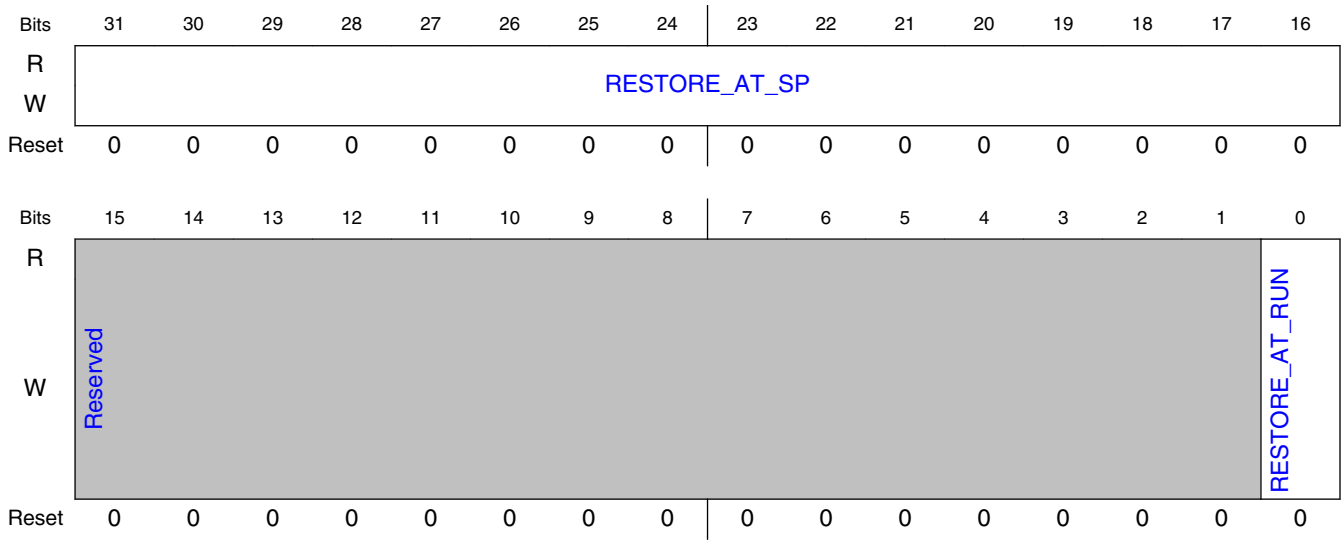
20.6.1.7.1 Offset

Register	Offset
BPC_SSAR_RESTORE_CTRL	44h

20.6.1.7.2 Function

State restore control for basic power controller.

20.6.1.7.3 Diagram



20.6.1.7.4 Fields

Field	Function
31-16 RESTORE_AT_SP	Restore data when system enters a Setpoint. Valid when CTRL_MODE is 0x2. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-1 —	Reserved
0 RESTORE_AT_RUN	Restore data at RUN mode Writing 0x1 triggers SSARC to restore the data

20.6.2 PGMC_CPC register descriptions

20.6.2.1 pgmc_cpc memory map

PGMC_CPC0 base address: 40C8_9000h

PGMC_CPC1 base address: 40C8_9400h

Offset	Register	Width (In bits)	Access	Reset value
4h	CPC Authentication Control (CPC_AUTHEN_CTRL)	32	Table 20-5	0000_0F00h
10h	CPC Core Mode (CPC_CORE_MODE)	32	Table 20-5	0000_0000h
14h	CPC core power control (CPC_CORE_POWER_CTRL)	32	Table 20-5	0000_0000h
2Ch	CPC flag (CPC_FLAG)	32	Table 20-5	0000_0000h
40h	CPC Cache Mode (CPC_CACHE_MODE)	32	Table 20-5	0000_0000h
44h	CPC cache CPU mode control (CPC_CACHE_CM_CTRL)	32	Table 20-5	0000_3330h
48h	CPC cache Setpoint control 0 (CPC_CACHE_SP_CTRL_0)	32	RW	3333_3333h
4Ch	CPC cache Setpoint control 1 (CPC_CACHE_SP_CTRL_1)	32	RW	3333_3333h
C0h	CPC local memory Mode (CPC_LMEM_MODE)	32	Table 20-5	0000_0000h
C4h	CPC local memory CPU mode control (CPC_LMEM_CM_CTRL)	32	Table 20-5	0000_0000h
C8h	CPC local memory Setpoint control 0 (CPC_LMEM_SP_CTRL_0)	32	RW	0000_0000h
CCh	CPC local memory Setpoint control 1 (CPC_LMEM_SP_CTRL_1)	32	RW	0000_0000h

20.6.2.2 CPC Authentication Control (CPC_AUTHEN_CTRL)

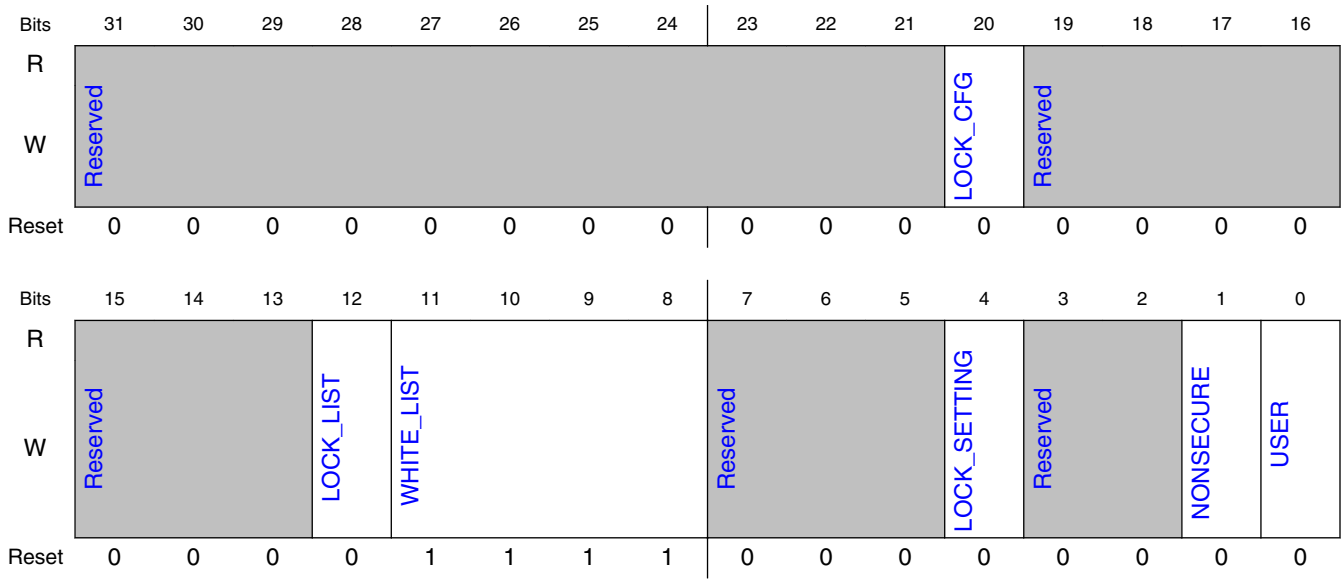
20.6.2.2.1 Offset

Register	Offset
CPC_AUTHEN_CTRL	4h

20.6.2.2.2 Function

Authentication control of CPU power controller.

20.6.2.2.3 Diagram



20.6.2.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields
19-13 —	Reserved
12 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field
11-8 WHITE_LIST	Domain ID white list when bit setting to 1, the corresponding domain ID can access CPU power control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Muliple bits with a value of 1 is permitted..
7-5 —	Reserved
4 LOCK_SETTIN G	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
3-2 —	Reserved
1 NONSECURE	Allow non-secure mode access Allow both secure and non-secure mode to access CPU power control registers

Table continues on the next page...

Field	Function
0 USER	Allow user mode access Allow both privilege and user mode to access CPU mode control registers

20.6.2.3 CPC Core Mode (CPC_CORE_MODE)

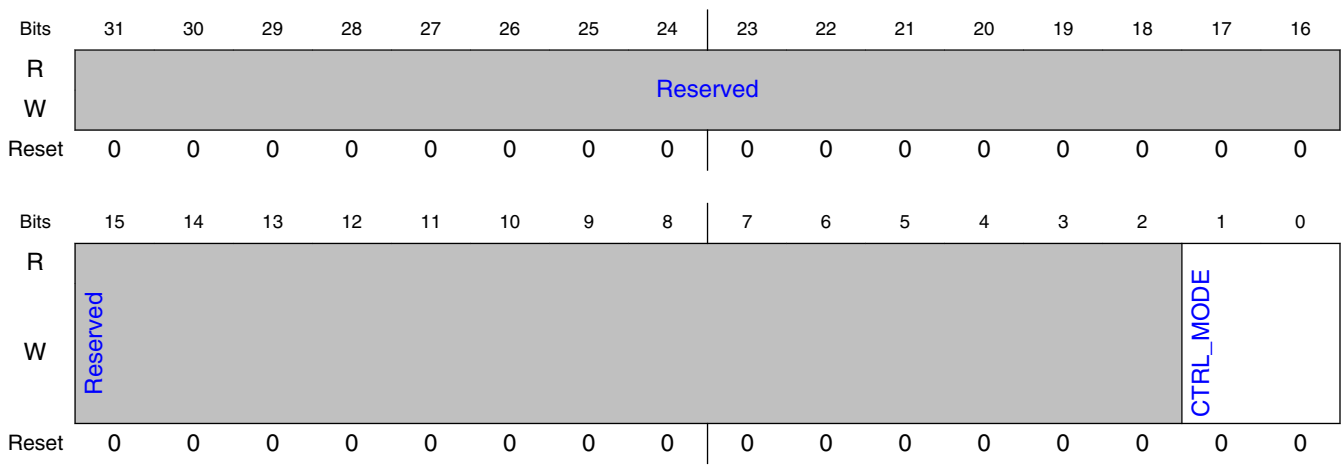
20.6.2.3.1 Offset

Register	Offset
CPC_CORE_MODE	10h

20.6.2.3.2 Function

Core power mode register.

20.6.2.3.3 Diagram



20.6.2.3.4 Fields

Field	Function
31-2 —	Reserved
1-0 CTRL_MODE	Control mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Not affected by any low power mode 01b - Controlled by CPU power mode of the domain

Field	Function
	10b - Reserved 11b - Reserved

20.6.2.4 CPC core power control (CPC_CORE_POWER_CTRL)

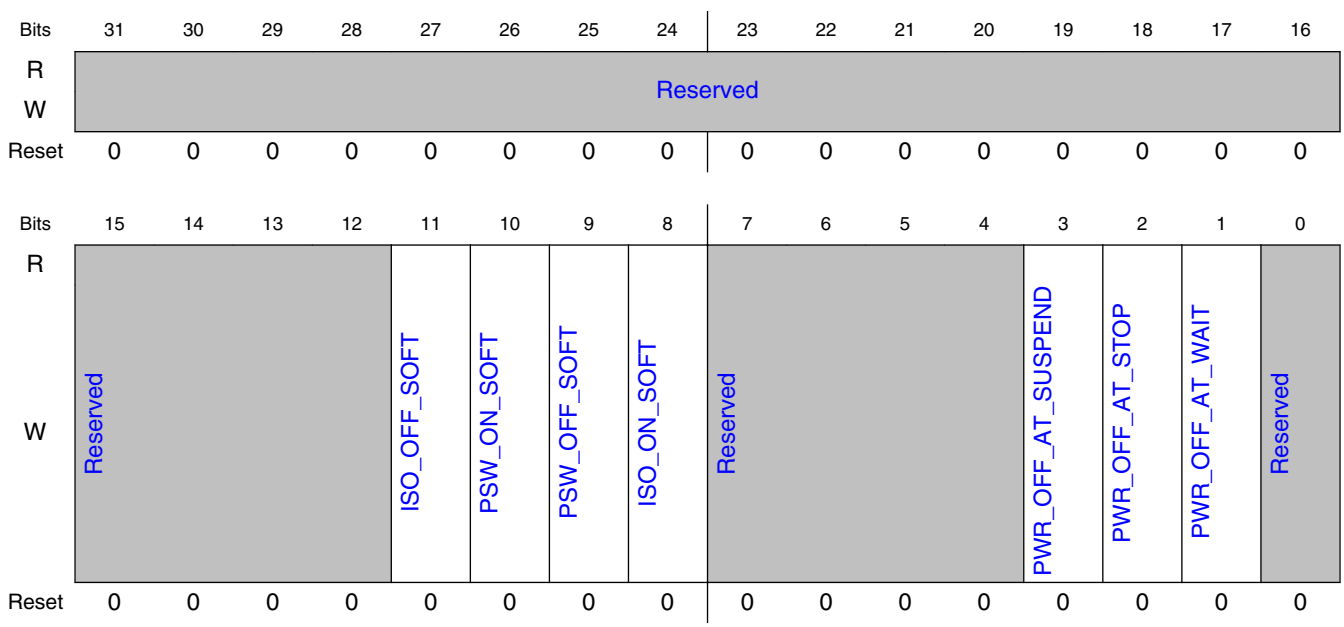
20.6.2.4.1 Offset

Register	Offset
CPC_CORE_POWER_CTRL	14h

20.6.2.4.2 Function

.

20.6.2.4.3 Diagram



20.6.2.4.4 Fields

Field	Function
31-12	Reserved

Table continues on the next page...

Field	Function
—	
11 ISO_OFF_SOFT	Software isolation off trigger
10 PSW_ON_SOFT	Software power on trigger
9 PSW_OFF_SOFT	Software power off trigger
8 ISO_ON_SOFT	Software isolation on trigger
7-4 —	Reserved
3 PWR_OFF_AT_SUSPEND	Power off when domain enters SUSPEND mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
2 PWR_OFF_AT_STOP	Power off when domain enters STOP mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
1 PWR_OFF_AT_WAIT	Power off when domain enters WAIT mode Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
0 —	Reserved

20.6.2.5 CPC flag (CPC_FLAG)

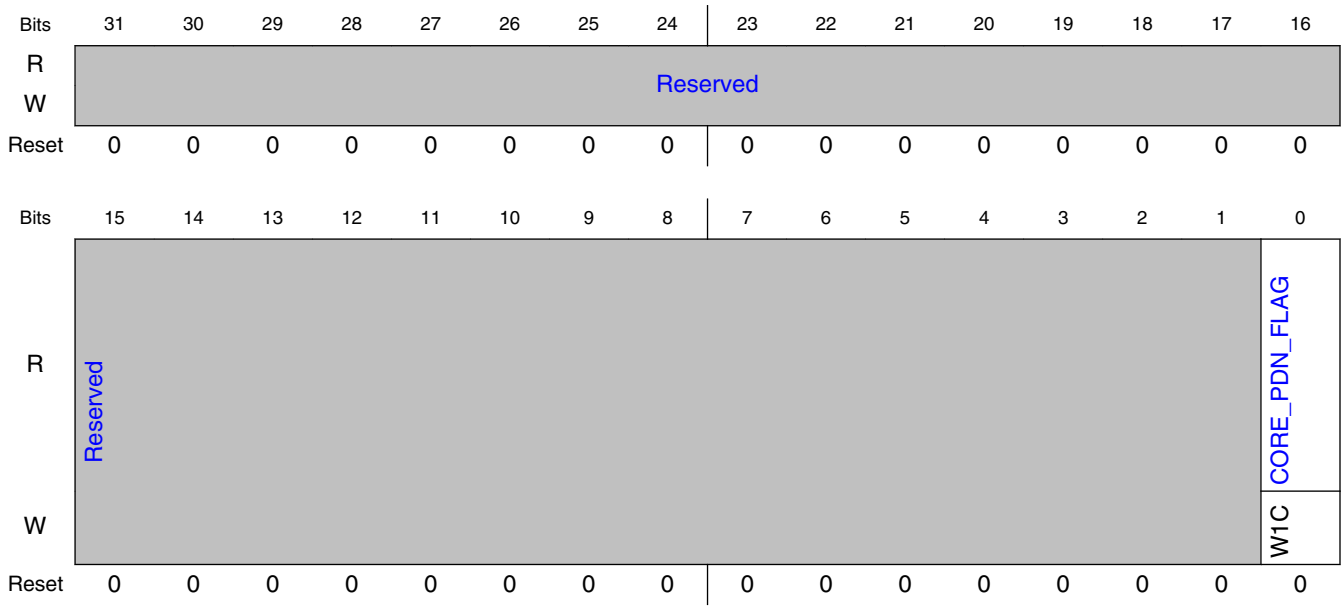
20.6.2.5.1 Offset

Register	Offset
CPC_FLAG	2Ch

20.6.2.5.2 Function

.

20.6.2.5.3 Diagram



20.6.2.5.4 Fields

Field	Function
31-1 —	Reserved
0 CORE_PDN_FLAG	set to 1 after core power switch off, cleared by writing 1

20.6.2.6 CPC Cache Mode (CPC_CACHE_MODE)

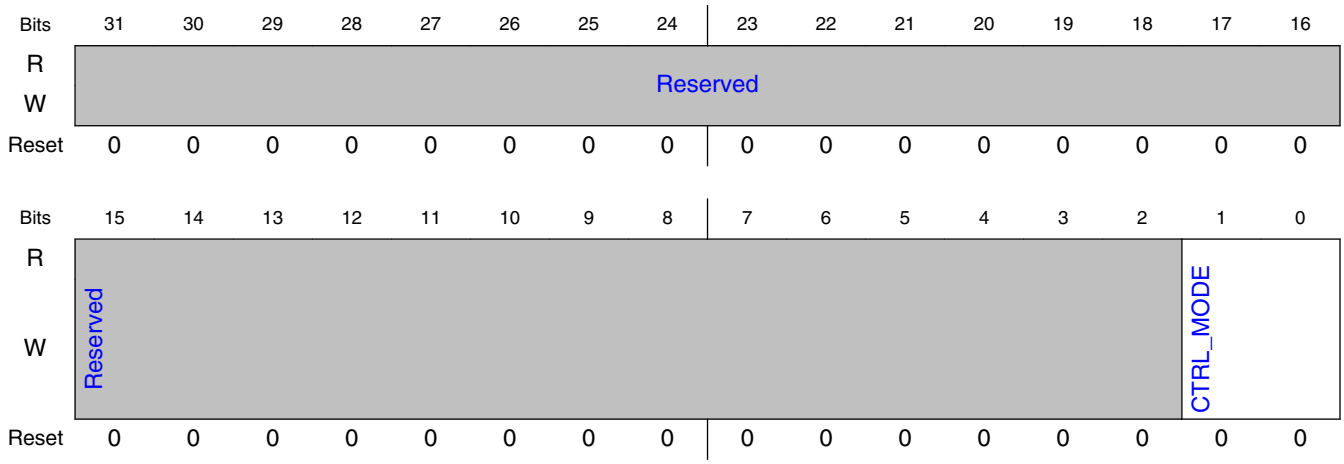
20.6.2.6.1 Offset

Register	Offset
CPC_CACHE_MODE	40h

20.6.2.6.2 Function

Cache power mode control register.

20.6.2.6.3 Diagram



20.6.2.6.4 Fields

Field	Function
31-2 —	Reserved
1-0 CTRL_MODE	Control mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Not affected by any low power mode 01b - Controlled by CPU power mode of the domain 10b - Controlled by Setpoint 11b - Reserved

20.6.2.7 CPC cache CPU mode control (CPC_CACHE_CM_CTRL)

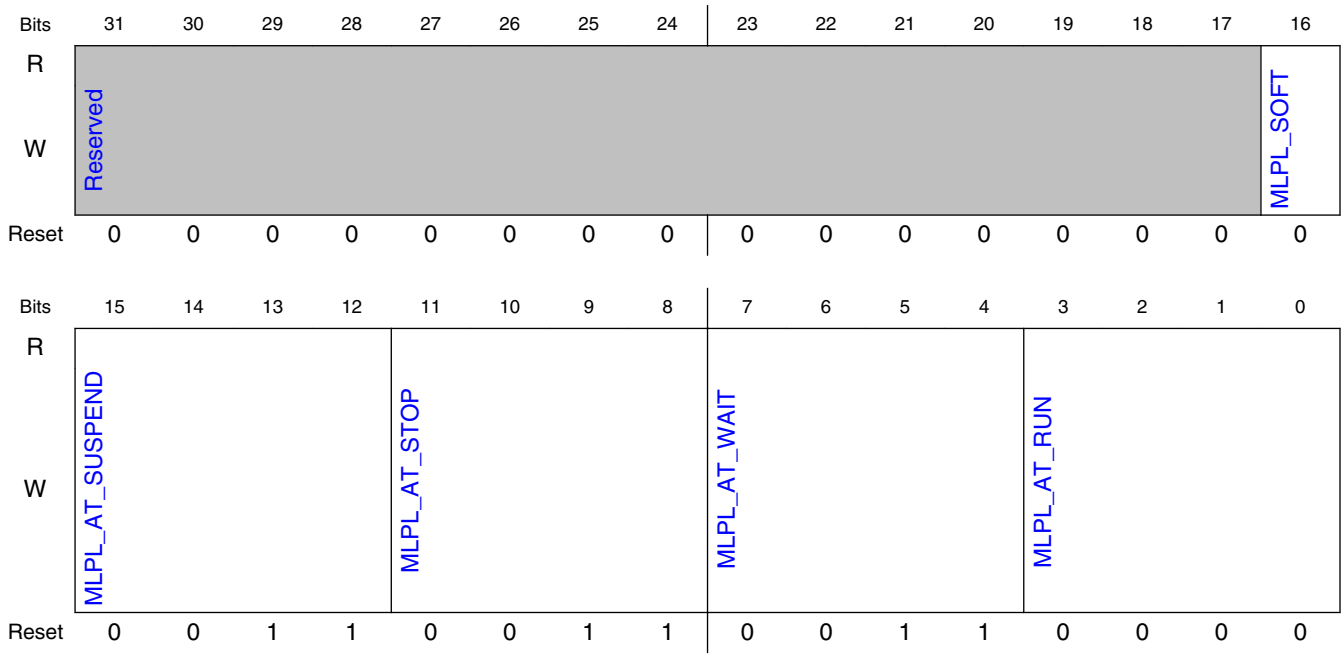
20.6.2.7.1 Offset

Register	Offset
CPC_CACHE_CM_CTRL	44h

20.6.2.7.2 Function

Cache memory low power level under CPU mode.

20.6.2.7.3 Diagram



20.6.2.7.4 Fields

Field	Function
31-17 —	Reserved
16 MLPL_SOFT	Memory Low Power Level (MLPL) software change request, keep 1 until MLPL transition complete
15-12 MLPL_AT_SUSPEND	Memory Low Power Level (MLPL) at SUSPEND mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8 MLPL_AT_STOP	Memory Low Power Level (MLPL) at STOP mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
7-4 MLPL_AT_WAIT	Memory Low Power Level (MLPL) at WAIT mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_RUN	Memory Low Power Level (MLPL) at RUN mode

20.6.2.8 CPC cache Setpoint control 0 (CPC_CACHE_SP_CTRL_0)

20.6.2.8.1 Offset

Register	Offset
CPC_CACHE_SP_CTRL_0	48h

20.6.2.8.2 Function

Cache memory low power level under setpoint register 0.

20.6.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MLPL_AT_SP7				MLPL_AT_SP6				MLPL_AT_SP5				MLPL_AT_SP4			
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_AT_SP3				MLPL_AT_SP2				MLPL_AT_SP1				MLPL_AT_SP0			
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

20.6.2.8.4 Fields

Field	Function
31-28 MLPL_AT_SP7	Memory Low Power Level (MLPL) at Setpoint 7. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
27-24 MLPL_AT_SP6	Memory Low Power Level (MLPL) at Setpoint 6. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
23-20 MLPL_AT_SP5	Memory Low Power Level (MLPL) at Setpoint 5. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
19-16 MLPL_AT_SP4	Memory Low Power Level (MLPL) at Setpoint 4. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-12 MLPL_AT_SP3	Memory Low Power Level (MLPL) at Setpoint 3. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8 MLPL_AT_SP2	Memory Low Power Level (MLPL) at Setpoint 2. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
7-4 MLPL_AT_SP1	Memory Low Power Level (MLPL) at Setpoint 1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_SP0	Memory Low Power Level (MLPL) at Setpoint 0. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.2.9 CPC cache Setpoint control 1 (CPC_CACHE_SP_CTRL_1)

20.6.2.9.1 Offset

Register	Offset
CPC_CACHE_SP_CTRL_1	4Ch

20.6.2.9.2 Function

Cache memory low power level under setpoint register 0.

20.6.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MLPL_AT_SP15				MLPL_AT_SP14				MLPL_AT_SP13				MLPL_AT_SP12			
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_AT_SP11				MLPL_AT_SP10				MLPL_AT_SP9				MLPL_AT_SP8			
Reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

20.6.2.9.4 Fields

Field	Function
31-28 MLPL_AT_SP15	Memory Low Power Level (MLPL) at Setpoint 15. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
27-24 MLPL_AT_SP14	Memory Low Power Level (MLPL) at Setpoint 14. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
23-20 MLPL_AT_SP13	Memory Low Power Level (MLPL) at Setpoint 13. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
19-16 MLPL_AT_SP12	Memory Low Power Level (MLPL) at Setpoint 12. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-12 MLPL_AT_SP11	Memory Low Power Level (MLPL) at Setpoint 11. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8 MLPL_AT_SP10	Memory Low Power Level (MLPL) at Setpoint 10. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
7-4 MLPL_AT_SP9	Memory Low Power Level (MLPL) at Setpoint 9. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_SP8	Memory Low Power Level (MLPL) at Setpoint 8. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.2.10 CPC local memory Mode (CPC_LMEM_MODE)

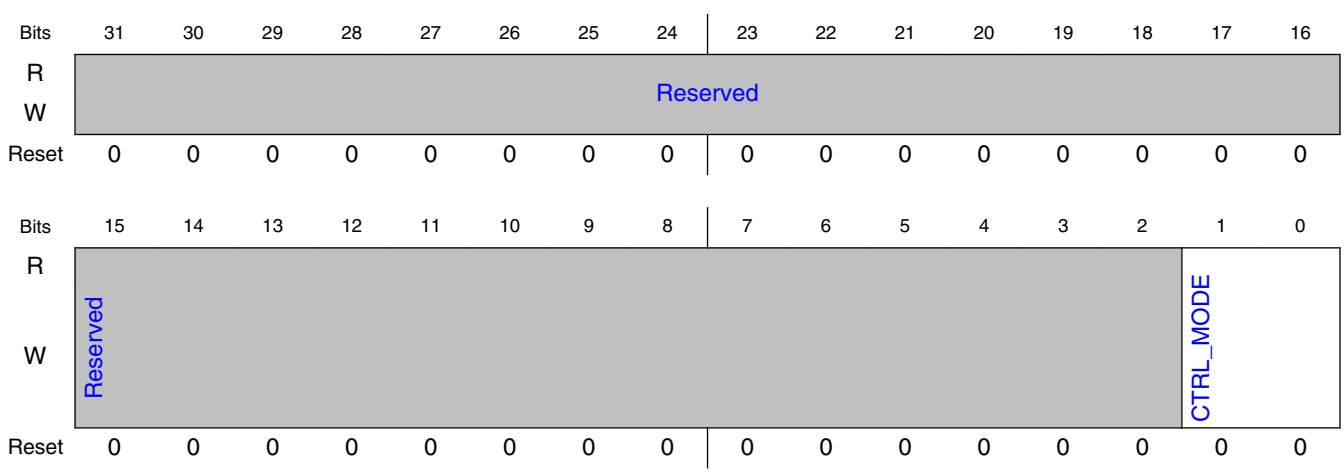
20.6.2.10.1 Offset

Register	Offset
CPC_LMEM_MODE	C0h

20.6.2.10.2 Function

Local memory mode control.

20.6.2.10.3 Diagram



20.6.2.10.4 Fields

Field	Function
31-2 —	Reserved
1-0 CTRL_MODE	Control mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Not affected by any low power mode 01b - Controlled by CPU power mode of the domain 10b - Controlled by Setpoint 11b - Reserved

20.6.2.11 CPC local memory CPU mode control (CPC_LMEM_CM_CTRL)

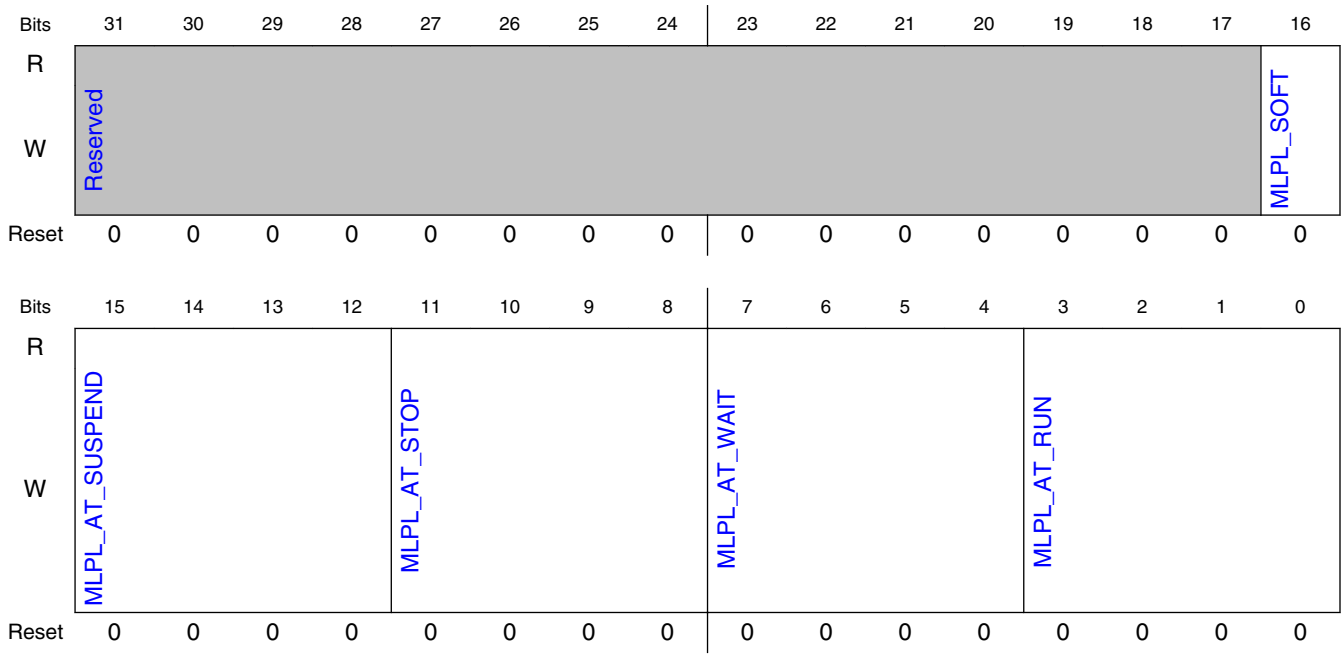
20.6.2.11.1 Offset

Register	Offset
CPC_LMEM_CM_CTRL	C4h

20.6.2.11.2 Function

Local memory low power level under CPU mode.

20.6.2.11.3 Diagram



20.6.2.11.4 Fields

Field	Function
31-17 —	Reserved
16 MLPL_SOFT	Memory Low Power Level (MLPL) software change request, keep 1 until MLPL transition complete
15-12 MLPL_AT_SUSPEND	Memory Low Power Level (MLPL) at SUSPEND mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8 MLPL_AT_STOP	Memory Low Power Level (MLPL) at STOP mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
7-4 MLPL_AT_WAIT	Memory Low Power Level (MLPL) at WAIT mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_RUN	Memory Low Power Level (MLPL) at RUN mode

20.6.2.12 CPC local memory Setpoint control 0 (CPC_LMEM_SP_CTRL_0)

20.6.2.12.1 Offset

Register	Offset
CPC_LMEM_SP_CTRL_0	C8h

20.6.2.12.2 Function

Local memory low power level under setpoint register 0.

20.6.2.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MLPL_AT_SP7				MLPL_AT_SP6				MLPL_AT_SP5				MLPL_AT_SP4			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_AT_SP3				MLPL_AT_SP2				MLPL_AT_SP1				MLPL_AT_SP0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.6.2.12.4 Fields

Field	Function
31-28 MLPL_AT_SP7	Memory Low Power Level (MLPL) at Setpoint 7. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
27-24 MLPL_AT_SP6	Memory Low Power Level (MLPL) at Setpoint 6. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
23-20 MLPL_AT_SP5	Memory Low Power Level (MLPL) at Setpoint 5. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
19-16 MLPL_AT_SP4	Memory Low Power Level (MLPL) at Setpoint 4. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-12 MLPL_AT_SP3	Memory Low Power Level (MLPL) at Setpoint 3. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8	Memory Low Power Level (MLPL) at Setpoint 2. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
MLPL_AT_SP2	
7-4 MLPL_AT_SP1	Memory Low Power Level (MLPL) at Setpoint 1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_SP0	Memory Low Power Level (MLPL) at Setpoint 0. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.2.13 CPC local memory Setpoint control 1 (CPC_LMEM_SP_CTRL_1)

20.6.2.13.1 Offset

Register	Offset
CPC_LMEM_SP_CTRL_1	CCh

20.6.2.13.2 Function

Local memory low power level under setpoint register 1.

20.6.2.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MLPL_AT_SP15				MLPL_AT_SP14				MLPL_AT_SP13				MLPL_AT_SP12			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_AT_SP11				MLPL_AT_SP10				MLPL_AT_SP9				MLPL_AT_SP8			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.6.2.13.4 Fields

Field	Function
31-28 MLPL_AT_SP15	Memory Low Power Level (MLPL) at Setpoint 15. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
27-24 MLPL_AT_SP14	Memory Low Power Level (MLPL) at Setpoint 14. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
23-20 MLPL_AT_SP13	Memory Low Power Level (MLPL) at Setpoint 13. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
19-16 MLPL_AT_SP12	Memory Low Power Level (MLPL) at Setpoint 12. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
15-12 MLPL_AT_SP11	Memory Low Power Level (MLPL) at Setpoint 11. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
11-8 MLPL_AT_SP10	Memory Low Power Level (MLPL) at Setpoint 10. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
7-4 MLPL_AT_SP9	Memory Low Power Level (MLPL) at Setpoint 9. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
3-0 MLPL_AT_SP8	Memory Low Power Level (MLPL) at Setpoint 8. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3 PGMC_MIF register descriptions

20.6.3.1 pgmc_mif memory map

PGMC_CPC0_MIF0 base address: 40C8_9100h

PGMC_CPC0_MIF1 base address: 40C8_9200h

PGMC_CPC1_MIF0 base address: 40C8_9500h

PGMC_CPC1_MIF1 base address: 40C8_9600h

Offset	Register	Width (In bits)	Access	Reset value
4h	MIF Authentication Control (MIF_AUTHEN_CTRL)	32	Table 20-5	0000_0000h
10h	MIF MLPL control of SLEEP (MIF_MLPL_SLEEP)	32	Table 20-5	0000_FF00h
20h	MIF MLPL control of IG (MIF_MLPL_IG)	32	Table 20-5	0000_0060h
30h	MIF MLPL control of LS (MIF_MLPL_LS)	32	Table 20-5	0000_0010h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
40h	MIF MLPL control of HS (MIF_MLPL_HS)	32	Table 20-5	0000_0002h
50h	MIF MLPL control of STDBY (MIF_MLPL_STDBY)	32	Table 20-5	0000_0040h
60h	MIF MLPL control of array power down (MIF_MLPL_ARR_PDN)	32	Table 20-5	0000_0C00h
70h	MIF MLPL control of peripheral power down (MIF_MLPL_PER_PDN)	32	Table 20-5	0000_4A00h
80h	MIF MLPL control of INITN (MIF_MLPL_INITN)	32	Table 20-5	0000_35FFh
B0h	MIF MLPL control of isolation enable (MIF_MLPL_ISO)	32	Table 20-5	0000_CA00h

20.6.3.2 MIF Authentication Control (MIF_AUTHEN_CTRL)

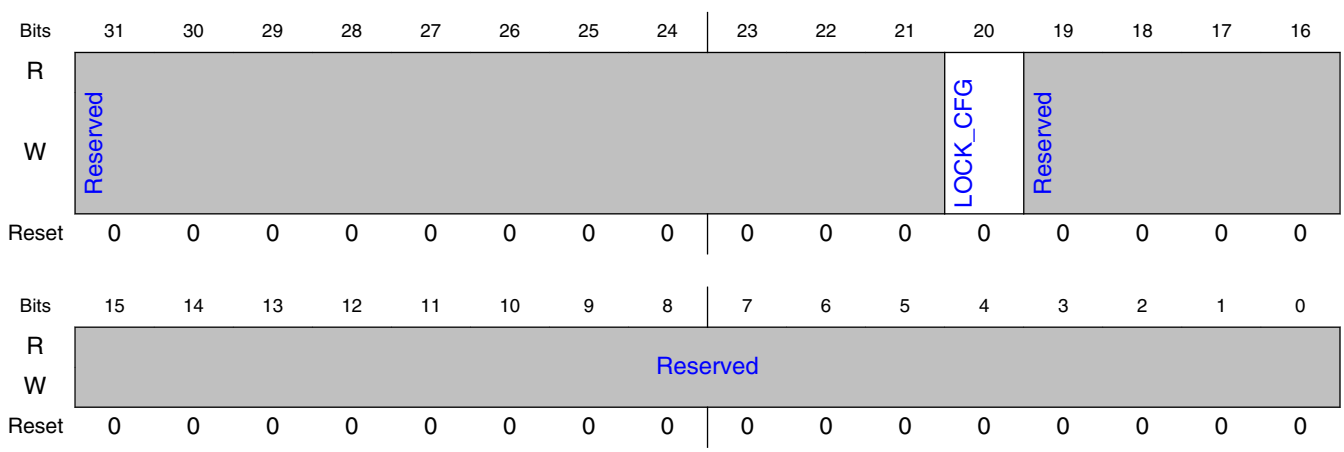
20.6.3.2.1 Offset

Register	Offset
MIF_AUTHEN_CTRL	4h

20.6.3.2.2 Function

Authentication control of memory interface.

20.6.3.2.3 Diagram



20.6.3.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields
19-0 —	Reserved

20.6.3.3 MIF MLPL control of SLEEP (MIF_MLPL_SLEEP)

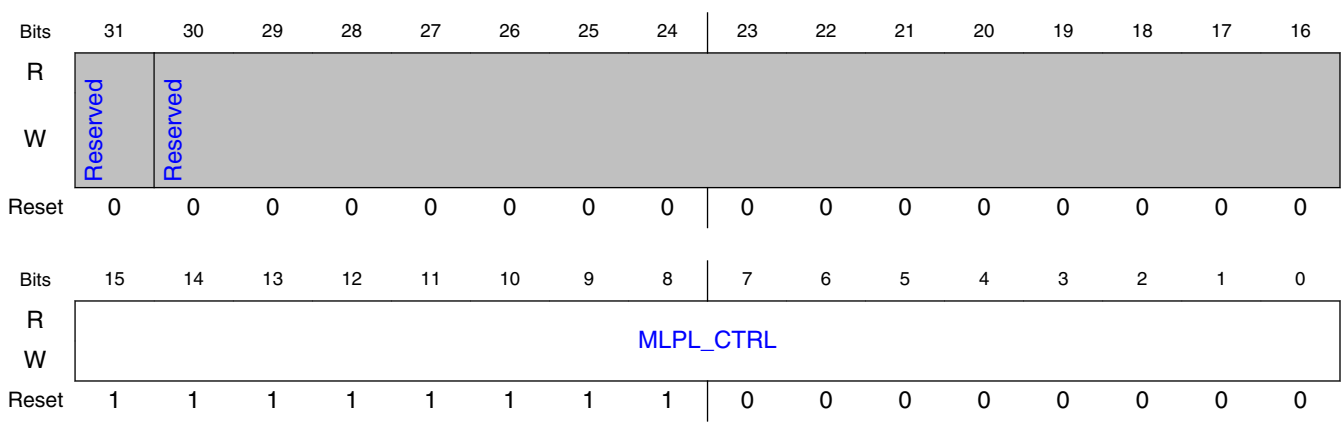
20.6.3.3.1 Offset

Register	Offset
MIF_MLPL_SLEEP	10h

20.6.3.3.2 Function

SLEEP pin control.

20.6.3.3.3 Diagram



20.6.3.3.4 Fields

Field	Function
31 —	Reserved
30-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.4 MIF MLPL control of IG (MIF_MLPL_IG)

20.6.3.4.1 Offset

Register	Offset
MIF_MLPL_IG	20h

20.6.3.4.2 Function

IG pin control.

20.6.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

20.6.3.4.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.5 MIF MLPL control of LS (MIF_MLPL_LS)

20.6.3.5.1 Offset

Register	Offset
MIF_MLPL_LS	30h

20.6.3.5.2 Function

LS pin control.

20.6.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

20.6.3.5.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.6 MIF MLPL control of HS (MIF_MLPL_HS)

20.6.3.6.1 Offset

Register	Offset
MIF_MLPL_HS	40h

20.6.3.6.2 Function

HS pin control.

20.6.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

20.6.3.6.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.7 MIF MLPL control of STDBY (MIF_MLPL_STDBY)

20.6.3.7.1 Offset

Register	Offset
MIF_MLPL_STDBY	50h

20.6.3.7.2 Function

STDBY pin control.

20.6.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

20.6.3.7.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.8 MIF MLPL control of array power down (MIF_MLPL_ARR_PDN)

20.6.3.8.1 Offset

Register	Offset
MIF_MLPL_ARR_PDN	60h

20.6.3.8.2 Function

Array power down control.

20.6.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

20.6.3.8.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means power down at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.9 MIF MLPL control of peripheral power down (MIF_MLPL_PER_PDN)

20.6.3.9.1 Offset

Register	Offset
MIF_MLPL_PER_PDN	70h

20.6.3.9.2 Function

Peripheral power down control.

20.6.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

20.6.3.9.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means power down at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.10 MIF MLPL control of INITN (MIF_MLPL_INITN)

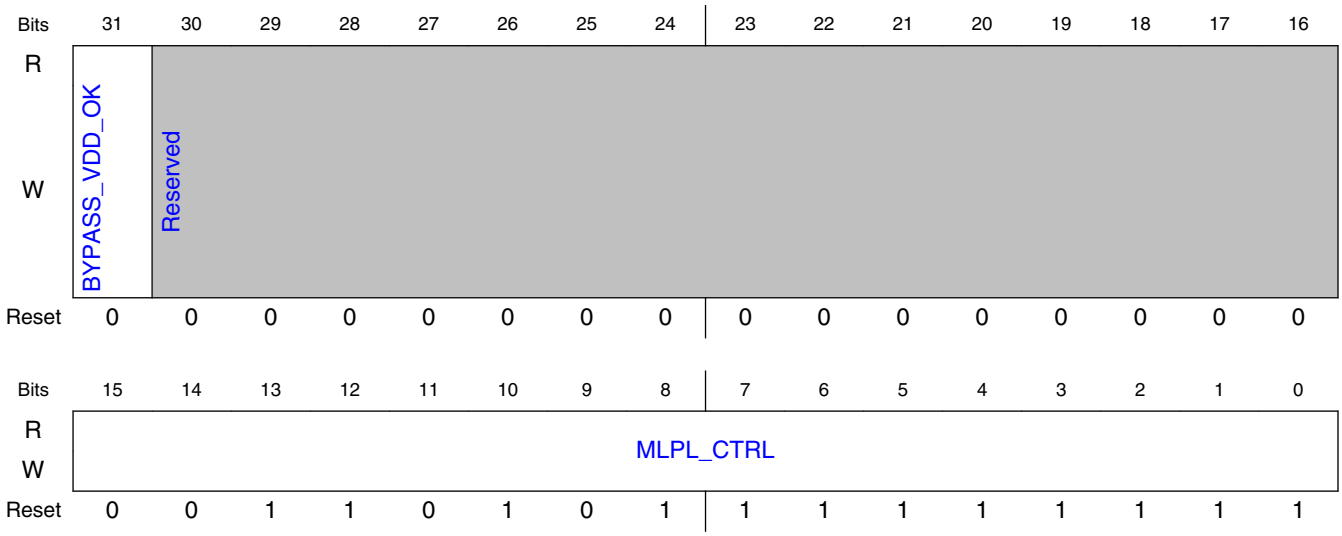
20.6.3.10.1 Offset

Register	Offset
MIF_MLPL_INITN	80h

20.6.3.10.2 Function

INITN pin control.

20.6.3.10.3 Diagram



20.6.3.10.4 Fields

Field	Function
31 BYPASS_VDD_OK	Bypass vdd_ok. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
30-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means assert the signal at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.3.11 MIF MLPL control of isolation enable (MIF_MLPL_ISO)

20.6.3.11.1 Offset

Register	Offset
MIF_MLPL_ISO	B0h

20.6.3.11.2 Function

Isolation control.

20.6.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPL_CTRL															
Reset	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

20.6.3.11.4 Fields

Field	Function
31-16 —	Reserved
15-0 MLPL_CTRL	Signal behavior at each MLPL 1 means enable the isolation at the MLPL. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

20.6.4 PGMC_PPC register descriptions

20.6.4.1 pgmc_ppc memory map

PGMC_PPC0 base address: 40C8_B000h

Offset	Register	Width (In bits)	Access	Reset value
4h	PPC Authentication Control (PPC_AUTHEN_CTRL)	32	Table 20-5	0000_0F00h
10h	PPC Mode (PPC_MODE)	32	Table 20-5	0000_0000h
14h	PPC standby CPU mode control (PPC_STBY_CM_CTRL)	32	Table 20-5	0000_0000h
18h	PPC standby Setpoint control (PPC_STBY_SP_CTRL)	32	RW	0000_0000h

20.6.4.2 PPC Authentication Control (PPC_AUTHEN_CTRL)

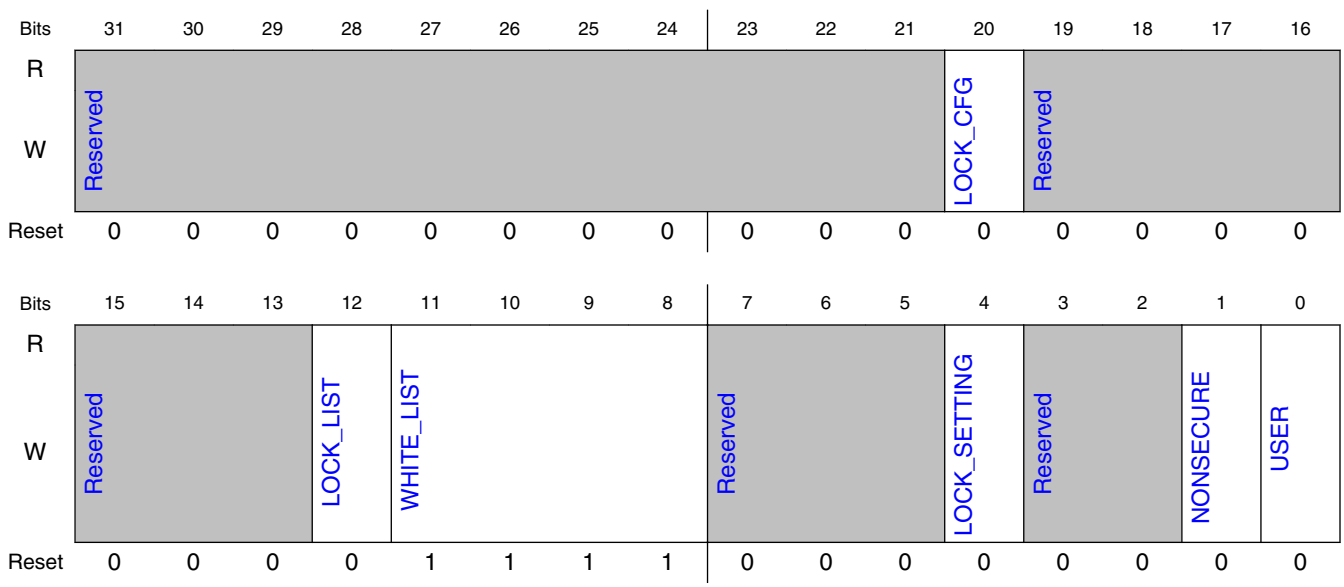
20.6.4.2.1 Offset

Register	Offset
PPC_AUTHEN_CTRL	4h

20.6.4.2.2 Function

Authentication control of PMIC power controller.

20.6.4.2.3 Diagram



20.6.4.2.4 Fields

Field	Function
31-21 —	Reserved
20 LOCK_CFG	Configuration lock this field is write once and can lock value of low power configuration fields
19-13 —	Reserved
12 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field

Table continues on the next page...

Field	Function
11-8 WHITE_LIST	Domain ID white list when bit setting to 1, the corresponding domain ID can access PMIC power control registers. bit[0] for Domain 0, bit[1] for Domain 1, bit[2] for Domain 2, and bit[3] for Domain 3. Muliple bits with a value of 1 is permitted.
7-5 —	Reserved
4 LOCK_SETTING	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
3-2 —	Reserved
1 NONSECURE	Allow non-secure mode access Allow both secure and non-secure mode to access PMIC power control registers
0 USER	Allow user mode access Allow both privilege and user mode to access PMIC power control registers

20.6.4.3 PPC Mode (PPC_MODE)

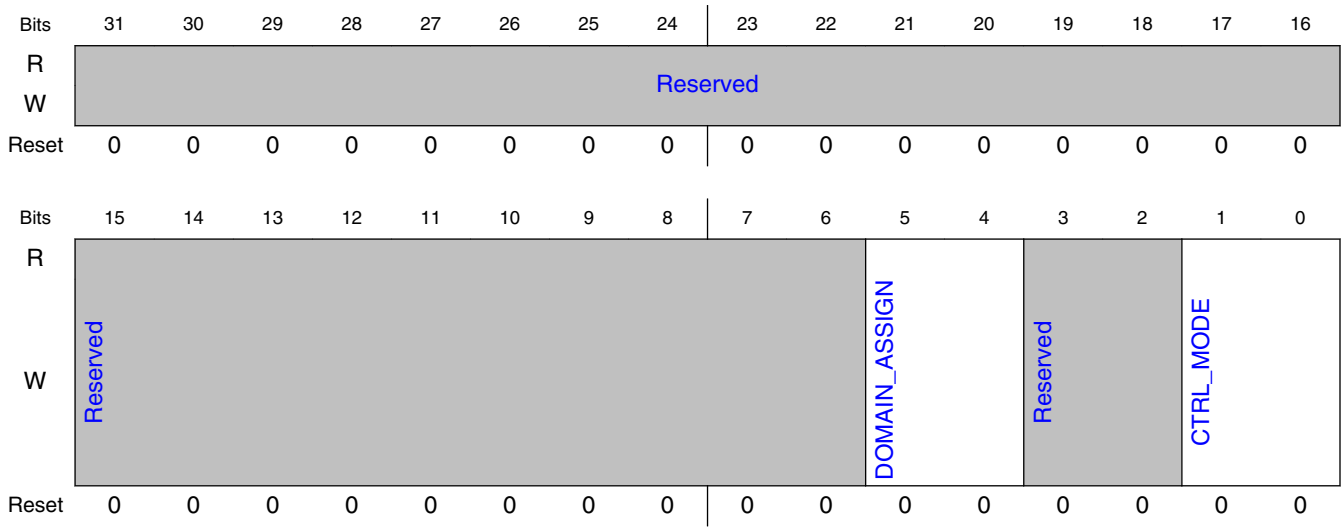
20.6.4.3.1 Offset

Register	Offset
PPC_MODE	10h

20.6.4.3.2 Function

PMIC power control mode.

20.6.4.3.3 Diagram



20.6.4.3.4 Fields

Field	Function
31-6 —	Reserved
5-4 DOMAIN_ASSIGN	Domain assignment of the BPC Valid when CTRL_MODE is 0x1. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Domain 0 01b - Domain 1 10b - Domain 2 11b - Domain 3
3-2 —	Reserved
1-0 CTRL_MODE	Control mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. 00b - Not affected by any low power mode 01b - Controlled by CPU power mode of the domain 10b - Controlled by Setpoint and system standby 11b - Reserved

20.6.4.4 PPC standby CPU mode control (PPC_STBY_CM_CTRL)

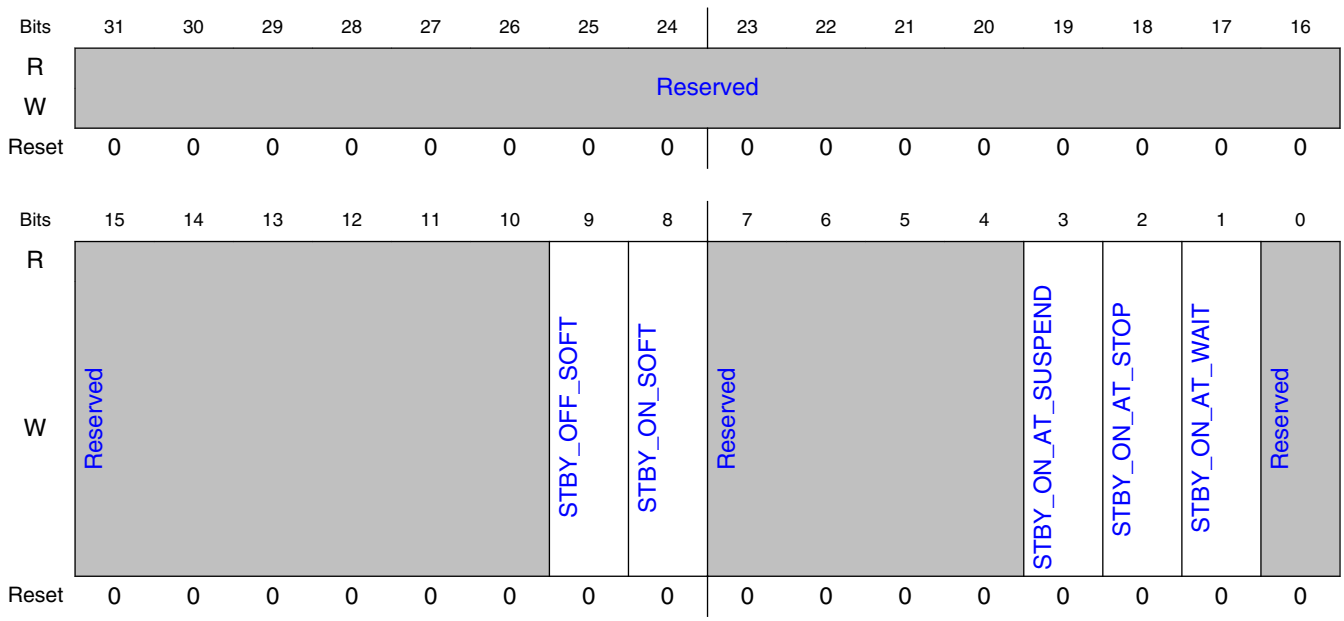
20.6.4.4.1 Offset

Register	Offset
PPC_STBY_CM_CTRL	14h

20.6.4.4.2 Function

PMIC standby control under CPU mode.

20.6.4.4.3 Diagram



20.6.4.4.4 Fields

Field	Function
31-10 —	Reserved
9 STBY_OFF_SOFT	Software PMIC standby off trigger Writing 0x1 to trigger PMIC standby off
8 STBY_ON_SOFT	Software PMIC standby on trigger Writing 0x1 to trigger PMIC standby on
7-4 —	Reserved
3 STBY_ON_AT_SUSPEND	PMIC Standby on when domain enters SUSPEND mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
2	PMIC Standby on when domain enters STOP mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.

Table continues on the next page...

Field	Function
STBY_ON_AT_STOP	
1 STBY_ON_AT_WAIT	PMIC Standby on when domain enters WAIT mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field.
0 —	Reserved

20.6.4.5 PPC standby Setpoint control (PPC_STBY_SP_CTRL)

20.6.4.5.1 Offset

Register	Offset
PPC_STBY_SP_CTRL	18h

20.6.4.5.2 Function

PMIC standby control under setpoint.

20.6.4.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STBY_ON_AT_SP_SLEEP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_ON_AT_SP_ACTIVE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.6.4.5.4 Fields

Field	Function
31-16 STBY_ON_AT_SP_SLEEP	PMIC standby on when system enters Setpoint number and system is in standby mode. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. Valid when CTRL_MODE is 0x2

Table continues on the next page...

Field	Function
15-0 STBY_ON_AT_SP_ACTIVE	PMIC standby on when system enters Setpoint number. This field is locked by AUTHEN_CTRL[LOCK_CFG] field. Valid when CTRL_MODE is 0x2

Chapter 21
DCDC Converter (DCDC)

21.1 Chip-specific DCDC information

Table 21-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

The on-chip DCDC converter is suitable for consumer and industrial applications up to the rated temperature specified in their respective market datasheets. For automotive applications, please check the automotive datasheet or consult with your NXP representative for appropriate power supply requirements.

21.2 Overview

The DCDC Converter (DCDC) is a synchronous buck mode DC-DC converter with two outputs. The DCDC is used for generating the power supply for the chip's core logic.

21.2.1 Block diagram

The block diagram of the DCDC module is provided below:

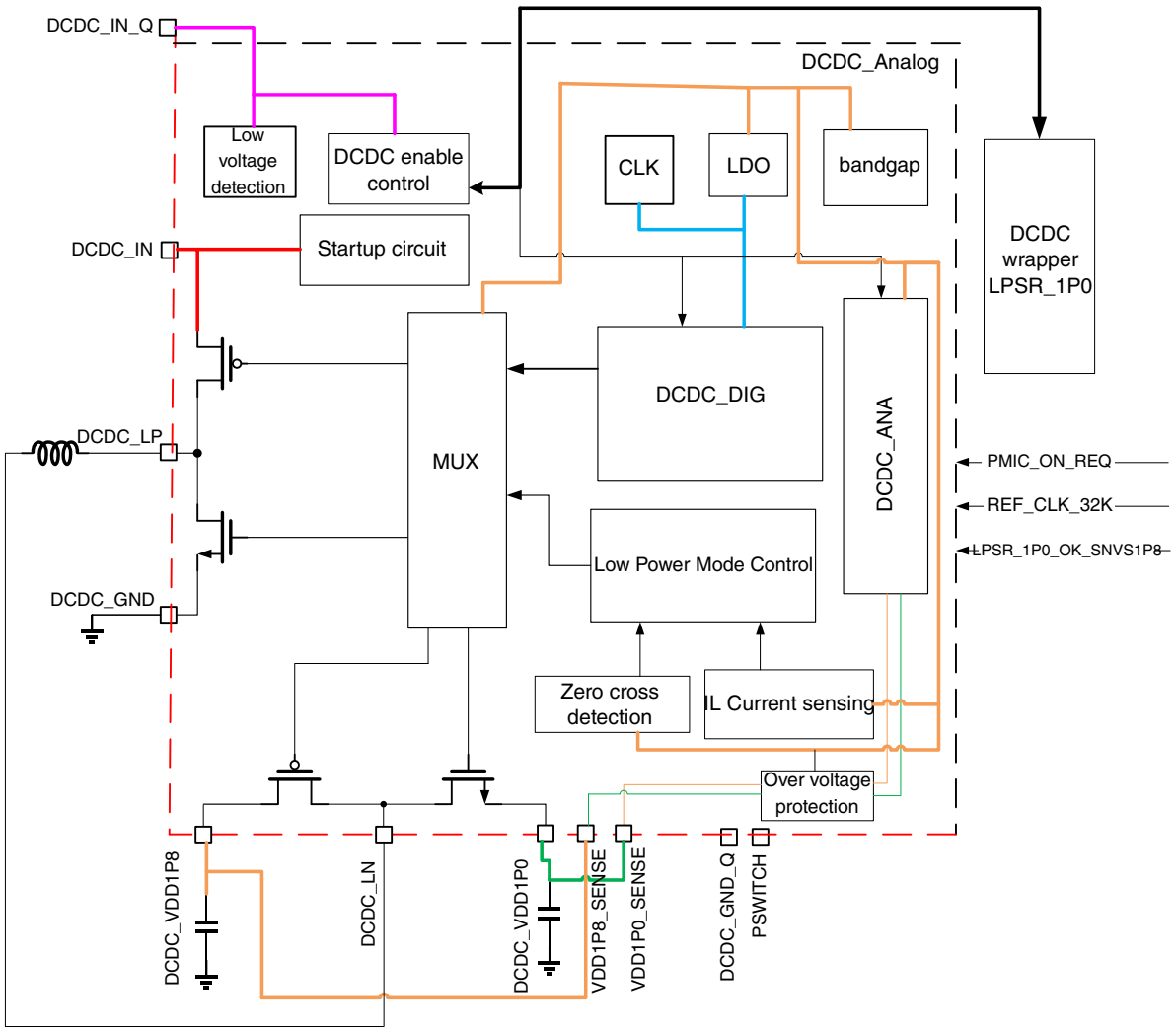


Figure 21-1. DCDC block diagram

21.2.2 Features

The DCDC includes the following features:

- Buck mode only, 3.3V input with 1.8V and 1.0V outputs.
- DCDC_PSWITCH pin to turn on DCDC
- Continuous conduction mode (CCM) and discontinuous conduction mode (DCM) operation.
- RUN mode and LP (low-power) mode

    - RUN mode supports CCM and DCM
- Automatic pulse skipping
- Overcurrent and overvoltage protection
- Low-input voltage detection

21.3 Functional description

The DCDC converter is a synchronous step-down converter with two outputs using one inductor. In run mode, it can operate on continuous conduction mode or discontinuous conduction mode, depending on the level of the load current. Discontinuous conduction mode needs to be enabled through register configuration.

DCDC can be configured to operate on LP (low-power) mode when the load current is less than 50mA. During the LP mode, the converter operates in pulse mode with a minimum quiescent current to maintain high efficiency.

21.3.1 Application mode

DCDC has one application mode, which is configured by the DCDC_MODE Pad:

- **Application mode 0**: DCDC_IN is connected to 3.3 V, and DCDC output is 1.8 V and 1.0 V. DCDC_MODE is tied to ground.

21.3.2 Functional Modes

- Run Mode
    - Continuous Conduction Mode
    - Discontinuos Conduction Mode - improved efficency and same working state as Continuous Conduction Mode
- Low-Power Mode
    - Improved efficiency
    - Ripple is a little bit higher than Run Mode

21.3.3 Operation

The DCDC is a synchronous buck converter. An illustration of the power switch arrangment is shown below.

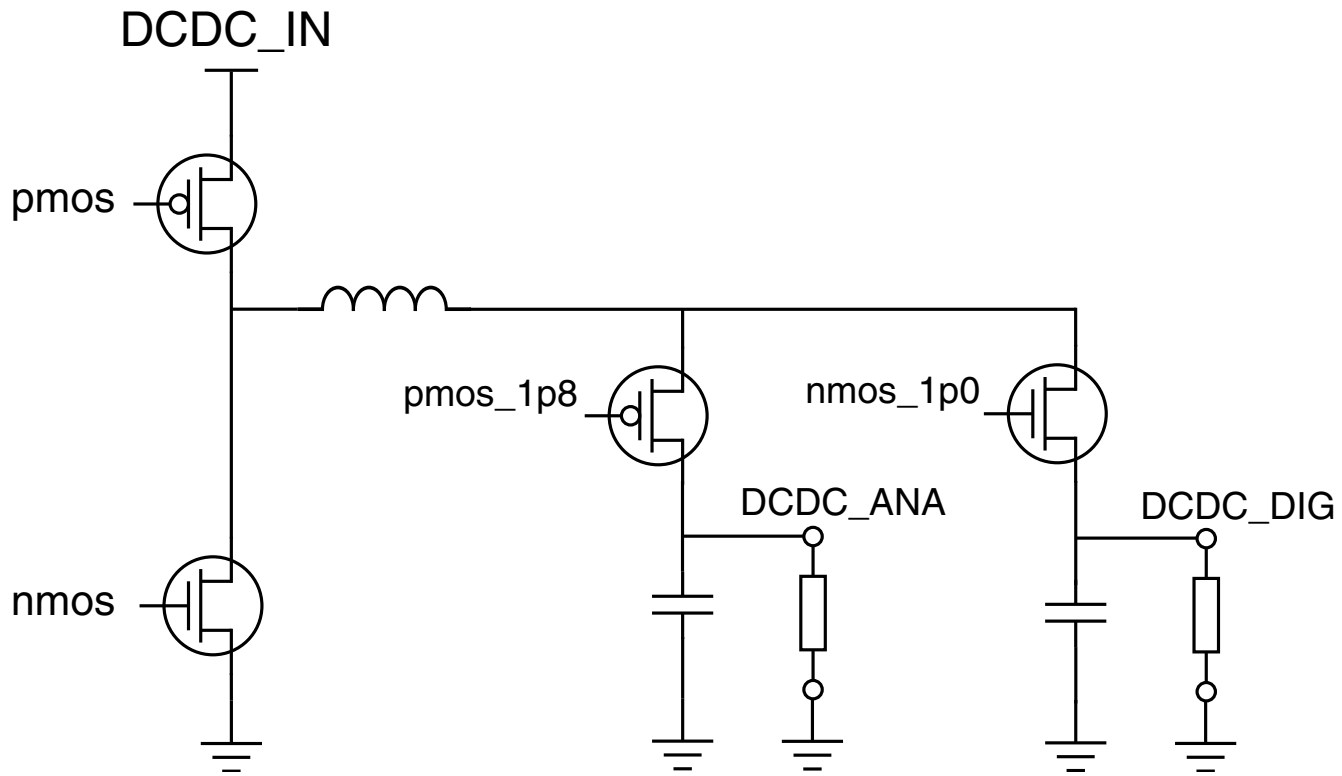


Figure 21-2. DCDC Power Switches

The DCDC features zero cross detection for Discontinuos Conduction Mode. This functionality is enabled by the REG0[PWD_ZCD] bit. pmos_1p8 and nmos_1p0 would not be turned on simultaneously. The DCDC control loop will determine which output is more starved for charging, and turn on it's switch. When one output is charged, the other one is sustained by the big capacitor.

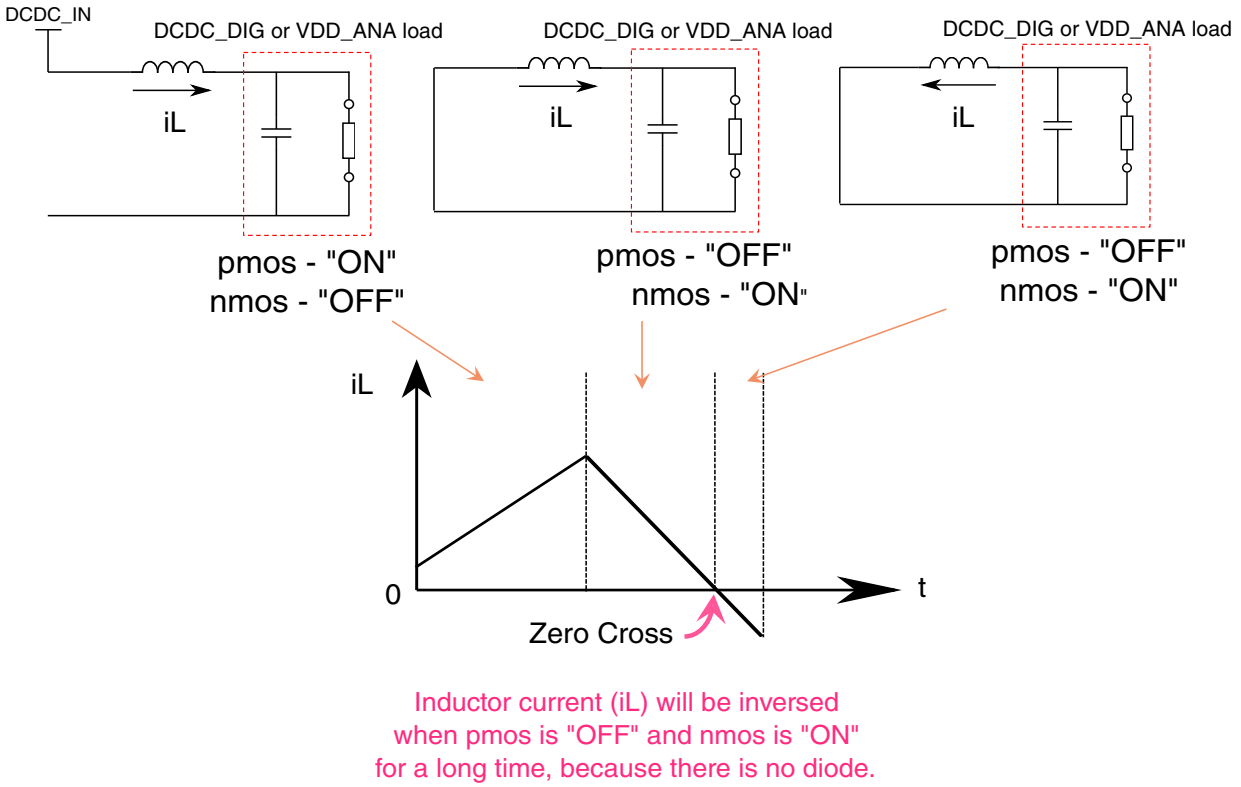


Figure 21-3. Zero Cross Detection

Working as the CCM, the inductor current (iL) will be inversed during the cycle of pmos is "OFF" and nmos is "ON" if the current loading is half of the current ripple of the inductor. Working as the DCM, zero cross detector will be turned on, it will turn off nmos to stop the current of inductor from being inversed, so the current of the inductor will remain zero until the next charging cycle starts. Therefore DCM can improve the efficiency.

21.3.4 Overcurrent, Overvoltage, and Low Voltage Detection

DCDC supports the following three protection functions:

1. Overcurrent protection
   - In run mode, DCDC will shut down when detecting abnormal large current in the P-type power switch (threshold - 1.5A or 2A, which can be programmed by register).
   - In power-save mode, DCDC will stop charging the inductor when detecting large current in the P-type power switch (threshold - 200 to 250mA).
2. Overvoltage protection - DCDC will shut down when detecting high voltage on either of the two outputs (typical 2.5V for DCDC_ANA and 1.5V for DCDC_DIG).
3. Low-Voltage Detection - DCDC will shut down when detecting the input voltage is too low (typical 2.6V for DCDC_IN).

NOTE

Please see the Chip Datasheet for the specified values.

21.3.5 Clocks

This section describes clocks and special clocking requirements for the DCDC module.

Table 21-2. Clocks

Clock	Description
ipg_clk	Peripheral Clock
clk_24M	24 MHz reference clock

The diagram below illustrates the logical selection of the 24M clock when auto clock switching is disabled:

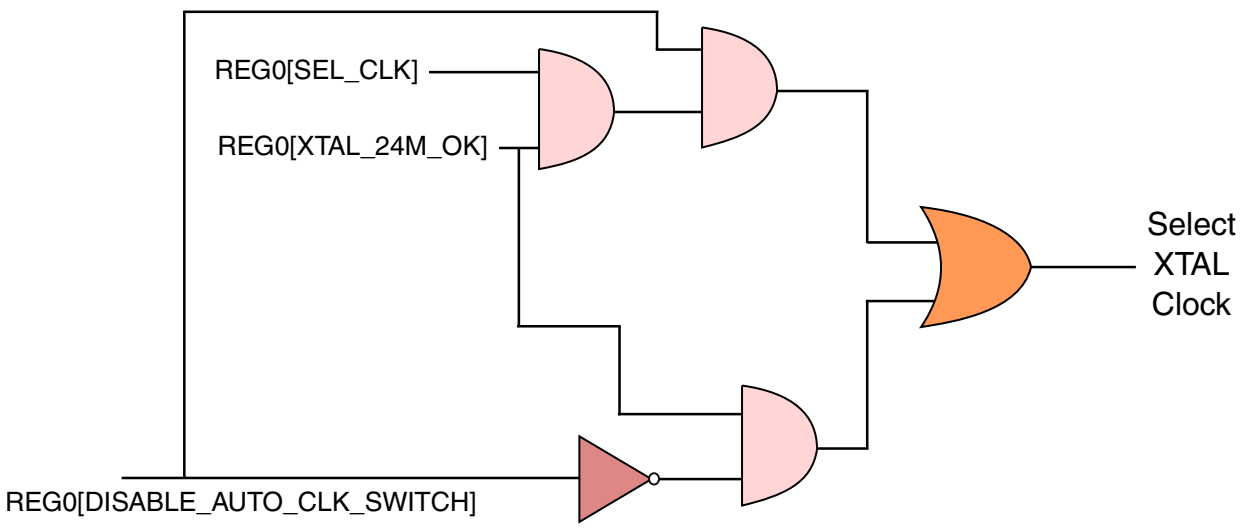


Figure 21-4. 24M Clock Selection

21.3.6 Interrupts

Please see the IRQ chapter for any assigned interrupts.

21.4 External Signals

Table 21-3. External Signals

External Signals	Power Domain	Description	Direction
DCDC_ANA	DCDC_ANA	DCDC 1P8 output	O
DCDC_ANA_SENSE	DCDC_ANA	The PAD used to provide 1P8V power to DCDC analog module and the sense point for DCDC feedback loop. It is connected to the 1.8V output of DCDC on the board.	I
DCDC_DIG	DCDC_DIG	DCDC 1P0 output	O
DCDC_DIG_SENSE	DCDC_DIG	The PAD used to provide 1P0V power to DCDC analog module and the sense point for DCDC feedback loop. It is connected to the 1.0V output of DCDC on the board.	I
DCDC_IN	DCDC_IN	The DCDC input which is connected to the power switch of DCDC	I
DCDC_LP	DCDC_IN	The PAD connected to one terminal of the external inductor on the board.	IO
DCDC_LN	DCDC_IN	The PAD connected to another terminal of the external inductor on the board.	IO
DCDC_GND	DCDC_IN	Ground PAD which is connected to DCDC power switch	O
DCDC_IN_Q	DCDC_IN	Analog power supply used by some analog modules of DCDC. It is from the same source as DCDC_IN, but is split from DCDC_IN, in order to make it more quiet.	I
DCDC_GND_Q	DCDC_IN	Ground PAD used by the analog module of DCDC	O
DCDC_PSWITCH	DCDC_IN	The PAD which can bypass the DCDC	I
DCDC_MODE	DCDC_IN	The PAD which is used to distinguish between application mode “0” and “1”. Please refer to the Application Mode section for more information.	I
lpsr_1p0_dig_ok	SNVS_1P8	The signal in SNVS 1P8 domain. It indicates that LPSR 1P0 power is OK.	I
PMIC_ON_REQ	SNVS_1P8	The request signal from SNVS 1P8 to turn on DCDC	I

21.5 Initialization

To enable DCDC, the following conditions must be met:

1. Power supply SNVS_1P8, LPSR_1P0 and DCDC_IN are available
2. 32kHz Clock in the SNVS_1P8 domain is available
3. Provide DCDC with 1ms of reset time after DCDC_IN powers up

4. PMIC_ON_REQ && lpsr_1p0_dig_ok && CTRL0[ENABLE] && DCDC_PSWITCH = 1'b1

The Power up sequence is:

- Power SNVS_1P8
- DCDC_IN powered up
- Delay 1ms reset time
- PMIC_ON_REQ && LPSR_1P0_OK_SNVS1P8 && DCDC_EN && DCDC_PSWITCH = 1'b1
- DCDC startup

21.6 Application Information

21.6.1 Typical Application

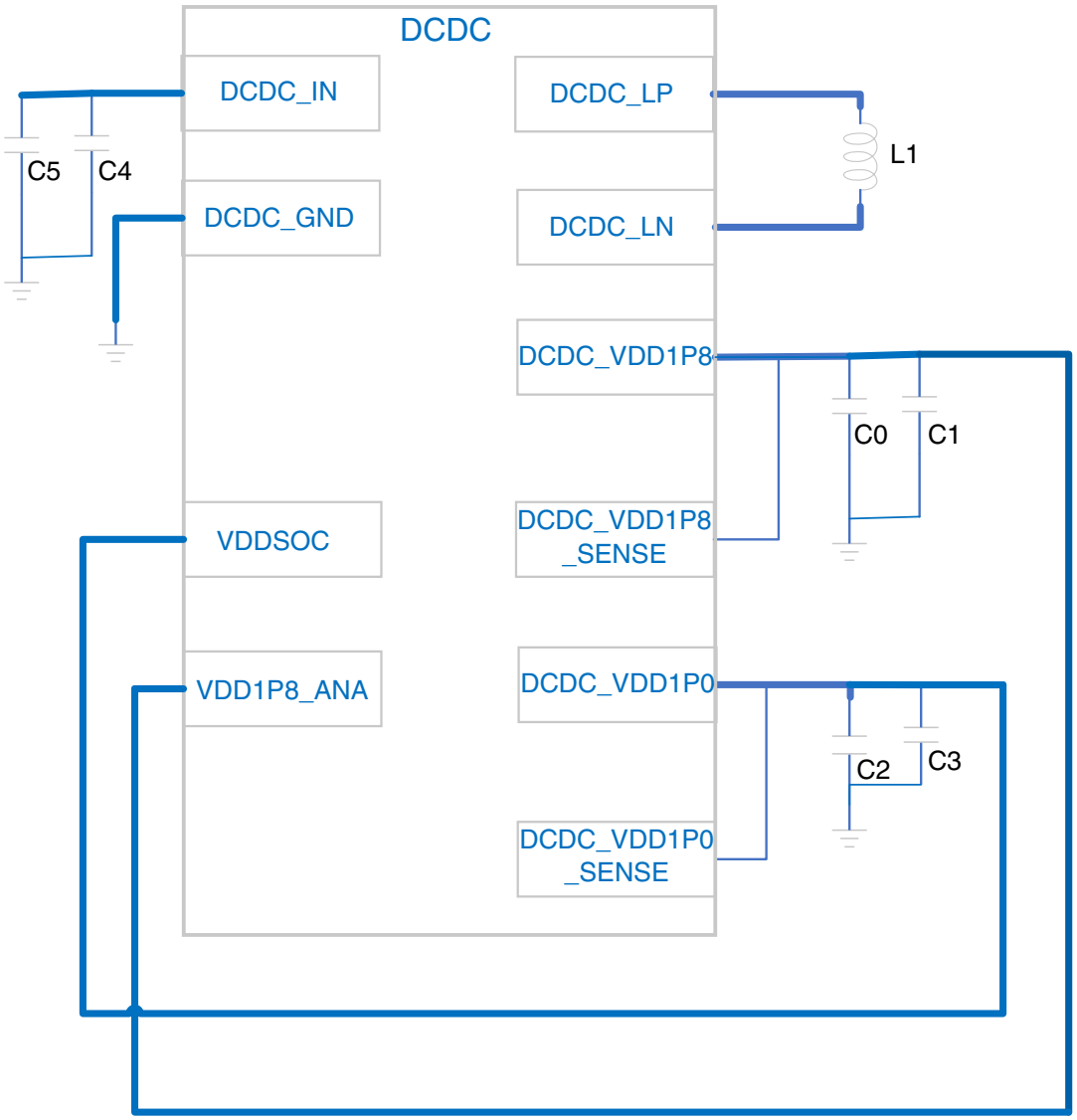


Figure 21-5. DCDC

Table 21-4. DCDC Component List

Component	Description	Recommendation
L1	Inductance needed by DCDC	4.7uH with low DC resistance (less than 0.1 ohm ESR) inductor to minimize conduction losses.
C0	High-frequency decoupling capacitor	At least 0.1uF.
C1	Filter capacitor on DCDC_ANA output	A 33uF with low ESR (less than 0.02 ohm ESR) capacitor for small ripple.
C2	High frequency decoupling capacitor	At least 0.1uF.
C3	Filter capacitors DCDC_DIG output	A 33uF with low ESR (less than 0.02 ohm ESR) capacitor for small ripple. Need a 66uF on DCDC_DIG output if the loading on it is larger than 500mA.

Table continues on the next page...

Table 21-4. DCDC Component List (continued)

Component	Description	Recommendation
C4	High-frequency decoupling capacitor for DCDC_IN	At least 0.1uF. The parasitic inductance of the routing from C4 to DCDC_IN, of the chip on the board, should be less than 0.5nH.
C5	Filter capacitor of DCDC_IN.	A 10uF (or greater) with low ESR capacitor. Increase capacitance (no limit) for better input-voltage filtering.

NOTE

It is very important to place C0, C1, C2, C3, C4 as close to the pin as possible.

NOTE

Please see the chip datasheet for specification details.

21.6.2 Recommended configuration

To improve the efficiency for light loading in Run mode:

- Set DCDC to DCM mode:
  - REG1[RLOAD_REG_EN_LPSR] = 1'b0
  - REG0[PWD_ZCD] = 1'b0
  - REG3[DISABLE_IDLE_SKIP] = 1'b0
  - REG3[DISABLE_PULSE_SKIP] = 1'b0
  - REG0[PWD_CMP_OFFSET] = 1'b0
  - REG2[LOOPCTRL_EN_RCSCALE] = 3'b101;
  - REG1[DM_CTRL] = 1’b1;
- To improve transient performance with a big loading step:
  - Set DCDC to DCM Mode
  - Set the ANA_STYBY_TRG_SP[3:0] and DIG_STBY_TRG_SP[3:0] one step lower than the Run Mode targets (e.g. If Run Mode targets are 1.0V and 1.8V, set Low-Power Mode targets to 0.975V and 1.775V).
  - Set REG3[ENABLE_FF] = 1’b1

NOTE

The STBY_TRG_SP setting does not affect DCDC in Run Mode. Setting these fields improves ripple when the Low-Voltage Detection (LVD) module detects outputs that are lower than the STBY_TRG_SP values.

- To improve loading ability under heavy load:
  - Set CTRL0[DEBUG_BITS] = 12'b000000000001

21.7 Memory Map and register definition

This section includes the DCDC module memory map and detailed descriptions of all registers.

21.7.1 DCDC register descriptions

21.7.1.1 DCDC memory map

DCDC base address: 40CA_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DCDC Control Register 0 (CTRL0)	32	RW	0000_0003h
4h	DCDC Control Register 1 (CTRL1)	32	RW	0F0B_100Ch
8h	DCDC Register 0 (REG0)	32	RW	0403_0511h
Ch	DCDC Register 1 (REG1)	32	RW	01CD_5430h
10h	DCDC Register 2 (REG2)	32	RW	0210_8089h
14h	DCDC Register 3 (REG3)	32	RW	0098_0000h
18h	DCDC Register 4 (REG4)	32	RW	0000_0000h
1Ch	DCDC Register 5 (REG5)	32	RW	0000_0000h
20h	DCDC Register 6 (REG6)	32	RW	0000_0000h
24h	DCDC Register 7 (REG7)	32	RW	0000_0000h
28h	DCDC Register 7 plus (REG7P)	32	RW	0000_0000h
2Ch	DCDC Register 8 (REG8)	32	RW	0000_0000h
30h	DCDC Register 9 (REG9)	32	RW	0000_0000h
34h	DCDC Register 10 (REG10)	32	RW	0000_0000h
38h	DCDC Register 11 (REG11)	32	RW	0000_0000h
3Ch	DCDC Register 12 (REG12)	32	RW	0000_0000h
40h	DCDC Register 13 (REG13)	32	RW	0000_0000h
44h	DCDC Register 14 (REG14)	32	RW	0000_0000h
48h	DCDC Register 15 (REG15)	32	RW	0000_0000h
4Ch	DCDC Register 16 (REG16)	32	RW	0000_0000h
50h	DCDC Register 17 (REG17)	32	RW	0000_0000h
54h	DCDC Register 18 (REG18)	32	RW	0000_0000h
58h	DCDC Register 19 (REG19)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5Ch	DCDC Register 20 (REG20)	32	RW	0000_0000h
60h	DCDC Register 21 (REG21)	32	RW	0000_0000h
64h	DCDC Register 22 (REG22)	32	RW	0000_0000h
68h	DCDC Register 23 (REG23)	32	RW	0000_0000h
6Ch	DCDC Register 24 (REG24)	32	RW	0010_0000h

21.7.1.2 DCDC Control Register 0 (CTRL0)

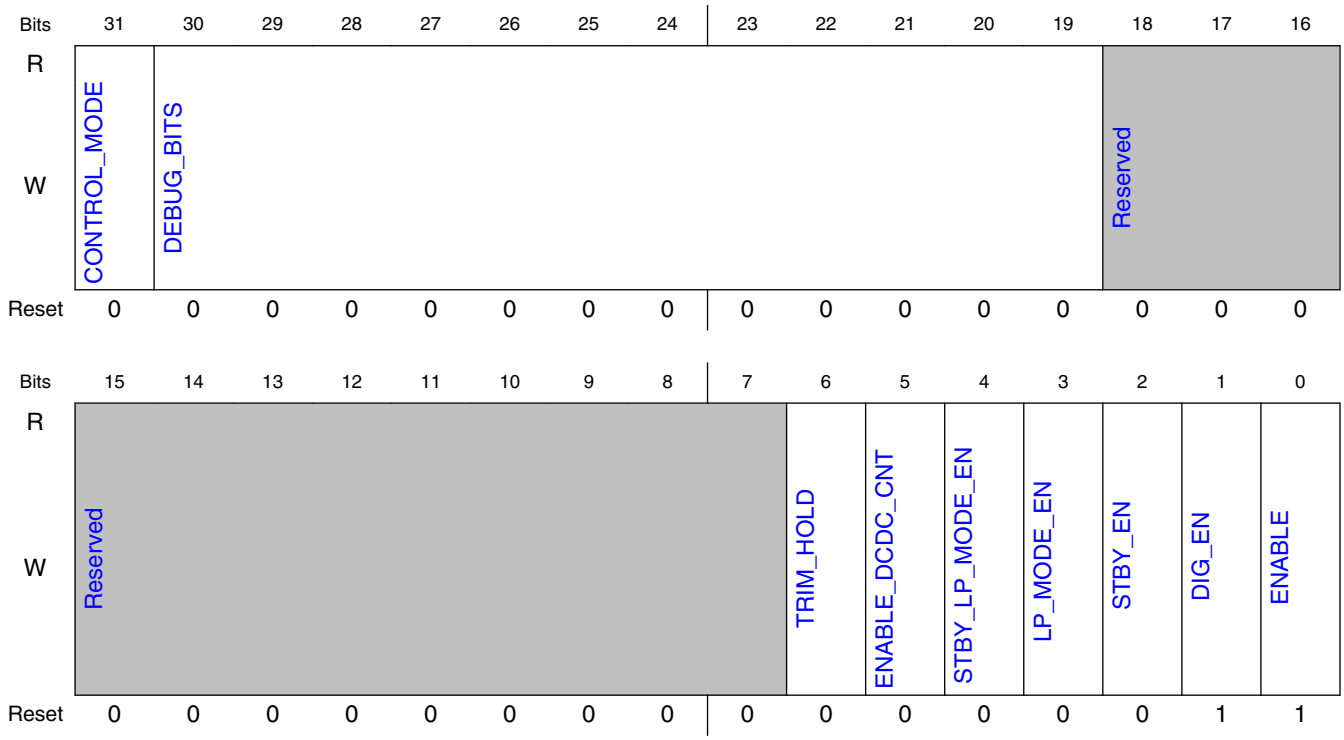
21.7.1.2.1 Offset

Register	Offset
CTRL0	0h

21.7.1.2.2 Function

DCDC Control register 0

21.7.1.2.3 Diagram



21.7.1.2.4 Fields

Field	Function
31 CONTROL_MODE	Control mode 0b - Software control mode 1b - Hardware control mode (controlled by GPC Setpoints)
30-19 DEBUG_BITS	DEBUG_BITS[11:0] debug_bits[0]: "1" can improve loading ability but the leakage on DCDC_IN will also increase, so set it to "1" only in the case of heavy loading which is larger than 500mA. debug_bits[1]: For debug, "1" can double the charge current of 1.8V output during startup, just keep it as default value "0". debug_bits[2]: For debug, "1" can disable the low voltage detector for 1.8V output, just keep it as default value "0" debug_bits[3]: "1" can double the refresh frequency of internal bandgap of DCDC in low power mode.
18-7 —	Reserved
6 TRIM_HOLD	Hold trim input 0b - Sample trim input 1b - Hold trim input
5	Enable internal count for DCDC_OK timeout

Table continues on the next page...

Field	Function
ENABLE_DCDC_CNT	0b - Wait DCDC_OK for ACK 1b - Enable internal count for DCDC_OK timeout
4 STBY_LP_MODE_EN	DCDC low-power mode enable by GPC standby request 0b - Disable DCDC entry into low-power mode from a GPC standby request 1b - Enable DCDC to enter into low-power mode from a GPC standby request
3 LP_MODE_EN	DCDC low-power (LP) mode enable **NOTE:** DCDC can't start up directly into LP mode. REG0[DCDC_EN] must be set first, before setting REG0[LP_MODE_EN] 1b - Enter into low-power mode
2 STBY_EN	DCDC standby mode enable 1b - Enter into standby mode
1 DIG_EN	Enable the DCDC_DIG switching converter output. The switching converter is enabled by default. Always set to 1. 0b - Reserved 1b - Enable
0 ENABLE	DCDC Enable 0b - Disable (Bypass) 1b - Enable

21.7.1.3 DCDC Control Register 1 (CTRL1)

21.7.1.3.1 Offset

Register	Offset
CTRL1	4h

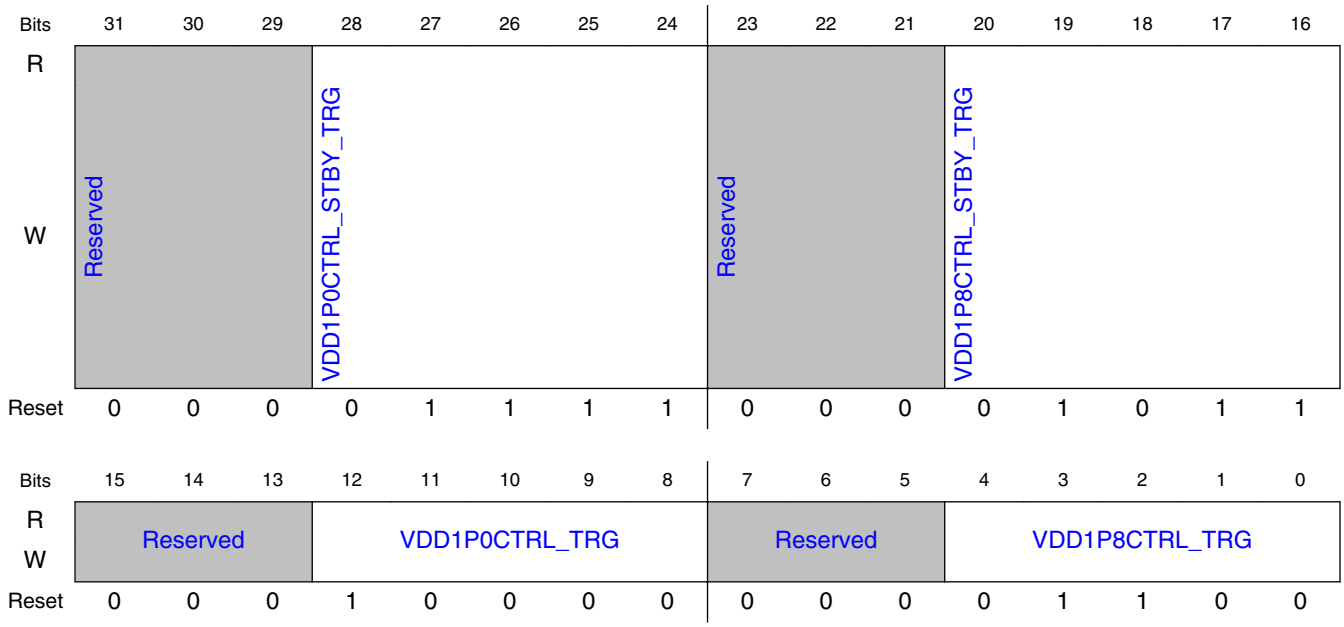
21.7.1.3.2 Function

DCDC Control Register 1

NOTE

VDD1P0CTRL_STBY_TRG and VDD1P8CTRL_STBY_TRG are the target reg in low-power mode, not in standby mode.

21.7.1.3.3 Diagram



21.7.1.3.4 Fields

Field	Function
31-29 —	Reserved
28-24 VDD1P0CTRL_STBY_TRG	Target value of VDD1P0 in standby mode, 25mV each step from 0x00 to 0x1F: 0_0000b - 0.625V 0_1111b - 1.0V 1_1111b - 1.4V
23-21 —	Reserved
20-16 VDD1P8CTRL_STBY_TRG	Target value of VDD1P8 in standby mode, 25mV each step from 0x00 to 0x1F: 0_0000b - 1.525V 0_1011b - 1.8V 1_1111b - 2.3V
15-13 —	Reserved
12-8 VDD1P0CTRL_TRG	Target value of VDD1P0 in buck mode, 25mV each step from 0x00 to 0x1F: 0_0000b - 0.6V 1_0000b - 1.0V 1_1111b - 1.375V
7-5 —	Reserved
4-0	Target value of VDD1P8 in buck mode, 25mV each step from 0x00 to 0x1F:

Field	Function
VDD1P8CTRL_TRG	0_0000b - 1.5V 0_1100b - 1.8V 1_1111b - 2.275V

21.7.1.4 DCDC Register 0 (REG0)

21.7.1.4.1 Offset

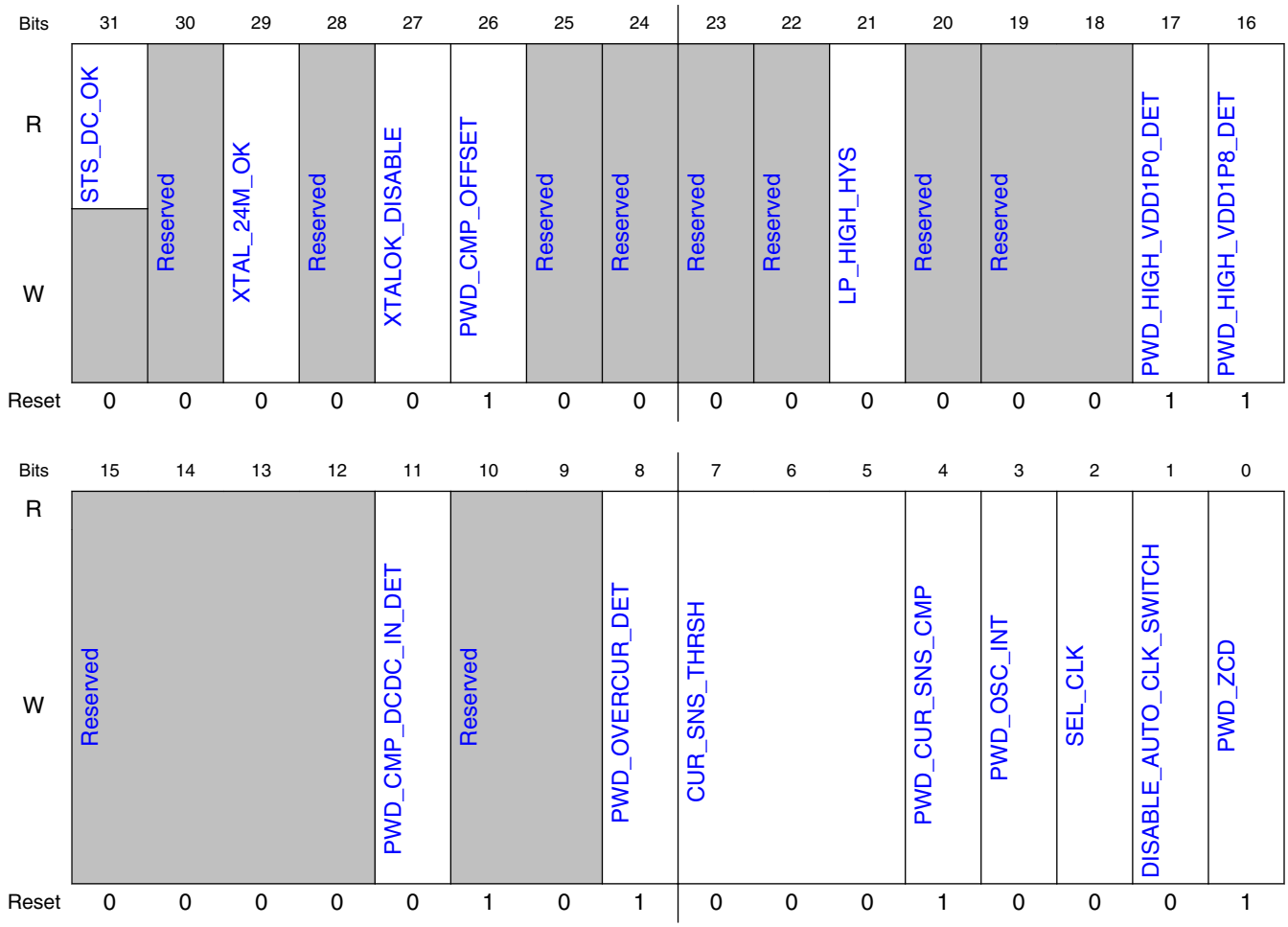
Register	Offset
REG0	8h

21.7.1.4.2 Function

DCDC register 0

This register controls various high-level functions of the DCDC

21.7.1.4.3 Diagram



21.7.1.4.4 Fields

Field	Function
31 STS_DC_OK	DCDC Output OK This is the status bit indicates if the DCDC output voltages (ANA and DIG) are stable. The lock time depends on the loading and the DCDC mode. 0b - DCDC is settling 1b - DCDC already settled
30 —	Reserved
29 XTAL_24M_OK	24M XTAL OK This bit determines if the DCDC uses the internal ring oscillator or the xtal 24M. 0b - DCDC uses internal ring oscillator 1b - DCDC uses xtal 24M
28	Reserved

Table continues on the next page...

Field	Function
—	
27 XTALOK_DISABLE	Disable xtalok detection circuit This is a debug bit which should not be changed in real case. This bit disables and bypasses the clock 24 MHz reference clock detector. 0b - Enable xtalok detection circuit 1b - Disable xtalok detection circuit and always outputs OK signal "1"
26 PWD_CMP_OFFSET	power down the out-of-range detection comparator The out-of-range comparator monitors the output voltage of DCDC. When the DCDC output voltage is out of range, this comparator will speed up DCDC control loop in an attempt to bring the DCDC output voltage back in range. PWD_CMP_OFFSET must be set to 0 (out-of-range comparator on) if LOOPCTRL_EN_RCSCALE is to be used to speed up the transient response. 0b - Out-of-range comparator powered up 1b - Out-of-range comparator powered down
25 —	Reserved
24 —	Reserved
23 —	Reserved
22 —	Reserved
21 LP_HIGH_HYS	Low Power High Hysteric Value Adjust hysteretic value in low power from 12.5mV to 25mV 0b - Adjust hysteretic value in low power to 12.5mV 1b - Adjust hysteretic value in low power to 25mV
20 —	Reserved
19-18 —	Reserved
17 PWD_HIGH_VDD1P0_DET	Power Down High Voltage Detection for VDD1P0 Power down overvoltage detection comparator for the VDD1P0 output. 0b - Overvoltage detection comparator for the VDD1P0 output is enabled 1b - Overvoltage detection comparator for the VDD1P0 output is disabled
16 PWD_HIGH_VDD1P8_DET	Power Down High Voltage Detection for VDD1P8 Power down overvoltage detection comparator for the VDD1P8 output. 0b - Overvoltage detection comparator for the VDD1P8 output is enabled 1b - Overvoltage detection comparator for the VDD1P8 output is disabled
15-12 —	Reserved
11 PWD_CMP_DCDC_IN_DET	Set to "1" to power down the low voltage detection comparator 0b - Low voltage detection comparator is enabled 1b - Low voltage detection comparator is disabled

Table continues on the next page...

Field	Function
10-9 —	Reserved
8 PWD_OVERCUR_DET	Power down overcurrent detection comparator 0b - Overcurrent detection comparator is enabled 1b - Overcurrent detection comparator is disabled
7-5 CUR_SNS_THRSH	Current Sense (detector) Threshold Set the threshold of current detector, if the peak current of the inductor exceeds the threshold, the current detector will assert. CUR_SNS_THRSH[1] is used to set overcurrent threshold in RUN mode - 0 is 1.5A, 1 is 2A CUR_SNS_THRSH[0] is used to set overcurrent threshold in LP (low-power) mode - 0 is 250mA, 1 is 200mA
4 PWD_CUR_SNS_CMP	Power down signal of the current detector Power down the control signal of the current detector. 0b - Current Detector powered up 1b - Current Detector powered down
3 PWD_OSC_INT	Power down internal ring oscillator Only set this bit when 24 MHz xtal oscillator is available. 0b - Internal ring oscillator powered up 1b - Internal ring oscillator powered down
2 SEL_CLK	Select Clock Select 24 MHz Crystal clock for DCDC, **when DISABLE_AUTO_CLK_SWITCH is set.** If DISABLE_AUTO_CLK_SWITCH is set to 0 and 24M xtal is OK, the clock source will switch from internal ring oscillator to 24M xtal automatically. If DISABLE_AUTO_CLK_SWITCH is set to 1, SEL_CLK will determine which clock source the DCDC uses: if SEL_CLK=0, DCDC will use internal ring oscillator, if SEL_CLK=1, DCDC will use 24M xtal. 0b - DCDC uses internal ring oscillator 1b - DCDC uses 24M xtal
1 DISABLE_AUTO_CLK_SWITCH	Disable Auto Clock Switch Disable automatic clock switch from internal oscillator to xtal clock. 0b - If DISABLE_AUTO_CLK_SWITCH is set to 0 and 24M xtal is OK, the clock source will switch from internal ring oscillator to 24M xtal automatically 1b - If DISABLE_AUTO_CLK_SWITCH is set to 1, SEL_CLK will determine which clock source the DCDC uses
0 PWD_ZCD	Power Down Zero Cross Detection Power down the zero cross detection function for discontinuous conductor mode. 0b - Zero cross detetion function powered up 1b - Zero cross detetion function powered down

21.7.1.5 DCDC Register 1 (REG1)

21.7.1.5.1 Offset

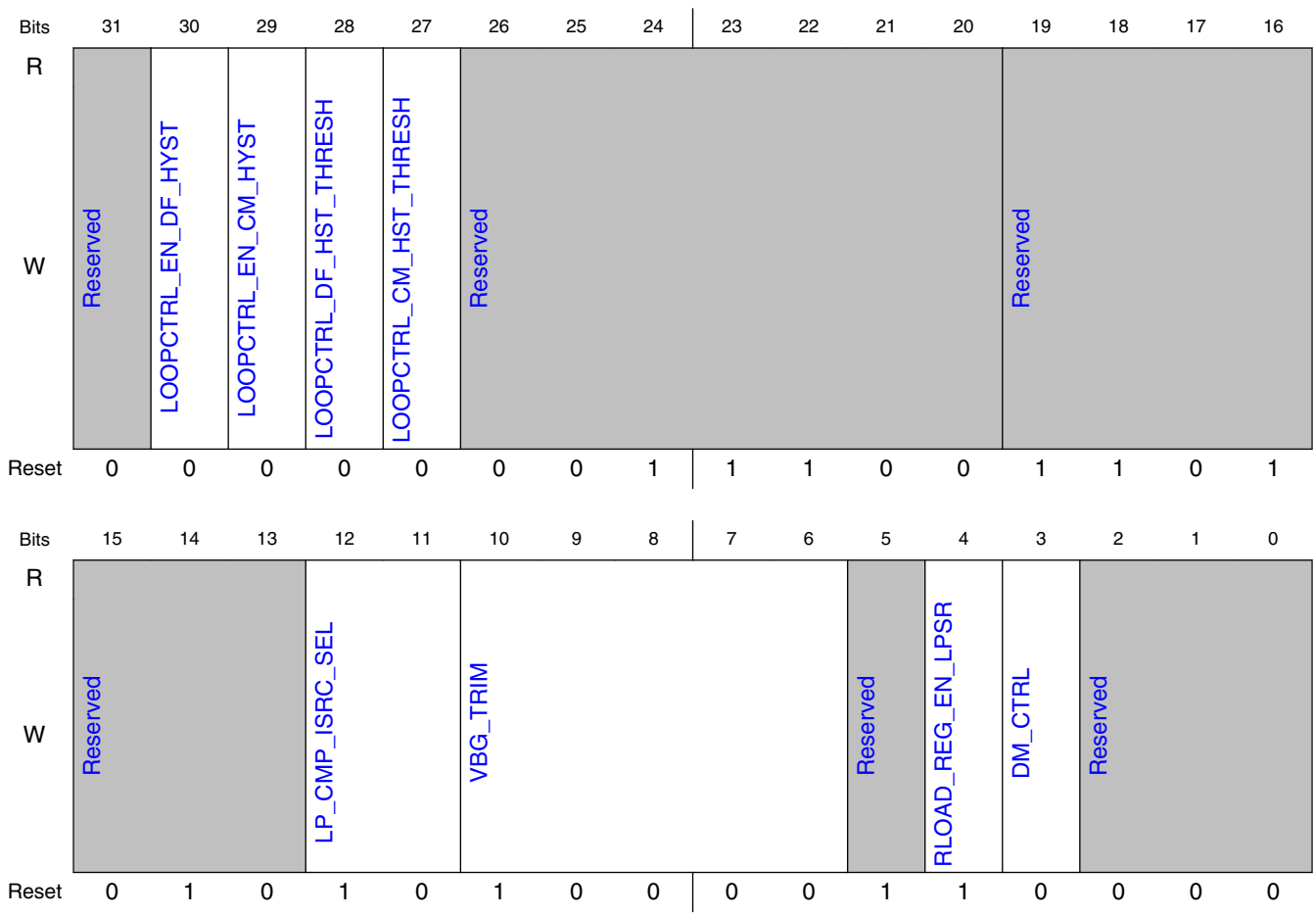
Register	Offset
REG1	Ch

21.7.1.5.2 Function

DCDC register 1

This register controls various high-level functions of the DCDC

21.7.1.5.3 Diagram



21.7.1.5.4 Fields

Field	Function
31	Reserved

Table continues on the next page...

Field	Function
—	
30 LOOPCTRL_EN_DF_HYST	Enable hysteresis in switching converter differential mode analog comparators. This feature will improve transient supply ripple and efficiency 0b - Disable hysteresis in switching converter differential mode analog comparators 1b - Enable hysteresis in switching converter differential mode analog comparators
29 LOOPCTRL_EN_CM_HYST	Enable hysteresis in switching converter common mode analog comparators. This feature will improve transient supply ripple and efficiency 0b - Disable hysteresis in switching converter common mode analog comparators 1b - Enable hysteresis in switching converter common mode analog comparators
28 LOOPCTRL_DF_HST_THRESH	Increase Threshold Detection Increase the threshold detection for differential mode analog comparators.
27 LOOPCTRL_CM_HST_THRESH	Increase Threshold Detection Increase the threshold detection for common mode analog comparators.
26-20 —	Reserved
19-13 —	Reserved
12-11 LP_CMP_ISRC_SEL	Low Power Comparator Current Bias set the current bias of low power comparator. 00b - 50nA 01b - 100nA 10b - 200nA 11b - 400nA
10-6 VBG_TRIM	Trim Bandgap Voltage Trim step is 3mV. 0_0000b - 0.452V 1_0000b - 0.5V 1_1111b - 0.545V
5 —	Reserved
4 RLOAD_REG_EN_LPSR	Load Resistor Enable This controls the load resistor of the internal regulator of DCDC. Connecting the load resistor to the internal LDO in RUN mode, improves the startup ripple of the LDO, but the load resistor will consume additional current. Once the STS_DC_OK asserts, the load resistor can be disconnected. 0b - Disconnect load resistor 1b - Connect load resistor
3 DM_CTRL	DM Control Set it to "1" when DCM is enabled. 0b - No change to ripple when the discontinuous current is present in DCM. 1b - Improves ripple when the inductor current goes to zero in DCM.
2-0 —	Reserved

21.7.1.6 DCDC Register 2 (REG2)

21.7.1.6.1 Offset

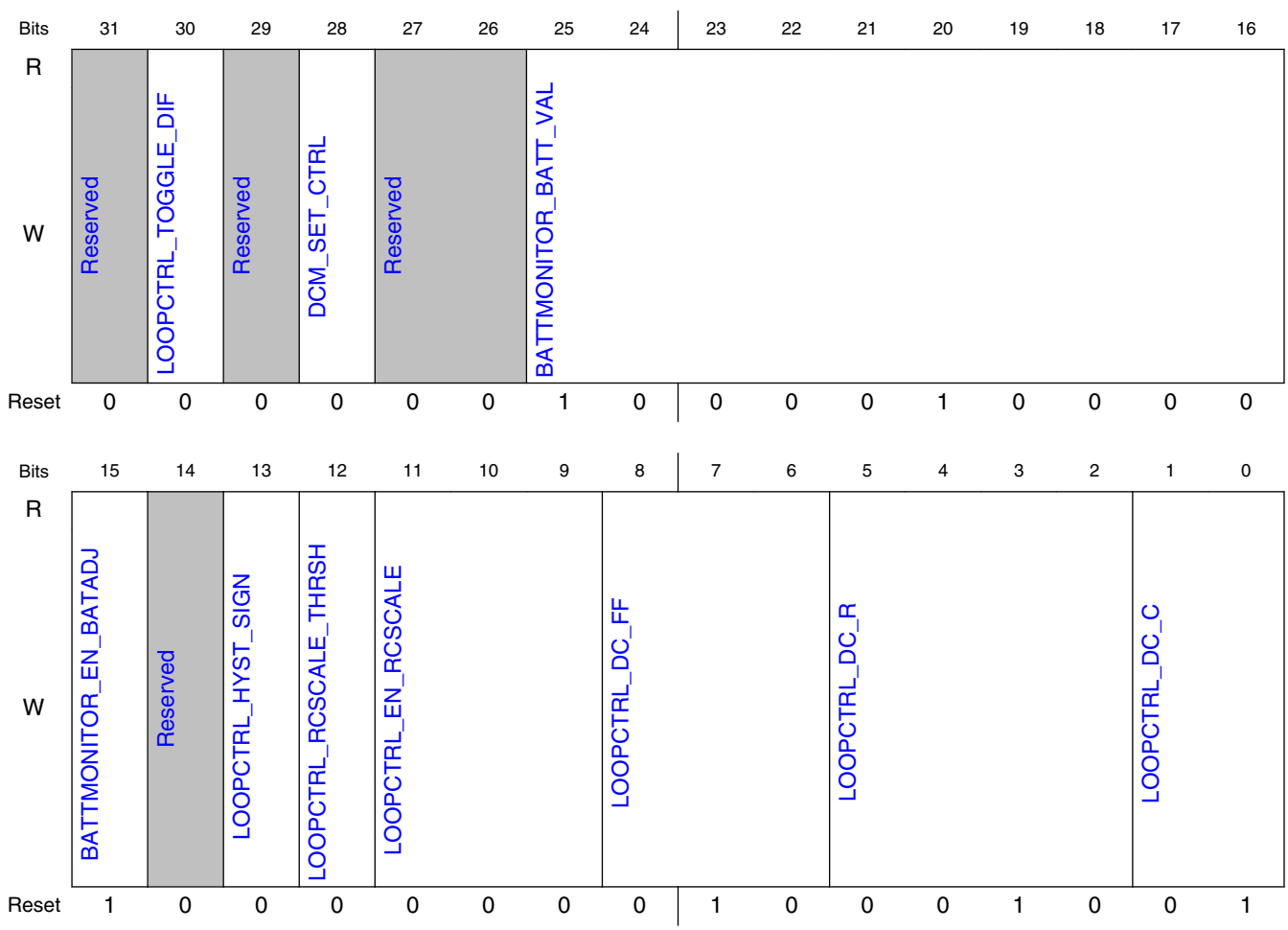
Register	Offset
REG2	10h

21.7.1.6.2 Function

DCDC register 2

This register controls various high-level functions of the DCDC

21.7.1.6.3 Diagram



21.7.1.6.4 Fields

Field	Function
31 —	Reserved
30 LOOPCTRL_TOGGLE_DIF	Set high to enable supply stepping to change only after the differential control loop has toggled as well. This should eliminate any chance of large transients when supply voltage changes are made.
29 —	Reserved
28 DCM_SET_CTRL	DCM Set Control Set high to improve the transition from heavy load to light load. **NOTE:** The REG2[DCM_SET_CTRL] should be set to 1'b1 if there is a current backflow to DCDC output in an application.
27-26 —	Reserved
25-16 BATTMONITOR_BATT_VAL	Software should be configured to place the battery voltage in this register measured with an 8-mV LSB resolution through the ADC. This value is used by the DCDC converter and must be correct before setting BATTMONITOR_EN_BATADJ.
15 BATTMONITOR_EN_BATADJ	This bit enables the DCDC to improve efficiency and minimize ripple using the information from the BATT_VAL field. It is very important that the BATT_VAL contain accurate information before setting EN_BATADJ.
14 —	Reserved
13 LOOPCTRL_HYST_SIGN	Invert the sign of the hysteresis in DCDC analog comparators.
12 LOOPCTRL_RCSCALE_THRSH	Increase the threshold detection for RC scale circuit.
11-9 LOOPCTRL_EN_RCSCALE	Enable RC Scale This enables the analog circuit of the DCDC to respond faster under transient load conditions. Higher the value, faster the response. 0x5 is the highest setting. 0x6 - 0x7: Not available
8-6 LOOPCTRL_DC_FF	Two's complement feed forward step in duty cycle in the switching DCDC converter. Each time this field makes a transition from 0x0, the loop filter of the DCDC converter is stepped once by a value proportional to the change. This can be used to force a certain control loop behavior, such as improving response under known heavy load transients.
5-2 LOOPCTRL_DC_R	Magnitude of proportional control parameter in the switching DCDC converter control loop.
1-0 LOOPCTRL_DC_C	Ratio of integral control parameter to proportional control parameter in the switching DCDC converter, and can be used to optimize efficiency and loop response.

21.7.1.7 DCDC Register 3 (REG3)

21.7.1.7.1 Offset

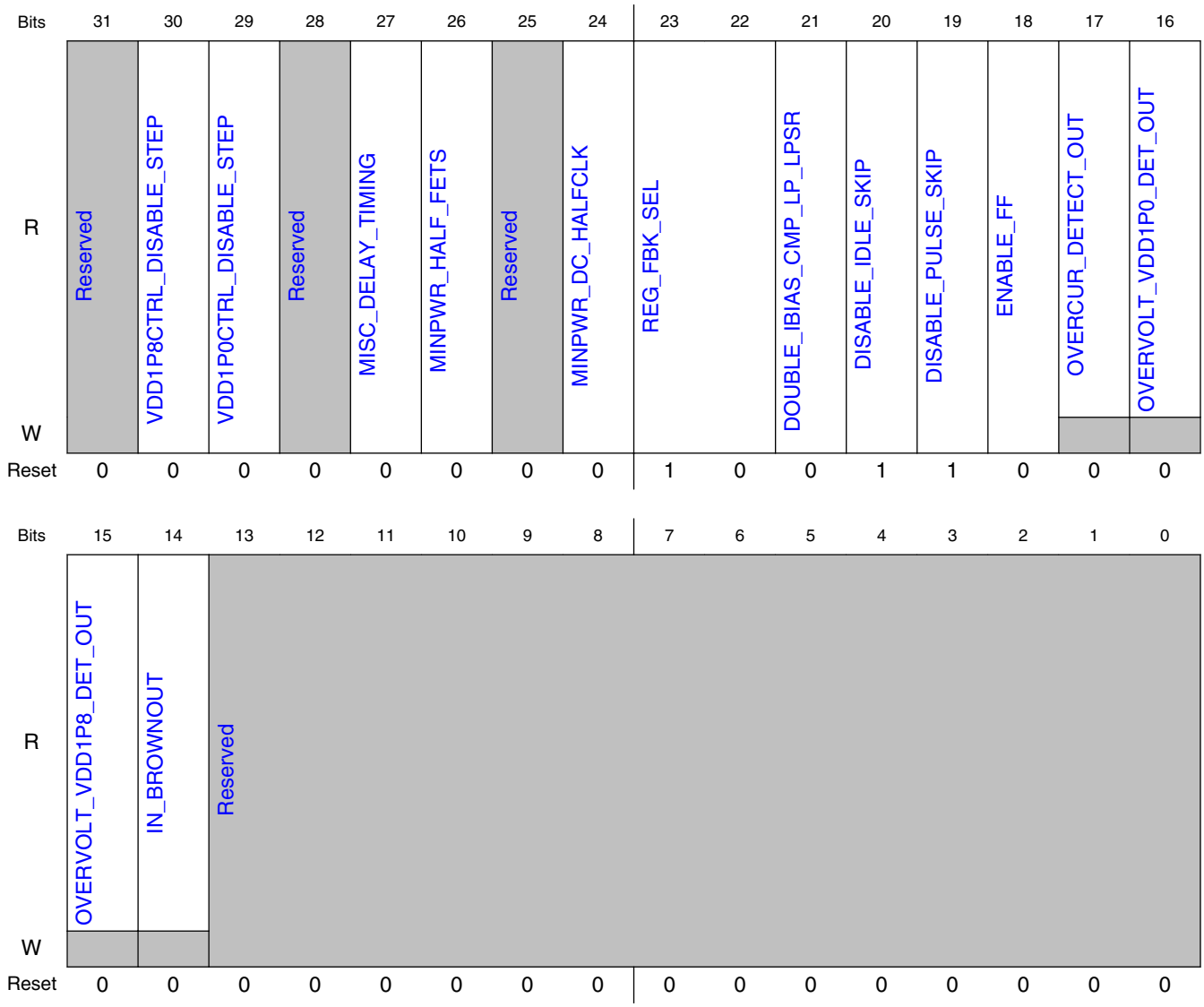
Register	Offset
REG3	14h

21.7.1.7.2 Function

DCDC register 3

This register controls various high-level functions of the DCDC

21.7.1.7.3 Diagram



21.7.1.7.4 Fields

Field	Function
31 —	Reserved
30 VDD1P8CTRL_DISABLE_STEP	Disable Step for VDD1P8 Disable stepping for VDD1P8. Must set this bit before entering low power modes. 0b - Enable stepping for VDD1P8 1b - Disable stepping for VDD1P8
29	Disable Step for VDD1P0 Disable stepping for VDD1P0. Must set this bit before entering low power modes.

Table continues on the next page...

Field	Function
VDD1P0CTRL_DISABLE_STEP	0b - Enable stepping for VDD1P0 1b - Disable stepping for VDD1P0
28 —	Reserved
27 MISC_DELAY_TIMING	Miscellaneous Delay Timing Ajust delay to reduce ground noise.
26 MINPWR_HALF_FETS	Use half switch FET
25 —	Reserved
24 MINPWR_DC_HALFCLK	Set DCDC clock to half freqeuncy for continuous mode. 0b - DCDC clock remains at full frequency for continuous mode 1b - DCDC clock set to half frequency for continuous mode
23-22 REG_FBK_SEL	Select the feedback point of the internal regulator
21 DOUBLE_IBIAS_CMP_LP_LPSR	1b - Double the bias current of the comparator for low-voltage detector in LP (low-power) mode
20 DISABLE_IDLE_SKIP	0b - Enable the idle skip function. The DCDC will be idle when out-of-range comparator detects the output voltage is higher than the target by 25mV. This function requires the out-of-range comparator to be enabled (PWD_CMP_OFFSET=0).
19 DISABLE_PULSE_SKIP	Disable Pulse Skip 0b - Stop charging if the duty cycle is lower than what is set by NEGLIMIT_IN
18 ENABLE_FF	1b - Enable feed-forward (FF) function that can speed up transient settling.
17 OVERCUR_DETECT_OUT	signal "1" when overcurrent happens. 1b - Overcurrent
16 OVERVOLT_VDD1P0_DET_OUT	signal "1" when overvoltage on the VDD1P0 output happens 1b - VDD1P0 Overvoltage
15 OVERVOLT_VDD1P8_DET_OUT	signal "1" when overvoltage on the VDD1P8 output happens 1b - VDD1P8 Overvoltage
14 IN_BROWNOUT	signal "1" when the voltage on DCDC_IN is lower than 2.6V 1b - DCDC_IN is lower than 2.6V
13-0 —	Reserved

21.7.1.8 DCDC Register 4 (REG4)

21.7.1.8.1 Offset

Register	Offset
REG4	18h

21.7.1.8.2 Function

Setpoint mode register

21.7.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENABLE_SP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.8.4 Fields

Field	Function
31-16 —	Reserved
15-0 ENABLE_SP	Configures CTRL0[ENABLE] (DCDC Enable) for Setpoints 0-15. Setting bit(s) to 1 enables DCDC for Setpoint.

21.7.1.9 DCDC Register 5 (REG5)

21.7.1.9.1 Offset

Register	Offset
REG5	1Ch

21.7.1.9.2 Function

Setpoint mode register

21.7.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_EN_SP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.9.4 Fields

Field	Function
31-16 —	Reserved
15-0 DIG_EN_SP	Configures CTRL0[DIG_EN] (DCDC_DIG Enable) for Setpoints 0-15. Always set these bits to 1.

21.7.1.10 DCDC Register 6 (REG6)

21.7.1.10.1 Offset

Register	Offset
REG6	20h

21.7.1.10.2 Function

Setpoint mode register

21.7.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LP_MODE_SP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.10.4 Fields

Field	Function
31-16 —	Reserved
15-0 LP_MODE_SP	Configures CTRL0[LP_MODE_EN] (LP Mode Enable) for Setpoints 0-15. Setting bit(s) to 1 enables LP mode for Setpoint.

21.7.1.11 DCDC Register 7 (REG7)

21.7.1.11.1 Offset

Register	Offset
REG7	24h

21.7.1.11.2 Function

Setpoint mode register

21.7.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_EN_SP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.11.4 Fields

Field	Function
31-16 —	Reserved
15-0 STBY_EN_SP	Configures CTRL0[STBY_EN] (Standby Enable) for Setpoints 0-15. Setting bit(s) to 1 enables DCDC to enter standby mode for Setpoint.

21.7.1.12 DCDC Register 7 plus (REG7P)

21.7.1.12.1 Offset

Register	Offset
REG7P	28h

21.7.1.12.2 Function

Setpoint mode register

21.7.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STBY_LP_MODE_SP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.12.4 Fields

Field	Function
31-16 —	Reserved
15-0 STBY_LP_MODE_SP	Configures CTRL0[STBY_LP_MODE_EN] (LP Mode via GPC Enable) for Setpoints 0-15. Setting bit(s) to 1 enables standby request from GPC to enter LP mode for Setpoint.

21.7.1.13 DCDC Register 8 (REG8)

21.7.1.13.1 Offset

Register	Offset
REG8	2Ch

21.7.1.13.2 Function

Setpoint mode register

21.7.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.13.4 Fields

Field	Function
31-0 ANA_TRG_SP0	Configures CTRL1[VDD1P8CTRL_TRG] FOR Setpoints 0-3. bit0~4 : DCDC 1.8V target as Setpoint 0 bit8~12 : DCDC 1.8V target as Setpoint 1 bit16~20 : DCDC 1.8V target as Setpoint 2 bit24~28 : DCDC 1.8V target as Setpoint 3

21.7.1.14 DCDC Register 9 (REG9)

21.7.1.14.1 Offset

Register	Offset
REG9	30h

21.7.1.14.2 Function

Setpoint mode register

21.7.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.14.4 Fields

Field	Function
31-0 ANA_TRG_SP1	Configures CTRL1[VDD1P8CTRL_TRG] FOR Setpoints 4-7. bit0~4 : DCDC 1.8V target as Setpoint 4 bit8~12 : DCDC 1.8V target as Setpoint 5 bit16~20 : DCDC 1.8V target as Setpoint 6 bit24~28 : DCDC 1.8V target as Setpoint 7

21.7.1.15 DCDC Register 10 (REG10)

21.7.1.15.1 Offset

Register	Offset
REG10	34h

21.7.1.15.2 Function

Setpoint mode register

21.7.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.15.4 Fields

Field	Function
31-0 ANA_TRG_SP2	Configures CTRL1[VDD1P8CTRL_TRG] FOR Setpoints 8-11. bit0~4 : DCDC 1.8V target as Setpoint 8 bit8~12 : DCDC 1.8V target as Setpoint 9 bit16~20 : DCDC 1.8V target as Setpoint 10 bit24~28 : DCDC 1.8V target as Setpoint 11

21.7.1.16 DCDC Register 11 (REG11)

21.7.1.16.1 Offset

Register	Offset
REG11	38h

21.7.1.16.2 Function

Setpoint mode register

21.7.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.16.4 Fields

Field	Function
31-0 ANA_TRG_SP3	Configures CTRL1[VDD1P8CTRL_TRG] FOR Setpoints 12-15. bit0~4 : DCDC 1.8V target as Setpoint 12 bit8~12 : DCDC 1.8V target as Setpoint 13 bit16~20 : DCDC 1.8V target as Setpoint 14 bit24~28 : DCDC 1.8V target as Setpoint 15

21.7.1.17 DCDC Register 12 (REG12)

21.7.1.17.1 Offset

Register	Offset
REG12	3Ch

21.7.1.17.2 Function

Setpoint mode register

21.7.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.17.4 Fields

Field	Function
31-0 DIG_TRG_SP0	Configures CTRL1[VDD1P0CTRL_TRG] FOR Setpoints 0-3. bit0~4 : DCDC 1.0V target at Setpoint 0 bit8~12 : DCDC 1.0V target at Setpoint 1 bit16~20 : DCDC 1.0V target at Setpoint 2 bit24~28 : DCDC 1.0V target at Setpoint 3

21.7.1.18 DCDC Register 13 (REG13)

21.7.1.18.1 Offset

Register	Offset
REG13	40h

21.7.1.18.2 Function

Setpoint mode register

21.7.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.18.4 Fields

Field	Function
31-0 DIG_TRG_SP1	Configures CTRL1[VDD1P0CTRL_TRG] FOR Setpoints 4-7. bit0~4 : DCDC 1.0V target at Setpoint 4 bit8~12 : DCDC 1.0V target at Setpoint 5 bit16~20 : DCDC 1.0V target at Setpoint 6 bit24~28 : DCDC 1.0V target at Setpoint 7

21.7.1.19 DCDC Register 14 (REG14)

21.7.1.19.1 Offset

Register	Offset
REG14	44h

21.7.1.19.2 Function

Setpoint mode register

21.7.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.19.4 Fields

Field	Function
31-0 DIG_TRG_SP2	Configures CTRL1[VDD1P0CTRL_TRG] FOR Setpoints 8-11. bit0~4 : DCDC 1.0V target at Setpoint 8 bit8~12 : DCDC 1.0V target at Setpoint 9 bit16~20 : DCDC 1.0V target at Setpoint 10 bit24~28 : DCDC 1.0V target at Setpoint 11

21.7.1.20 DCDC Register 15 (REG15)

21.7.1.20.1 Offset

Register	Offset
REG15	48h

21.7.1.20.2 Function

Setpoint mode register

21.7.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.20.4 Fields

Field	Function
31-0 DIG_TRG_SP3	Configures CTRL1[VDD1P0CTRL_TRG] FOR Setpoints 12-15. bit0~4 : DCDC 1.0V target at Setpoint 12 bit8~12 : DCDC 1.0V target at Setpoint 13 bit16~20 : DCDC 1.0V target at Setpoint 14 bit24~28 : DCDC 1.0V target at Setpoint 15

21.7.1.21 DCDC Register 16 (REG16)

21.7.1.21.1 Offset

Register	Offset
REG16	4Ch

21.7.1.21.2 Function

Setpoint mode register

21.7.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_STBY_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_STBY_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.21.4 Fields

Field	Function
31-0 ANA_STBY_TRG_SP0	Configures CTRL1[VDD1P8CTRL_STBY_TRG] FOR Setpoints 0-3. bit0~4 : DCDC 1.8V target at Setpoint 0 and system standby bit8~12 : DCDC 1.8V target at Setpoint 1 and system standby bit16~20 : DCDC 1.8V target at Setpoint 2 and system standby bit24~28 : DCDC 1.8V target at Setpoint 3 and system standby

21.7.1.22 DCDC Register 17 (REG17)

21.7.1.22.1 Offset

Register	Offset
REG17	50h

21.7.1.22.2 Function

Setpoint mode register

21.7.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_STBY_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_STBY_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.22.4 Fields

Field	Function
31-0 ANA_STBY_TRG_SP1	Configures CTRL1[VDD1P8CTRL_STBY_TRG] FOR Setpoints 4-7. bit0~4 : DCDC 1.8V target at Setpoint 4 and system standby bit8~12 : DCDC 1.8V target at Setpoint 5 and system standby bit16~20 : DCDC 1.8V target at Setpoint 6 and system standby bit24~28 : DCDC 1.8V target at Setpoint 7 and system standby

21.7.1.23 DCDC Register 18 (REG18)

21.7.1.23.1 Offset

Register	Offset
REG18	54h

21.7.1.23.2 Function

Setpoint mode register

21.7.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_STBY_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_STBY_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.23.4 Fields

Field	Function
31-0 ANA_STBY_TRG_SP2	Configures CTRL1[VDD1P8CTRL_STBY_TRG] FOR Setpoints 8-11. bit0~4 : DCDC 1.8V target at Setpoint 8 and system standby bit8~12 : DCDC 1.8V target at Setpoint 9 and system standby bit16~20 : DCDC 1.8V target at Setpoint 10 and system standby bit24~28 : DCDC 1.8V target at Setpoint 11 and system standby

21.7.1.24 DCDC Register 19 (REG19)

21.7.1.24.1 Offset

Register	Offset
REG19	58h

21.7.1.24.2 Function

Setpoint mode register

21.7.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_STBY_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ANA_STBY_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.24.4 Fields

Field	Function
31-0 ANA_STBY_TRG_SP3	Configures CTRL1[VDD1P8CTRL_STBY_TRG] FOR Setpoints 12-15. bit0~4 : DCDC 1.8V target at Setpoint 12 and system standby bit8~12 : DCDC 1.8V target at Setpoint 13 and system standby bit16~20 : DCDC 1.8V target at Setpoint 14 and system standby bit24~28 : DCDC 1.8V target at Setpoint 15 and system standby

21.7.1.25 DCDC Register 20 (REG20)

21.7.1.25.1 Offset

Register	Offset
REG20	5Ch

21.7.1.25.2 Function

Setpoint mode register

21.7.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_STBY_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_STBY_TRG_SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.25.4 Fields

Field	Function
31-0 DIG_STBY_TRG_SP0	Configures CTRL1[VDD1P0CTRL_STBY_TRG] FOR Setpoints 0-3. bit0~4 : DCDC 1.0V target at Setpoint 0 and system standby bit8~12 : DCDC 1.0V target at Setpoint 1 and system standby bit16~20 : DCDC 1.0V target at Setpoint 2 and system standby bit24~28 : DCDC 1.0V target at Setpoint 3 and system standby

21.7.1.26 DCDC Register 21 (REG21)

21.7.1.26.1 Offset

Register	Offset
REG21	60h

21.7.1.26.2 Function

Setpoint mode register

21.7.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_STBY_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_STBY_TRG_SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.26.4 Fields

Field	Function
31-0 DIG_STBY_TRG_SP1	Configures CTRL1[VDD1P0CTRL_STBY_TRG] FOR Setpoints 4-7. bit0~4 : DCDC 1.0V target at Setpoint 4 and system standby bit8~12 : DCDC 1.0V target at Setpoint 5 and system standby bit16~20 : DCDC 1.0V target at Setpoint 6 and system standby bit24~28 : DCDC 1.0V target at Setpoint 7 and system standby

21.7.1.27 DCDC Register 22 (REG22)

21.7.1.27.1 Offset

Register	Offset
REG22	64h

21.7.1.27.2 Function

Setpoint mode register

21.7.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_STBY_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_STBY_TRG_SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.27.4 Fields

Field	Function
31-0 DIG_STBY_TRG_SP2	Configures CTRL1[VDD1P0CTRL_STBY_TRG] FOR Setpoints 8-11. bit0~4 : DCDC 1.0V target at Setpoint 8 and system standby bit8~12 : DCDC 1.0V target at Setpoint 9 and system standby bit16~20 : DCDC 1.0V target at Setpoint 10 and system standby bit24~28 : DCDC 1.0V target at Setpoint 11 and system standby

21.7.1.28 DCDC Register 23 (REG23)

21.7.1.28.1 Offset

Register	Offset
REG23	68h

21.7.1.28.2 Function

Setpoint mode register

21.7.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIG_STBY_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIG_STBY_TRG_SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.28.4 Fields

Field	Function
31-0 DIG_STBY_TRG_SP3	Configures CTRL1[VDD1P0CTRL_STBY_TRG] FOR Setpoints 12-15. bit0~4 : DCDC 1.0V target at Setpoint 12 and system standby bit8~12 : DCDC 1.0V target at Setpoint 13 and system standby bit16~20 : DCDC 1.0V target at Setpoint 14 and system standby bit24~28 : DCDC 1.0V target at Setpoint 15 and system standby

21.7.1.29 DCDC Register 24 (REG24)

21.7.1.29.1 Offset

Register	Offset
REG24	6Ch

21.7.1.29.2 Function

DCDC_OK Count Value

21.7.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OK_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OK_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

21.7.1.29.4 Fields

Field	Function
31-0 OK_COUNT	Internal count for dcdc_ok timeout

Chapter 22
Temperature Sensor (TMPSNS)

22.1 Chip-specific TMPSNS information

Table 22-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map.
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

The temperature sensor registers are accessed through the Analog IP (AI) Interface.

22.2 Overview

The Temperature Sensor (TMPSNS) implements a temperature sensor/conversion function based on a temperature-dependent voltage to time conversion.

22.2.1 Block diagram

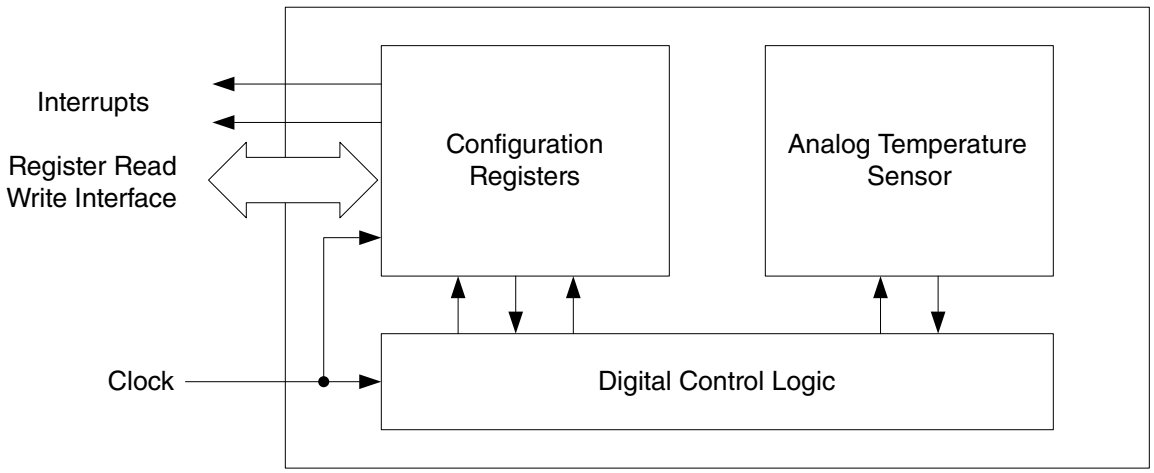


Figure 22-1. Temperature sensor block diagram

22.2.2 Features

The Temperature Sensor supports the following features:

- Temperature Alarm Interrupts to indicate if temperature is beyond the programmed thresholds.
  - Three tier alarms supported viz., Low, High and Panic.
- Single, and Continuous Measurement Mode

22.3 Functional description

The temperature sensor module features alarm functions that can raise independent interrupt signals if the temperature is above two high-temperature thresholds and below a low temperature threshold. These temperature thresholds are programmable and designated as low, high and panic temperature. The panic threshold is a special programmable threshold in that if the temperature increases above this value and the temperature-panic-reset interrupt is enabled in the System Reset Controller, the hardware will assume that software no longer has control over the thermal situation and will initiate a reset of the chip. In order to avoid false panic temperature initiated resets, the panic alarm will not fire until the temperature panic condition has been met for four consecutive conversion cycles. A self-repeating mode can also be programmed which executes a temperature sensing operation based on a programmed delay.

Since the high and low temperature thresholds are programmable, they form a sliding temperature bracket that can be tailored to the application's needs. For example, at startup software can set the low temperature threshold to the minimum temperature code and the high temperature threshold to a maximum operating temperature for the system.

The system can then use this module to monitor the on-die temperature and take appropriate actions such as throttling back the core frequency when a the high temperature interrupt is set. Once the high temperature interrupt is set then the system can program the low temperature threshold to a desired cool down temperature. The system would then switch to monitoring the low temperature alarm and wait for its interrupt to be set. With this scheme, once the low temperature interrupt is set, then software could be assured that the temperature has cooled down to a safe level and the process could be repeated.

22.3.1 Modes of Operation

The following modes are supported in Temperature Sensor:

- Single Measurement Mode
  - Performs one measurement per Start indication from software.
  - Finish bit asserted on every measurement completion.
- Continuous Measurement Mode
  - Performs continuous measurements (once initiated by software) at the rate defined by CTRL1[FREQ]
  - Frequency of measurements programmable by software
  - Finish bit asserts on every measurement completion, if previously cleared.

22.3.2 Temperature Sensing

The value returned by the Temperature Sensor in STATUS0[TEMP_VAL] does not indicate the actual temperature. The temperature value can be obtained by the following equation:

$$temp_{diode} = - ts_{21} - \frac{\sqrt{ts_{21}^2 - 4 * ts_{22} * (ts_{20} + ts_{25c} - ts_{meas})}}{2 * ts_{22}}$$

Where:

$ts_{22} = 0.002$

$ts_{21} = - 5.39$

$ts_{20} = 133.6$

ts_{meas} is the measured temperature value from STATUS0[TEMP_VAL]

ts_{25c} comes from fuse

22.3.3 Clocks

This module uses a 24MHz reference clock from the 24MHz crystal oscillator.

22.3.4 Reset

All resets are controlled through the Analog IP (AI) Interface.

22.3.5 Interrupts

All interrupts are controlled through the Analog IP (AI) Interface.

22.4 External Signals

This module does not have any SoC level pins.

22.5 Initialization

There are no special considerations for initializing this module. The CTRL[PWD] and CTRL[PWD_FUL] bits are used to power down the temperature sensor.

22.6 Application Information

22.6.1 Configuration Steps

The following steps should be followed when configuring the Temperature Sensor in Single Measurement Mode.

1. Set CTRL0[V_SEL]=0 to select normal temperature measuring mode
2. Write the appropriate threshold values in RANGE0 and RANGE1 registers for the Panic, High and Low Temperature Alarms to trigger. Please refer to section "Temperature Sensing" to determine the value to be programmed based on desired temperature value.
3. Enable appropriate interrupts by setting the corresponding interrupt enable bits to '1' in the CTRL1 register.
4. Power up the analog section by setting the CTRL1[PWD] and CTRL1[PWD_FULL] bits to '0'.
5. Ensure that the CTRL1[FREQ] field is set to 0.
6. Set the CTRL1[START] bit to 1 to enable the measurement process.
7. Poll the STATUS0[FINISH] bit to determine the completion of the measurement.
8. Read the measured value in the STATUS0[TEMP_VAL] field. Refer to Temperature Sensing to determine the actual temperature value based on the measured value.
9. Clear the CTRL1[FINISH] bit by writing '1' to it.
10. Clear the CTRL1[START] bit by writing '0' to it. This is important to start the next single measurement.

Repeat the steps 6 to 10 to start another measurement cycle.

The following steps change when configuring the Temperature Sensor in Multiple Measurement Mode.

- In Step 5, set the desired frequency of measurement by writing a non-zero value in CTRL1[FREQ] field. This field determines the number of idle clock cycles between two conversions. The next measurement starts after counting these clock cycles.
- Repeat Steps 7 to 9 to keep getting continuous measurements.
- Perform Step 10 only when continuous mode is to be exited.

22.7 Memory Map and register definition

The memory map and descriptions of all registers are shown below.

NOTE

For related IPS Domain Slot control, please see IPS Domain Registers in Power Management Unit (PMU) chapter.

22.7.1 Temperature Sensor Memory Map register descriptions

The memory map comprises of 32-bit aligned registers which can be accessed via 8, or 16 or 32-bit accesses only. Write access to reserved locations will generate transfer error. Read access to reserved locations will also generate transfer error and the read data bus will show all 0s.

The module will not check for correctness of programmed values in the registers and software must ensure that correct values are being written.

NOTE

All register bit fields are shadowed except CTRL0, and STATUS0 . Shadowed bit fields can be changed only when CTRL1[START] = '0' or CTRL1[PWD] = '1'. Changing shadowed bit fields at other time will take effect only when START bit is toggled from '0' to '1' or PWD bit is toggled from '1' to '0'. CTRL0, STATUS0 can be changed irrespective of the START and PWD bit values.

22.7.1.1 TMPSNS memory map

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Temperature Sensor Control Register 0 (CTRL0)	32	RW	0000_8020
4	Temperature Sensor Control Register 0 (CTRL0_SET)	32	RW	0000_8020
8	Temperature Sensor Control Register 0 (CTRL0_CLR)	32	RW	0000_8020
C	Temperature Sensor Control Register 0 (CTRL0_TOG)	32	RW	0000_8020
10	Temperature Sensor Control Register 1 (CTRL1)	32	RW	0080_0000
14	Temperature Sensor Control Register 1 (CTRL1_SET)	32	RW	0080_0000
18	Temperature Sensor Control Register 1 (CTRL1_CLR)	32	RW	0080_0000
1C	Temperature Sensor Control Register 1 (CTRL1_TOG)	32	RW	0080_0000
20	Temperature Sensor Range Register 0 (RANGE0)	32	RW	0000_0000
24	Temperature Sensor Range Register 0 (RANGE0_SET)	32	RW	0000_0000
28	Temperature Sensor Range Register 0 (RANGE0_CLR)	32	RW	0000_0000
2C	Temperature Sensor Range Register 0 (RANGE0_TOG)	32	RW	0000_0000
30	Temperature Sensor Range Register 1 (RANGE1)	32	RW	0000_0000
34	Temperature Sensor Range Register 1 (RANGE1_SET)	32	RW	0000_0000
38	Temperature Sensor Range Register 1 (RANGE1_CLR)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
3C	Temperature Sensor Range Register 1 (RANGE1_TOG)	32	RW	0000_0000
50	Temperature Sensor Status Register 0 (STATUS0)	32	W1C	0000_0000

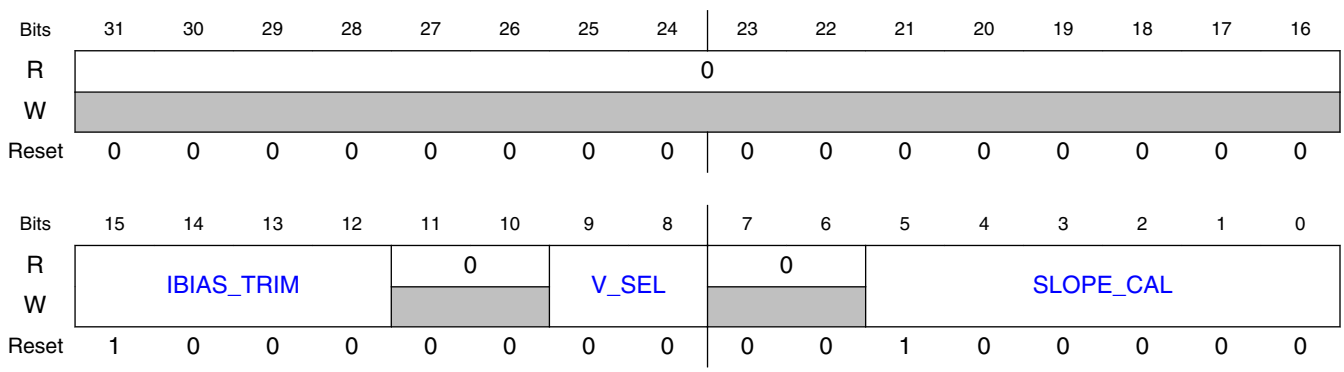
22.7.1.2 Temperature Sensor Control Register 0 (CTRL0)

Configures the analog temperature sensor control.

22.7.1.2.1 Offset

Register	Offset	Description
CTRL0	0h	Temperature Sensor Control Register 0
CTRL0_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL0
CTRL0_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL0
CTRL0_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL0

22.7.1.2.2 Diagram



22.7.1.2.3 Fields

Field	Description
31-16 —	Reserved field. Reads only zeros.

Table continues on the next page...

Field	Description
15-12 IBIAS_TRIM	Current bias trim value This field defines the trim setting for the bias current that's used for ramp slope calibration. Note: The trim value is loaded from the fuse.
11-10 —	Reserved field. Reads only zeros.
9-8 V_SEL	Voltage Select Determines the voltage that will be detected by the temperature sensor. 00 - Normal temperature measuring mode 01-10 - Reserved
7-6 —	Reserved field. Reads only zeros.
5-0 SLOPE_CAL	Ramp slope calibration control This field provides the calibration for the ramp slope of temperature sensor with trimming capacitor value. Note: This value is loaded from the fuse.

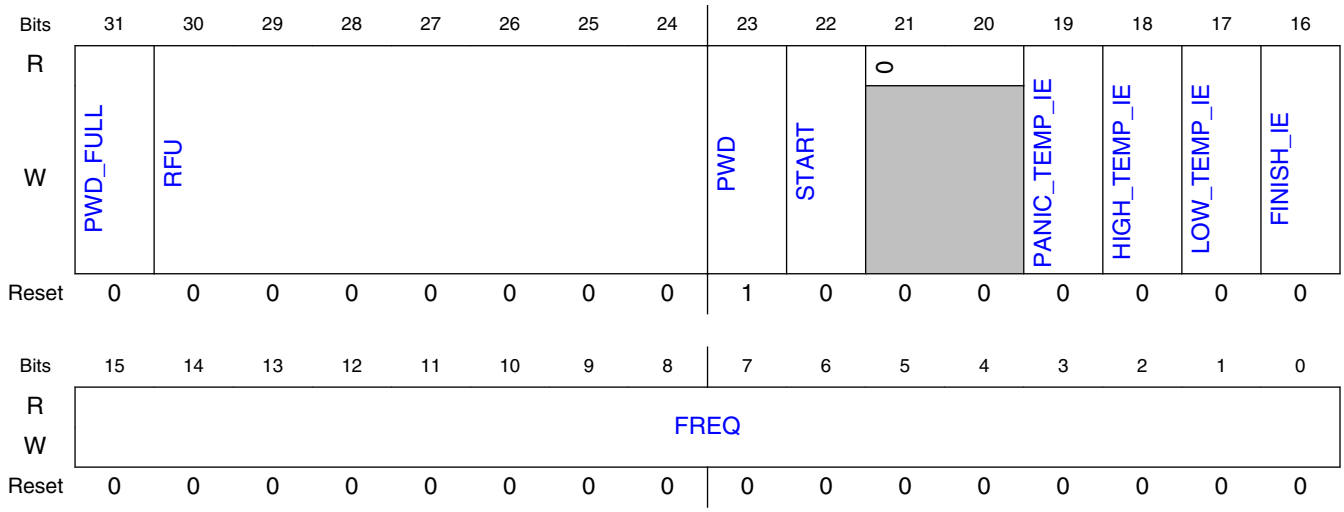
22.7.1.3 Temperature Sensor Control Register 1 (CTRL1)

Configures the digital temperature sensor control and mode of operation (Single Read or Continuous Read).

22.7.1.3.1 Offset

Register	Offset	Description
CTRL1	10h	Temperature Sensor Control Register 1
CTRL1_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL1
CTRL1_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL1
CTRL1_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL1

22.7.1.3.2 Diagram



22.7.1.3.3 Fields

Field	Description
31 PWD_FULL	Temperature Sensor Full Power Down Full power down of the temperature sensor including bias. 0 - Sensor is active 1 - Sensor is powered down
30-24 RFU	Read/Writeable field. Reserved for future use
23 PWD	Temperature Sensor Power Down Powers down the temperature sensor except bias circuit. 0 - Sensor is active 1 - Sensor is powered down
22 START	Start Temperature Measurement Initiates the temperature reading in either Single Read or Continuous Read modes. For single reading, this bit must be set to 0 and then to 1 for every reading. For continuous reading, this bit can be left at '1' and set to '0' only when measurement is to be stopped. 0 - No new temperature reading taken 1 - Initiate a new temperature reading
21-20 —	Reserved field. Reads only zeros
19 PANIC_TEMP_IE	Panic temperature interrupt enable Enables assertion of interrupt when measured temperature is greater than or equal to the programmed panic temperature value. 0 - Interrupt is disabled 1 - Interrupt is enabled

Table continues on the next page...

Field	Description
18 HIGH_TEMP_IE	High temperature interrupt enable Enables assertion of interrupt when measured temperature is greater than or equal to the programmed high temperature value. 0 - Interrupt is disabled 1 - Interrupt is enabled
17 LOW_TEMP_IE	Low temperature interrupt enable Enables assertion of interrupt when measured temperature is lower than or equal to the programmed low temperature value. 0 - Interrupt is disabled 1 - Interrupt is enabled
16 FINISH_IE	Measurement finished interrupt enable Enables assertion of interrupt when temperature measurement is complete. 0 - Interrupt is disabled 1 - Interrupt is enabled
15-0 FREQ	Temperature Measurement Frequency Number of clock cycles of 24MHz clock that will be counted between two temperature readings. If this field is set to 0, the sensor will work in Single Reading mode, where a new temperature reading will be available, every time the CTRL1[START] bit is set to 1. For non-zero values, the sensor will work in Continuous Reading mode, where a new temperature reading will be initiated once the programmed period expires after the end of current temperature reading. CTRL1[START] bit should be kept 1 for continuous conversion. 0000000000000000 - Single Reading Mode. New reading available every time CTRL1[START] bit is set to 1 from 0. 0000000000000001-1111111111111111 - Continuous Reading Mode. Next temperature reading taken after programmed number of cycles after current reading is complete.

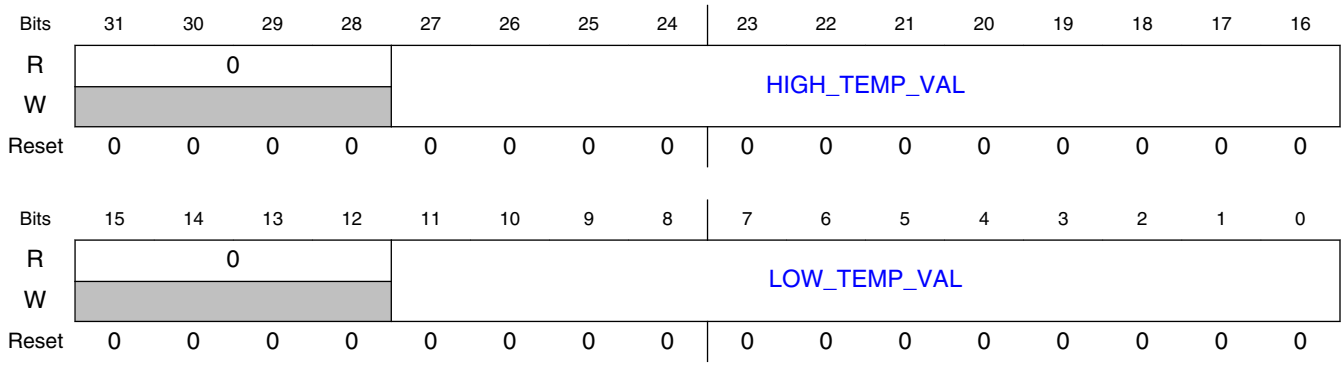
22.7.1.4 Temperature Sensor Range Register 0 (RANGE0)

Configures the ranges for generation of High and Low temperature alarms.

22.7.1.4.1 Offset

Register	Offset	Description
RANGE0	20h	Temperature Sensor Range Register 0
RANGE0_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in RANGE0
RANGE0_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in RANGE0
RANGE0_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RANGE0

22.7.1.4.2 Diagram



22.7.1.4.3 Fields

Field	Description
31-28 —	Reserved field. Reads only zeros.
27-16 HIGH_TEMP_VAL	High temperature threshold value High temperature threshold value (number of clock counts) below which the STATUS0[HIGH_TEMP] status bit will be set.
15-12 —	Reserved field. Reads only zeros.
11-0 LOW_TEMP_VAL	Low temperature threshold value Low temperature threshold value (number of clock counts) above which the LOW_TEMP status bit will be set.

22.7.1.5 Temperature Sensor Range Register 1 (RANGE1)

Configures the ranges for generation of Low temperature alarm.

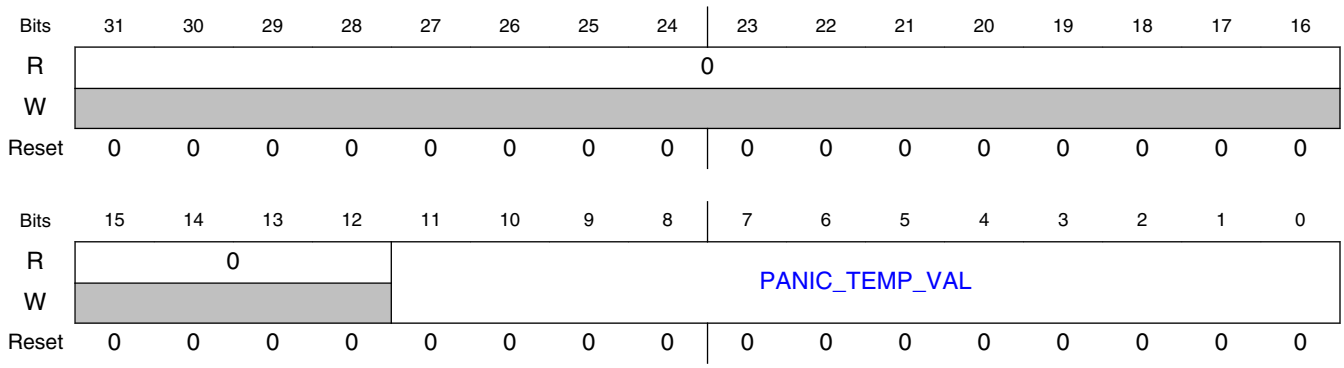
22.7.1.5.1 Offset

Register	Offset	Description
RANGE1	30h	Temperature Sensor Range Register 1
RANGE1_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in RANGE1
RANGE1_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in RANGE1

Table continues on the next page...

Register	Offset	Description
RANGE1_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RANGE1

22.7.1.5.2 Diagram



22.7.1.5.3 Fields

Field	Description
31-12 —	Reserved field. Reads only zeros.
11-0 PANIC_TEMP_VAL	Panic temperature threshold value Panic temperature threshold value (number of clock counts) below which the STATUS0[PANIC_TEMP] status bit will be set.

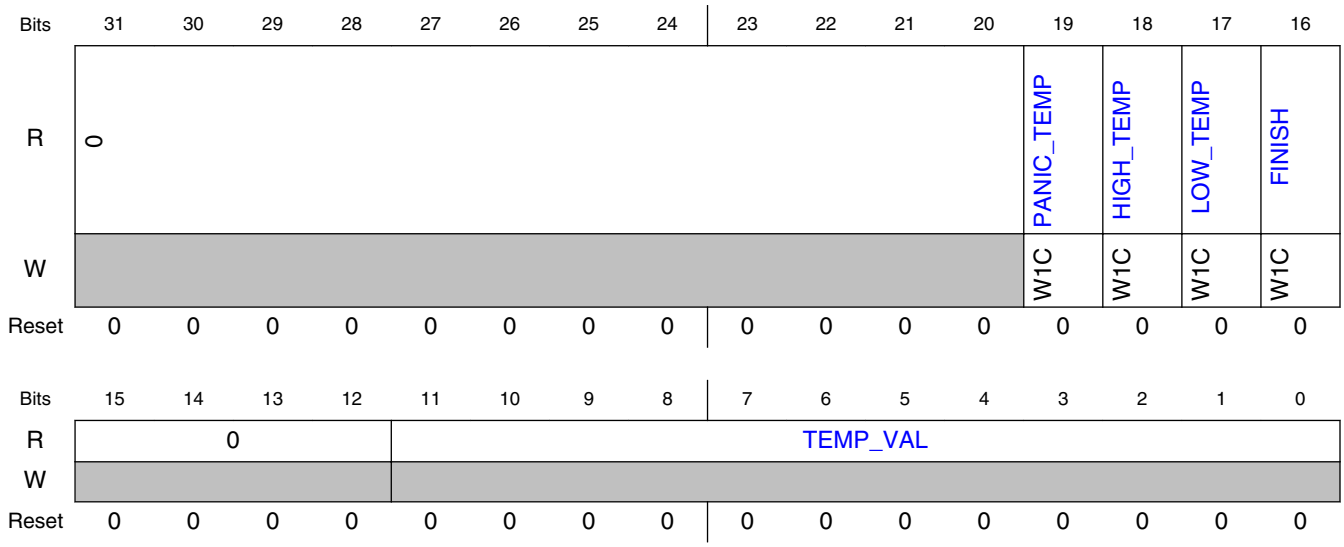
22.7.1.6 Temperature Sensor Status Register 0 (STATUS0)

Provides status and measured value from the temperature sensor.

22.7.1.6.1 Offset

Register	Offset
STATUS0	50h

22.7.1.6.2 Diagram



22.7.1.6.3 Fields

Field	Description
31-20 —	Reserved
19 PANIC_TEMP	Panic temperature alarm bit Asserts when measured temperature is above the programmed panic temperature threshold. 0 - No Panic temperature alert 1 - Panic temperature alert
18 HIGH_TEMP	High temperature alarm bit Asserts when measured temperature is above the programmed high temperature threshold. 0 - No High temperature alert 1 - High temperature alert
17 LOW_TEMP	Low temperature alarm bit Asserts when measured temperature is below the programmed low temperature threshold. 0 - No Low temperature alert 1 - Low temperature alert
16 FINISH	Temperature measurement complete Indicates that the temperature sensor has completed measuring the temperature value and the measured value is available in the STATUS0[TEMP_VAL] for reading. This bit should be cleared by writing 1 to acknowledge that the temperature reading has been read out. 0 - Temperature sensor is busy (if CTRL1[START] = 1)or no new reading has been initiated (if CTRL1[START] = 0) 1 - Temperature reading is complete and new temperature value available for reading
15-12	Reserved field. Reads only zeros

Table continues on the next page...

Field	Description
—	
11-0 TEMP_VAL	Measured temperature value Clock counts corresponding to the measured temperature by the sensor. Valid when STATUS0[FINISH] bit is asserted. This count needs to be converted as per formula (refer to Temperature Sensing) to get the actual temperature value.

22.7.2 Temperature Sensor Analog register descriptions

22.7.2.1 TEMPSENSOR memory map

TEMPSENSOR Analog base address: 40C8_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
400	Tempsensor Register (TEMPSENSOR)	32	RW	0000_0000
430	TEMPSNS_OTP_TRIM_VALUE_REGISTER (TEMPSNS_OTP_TRIM_VALUE)	32	RO	0000_0000

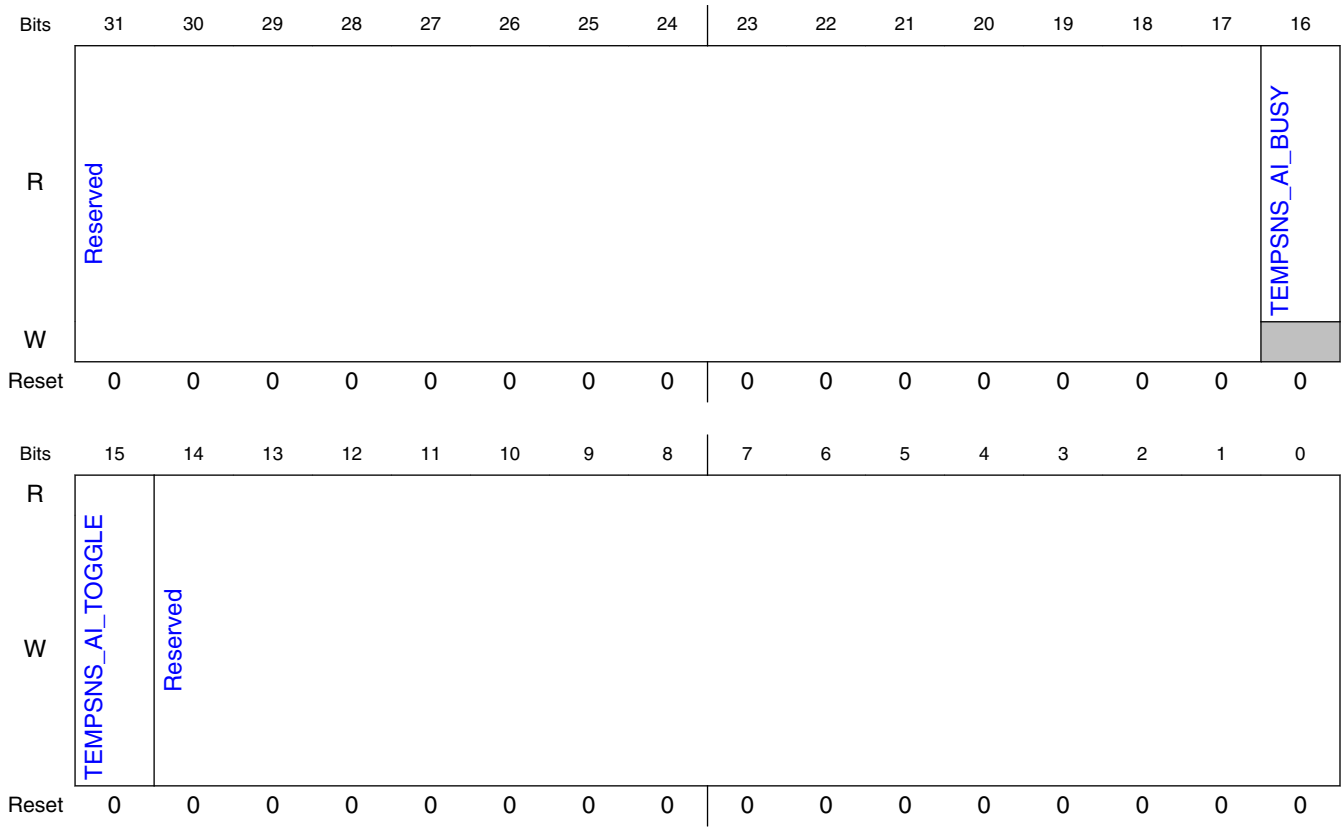
22.7.2.2 Tempsensor Register (TEMPSENSOR)

This register defines the basic settings for access to the temperature sensor through Analog IP (AI) Interface.

22.7.2.2.1 Offset

Register	Offset
TEMPSENSOR	400h

22.7.2.2.2 Diagram



22.7.2.2.3 Fields

Field	Description
31-17 —	Always set to zero
16 TEMPSNS_AI_BUSY	AI Busy monitor Please see 'Analog IP (AI) Interface' topic for details on how to use this bit.
15 TEMPSNS_AI_TOGGLE	AI toggle Please see 'Analog IP (AI) Interface' topic for details on how to use this bit.
14-0 —	Reserved

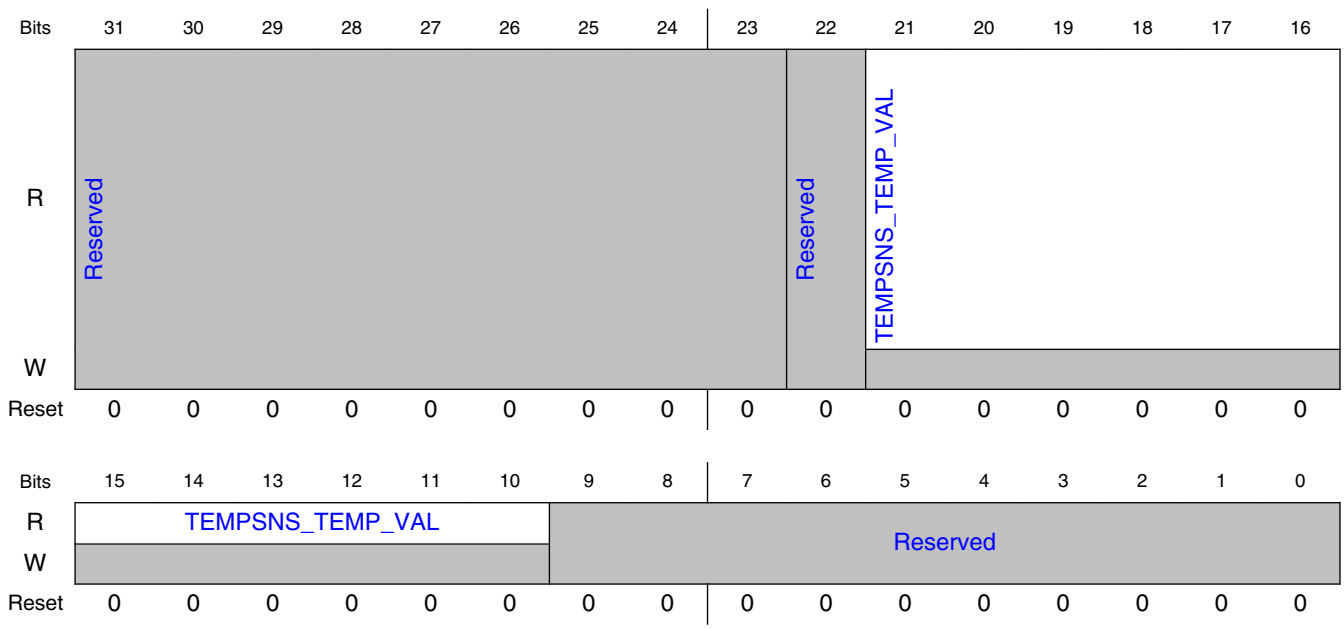
22.7.2.3 TEMPSNS_OTP_TRIM_VALUE_REGISTER (TEMPSNS_OTP_TRIM_VALUE)

This is the control for TRIM bus control. For ANATOP, this is the source from software register inside ANATOP.

22.7.2.3.1 Offset

Register	Offset
TEMPSNS_OTP_TRIM_VALUE	430h

22.7.2.3.2 Diagram



22.7.2.3.3 Fields

Field	Description
31-23 —	Reserved
22 —	Reserved
21-10	Temperature Value at 25C The value for this bit field comes from the fuses.

Table continues on the next page...

Field	Description
TEMPSNS_TEMP_VAL	
9-0 —	Reserved

Chapter 23
State Save and Restore Controller (SSARC)

23.1 Chip-specific State Save and Restore Controller (SSARC) information

Table 23-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

23.2 Overview

The State Save and Restore Controller (SSARC) saves the registers of functional modules in memory before power down and restores the registers from memory after the module is powered up.

23.2.1 Block diagram

The block diagram of the SSARC module is provided below:

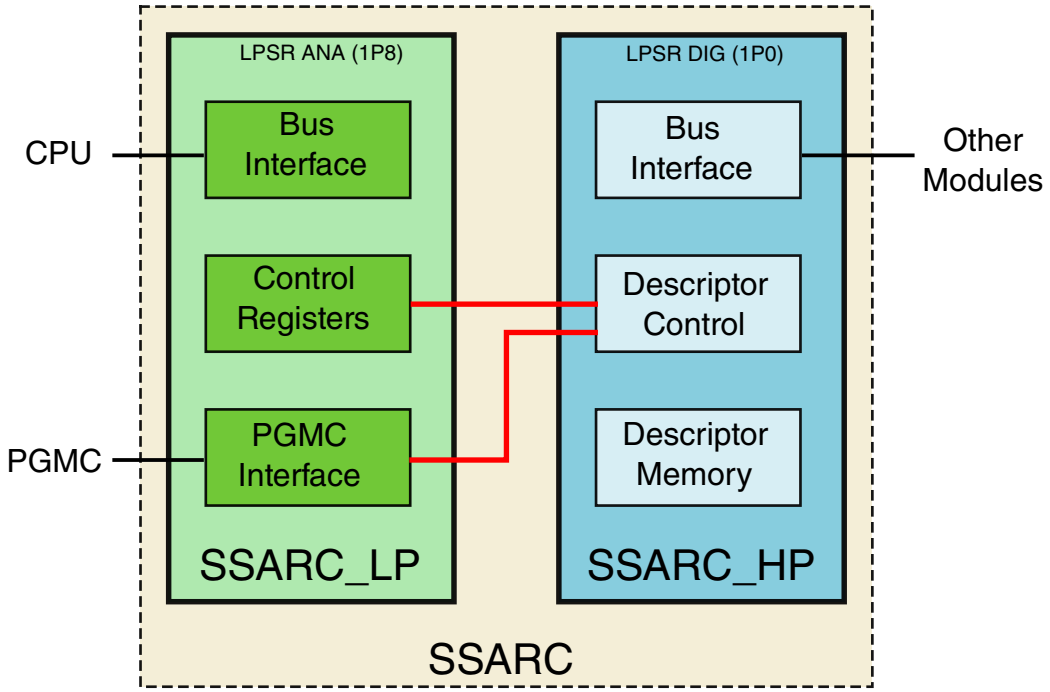


Figure 23-1. SSARC block diagram

23.2.2 Features

The SSARC features the following interfaces:

- The SSARC_HP interface, for read and write access to other functional module registers.
- The SSARC_LP interface, for CPU to configure the SSARC module.

The SSARC features a PGMC interface for the following:

- Save request for each power domain (save_req)
- Save done for each power domain (save_done)
- Restore request for each power domain (restore_req)
- Restore done for each power domain (restore_done)

NOTE

The save and restore signals are mapped to the PGMC BPCs.

23.3 Functional description

The operation of the SSARC is described in the following sections.

23.3.1 Descriptors

The SSARC register save and restore operations are handled as descriptors. Software needs to configure these descriptors like CPU instructions, before using SSARC. Each descriptor acts as one instruction that can save, restore, write value, and delay. See the SRAM registers for more information.

This module supports up to 1024 descriptors. These descriptors are stored in memory and 1024 descriptors requires 1024x72-bit of memory. Each descriptor is mapped to a 16B (4 words) aligned address. And this is allocated from the 4KB offset of the overall SSARC address.

The descriptors used by the system require initialization. If a descriptor is not initialized, its value could be random data. However, if descriptors are not used, there is no need to initialize them.

23.3.2 Descriptor function

The 1024 descriptors are divided into 16 groups and each group contains a number of continuous descriptors. Each group has a set of control registers - DESC_CTRL0_n, DESC_CTRL1_n, DESC_ADDR_UP_n, and DESC_ADDR_DOWN_n (n represents 0-15 groups)

The DESC_CTRL0_n registers are used to control the descriptors are mapped to a given group and set the save and restore order used to process the descriptors.

NOTE

There are no limitations on the number of descriptors that can be designated as a part of a single group. For example, SW can designate one group of 1024 descriptors.

The descriptors are controlled based on the group. Each group can be mapped to save and restore request from one power domain. When a request is asserted externally, the module will perform a state save/restore function for every descriptor in this group.

The DESC_CTRL1_n register is used to map each group to a particular power domain 1-7.

Each group also allows SW to trigger its save/restore operation.

Each group has two configurable priority settings:

- PRIORITY_SAVE[3:0]: 0 has highest priority, and 15 has lowest priority
- PRIORITY_RESTORE[3:0]: 0 has highest priority, and 15 has lowest priority

NOTE

Multiple groups can be configured with the same DESC_CTRL1_n_[POWER_DOMAIN] value.

Save operation usually does not have a very strong requirement on the priority, since the register values are always available before power down. However, Restore operation need priority, because security settings need to be restored first, such are RDC and XRDC.

23.3.3 Permission Control

After lock, the register for this group can only be accessed by the CPU from the domain. The access read/write from the bus interface will use the setting as its Domain ID (DID[1:0]). More details are provided in the register section.

23.3.4 Clocks

The table found here describes the clock sources for SSARC. Please see the Clock Control chapter for clock setting, configuration, and gating information.

Table 23-2. Clocks

Clock Name	Description
ipg_clk	Peripheral access clock connected to SSARC_LP
ahb_clk	SSARC_HP and SSARC_LP clock

23.3.5 Reset

The SSARC module has two resets:

- System Reset
- SSARC_LP Reset

Please see the System Reset Controller (SRC) chapter for more details.

23.3.6 Interrupts

Please see the INT_STATUS register for more information on local interrupt generation.

23.4 External Signals

There are no external signals pinned out for the SSARC. Please see the Features section for details on external SSARC interfaces.

23.5 Initialization

The following steps are required for the initialization of the SSARC save and restore functions:

1. PGMC asserts save request signal (save_req)
2. SSARC gets the request, and checks if any group is assigned to this power domain
3. If multiple groups are assigned, the higher priority group will be triggered first
4. SSARC reads the descriptors belonging to this group
5. SSARC issues read/write operation through the bus interface based on the descriptor
6. SSARC moves to the next descriptor, and so on, until all the descriptors in this group are triggered
7. If there are other groups pending, SSARC moves on to the next group
8. After all the groups are triggered, SSARC asserts save done signal (save_done)

23.6 Memory Map and register definition

This section includes the SSARC module memory map and detailed descriptions of all registers.

23.6.1 SRAM Registers register descriptions

23.6.1.1 SRAM memory map

sram base address: 40CB_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0 - 3FF0	Description Address Register (SRAM0_0 - SRAM0_1023)	32	RW	Table 23-2

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4 - 3FF4	Description Data Register (SRAM1_0 - SRAM1_1023)	32	RW	Table 23-2
8 - 3FF8	Description Control Register (SRAM2_0 - SRAM2_1023)	32	RW	Table 23-2

23.6.1.2 Description Address Register (SRAM0_0 - SRAM0_1023)

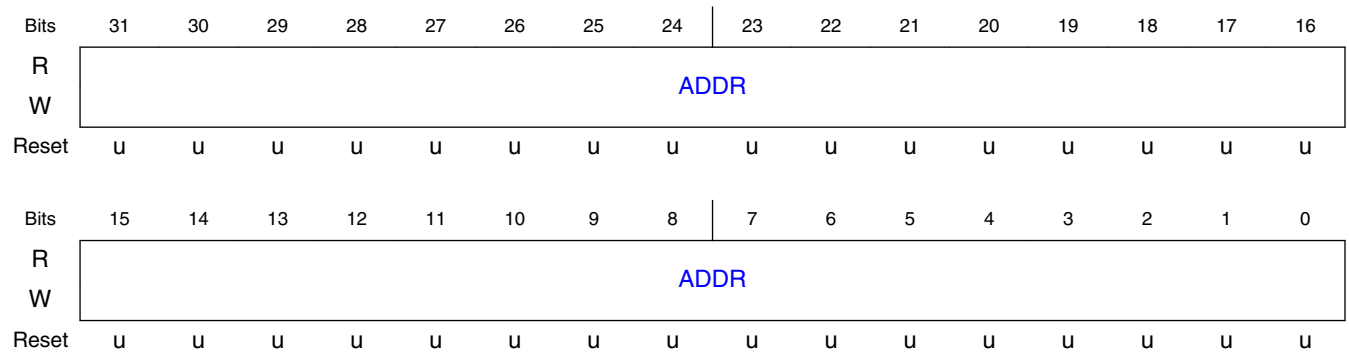
The SRAM array is 1024x72 bits and is split 3 ways between SRAM0[31:0]_n, SRAM1[63:32]_n, and SRAM2[71:64]_n, where n = 0 to 1024. This register represents the first 32 bits (1024x[31:0]) of the SRAM space.

23.6.1.2.1 Offset

For a = 0 to 1023:

Register	Offset
SRAM0_a	0h + (a × 10h)

23.6.1.2.2 Diagram



23.6.1.2.3 Fields

Field	Description
31-0 ADDR	Address field This field describes the 32-bit address of the register to be saved/restored

23.6.1.3 Description Data Register (SRAM1_0 - SRAM1_1023)

The SRAM array is 1024x72 bits and is split 3 ways between SRAM0[31:0]_n, SRAM1[63:32]_n, and SRAM2[71:64]_n, where n = 0 to 1024. This register represents the second 32 bits (1024x[63:32]) of the SRAM space.

23.6.1.3.1 Offset

For a = 0 to 1023:

Register	Offset
SRAM1_a	4h + (a × 10h)

23.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

23.6.1.3.3 Fields

Field	Description
31-0 DATA	Data field This field depends on the setting of the SRAM2_n[TYPE] bitfield. When SRAM2_n[TYPE]=W0 (0x1 - W0 - Write), this field retains the value to be written to the register on restore When SRAM2_n[TYPE]=SR (0x0 - SR - Save Read), this field will be loaded with the register value on save operation

23.6.1.4 Description Control Register (SRAM2_0 - SRAM2_1023)

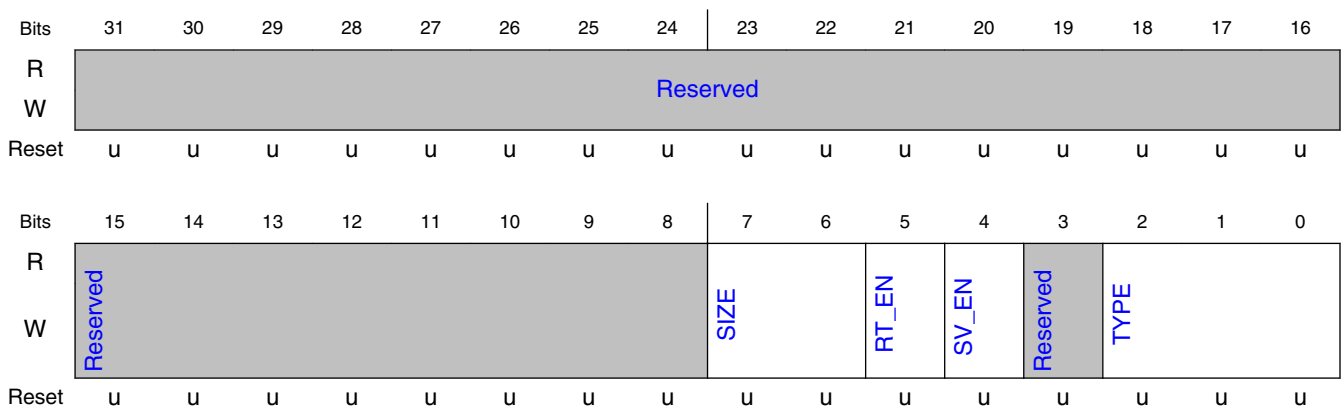
The SRAM array is 1024x72 bits and is split 3 ways between SRAM0[31:0]_n, SRAM1[63:32]_n, and SRAM2[71:64]_n, where n = 0 to 1024. This register represents the last 8 bits (1024x[71:64]) of the SRAM space.

23.6.1.4.1 Offset

For a = 0 to 1023:

Register	Offset
SRAM2_a	8h + (a × 10h)

23.6.1.4.2 Diagram



23.6.1.4.3 Fields

Field	Description
31-8 —	Reserved
7-6 SIZE	Size field 00 - 8-bit 01 - 16-bit 10 - 32-bit 11 - Reserved
5 RT_EN	Restore Enable Software can allocate one descriptor for both save and restore by setting both RT_EN and SV_EN. 0 - Do not use this descriptor for the restore operation

Table continues on the next page...

Field	Description
	1 - Use this descriptor for the restore operation
4 SV_EN	Save Enable Software can allocate one descriptor for both save and restore by setting both RT_EN and SV_EN. 0 - Do not use this descriptor in the save operation 1 - Use this descriptor in the save operation
3 —	Reserved
2-0 TYPE	Type field This field describes the type of operation on SRAM1_n[DATA] 000 - Read the register value on save operation and write it back on restore operation 001 - Always write a fixed value from SRAM1_n[DATA] 010 - Read register, OR with the SRAM1_n[DATA], and write it back 011 - Read register, AND with the SRAM1_n[DATA], and write it back 100 - Delay for number of cycles based on the SRAM1_n[DATA] 101 - Read the register until read_data[31:0] & SRAM1_n[DATA] == 0 110 - Read the register until read_data[31:0] & SRAM1_n[DATA] != 0 111 - Reserved

23.6.2 SSARC Registers register descriptions

23.6.2.1 SSARC memory map

ssarc base address: 40CB_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0 - 1E0	Descriptor Control0 a Register (DESC_CTRL0_0 - DESC_CTRL0_15)	32	RW	0000_0000
4 - 1E4	Descriptor Control1 a Register (DESC_CTRL1_0 - DESC_CTRL1_15)	32	RW	0000_0000
8 - 1E8	Descriptor Address Up a Register (DESC_ADDR_UP_0 - DESC_ADDR_UP_15)	32	RW	0000_0000
C - 1EC	Descriptor Address Down a Register (DESC_ADDR_DOWN_0 - DESC_ADDR_DOWN_15)	32	RW	0000_0000
200	Control Register (CTRL)	32	RW	0000_0000
204	Interrupt Status Register (INT_STATUS)	32	W1C	0000_0000
20C	HP Timeout Register (HP_TIMEOUT)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
21C	Hardware Request Pending Register (HW_GROUP_PENDING)	32	RO	0000_0000
220	Software Request Pending Register (SW_GROUP_PENDING)	32	RO	0000_0000

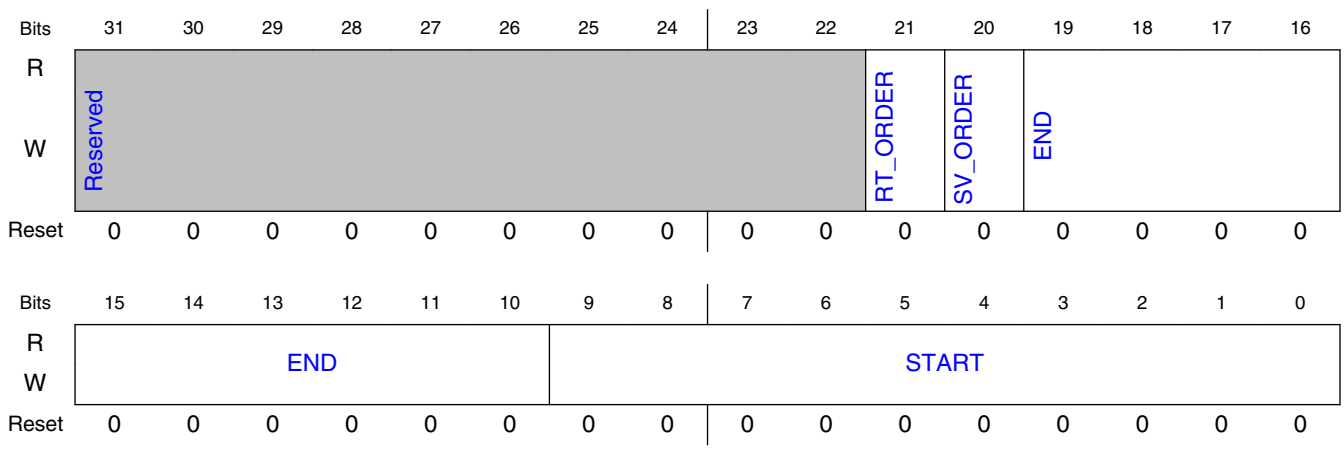
23.6.2.2 Descriptor Control0 a Register (DESC_CTRL0_0 - DESC_CTRL0_15)

23.6.2.2.1 Offset

For a = 0 to 15:

Register	Offset
DESC_CTRL0_a	0h + (a × 20h)

23.6.2.2.2 Diagram



23.6.2.2.3 Fields

Field	Description
31-22 —	Reserved

Table continues on the next page...

Field	Description
21 RT_ORDER	Restore order This field defines the order descriptors within the group are processed when restoring register values 0 - Descriptors within the group are processed from start to end 1 - Descriptors within the group are processed from end to start
20 SV_ORDER	Save Order This field defines the order descriptors within the group are processed when saving register values 0 - Descriptors within the group are processed from start to end 1 - Descriptors within the group are processed from end to start
19-10 END	End index This field describes the Index of the last descriptor that belongs to this group. It can be anywhere from 0-1023.
9-0 START	Start index This field describes the Index of the first descriptor that belongs to this group. It can be anywhere from 0-1023.

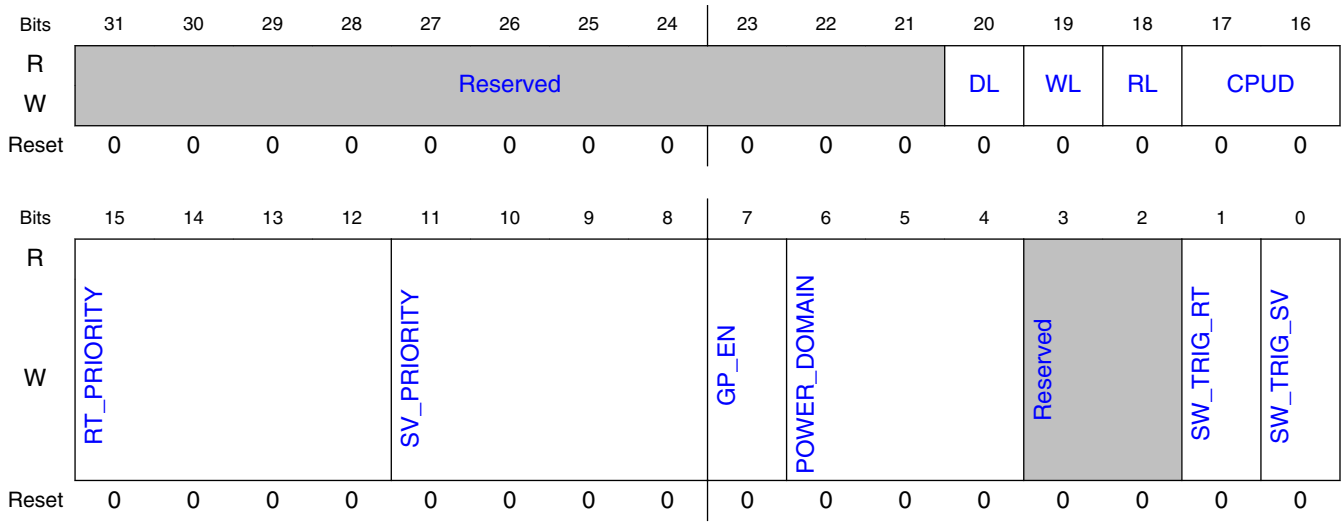
23.6.2.3 Descriptor Control1 a Register (DESC_CTRL1_0 - DESC_CTRL1_15)

23.6.2.3.1 Offset

For a = 0 to 15:

Register	Offset
DESC_CTRL1_a	4h + (a × 20h)

23.6.2.3.2 Diagram



23.6.2.3.3 Fields

Field	Description
31-21 —	Reserved
20 DL	Domain lock This bitfield locks the configuration of the CPU domain. Once locked, only the access from the same domain is allowed, access from other domain will be blocked. This includes both the control registers for the group and descriptors assigned to the group. This is a sticky bit. After it has been set, it can only be cleared by POR. 0 - Unlock 1 - Lock
19 WL	Write Lock This bit can be used to block all write accesses to the control registers and descriptors for the group. All writes are blocked--no dependency on domain. This is a sticky bit. After it has been set, it can only be cleared by POR. 0 - Group is unlocked (write access allowed) 1 - Group is locked (write access not allowed)
18 RL	Read Lock Lock the read access to the control register and descriptors in this group, no dependency on domain. Once set, the sticky bit can be only cleared by POR. 0 - Group is unlocked (read access allowed) 1 - Group is locked (read access not allowed)
17-16 CPUD	CPU Domain The ownership of this group: After lock, the register for this group can only be accessed by the CPU from the domain. The access read/write from the bus interface will use the setting as its Domain ID (DID[1:0]).
15-12	Restore Priority

Table continues on the next page...

Field	Description
RT_PRIORITY	Restore priority of current group, 0 has highest priority, 15 has lowest priority f two or more groups are configured with the same POWER_DOMAIN value, then the value of this field is used to determine the priority used to process each of the groups for restore operations.
11-8 SV_PRIORITY	Save Priority Save priority of current group, 0 has highest priority, 15 has lowest priority
7 GP_EN	Group Enable This field determines if the group is processed for save and restore operations 0 - Group disabled 1 - Group enabled
6-4 POWER_DOMAIN	This field describes the mapping (0-7) to external request signals from different domains. Please see the PGMC chapter for the BPC power domain assignments. 000 - PGMC_BPC0 001 - PGMC_BPC1 010 - PGMC_BPC2 011 - PGMC_BPC3 100 - PGMC_BPC4 101 - PGMC_BPC5 110 - PGMC_BPC6 111 - PGMC_BPC7
3-2 —	Reserved
1 SW_TRIG_RT	Software trigger restore Set this bit manually to trigger a restore operation. This bit will automatically clear when the restore operation is complete. 0 - No software restore request/software restore request complete 1 - Request a software restore operation/software restore operation in progress
0 SW_TRIG_SV	Software trigger save Set this bit to trigger save operation, hardware will clear it after assert interrupt 0 - No software save request/software restore request complete 1 - Request a software save operation/software restore operation in progress

23.6.2.4 Descriptor Address Up a Register (DESC_ADDR_UP_0 - DESC_ADDR_UP_15)

23.6.2.4.1 Offset

For a = 0 to 15:

Register	Offset
DESC_ADDR_UP_a	8h + (a × 20h)

23.6.2.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_UP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR_UP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.4.3 Fields

Field	Description
31-0 ADDR_UP	Address field (High) Highest address that can be accessed for the descriptors in this group If a descriptor attempts an access to an address larger than this value, the INT_STATUS[ADDR_ERR] bit will set and processing of additional descriptors in the group will abort.

23.6.2.5 Descriptor Address Down a Register (DESC_ADDR_DOWN_0 - DESC_ADDR_DOWN_15)

23.6.2.5.1 Offset

For a = 0 to 15:

Register	Offset
DESC_ADDR_DOWN_a	Ch + (a × 20h)

23.6.2.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_DOWN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR_DOWN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.5.3 Fields

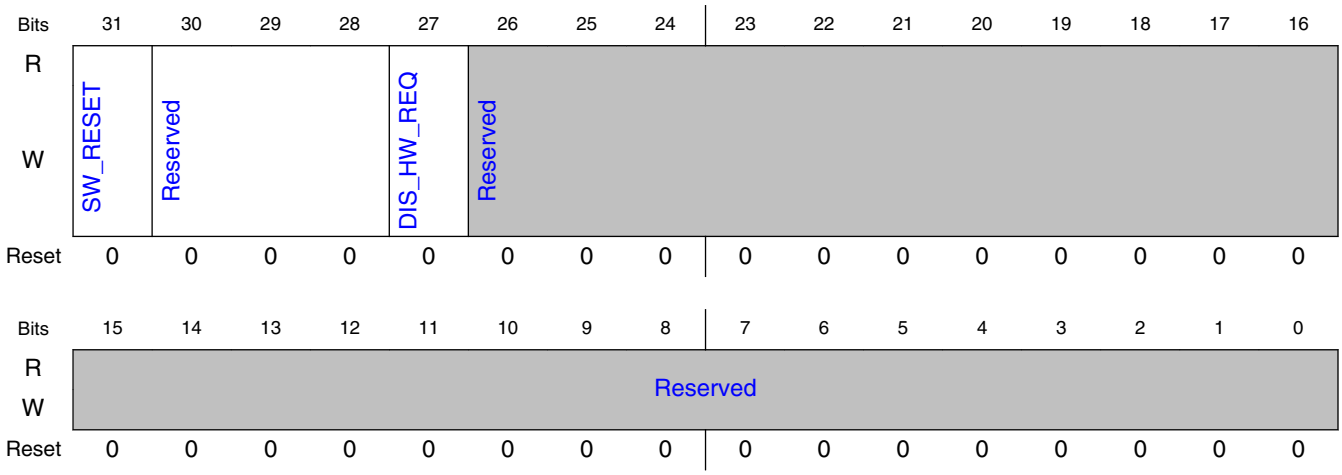
Field	Description
31-0 ADDR_DOWN	Address field (Low) Lowest address that can be accessed for the descriptors in this group If a descriptor attempts an access to an address lower than this value, the INT_STATUS[ADDR_ERR] bit will set and processing of additional descriptors in the group will abort.

23.6.2.6 Control Register (CTRL)

23.6.2.6.1 Offset

Register	Offset
CTRL	200h

23.6.2.6.2 Diagram



23.6.2.6.3 Fields

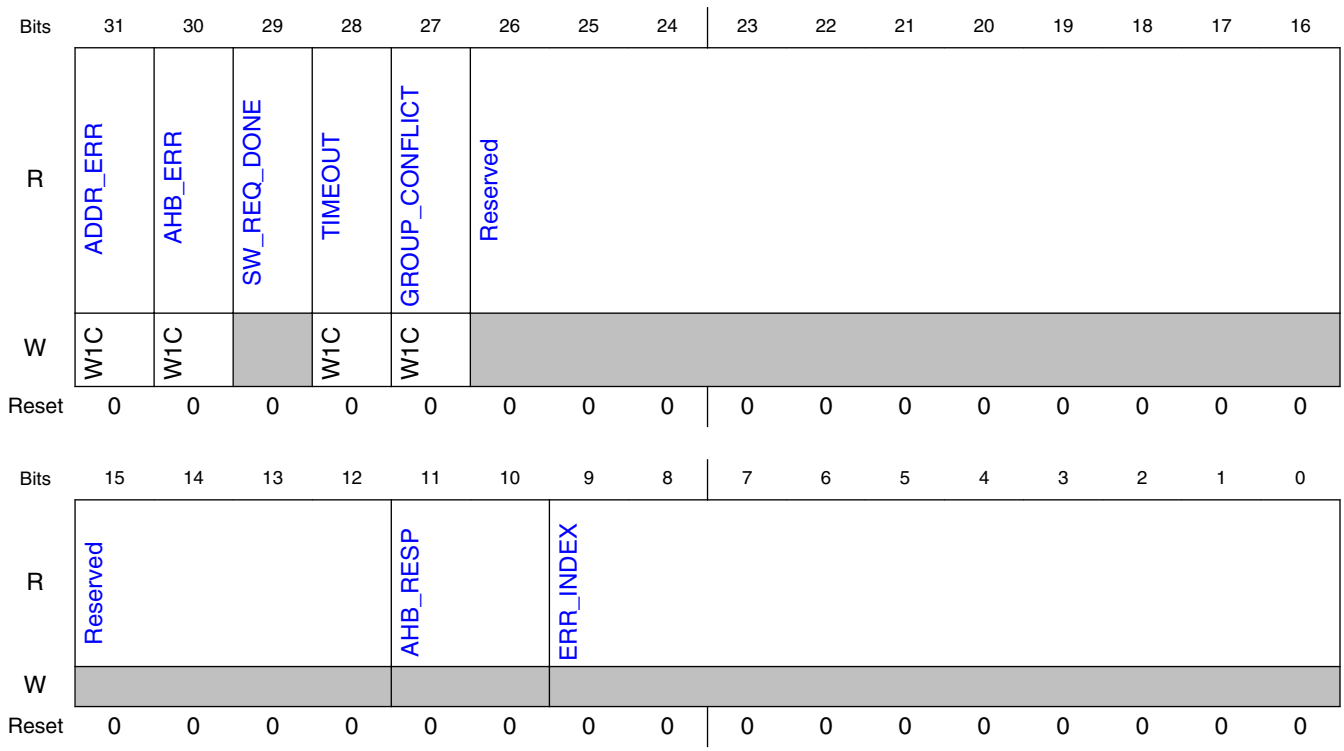
Field	Description
31 SW_RESET	Software reset This software reset is used to reset the logic inside SSARC_LP and SSARC_HP, including the internal state machine, which can be used when a fault may cause the SSARC block to hang. The SW_RESET will not reset the DESC_x Registers, since they have domain control/lock control registers. The SW_RESET will also not reset the Descriptors inside SRAM, which is not part of SSARC block. Please ensure enough delay is met to reset the SSARC block (at least 10us), then clear it.
30-28 —	Reserved
27 DIS_HW_REQ	Save/Restore request disable This bit can be used to disable save/restore requests from the PGMC module. Software triggered requests using SW_TRIGGER_SV and SW_TRIGGER_RT can still be used. 0 - PGMC save/restore requests enabled 1 - PGMC save/restore requests disabled
26-0 —	Reserved

23.6.2.7 Interrupt Status Register (INT_STATUS)

23.6.2.7.1 Offset

Register	Offset
INT_STATUS	204h

23.6.2.7.2 Diagram



23.6.2.7.3 Fields

Field	Description
31 ADDR_ERR	Address Error field If the descriptor address is not in the range (from desc_addr_low to desc_addr_up), assert addr_err. This bit is write 1 to clear. 0 - No address error 1 - An address error has occurred
30 AHB_ERR	AHB Error field If any AHB master access receives none-OKAY hresp, it will assert this bit when ssarc_hp_done. The error type can be seen in INT_STATUS[AHB_RESP]. Note, if multiple errors occur, only the last error will be kept 0 - No AHB error

Table continues on the next page...

Field	Description
	1 - An AHB error has occurred
29 SW_REQ_DONE	Software Request Done This field indicates if A software triggered save or restore process is completed. For more details see SW_DONE_STATUS register. This bit will only clear if all the bits in the SW_DONE_STATUS register are cleared. 0 - No software triggered requests or software triggered request still in progress 1 - Atleast one software triggered has been complete
28 TIMEOUT	Timeout field This field indicates if processing of a group has exceeded the time out value configured in the HP_TIMEOUT register 0 - No timeout event 1 - A timeout event has occurred
27 GROUP_CONFLICT	Group Conflict field This field gives error interrupt when one group is pending or in progress, and another request for same group is received from hardware. It should not occur in normal circumstances. Save and restore for same group should be treated as different request. If one group is pending or process, and another request for same group is received from software, it will be ignored. 0 - No group conflict error 1 - A group conflict error has occurred
26-12 —	Reserved
11-10 AHB_RESP	AHB Bus response field
9-0 ERR_INDEX	Error Index This field indicates which descriptor will trigger the AHB_ERR or ADDR_ERR interrupt. This block contains 1024 descriptors and this field represents the index of the descriptor that generated the error.

23.6.2.8 HP Timeout Register (HP_TIMEOUT)

23.6.2.8.1 Offset

Register	Offset
HP_TIMEOUT	20Ch

23.6.2.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TIMEOUT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TIMEOUT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.8.3 Fields

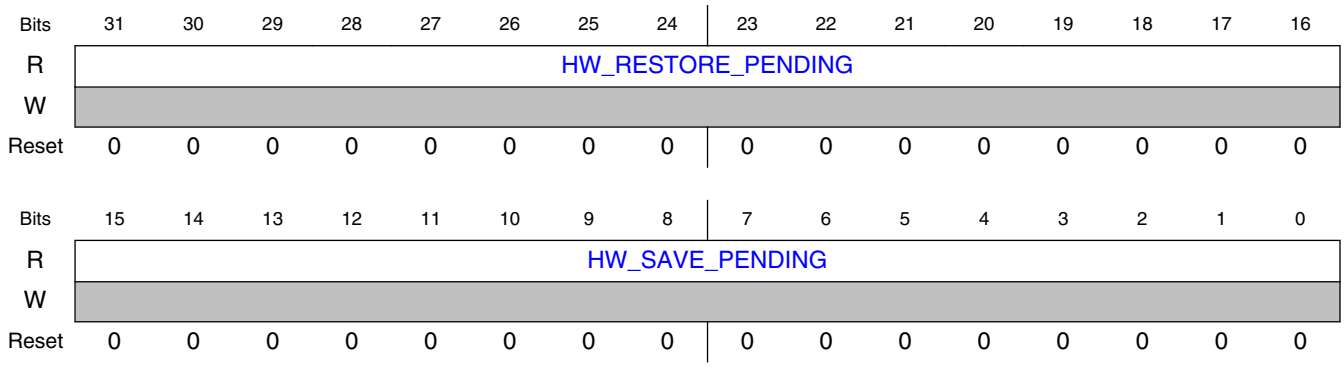
Field	Description
31-0 TIMEOUT_VALUE	Time out value This field contains the 32-bit timeout value for AHB clock. Clear to 0 will disable this feature.

23.6.2.9 Hardware Request Pending Register (HW_GROUP_PENDING)

23.6.2.9.1 Offset

Register	Offset
HW_GROUP_PENDING	21Ch

23.6.2.9.2 Diagram



23.6.2.9.3 Fields

Field	Description
31-16 HW_RESTORE_PENDING	This field indicates which groups are pending for restore from hardware request
15-0 HW_SAVE_PENDING	This field indicates which groups are pending for save from hardware request

23.6.2.10 Software Request Pending Register (SW_GROUP_PENDING)

23.6.2.10.1 Offset

Register	Offset
SW_GROUP_PENDING	220h

23.6.2.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_RESTORE_PENDING															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_SAVE_PENDING															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.10.3 Fields

Field	Description
31-16 SW_RESTORE_PENDING	This field indicates which groups are pending for restore from software request
15-0 SW_SAVE_PENDING	This field indicates which groups are pending for save from software request

Chapter 24
Secure Non-Volatile Storage (SNVS)

24.1 Chip-specific SNVS information

Table 24-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

24.2 Overview

The Secure Non-Volatile Storage (SNVS) is a companion module to the CAAM module.

The SNVS non-security functionality is described in this document, but the SNVS security functionality is described only in the Security Reference Manual.

24.2.1 Block diagram

The following figure illustrates the structure of SNVS.

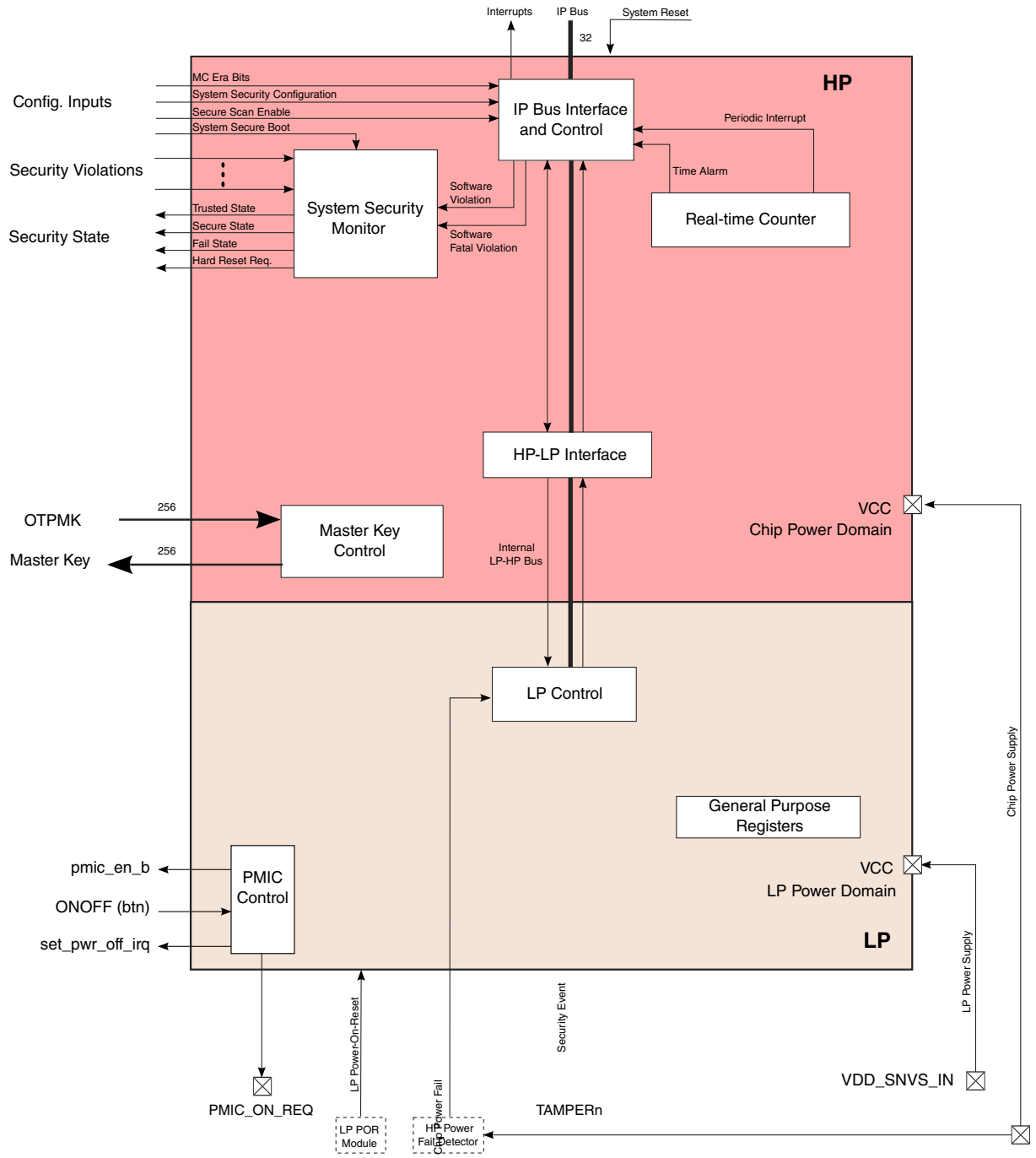


Figure 24-1. SNVS Block Diagram

24.2.2 Features

The following table summarizes the features of SNVS:

Table 24-2. SNVS feature list

Feature	Description	Links for Further Information
Real time counter (RTC)	• The RTC is driven by a dedicated clock, which is off when the system power is down. • Programmable time alarm interrupt • Periodic interrupt can be generated with software-selected frequency.	SNVS_HP Real Time Counter
General-purpose register	• The general-purpose register is available to software to store 128 bits of data. • The general-purpose register is zeroized when a security violation is detected. • If the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains), the general-purpose register value is retained even if the main chip is powered down.	Using the General-Purpose Register
Register access restrictions	• Some registers/values can be written only once per boot cycle.	privileged and non-privileged registers
Wakeup from power off	• Input signal from off chip requests SNVS_LP to power on the main SoC (Assuming that the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains). • Hardware debounces the input signal using software-specified signal bounce characteristics	LP Wake-Up Interrupt Enable

24.3 SNVS functional description

SNVS implements several non-security features that involve software interaction:

- reading or writing the Realtime Counter (RTC) (This is a non-privileged operation.) - software can also instruct SNVS to load the current SRTC value into the RTC
- reading or writing the General Purpose Register (GPR) (Note that there may be a significant delay when reading or writing registers in the LP section if the LP clock is different from the HP clock.)

The following sections describe in more detail the operation of SNVS.

24.3.1 SNVS power domains

In some versions of SNVS (including this version), the LP (Low Power) section is implemented in an independent power domain from the HP (High Power) section, and most other logic on the chip. Throughout the SNVS documentation whenever mention is made of "always-on" logic, this assumes a version of SNVS that implements an independent power domain for the LP section, and that the power for this section is supplied by an uninterrupted power supply. The purpose for the independent power domain is so that data can be retained and certain logic can remain functional even when the main chip logic is powered down. But this is possible only if the LP domain remains

powered via an uninterrupted power supply when the main chip power domain is powered off. Usually this uninterrupted power supply would be a battery, with possibly some power management logic to power the LP section from main power (and perhaps recharge the battery) when main power is on, and switch to battery power when the main power is off. In versions of SNVS with an independent LP power domain the LP section can be electrically isolated from the rest of the chip logic to ensure that its logic does not get corrupted when the main chip is powered down. If the battery runs down or is removed, an LP POR will occur when the LP section next powers up. Note that some OEMs may choose to connect LP power to HP/main chip power and dispense with a battery. In that case the SNVS will operate the same as an SNVS without an independent LP power domain. No state will be retained in the LP section when the chip is powered down, and an LP POR will occur whenever there is an HP POR.

24.3.2 Digital Low-Voltage Detector (LVD)

SNVS_LP incorporates a mechanism to detect interruptions of the SNVS_LP power supply that might cause the LP control, status, and secure counter values to change. The mechanism works as follows:

1. The LVD register (LPLVDR) is loaded with the known specific value 4173_6166h as part of the SNVS initialization process.
2. Subsequently, this register's value is continuously compared to the hardwired value 4173_6166h.
3. If the comparison indicates that any bit has changed, a low-voltage violation is asserted.

Digital low-voltage detection is always enabled and cannot be disabled. At LP POR this register is reset to all 0's, so the hardwired comparison fails and a low-voltage violation is reported. Therefore, before programming any feature in the SNVS the low-voltage violation should be cleared. The initialization software should write the proper value (4173_6166h) into LPLVDR (see SNVS_LP Digital Low-Voltage Detector Register (LPLVDR)) and should then clear the low-voltage event record in the LP status register (see SNVS_LP Status Register (LPSR)).

The following figure shows the LVD mechanism.

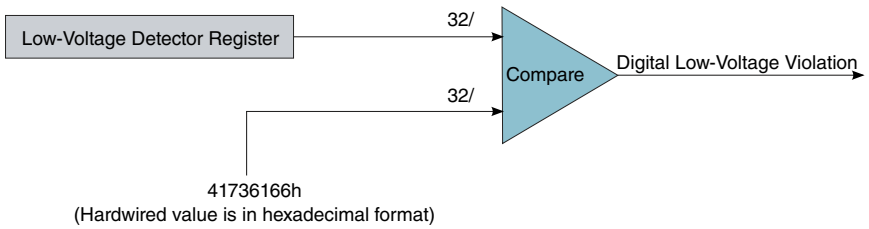


Figure 24-2. Digital low-voltage detector

24.3.3 Runtime Procedures

SNVS implements a number of features that are intended to be accessed by software at runtime (as opposed to accessed at boot time). These features include:

- Real Time Clock (see SNVS_HP Real Time Counter)
- General Purpose Register (see Using the General-Purpose Register)

Procedures for using these features are described in the following sections.

24.3.3.1 Using SNVS Timer Facilities

SNVS incorporates timer facilities that can optionally generate an interrupt at a specified time. As described in the following sections, SNVS_HP incorporates a Real Time Counter that is available for general use, and SNVS_LP incorporates a Secure Real Time Counter intended for security applications.

24.3.3.1.1 SNVS_HP Real Time Counter

SNVS_HP implements a real time counter that can be read or written by any application; it has no privileged software access restrictions. When the chip is powered down the RTC is not active and it is reset at chip POR. The RTC can be used to generate a functional interrupt request either at a specific time, or at a specific frequency, or both. To generate an interrupt request at a specific time HPTA_EN is set to 0, the desired time is written to HPTA_MS and HPTA_LS and then HPTA_EN is set to 1. HPTA_EN, HPTA_MS and HPTA_LS can be written by any software that has access to SNVS registers; there are no privileged access restrictions. The counter can be synchronized to the SNVS_LP SRTC by writing to the HP_TS bit of SNVS_HP Control Register. This is particularly useful if the SNVS_LP is powered from an uninterrupted power source because the RTC can then be set from a chip-internal time source.

24.3.3.1.2 RTC/SRTC control bits setting

All SNVS registers are programmed from the register bus, consequently any software-initiated changes are synchronized with the IP clock. Several registers can also change synchronously with the RTC/SRTC clock after they are programmed. To avoid IP clock and RTC/SRTC clock synchronization issues, the following values can be changed only when the corresponding function is disabled.

Table 24-3. RTC/SRTC synchronized values list

Function	Value/register	Control bit setting
HP section		
HP Real Time Counter	HPRTCMR and HPRTCLR Registers	RTC_EN = 0 : HPRTCMR/HPRTCLR **can** be programmed RTC_EN = 1 : HPRTCMR/HPRTCLR **cannot** be programmed
HP Time Alarm	HPTAMR and HPTALR Registers	HPTA_EN = 0 : HPTAMR/HPTALR **can** be programmed HPTA_EN = 1 : HPTAMR/HPTALR **cannot** be programmed
LP section		
LP Secure Real Time Counter	LPRTCMR and LPRTCLR Registers	SRTC_ENV = 0 : LPRTCMR/LPRTCLR **can** be programmed SRTC_ENV = 1 : LPRTCMR/LPRTCLR **cannot** be programmed
LP Time Alarm	LPTAR Register	LPTA_EN = 0 : LPTAR **can** be programmed LPTA_EN = 1 : LPTAR **cannot** be programmed

Use the following steps to program synchronized values:

1. Check the enable bit value. If set, clear it.

2. Verify that the enable bit is cleared. There are two reasons to verify the enable bit's setting:

- Enable bit clearing does not happen immediately; it takes three IP clock cycles and two RTC/SRTC clock cycles to change the enable bit's value.
- If the enable bit is locked for programming, it cannot be cleared.

3. Program the desired value.

4. Set the enable bit; it takes three IP clock cycles and two RTC/SRTC clock cycles for the bit to set.

NOTE

Incrementing the value programmed into RTC/SRTC registers by two compensates for the two RTC/SRTC clock cycle delay that is required to enable the counter.

24.3.3.1.3 Reading RTC and SRTC values

Software should follow the following procedure to ensure that it has read correct data from the RTC (HPRTCMR and HPRTCLR) and SRTC (LPSRTCMR and LPSRTCLR) registers:

- Read the most-significant half and the least-significant half of the RTC/SRTC and then read both halves again. If the values read are the same both times, the value is correct.
- If the two consecutive pairs of reads yield different results, perform two more reads.

The worst case scenario may require three sessions of two consecutive pairs of reads. There are several reasons that the values may be incorrectly read initially:

- Synchronization issues between the RTC/SRTC clock and the system clock
- Since the counter continues to increment, there may be a carry from the least-significant 32-bits to the most-significant bits in between reading the two halves of the counter

24.3.3.2 Using Other SNVS Registers

The sections below describe how to use the General Purpose Register.Monotonic Counter.The sections below describe how to use the General Purpose Register and the Monotonic Counter.

24.3.3.2.1 Using the General-Purpose Register

SNVS implements a 128-bit general-purpose register allows software to store a small amount of data. To maintain backward compatibility with versions of SNVS that implement only a 32-bit general purpose register, the most-significant word of the general purpose register is aliased to the original legacy address, and to maintain backward compatibility with versions of snvs_module_name that implement a 128-bit general purpose register, the most-significant half of the general purpose register is aliased to the previous legacy address address. The data in the GPR will be retained during system power-down mode as long as the SNVS_LP remains powered by an uninterrupted power source.

24.3.4 Clocks

The SNVS has the following clock sources:

- System peripheral clock input. This clock is used by the SNVS's internal logic, for example, the Security State Machine. This clock can be gated outside of the module when the SNVS indicates that it is not in use.
- HP RTC clock. This clock is used by SNVS_HP real-time counter. This clock does not need to be synchronous with other clocks.

24.3.5 Reset

The table below shows the different resets associated with this module.

Table 24-4. Reset summary

Reset	Source	Characteristics	Internally resets
HP hard	ipg_hard_async_reset_b	active-low, asynchronous	All SNVS_HP and SNVS_LP registers and flops.
LP Power On Reset (POR)	lp_por_b	active-low, asynchronous	All SNVS_LP registers and flops.
LP software reset	software	active-high, synchronous, one cycle	All SNVS_LP registers and flops. The LP software reset can be asserted if not disabled.

24.4 External Signals

Table 24-5. External Signals

Signal	Description	Direction
TAMPERn	External Tamper signals	I
PMIC_ON_REQ	PMIC ON Request signal	I

24.5 Initialization of SNVS

SNVS is implemented in two sections (HP and LP) that both must be initialized by software. If the SNVS_LP is powered by an uninterrupted power source that is separate from main SoC power, then SNVS can operate in either of two modes, depending upon whether the main SoC power is on or off.

- During main SoC power-down SNVS_HP is powered-down, but SNVS_LP is powered from the backup power supply and is electrically isolated from the rest of the chip. In this mode SNVS_LP keeps its registers' values and monitors the SNVS_LP tamper detection inputs, but the LP registers cannot be read or written.
- During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Both LP and HP registers can be read and written (locks and privilege modes permitting). Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational.

Since the HP and LP sections reside in different power domains, the POR for the two sections can occur at different times. If the SNVS_LP section remains powered by an uninterrupted power source when the main SoC power is off, SNVS_LP is initialized rarely, typically once when the device is first powered on and again whenever the battery is replaced. During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational. The SNVS_HP section is powered from the main SoC power, so it must be initialized after the device is powered on. If the SNVS_LP section is powered from the main SoC power rather than from an uninterrupted power source, the SNVS_LP section must also be initialized at SoC POR.

- Initializing the LP section
  - The following steps should be completed to properly initialize the SNVS LP section (required only on LP POR, i.e. when the battery is replaced):
    - Software should write the proper initialization value (41736166h) into the LP Digital Low-Voltage Detector Register and clear the low-voltage event record in the LP status register. See Digital Low-Voltage Detector (LVD) for more details.
- Initializing the HP section
  - The following steps should be completed to properly initialize the SNVS HP section (required on HP POR, i.e. SoC POR):
    - Perform normal boot to put the SNVS into a functional state (Non-secure, Trusted, Secure) (see HP Command Register, SSM_ST bitfield).
    - Program SNVS general functions/configurations (see HP Control Register).

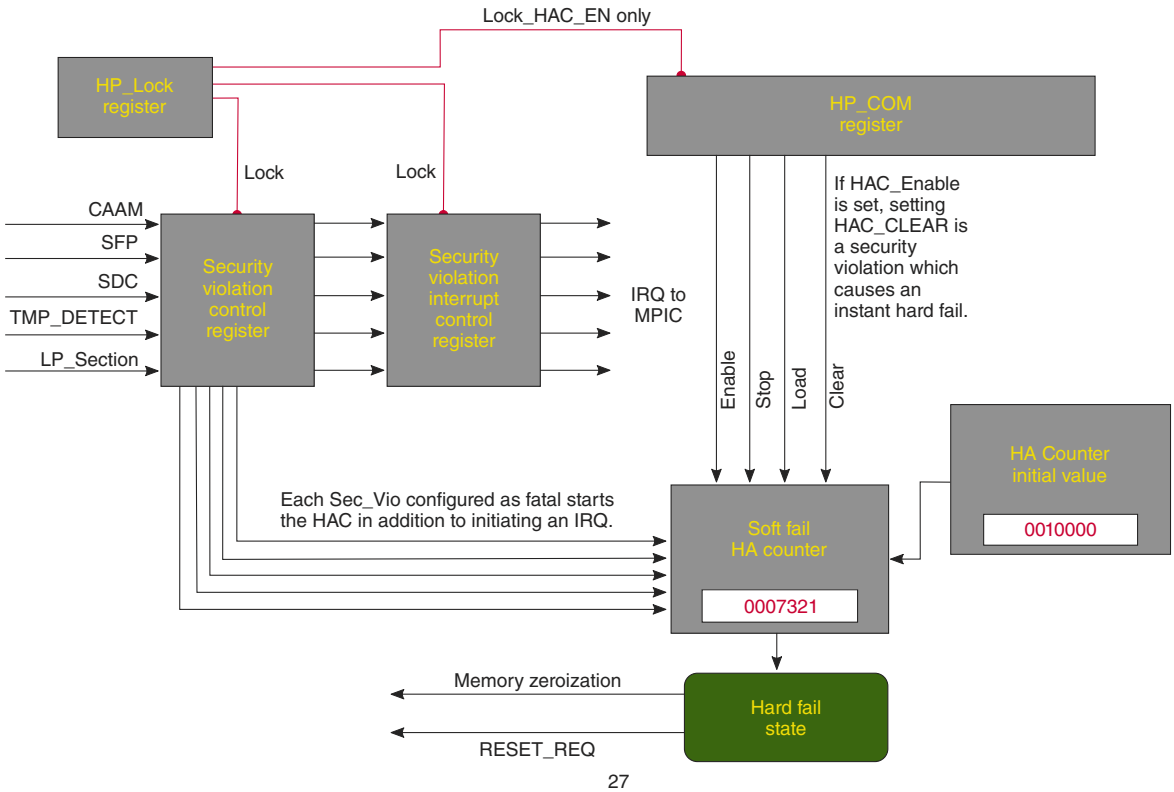


Figure 24-3. Relationship Between the Registers

24.6 Memory Map and register definition

This section contains detailed register descriptions for the SNVS registers. Each description includes a standard register diagram and register table. The register table provides detailed descriptions of the register bit and field functions, in bit order.

SNVS registers consist of two types:

- Privileged read/write accessible
- Non-privileged read/write accessible

Privileged read/write accessible registers can only be accessed for read/write by privileged software. Unauthorized write accesses are ignored, and unauthorized read accesses return zero. Non-privileged software can access privileged access registers when the non-privileged software access enable bit is set in the SNVS_HP Command Register.

- Non-Secure
- Trusted
- Secure

Non-privileged read/write accessible registers are read/write accessible by any software.

The LP register values are set only on LP POR and are unaffected by System (HP) POR. The HP registers are set only on System POR and are unaffected by LP POR.

The following table shows the SNVS main memory map.

NOTE

For more information on security-related bitfields, see the Security Reference Manual.

24.6.1 SNVS register descriptions

24.6.1.1 SNVS memory map

SNVS base address: 40C9_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4	SNVS_HP Command Register (HPCOMR)	32	RW	0000_0000
8	SNVS_HP Control Register (HPCR)	32	RW	0000_0000
14	SNVS_HP Status Register (HPSR)	32	RW	8000_0000
24	SNVS_HP Real Time Counter MSB Register (HPRTCMR)	32	RW	0000_0000
28	SNVS_HP Real Time Counter LSB Register (HPRTCLR)	32	RW	0000_0000
2C	SNVS_HP Time Alarm MSB Register (HPTAMR)	32	RW	0000_0000
30	SNVS_HP Time Alarm LSB Register (HPTALR)	32	RW	0000_0000
34	SNVS_LP Lock Register (LPLR)	32	RW	0000_0000
38	SNVS_LP Control Register (LPCR)	32	RW	0000_0020
4C	SNVS_LP Status Register (LPSR)	32	RW	0000_0008
5C	SNVS_LP Secure Monotonic Counter MSB Register (LPSMCMR)	32	RW	0000_0000
60	SNVS_LP Secure Monotonic Counter LSB Register (LPSMCLR)	32	RW	0000_0000
64	SNVS_LP Digital Low-Voltage Detector Register (LPLVDR)	32	RW	0000_0000
68	SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)	32	RW	0000_0000
90 - 9C	SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0_alias - LPGPR3_alias)	32	RW	0000_0000
100 - 10C	SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)	32	RW	0000_0000
BF8	SNVS_HP Version ID Register 1 (HPVIDR1)	32	RO	003E_0103
BFC	SNVS_HP Version ID Register 2 (HPVIDR2)	32	RO	0600_0500

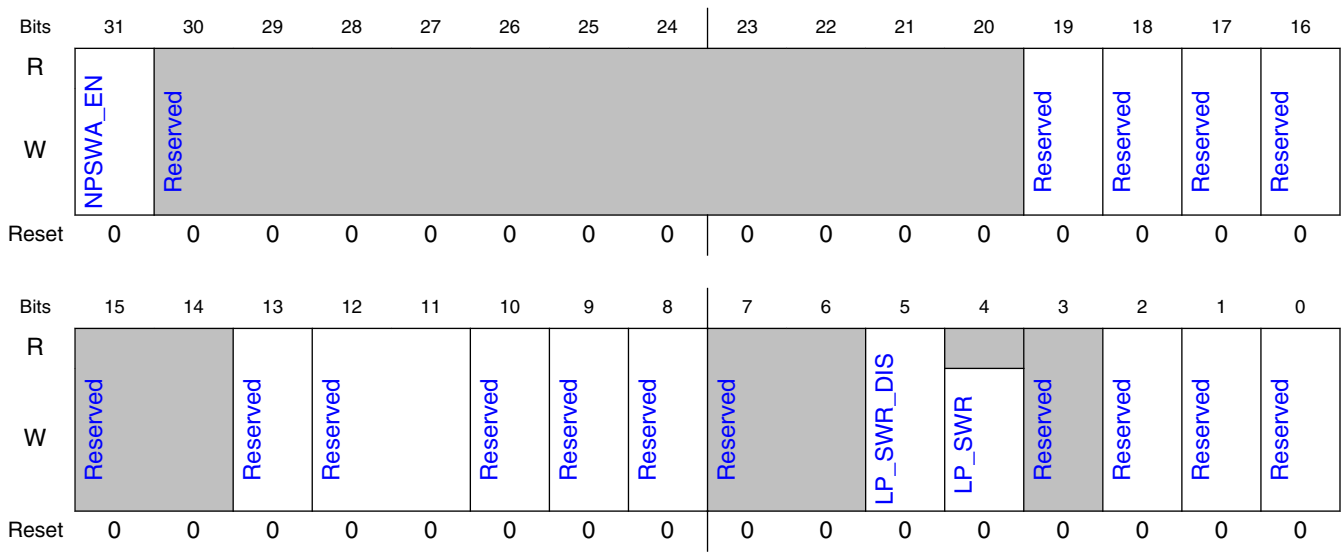
24.6.1.2 SNVS_HP Command Register (HPCOMR)

The SNVS_HP Command Register contains the command, configuration, and control bits for the SNVS block. This is a privileged write register.

24.6.1.2.1 Offset

Register	Offset
HPCOMR	4h

24.6.1.2.2 Diagram



24.6.1.2.3 Fields

Field	Description
31 NPSWA_EN	Non-Privileged Software Access Enable When set, allows non-privileged software to access all SNVS registers, including those that are privileged software read/write access only. 0 Only privileged software can access privileged registers 1 Any software can access privileged registers
30-20 —	Reserved
19	Reserved

Table continues on the next page...

Field	Description
—	
18 —	Reserved
17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13 —	Reserved
12-11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7-6 —	Reserved
5 LP_SWR_DIS	LP Software Reset Disable When set, disables the LP software reset. Once set, this bit can only be reset by the system reset. 0 - LP software reset is enabled 1 - LP software reset is disabled
4 LP_SWR	LP Software Reset When set to 1, the registers in the SNVS_LP section are reset. 0 - No Action 1 - Reset LP section
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

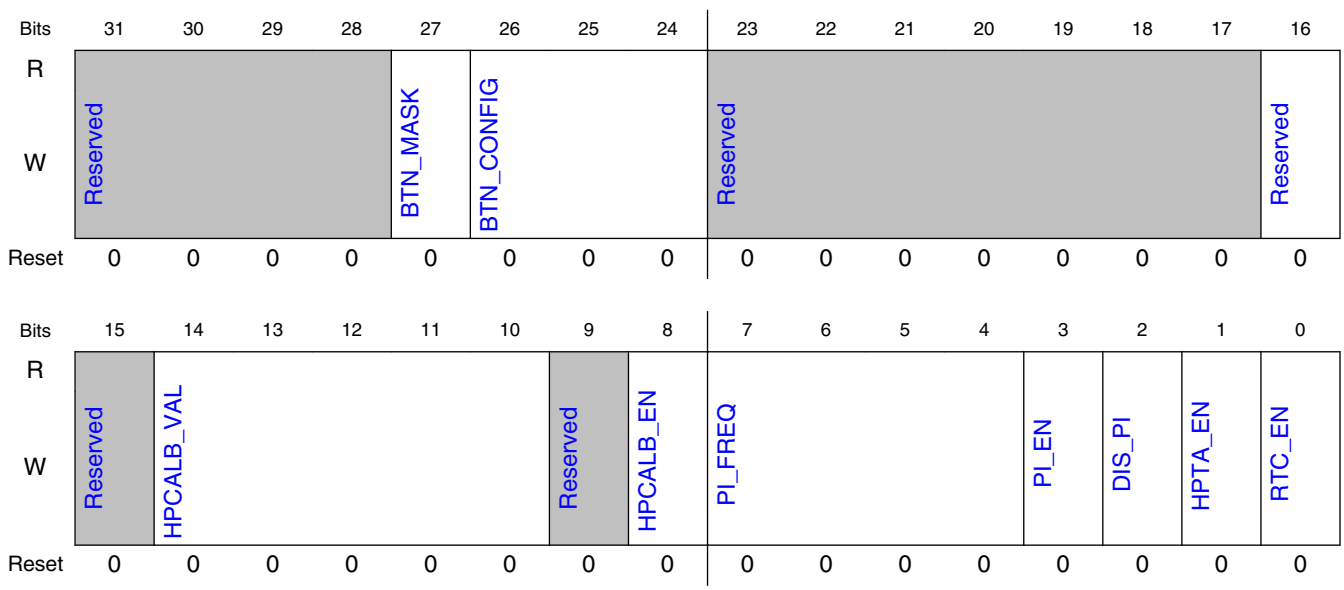
24.6.1.3 SNVS_HP Control Register (HPCR)

The SNVS_HP Control Register contains various control bits of the HP section of SNVS. This is *not* a privileged write register.

24.6.1.3.1 Offset

Register	Offset
HPCR	8h

24.6.1.3.2 Diagram



24.6.1.3.3 Fields

Field	Description
31-28 —	Reserved
27 BTN_MASK	Button interrupt mask. This bit is used to mask the button (BTN) interrupt request. 0: Interrupt disabled 1: Interrupt enabled
26-24	Button Configuration.

Table continues on the next page...

Field	Description
BTN_CONFIG	This field is used to configure which feature of the button (BTN) input signal constitutes "active". 000: Button signal is active high 001: Button signal is active low 010: Button signal is active on the falling edge 011: Button signal is active on the rising edge 100: Button signal is active on any edge All other patterns are Reserved
23-17 —	Reserved
16 —	Reserved
15 —	Reserved
14-10 HPCALB_VAL	HP Calibration Value Defines signed calibration value for the HP Real Time Counter. This field can be programmed only when RTC Calibration is disabled (HPCALB_EN is not set). This is a 5-bit 2's complement value, hence the allowable calibration values are in the range from -16 to +15 counts per 32768 ticks of the counter. 00000 - +0 counts per each 32768 ticks of the counter 00001 - +1 counts per each 32768 ticks of the counter 00010 - +2 counts per each 32768 ticks of the counter 01111 - +15 counts per each 32768 ticks of the counter 10000 - -16 counts per each 32768 ticks of the counter 10001 - -15 counts per each 32768 ticks of the counter 11110 - -2 counts per each 32768 ticks of the counter 11111 - -1 counts per each 32768 ticks of the counter
9 —	Reserved
8 HPCALB_EN	HP Real Time Counter Calibration Enabled Indicates that the time calibration mechanism is enabled. 0 - HP Timer calibration disabled 1 - HP Timer calibration enabled
7-4 PI_FREQ	Periodic Interrupt Frequency Defines frequency of the periodic interrupt. The interrupt is generated when a zero-to-one or one-to-zero transition occurs on the selected bit of the HP Real Time Counter and Real Time Counter and Periodic Interrupt are both enabled (RTC_EN and PI_EN are set). It is recommended to program this field when Periodic Interrupt is disabled (PI_EN is not set). The possible frequencies are: 0000 - - bit 0 of the HPRTCLR is selected as a source of the periodic interrupt 0001 - - bit 1 of the HPRTCLR is selected as a source of the periodic interrupt 0010 - - bit 2 of the HPRTCLR is selected as a source of the periodic interrupt 0011 - - bit 3 of the HPRTCLR is selected as a source of the periodic interrupt 0100 - - bit 4 of the HPRTCLR is selected as a source of the periodic interrupt

Table continues on the next page...

Field	Description
	0101 - - bit 5 of the HPRTCLR is selected as a source of the periodic interrupt 0110 - - bit 6 of the HPRTCLR is selected as a source of the periodic interrupt 0111 - - bit 7 of the HPRTCLR is selected as a source of the periodic interrupt 1000 - - bit 8 of the HPRTCLR is selected as a source of the periodic interrupt 1001 - - bit 9 of the HPRTCLR is selected as a source of the periodic interrupt 1010 - - bit 10 of the HPRTCLR is selected as a source of the periodic interrupt 1011 - - bit 11 of the HPRTCLR is selected as a source of the periodic interrupt 1100 - - bit 12 of the HPRTCLR is selected as a source of the periodic interrupt 1101 - - bit 13 of the HPRTCLR is selected as a source of the periodic interrupt 1110 - - bit 14 of the HPRTCLR is selected as a source of the periodic interrupt 1111 - - bit 15 of the HPRTCLR is selected as a source of the periodic interrupt
3 PI_EN	HP Periodic Interrupt Enable The periodic interrupt can be generated only if the HP Real Time Counter is enabled. 0 - HP Periodic Interrupt is disabled 1 - HP Periodic Interrupt is enabled
2 DIS_PI	Disable periodic interrupt in the functional interrupt 0 - Periodic interrupt will trigger a functional interrupt 1 - Disable periodic interrupt in the function interrupt
1 HPTA_EN	HP Time Alarm Enable When set, the time alarm interrupt is generated if the value in the HP Time Alarm Registers is equal to the value of the HP Real Time Counter. This bit syncs with the 32KHz clock. It won't update with the bus clock. 0 - HP Time Alarm Interrupt is disabled 1 - HP Time Alarm Interrupt is enabled
0 RTC_EN	HP Real Time Counter Enable. This bit syncs with the 32KHz clock. It won't update with the bus clock. 0 - RTC is disabled 1 - RTC is enabled

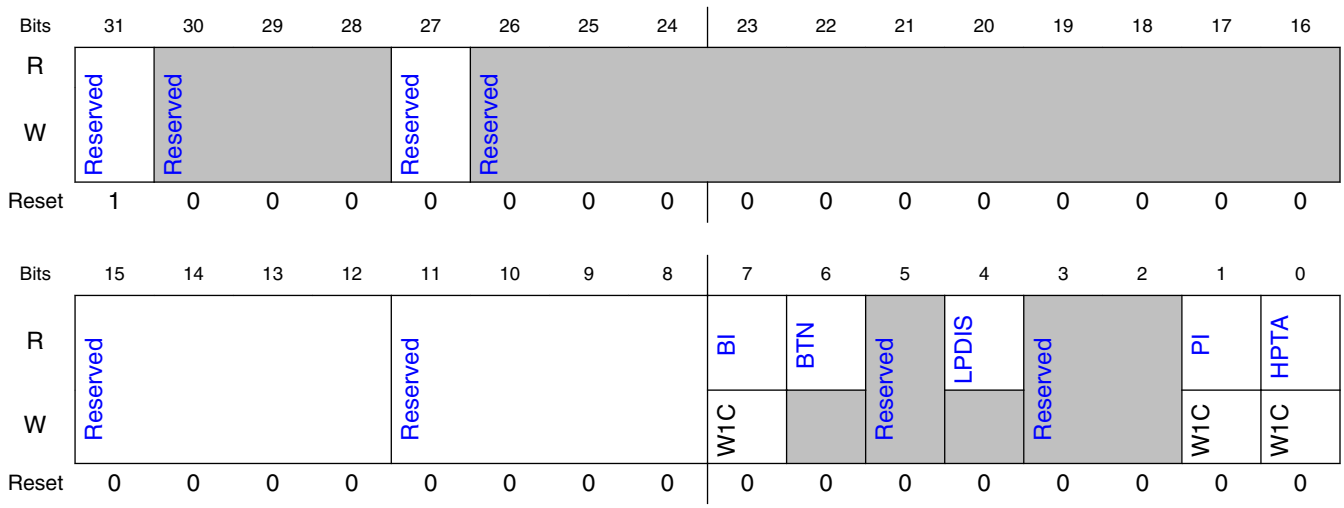
24.6.1.4 SNVS_HP Status Register (HPSR)

The HP Status Register reflects the internal state of the SNVS. This is *not* a privileged write register.

24.6.1.4.1 Offset

Register	Offset
HPSR	14h

24.6.1.4.2 Diagram



24.6.1.4.3 Fields

Field	Description
31 —	Reserved
30-28 —	Reserved
27 —	Reserved
26-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7 BI	Button Interrupt Signal ipi_snvs_btn_int_b was asserted.
6 BTN	Button Value of the BTN input. This is the external button used for PMIC control. 0: BTN not pressed 1: BTN pressed
5 —	Reserved

Table continues on the next page...

Field	Description
4 LPDIS	Low Power Disable If 1, the low power section has been disabled by means of an input signal to SNVS.
3-2 —	Reserved
1 PI	Periodic Interrupt Indicates that periodic interrupt has occurred since this bit was last cleared. 0 - No periodic interrupt occurred. 1 - A periodic interrupt occurred.
0 HPTA	HP Time Alarm Indicates that the HP Time Alarm has occurred since this bit was last cleared. 0 - No time alarm interrupt occurred. 1 - A time alarm interrupt occurred.

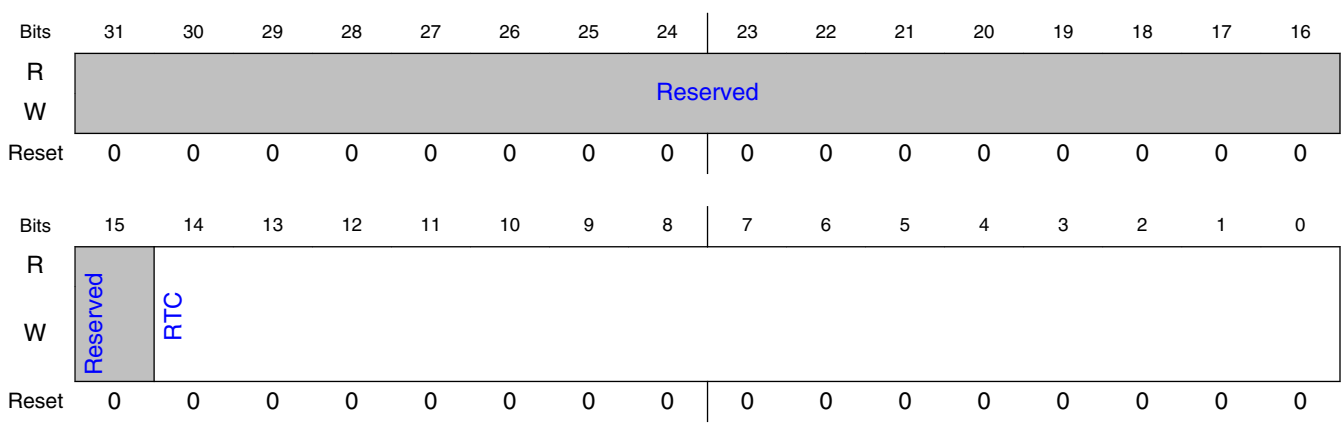
24.6.1.5 SNVS_HP Real Time Counter MSB Register (HPRTCMR)

The SNVS_HP Real Time Counter MSB register contains the 15 most-significant bits of the HP Real Time Counter. This is *not* a privileged write register.

24.6.1.5.1 Offset

Register	Offset
HPRTCMR	24h

24.6.1.5.2 Diagram



24.6.1.5.3 Fields

Field	Description
31-15 —	Reserved
14-0 RTC	HP Real Time Counter The most-significant 15 bits of the RTC. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

24.6.1.6 SNVS_HP Real Time Counter LSB Register (HPRTCLR)

The SNVS_HP Real Time Counter LSB register contains the 32 least-significant bits of the HP real time counter. This is *not* a privileged write register.

24.6.1.6.1 Offset

Register	Offset
HPRTCLR	28h

24.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.6.3 Fields

Field	Description
31-0	HP Real Time Counter

Field	Description
RTC	least-significant 32 bits. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

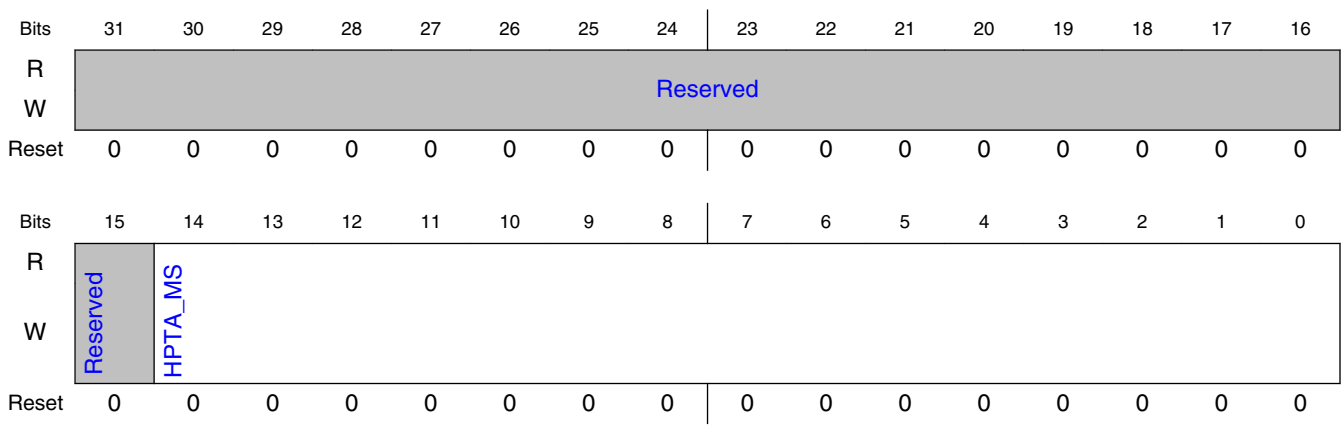
24.6.1.7 SNVS_HP Time Alarm MSB Register (HPTAMR)

The SNVS_HP Time Alarm MSB register contains the most-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

24.6.1.7.1 Offset

Register	Offset
HPTAMR	2Ch

24.6.1.7.2 Diagram



24.6.1.7.3 Fields

Field	Description
31-15 —	Reserved
14-0 HPTA_MS	HP Time Alarm, most-significant 15 bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

24.6.1.8 SNVS_HP Time Alarm LSB Register (HPTALR)

The SNVS_HP Time Alarm LSB register contains the 32 least-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

24.6.1.8.1 Offset

Register	Offset
HPTALR	30h

24.6.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.8.3 Fields

Field	Description
31-0 HPTA_LS	HP Time Alarm, 32 least-significant bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

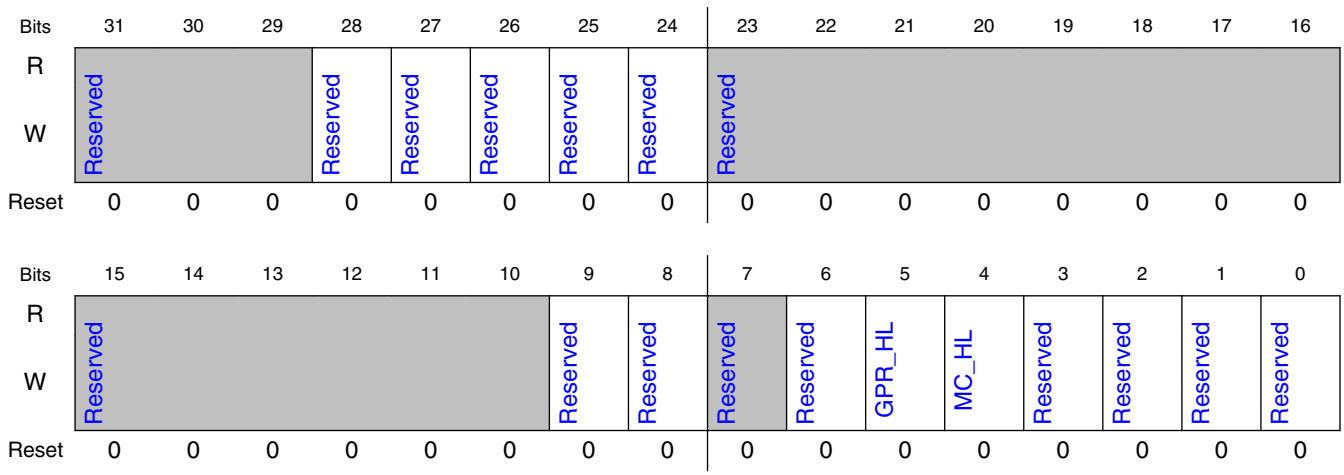
24.6.1.9 SNVS_LP Lock Register (LPLR)

The SNVS_LP Lock Register contains lock bits for the SNVS_LP registers. This is a privileged write register.

24.6.1.9.1 Offset

Register	Offset
LPLR	34h

24.6.1.9.2 Diagram



24.6.1.9.3 Fields

Field	Description
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23-10 —	Reserved
9 —	Reserved

Table continues on the next page...

Field	Description
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 GPR_HL	General Purpose Register Hard Lock When set, prevents any writes to the GPR. Once set, this bit can only be reset by the LP POR. 0 - Write access is allowed. 1 - Write access is not allowed.
4 MC_HL	Monotonic Counter Hard Lock When set, prevents any writes (increments) to the MC Registers and MC_ENV bit. Once set, this bit can only be reset by the LP POR. 0 - Write access (increment) is allowed. 1 - Write access (increment) is not allowed.
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

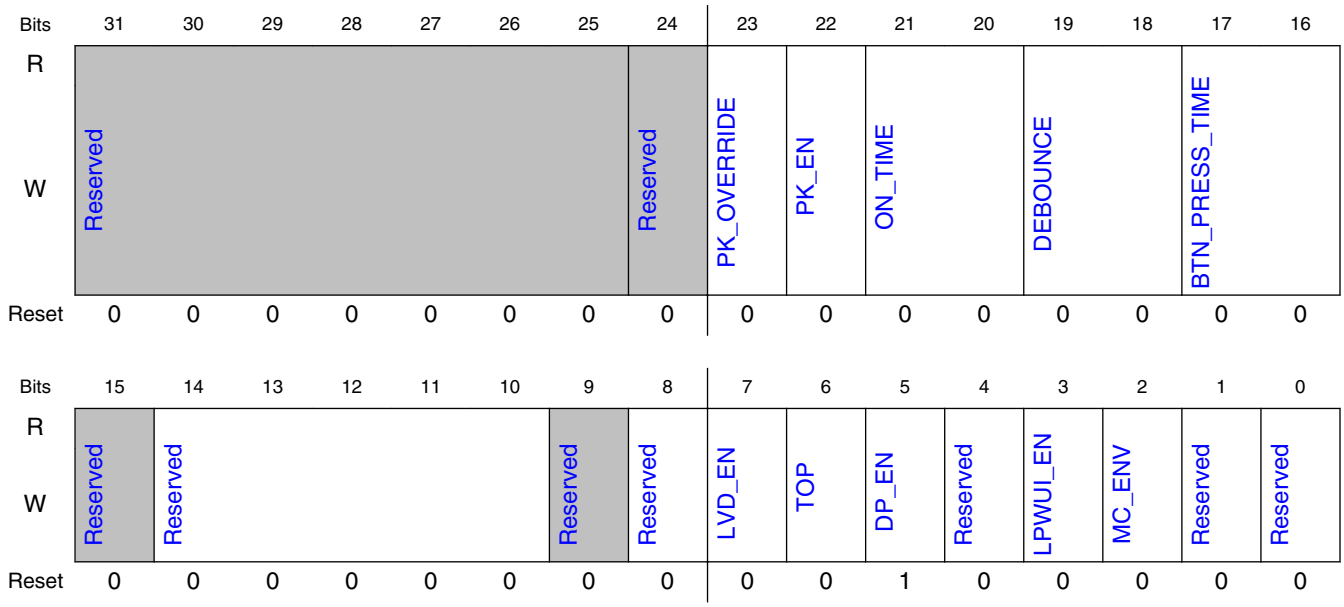
24.6.1.10 SNVS_LP Control Register (LPCR)

The SNVS_LP Control Register contains various control bits of the LP section of SNVS. This is a privileged write register.

24.6.1.10.1 Offset

Register	Offset
LPCR	38h

24.6.1.10.2 Diagram



24.6.1.10.3 Fields

Field	Description
31-25 —	Reserved
24 —	Reserved
23 PK_OVERRIDE	PMIC On Request Override The value written to PK_OVERRIDE will be asserted on output signal snvs_lp_pk_override. That signal is used to override the IOMUX control for the PMIC I/O pad.
22 PK_EN	PMIC On Request Enable The value written to PK_EN will be asserted on output signal snvs_lp_pk_en. That signal is used to turn off the pullup/pulldown circuitry in the PMIC I/O pad.
21-20 ON_TIME	The ON_TIME field is used to configure the period of time after BTN is asserted before pmic_en_b is asserted to turn on the SoC power. 00: 500msec off->on transition time 01: 50msec off->on transition time 10: 100msec off->on transition time 11: 0msec off->on transition time
19-18 DEBOUNCE	This field configures the amount of debounce time for the BTN input signal. 00: 50msec debounce 01: 100msec debounce 10: 500msec debounce

Table continues on the next page...

Field	Description
	11: 0msec debounce
17-16 BTN_PRESS_TIME	This field configures the button press time out values for the PMIC Logic. 00 : 5 secs 01 : 10 secs 10 : 15 secs 11 : long press disabled (pmic_en_b will not be asserted regardlessof how long BTN is asserted)
15 —	Reserved
14-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 LVD_EN	Digital Low-Voltage Event Enable By default the detection of a low-voltage event does not cause the pmic_en_b signal to be asserted. Setting the Digital Low-Voltage Event Enable bit to 1 enables the low-voltage event for the PMIC. 0 - disabled 1 - enabled
6 TOP	Turn off System Power Asserting this bit causes a signal to be sent to the Power Management IC to turn off the system power. This bit will clear once power is off. This bit is only valid when the Dumb PMIC is enabled. 0 - Leave system power on. 1 - Turn off system power.
5 DP_EN	Dumb PMIC Enabled When set, software can control the system power. When cleared, the system requires a Smart PMIC to automatically turn power off. 0 - Smart PMIC enabled. 1 - Dumb PMIC enabled.
4 —	Reserved
3 LPWUI_EN	LP Wake-Up Interrupt Enable This interrupt line should be connected to the external pin and is intended to inform the external chip about an SNVS_LP event (tamper event, MC rollover, SRTC rollover, or time alarm). This wake-up signal can be asserted only when the chip (HP section) is powered down, and the LP section is isolated. 0 LP wake-up interrupt is disabled. 1 LP wake-up interrupt is enabled.
2 MC_ENV	Monotonic Counter Enabled and Valid When set, the MC can be incremented (by write transaction to the LPSMCMR or LPSMCLR). Once MC_SL or MC_HL bit is set this bit can be changed only by LP POR. 0 - MC is disabled or invalid.

Table continues on the next page...

Field	Description
	1 - MC is enabled and valid.
1 —	Reserved
0 —	Reserved

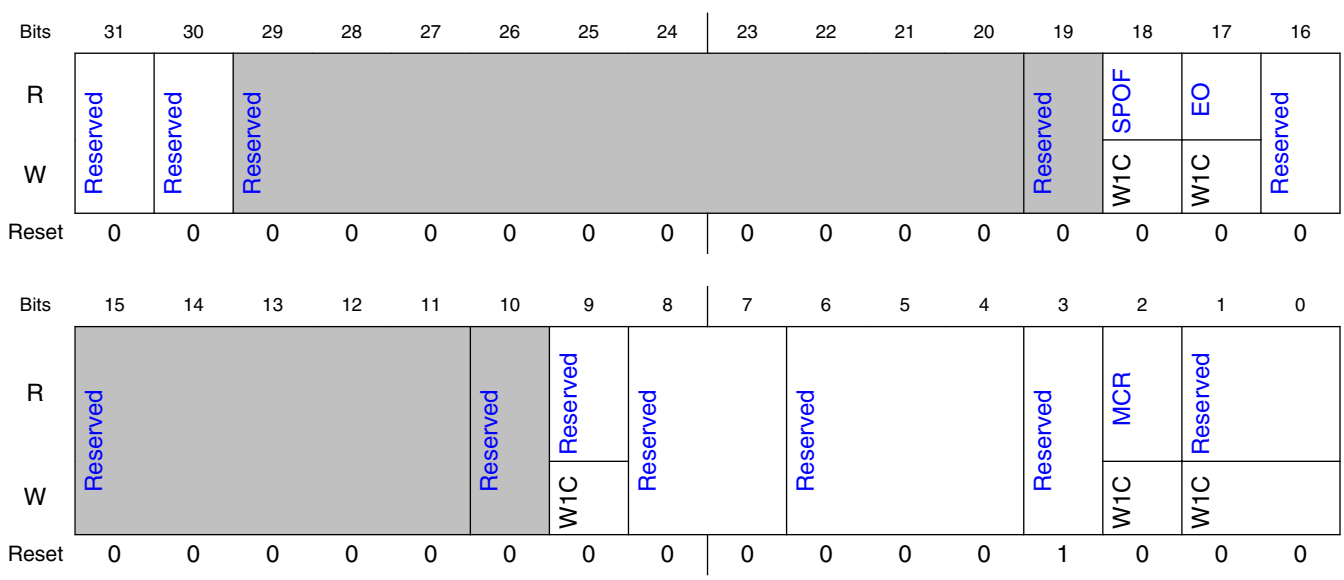
24.6.1.11 SNVS_LP Status Register (LPSR)

The SNVS_LP Status Register reflects the internal state and behavior of the SNVS_LP. This is a privileged write register.

24.6.1.11.1 Offset

Register	Offset
LPSR	4Ch

24.6.1.11.2 Diagram



24.6.1.11.3 Fields

Field	Description
31 —	Reserved
30 —	Reserved
29-20 —	Reserved
19 —	Reserved
18 SPOF	Set Power Off The SPO bit is set when the power button is pressed longer than the configured debounce time. Writing to the SPO bit will clear the set_pwr_off_irq interrupt. 0 - Set Power Off was not detected. 1 - Set Power Off was detected.
17 EO	Emergency Off This bit is set when a power off is requested. 0 - Emergency off was not detected. 1 - Emergency off was detected.
16 —	Reserved
15-11 —	Reserved
10 —	Reserved
9 —	Reserved
8-7 —	Reserved
6-4 —	Reserved
3 —	Reserved
2 MCR	Monotonic Counter Rollover 0 - MC has not reached its maximum value. 1 - MC has reached its maximum value.
1-0 —	Reserved

24.6.1.12 SNVS_LP Secure Monotonic Counter MSB Register (LPSMCMR)

The SNVS_LP Secure Monotonic Counter MSB Register contains the monotonic counter era bits and the most-significant 16 bits of the monotonic counter. The monotonic counter is incremented by one if there is a write command to the LPSMCMR or LPSMCLR register. This is a non-privileged read-only register.

24.6.1.12.1 Offset

Register	Offset
LPSMCMR	5Ch

24.6.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MC_ERA_BITS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MON_COUNTER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.12.3 Fields

Field	Description
31-16	Monotonic Counter Era Bits
MC_ERA_BITS	These bits are inputs to the module and typically connect to fuses. When the Monotonic Counter is in use (i.e. enabled and valid and powered by an uninterrupted power source), and the boot software detects that the Monotonic Counter most-significant 16 Bits and Monotonic Counter LSB Register have been reset (MC_ENV=0), the boot software can take action to ensure that the value in the monotonic counter remains monotonic (i.e. never decreasing). The action is to blow an additional MC_ERA_BITS fuse. Since the MC_ERA_BITS field forms the most-significant field of the monotonic counter, blowing an additional fuse guarantees that the new monotonic counter value is higher than any previous value.
15-0	Monotonic Counter most-significant 16 Bits

Field	Description
MON_COUNTER	**Note that writing to this register does *not* change the value of this field to the value that was written.** **The 48-bit monotonic counter value (consisting of LPSMCMR[MON_COUNTER] prepended to LPSMCLR[MON_COUNTER]) is incremented by one when:** • A write transaction to the LPSMCMR or LPSMCLR register is detected. • The MC_ENV bit is set. • MC_SL and MC_HL bits are not set. This value can be reset only by LP POR.

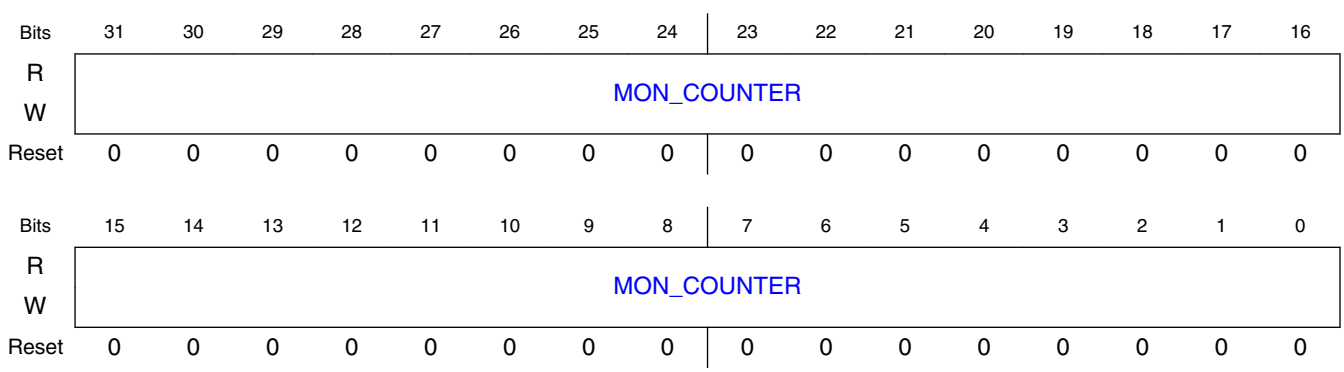
24.6.1.13 SNVS_LP Secure Monotonic Counter LSB Register (LPSMCLR)

The SNVS_LP Secure Monotonic Counter LSB Register contains the 32 least-significant bits of the monotonic counter. The MC is incremented by one if there is a write command to the LPSMCMR or LPSMCLR register. This is a non-privileged read-only register.

24.6.1.13.1 Offset

Register	Offset
LPSMCLR	60h

24.6.1.13.2 Diagram



24.6.1.13.3 Fields

Field	Description
31-0 MON_COUNTER	Monotonic Counter bits **Note that writing to this register does *not* change the value of this field to the value that was written.** **The 48-bit monotonic counter value (consisting of LPSMCMR[MON_COUNTER] prepended to LPSMCLR[MON_COUNTER]) is incremented by one when:** • A write transaction to the LPSMCMR or LPSMCLR register is detected. • The MC_ENV bit is set. • MC_SL and MC_HL bits are not set. This value can be reset only by LP POR.

24.6.1.14 SNVS_LP Digital Low-Voltage Detector Register (LPLVDR)

The SNVS_LP Digital Low-Voltage Detector Register is a 32-bit read/write register that is used for storing the low-voltage detector value, as described in Digital Low-Voltage Detector (LVD). This is a privileged write register.

24.6.1.14.1 Offset

Register	Offset
LPLVDR	64h

24.6.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LVD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LVD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.14.3 Fields

Field	Description
31-0 LVD	Low-Voltage Detector Value

24.6.1.15 SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)

See register SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3).

24.6.1.15.1 Offset

Register	Offset
LPGPR0_legacy_alias	68h

24.6.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.15.3 Fields

Field	Description
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

24.6.1.16 SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0_alias - LPGPR3_alias)

See register SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3).

24.6.1.16.1 Offset

Register	Offset
LPGPR0_alias	90h
LPGPR1_alias	94h
LPGPR2_alias	98h
LPGPR3_alias	9Ch

24.6.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.16.3 Fields

Field	Description
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

24.6.1.17 SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)

The SNVS_LP General Purpose Register is a 128-bit read/write register located in SNVS_LP, which can be used by any application for retaining data during an SoC power-down mode. This is a privileged read/write register. The full GPR register is accessed as 4 32-bit registers located in successive word addresses starting at offset 100h. For backward compatibility with earlier versions of SNVS, LPGPR0..LPGPR3 are aliased at the earlier offset of 90h and LPGPR0 is also aliased at its original offset of 68h. New software should access the GPR register at the preferred offset of 100h. The GPR will be automatically zeroized when an enabled security event occurs, unless GPR zeroization is disabled via the GPR_Z_DIS bit in the LP Control Register.

24.6.1.17.1 Offset

Register	Offset
LPGPR0	100h
LPGPR1	104h
LPGPR2	108h
LPGPR3	10Ch

24.6.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.6.1.17.3 Fields

Field	Description
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

24.6.1.18 SNVS_HP Version ID Register 1 (HPVIDR1)

The SNVS_HP Version ID Register 1 is a non-privileged read-only register that contains the current version of the SNVS. The version consists of a module ID, a major version number, and a minor version number.

24.6.1.18.1 Offset

Register	Offset
HPVIDR1	BF8h

24.6.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR_REV								MINOR_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

24.6.1.18.3 Fields

Field	Description
31-16 IP_ID	SNVS block ID
15-8 MAJOR_REV	SNVS block major version number
7-0 MINOR_REV	SNVS block minor version number

24.6.1.19 SNVS_HP Version ID Register 2 (HPVIDR2)

The SNVS_HP Version ID Register 2 is a non-privileged read-only register that indicates the current version of the SNVS. Version ID register 2 consists of the following fields: integration options, ECO revision, and configuration options.

24.6.1.19.1 Offset

Register	Offset
HPVIDR2	BFCh

24.6.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ERA								Reserved							
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECO_REV								Reserved							
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

24.6.1.19.3 Fields

Field	Description
31-24 IP_ERA	IP Era 00h - Era 1 or 2 03h - Era 3 04h - Era 4 05h - Era 5 06h - Era 6
23-16 —	Reserved
15-8 ECO_REV	SNVS ECO Revision The engineering change order revision number for this release of SNVS.
7-0 —	Reserved

Chapter 25
System Reset Controller (SRC)

25.1 Chip-specific SRC information

Table 25-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

The CPU2 and CPU3 fields are not used in this chip.

25.2 Overview

The System Reset Controller (SRC) is responsible for the generation of all the system reset signals and boot argument latching.

25.2.1 Block Diagram

The figure below shows a high-level block diagram of the SRC.

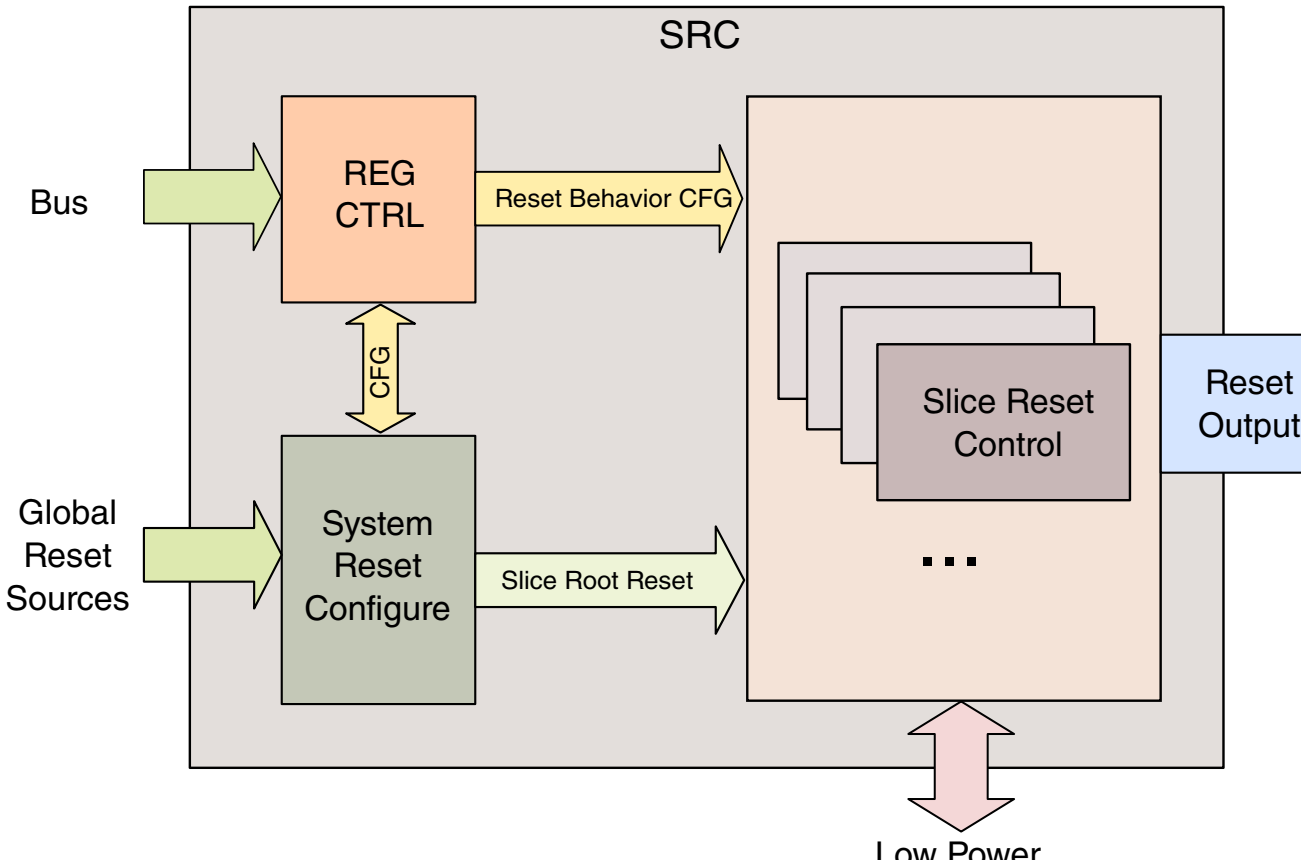


Figure 25-1. SRC High-Level Diagram

25.2.2 Features

The SRC includes the following features:

- Receives and handles the resets from all the reset sources
- Resets the appropriate domains based upon the reset sources and low-power behavior
- Latches the SRC_BOOT_MODE pins and boot configuration signals from the on-chip Fuse or the SRC_BT_CFG pins.

25.3 Functional Description

The reset control determines the source and type of reset, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire chip.

The boot argument latching includes the SRC_BOOT_MODE pin and SRC_BT_CFG pin latch, when the system comes out of reset.

25.3.1 Reset and Power Scheme

The chip presumes the following system reset and power schemes. Please see the power managament chapter for more details on the specific power sources, controls, and domains.

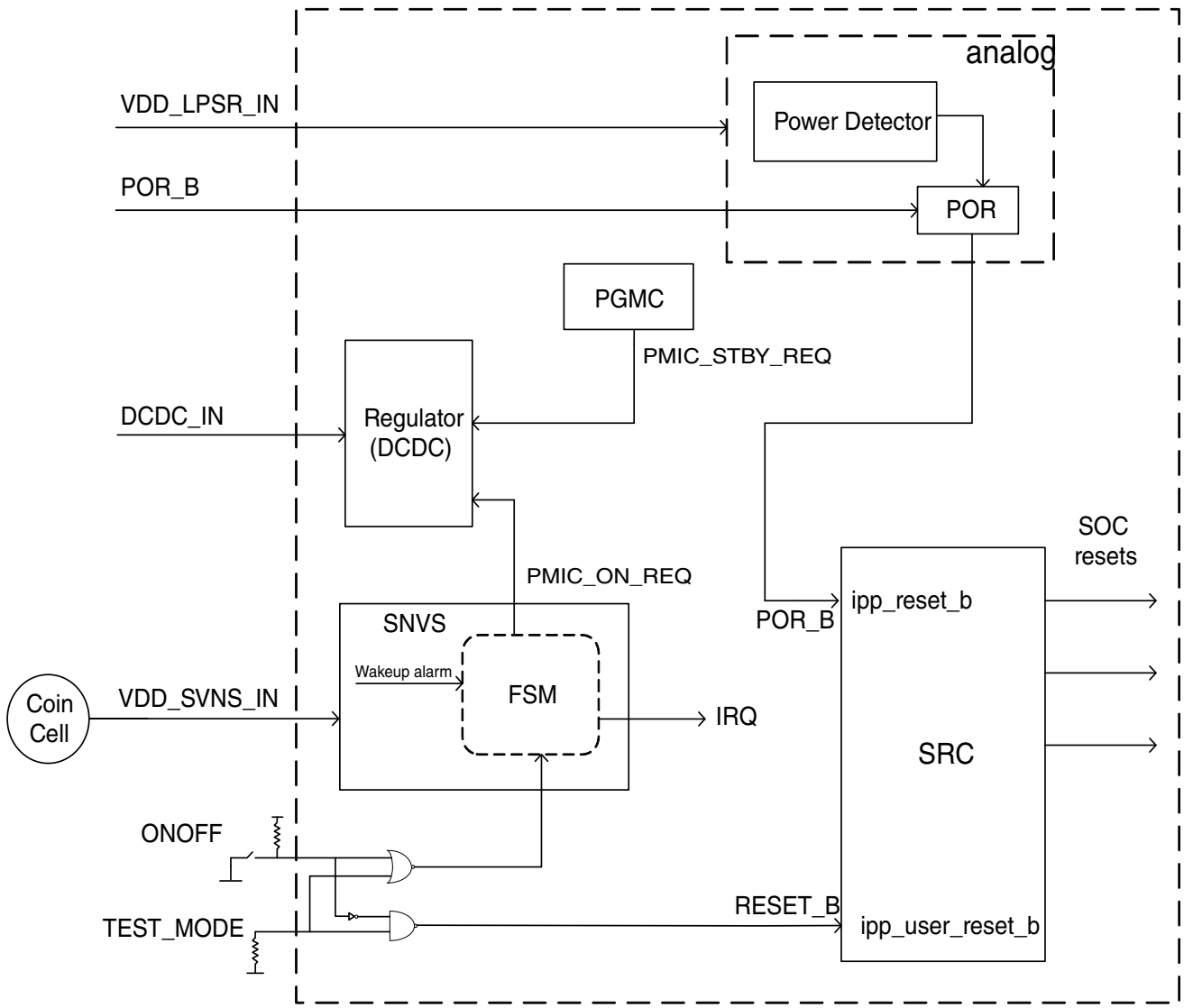


Figure 25-2. Chip Reset with on-chip DCDC

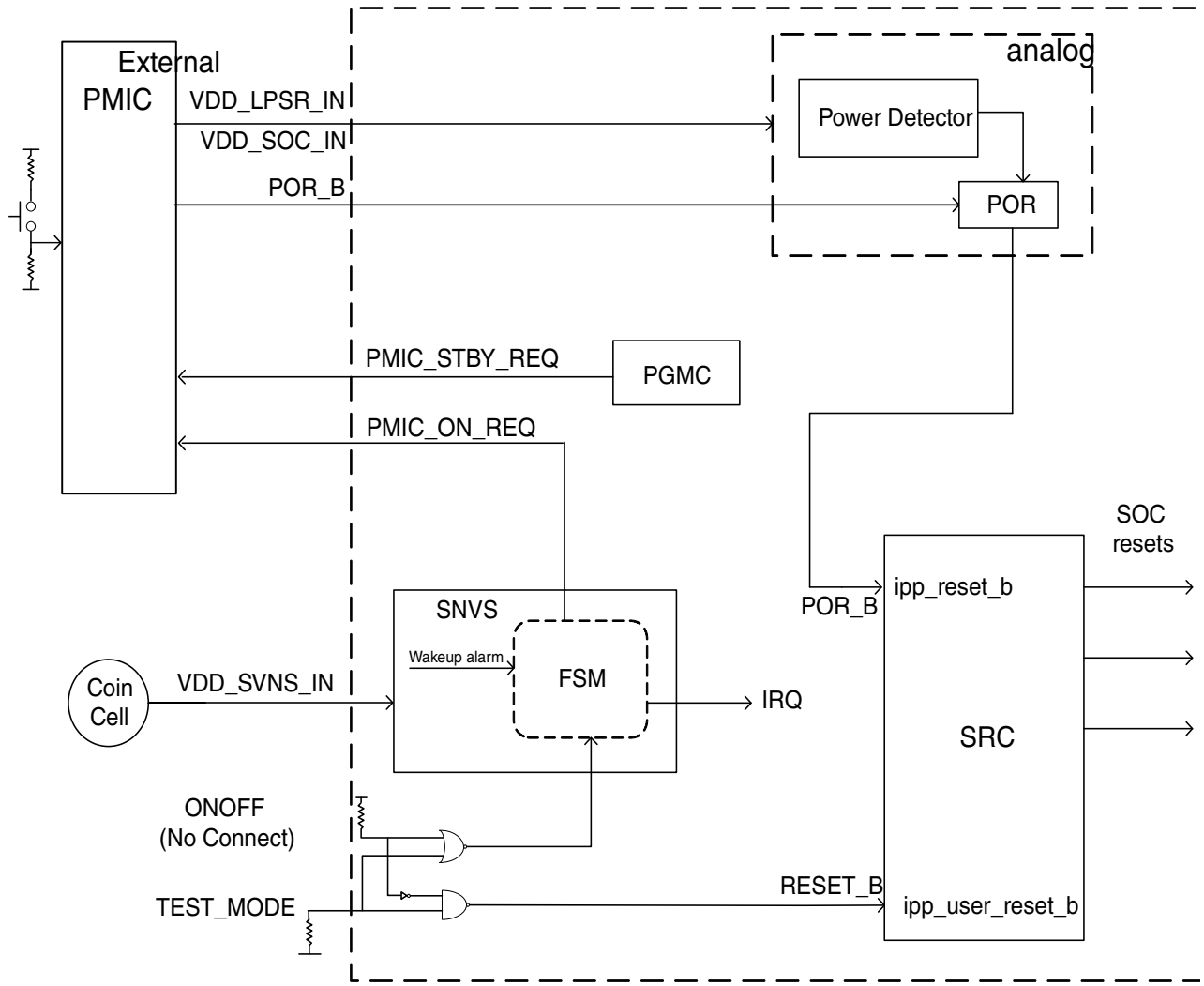


Figure 25-3. Chip Reset with external PMIC

25.3.2 Finite State Machine (FSM)

The FSM Operation of the SRC is illustrated below. The details of the states are explained in the table as well.

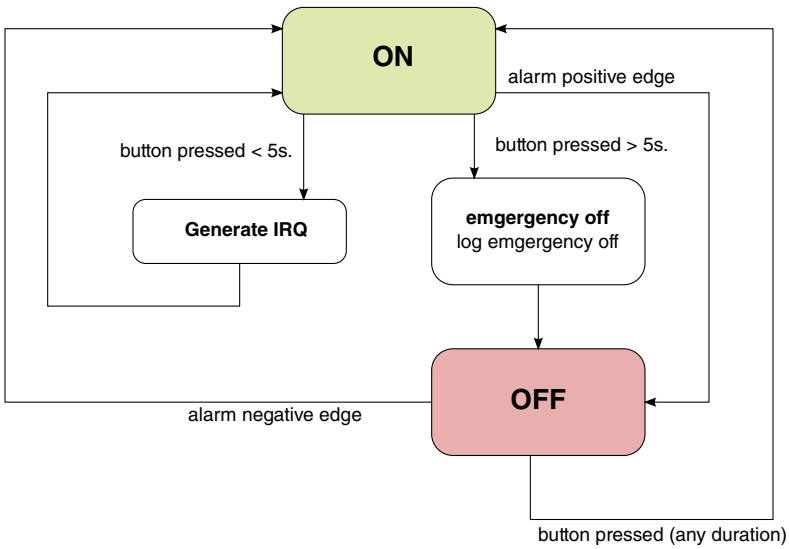


Figure 25-4. FSM Operation

The table below describes the ON-OFF state mechanism:

State	Configuration with external PMIC	Configuration with on-chip DCDC
ON, first time	1. SNVS is powered, PMIC_ON_REQ goes to '1'. 2. When PMIC button is pressed, PMIC powers on.	1. SNVS is powered, PMIC_ON_REQ goes to '1'. 2. 2VDD_LPSR_IN and DCDC_IN are powered. 3. Drive DCDC_PSWITCH to high, 1 ms after DCDC_IN is powered. 4. DCDC regulator is enabled.
Normal ON to OFF, by button (ONOFF pin)	1. Button is pressed (short duration) on the external PMIC. 2. IRQ is sent to SoC from the external PMIC. 3. SoC is programing PMIC for power off when PMIC_STBY_REQ is asserted. 4. In low-power mode, the PMIC_STBY_REQ is asserted according to the PGMC configuration. PMIC gates SoC supplies.	1. ONOFF button is pressed (short duration, less than 5s). 2. IRQ is sent to SoC from FSM. 3. Alarm timer is set by software routine and started. 4. Upon wakeup alarm assertion to '1', PMIC_ON_REQ goes '0'. 5. DCDC regulator goes OFF.
Emergency On to OFF, by button	1. Button is pressed long way on the external PMIC. 2. PMIC is powering off.	1. ONOFF Button is pressed for longer than 5 seconds on the SoC. 2. FSM validates button pressed for 5 seconds. 3. Emergency power off is logged, PMIC_ON_REQ goes '0' 4. DCDC regulator goes OFF.
OFF to ON, by button	1. Button is pressed on the external PMIC. 2. PMIC is powering on.	1. ONOFF button is pressed on the SoC. 2. PMIC_ON_REQ goes '1' 3. DCDC regulator powers ON.

Table continues on the next page...

State	Configuration with external PMIC	Configuration with on-chip DCDC
OFF to ON, by timer alarm	1. Timer alarm in SNVS is programmed by s/w before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, the wakeup alarm goes ‘0’. PMIC_ON_REQ goes ‘1’. 4. PMIC receives assertion of PMIC_ON_REQ and wakes up.	1. Timer alarm in SNVS is programmed by 2. software before SoC goes OFF. 3. SoC enters OFF mode. 4. Upon timer limit, wakeup alarm goes '0'. PMIC_ON_REQ goes '1'. 5. DCDC regulator is enabled by PMIC_ON_REQ= 1.

25.3.3 Reset Control

This section details the reset control of this device. There are four situations, which will cause a reset event. They are:

- Power-on reset (POR)
- Global system reset request
- Software configuration
- Low-power mode transaction

The System Reset Configure, as shown in the block diagram, receives reset requests from all the potential reset sources and generates the Slice Root Reset to the reset slices, named as Slice Reset Control. The Slice Reset Control will help determine the reset behavior according to the slice configuration and the system low-power status after the slice root reset de-assertion.

The independent reset domains in the system have dedicated reset slices, which generate the dedicated reset signal for each reset domain.

The table below shows the Slice Index and the Slice ID of each Slice Reset Control.

Table 25-2. Slice Reset Control List

Slice Index	Slice ID	Description
0	MEGA	Audio, ENET, uSDHC, USB
1	DISPLAY	Display
2	WAKEUP	Memory, Debug, Analog, Timers
3	LPSR	Power, Clocking, Reset, DMA
4	M4CORE	Cortex-M4 Platform
5	M7CORE	Cortex-M7 Platform
6	M4DEBUG	CM4 Debug
7	M7DEBUG	CM7 Debug
8	USBPHY1	USB PHY1
9	USBPHY2	USB PHY2
10	CM4MEM	LMEM

Table continues on the next page...

Table 25-2. Slice Reset Control List (continued)

Slice Index	Slice ID	Description
11	CM7MEM	FlexRAM
12	Reserved	Reserved
13	Reserved	Reserved
14	Reserved	Reserved
15	Reserved	Reserved

25.3.4 Reset behavior of the Power-on Reset

The Power-on Reset (POR) will assert when the chip is powered ON or the POR_B pin is active. The POR will reset all of the chip except the Secure Non-Volatile Storage (SNVS) block.

25.3.5 Reset behavior under the global system reset sources

The Slice Reset Control will determine the reset behavior based on the power mode transaction and the software reset request. The table below shows the global system reset sources.

Table 25-3. Global System Reset Sources

Reset sources	Description
M7_req_rst	Asserts when software writes SYSRESETREQ bit of register APPLICATION_INTERRUPT_CONTROL_REGISTER in CM7
M7_lockup_rst	Asserts when CM7 lock up
M4_req_rst	Asserts when software write SYSRESETREQ bit of register APPLICATION_INTERRUPT_CONTROL_REGISTER in CM4
M4_lockup_rst	Asserts when CM4 lock up
csu_rst	Asserts when the security violation happens in SNVS
ipp_user_reset_b	Asserts when the ONOFF pad is pressed down in test mode
wdog_rst	Asserts when the peripheral WDOG timeouts
jtag_rst_b	Assert when the debugger asserts the reset request in Boundary Scanning mode
jtag_sw_b	Assert when the debugger asserts software reset by JTAG
wdog3_rst	Asserts when the peripheral WDOG3 timeouts
wdog4_rst	Asserts when the peripheral WDOG4 timeouts
tempsense_rst	Assert when the Temper Sensor requests a software reset
overvolt_rst	Assert when the DCDC requests Over Voltage reset

All the reset sources can be disabled by writing the corresponding bit with 2'b11 in the SRC_SRMR. The reset generated by the global reset sources will reset all of the system except for the same part of WDOG which is reset when the POR asserted, and some IPs.

25.3.6 Reset behavior by software configuration

There are ten reset slices. One slice for each dedicated reset domain in the system. Software can write the CTRL_x[SW_RESET] bit of the Slice Control Register, for each reset slice, to request a reset to it's own reset domain.

The CTRL_xx[SW_RESET] bit is a self-clearing bit. The reset is asserted when the CTRL_x[SW_RESET] bit is set to 1'b1, and it will be cleared when the reset sequence is finished.

25.3.7 Reset behavior under the low-power mode transition

When some of the reset domains are powered-down then they need an accurate reset sequence for themselves. The user can define the behavior under different low-power modes for each reset domain, by configuring the Slice Setpoint Config Register and Slice Domain Config Register.

The user can determine if the reset behavior is controlled by CPU low-power mode or by Setpoint status, by configuring the AUTHEN_x[SETPOINT_MODE] bit and AUTHEN_x[DOMAIN_MODE] bit in the Slice Authentication Register.

When in Domain Mode, the reset will be asserted and held active if the system enters a CPU low-power mode, if it's reset domain is configured to be in reset. It will come out of reset when the system enters another CPU low-power mode, if it's reset domain is configured to be not in reset. The timing of the reset behavior under CPU low-power mode transaction is shown in the figure below.

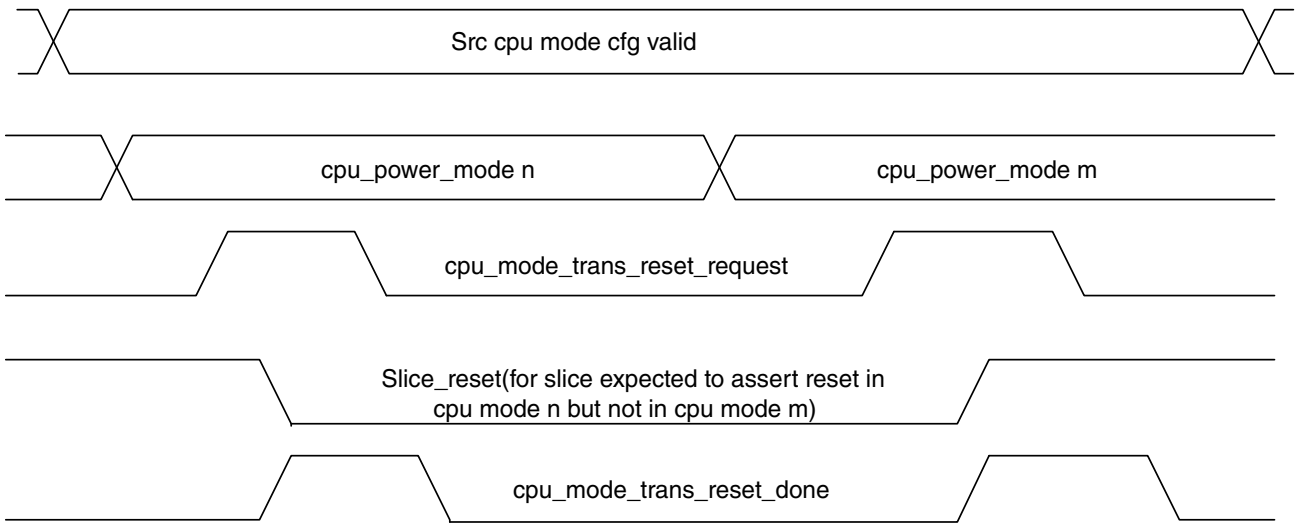


Figure 25-5. The timing of the reset behavior under CPU low-power mode transaction

The timing of the reset behavior under system Setpoint transaction is shown in the figure below.

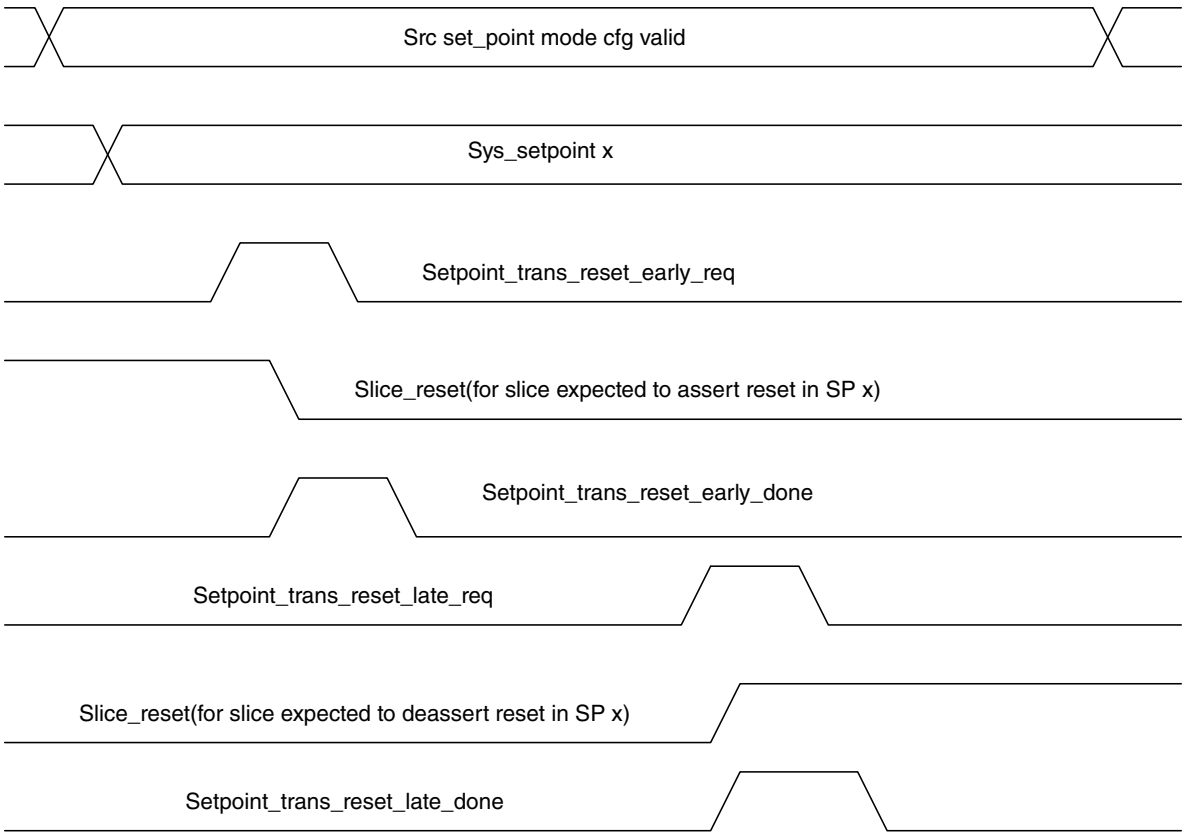


Figure 25-6. The timing of the reset behavior under the system Setpoint transaction

25.3.8 SRC_BOOT_MODE Pin and SRC_BT_CFG Pin Latching

The exact boot sequence is controlled by the values of the SRC_BOOT_MODE pins on this device. The value of the SRC_BOOT_MODE pins will be latched after values of the on-chip Fuses are latched.

After latching, the values of the SRC_BOOT_MODE pins and the BT_FUSE_SEL e-fuse are used to determine other booting options (named as BOOT_CFG) as described in the SBMR1 registers.

The BOOT_CFG bits can be provided to the SRC from either e-fuses named as BOOT_CFG or from SRC_BT_CFG pins. This is determined by the BT_FUSE_SEL fuse setting.

- If BT_FUSE_SEL fuse setting = 1’b1, the BOOT_CFG is provided from BOOT_CFG fuses.
- If BT_FUSE_SEL fuse setting = 1’b0, the BOOT_CFG is taken from SRC_BT_CFG and SRC_BOOT_MODE pins.

Please see the fusemap for more details.

25.3.9 Clocks

The table below describes the relevant SRC clock sources.

Table 25-4. SRC Clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock. This clock is typically the same source as the ipg_clk.

NOTE

Please see the clock controller for clock setting, configuration, and gating information.

25.4 External Signals

This table describes the SRC external signals.

Table 25-5. External Signals

Signal	Description	Direction
SRC_POR_B	Power on reset input pin, active low.	I
SRC_BT_CFGn	Boot configuration pins. The values are sampled during system reset and can be read in the SBMR1 register. See the boot chapter for specific boot configuration settings, and the pin mux chapter for pin assignments.	I
SRC_BOOT_MODE	Boot mode selection pins. The values are sampled during system reset and can be read in the SBMR2 register. See the boot chapter for specific boot configuration settings, and the pin mux chapter for pin assignments.	I
RESET_B (ONOFF)	Reset Signal to ONOFF pin, active low.	O

25.5 Initialization

25.5.1 Power-On Reset and power sequencing

The following conditions must be met for Power-Up:

- VDD_SNVS_IN supply must be turned on before any other power supply, or be connected (shorted) with VDD_LPSR_IN and DCDC_IN supply

- If a coin cell is used to power VDD_SNVS_IN, then ensure that it is connected before any other supply is switched on
- When the internal DCDC is enabled, an external delay circuit is required to delay the DCDC_PSWITCH signal at least 1ms after DCDC_IN is stable
- POR_B should be held low during the entire power up sequence

Similarly, for Power-Down, the following conditions must be met:

- VDD_SNVS_IN supply must be turned off after any other power supply, or be connected (shorted) with VDD_LPSR_IN and DCDC_IN supply
- If a coin cell is used to power VDD_SNVS_IN, then ensure that it is removed after any other supply is switched off

25.6 Memory Map and register definition

This section includes the SRC module memory map and detailed descriptions of all registers.

25.6.1 SRC register descriptions

25.6.1.1 SRC memory map

SRC base address: 40C0_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	SRC Control Register (SCR)	32	RW	0000_0000
4	SRC Reset Mode Register (SRMR)	32	RW	0000_0000
8	SRC Boot Mode Register 1 (SBMR1)	32	RO	0000_0000
C	SRC Boot Mode Register 2 (SBMR2)	32	RO	0000_0000
10	SRC Reset Status Register (SRSR)	32	W1C	0000_0000
14 - 60	SRC General Purpose Register (GPR1 - GPR20)	32	RW	0000_0000
200	Slice Authentication Register (AUTHEN_MEGA)	32	RW	0000_0000
204	Slice Control Register (CTRL_MEGA)	32	RW	0000_0000
208	Slice Setpoint Config Register (SETPOINT_MEGA)	32	RW	0000_0000
20C	Slice Domain Config Register (DOMAIN_MEGA)	32	RW	0000_0000
210	Slice Status Register (STAT_MEGA)	32	W1C	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
220	Slice Authentication Register (AUTHEN_DISPLAY)	32	RW	0000_0000
224	Slice Control Register (CTRL_DISPLAY)	32	RW	0000_0000
228	Slice Setpoint Config Register (SETPOINT_DISPLAY)	32	RW	0000_0000
22C	Slice Domain Config Register (DOMAIN_DISPLAY)	32	RW	0000_0000
230	Slice Status Register (STAT_DISPLAY)	32	W1C	0000_0000
240	Slice Authentication Register (AUTHEN_WAKEUP)	32	RW	0000_0000
244	Slice Control Register (CTRL_WAKEUP)	32	RW	0000_0000
248	Slice Setpoint Config Register (SETPOINT_WAKEUP)	32	RW	0000_0000
24C	Slice Domain Config Register (DOMAIN_WAKEUP)	32	RW	0000_0000
250	Slice Status Register (STAT_WAKEUP)	32	W1C	0000_0000
280	Slice Authentication Register (AUTHEN_M4CORE)	32	RW	0000_0000
284	Slice Control Register (CTRL_M4CORE)	32	RW	0000_0000
288	Slice Setpoint Config Register (SETPOINT_M4CORE)	32	RW	0000_0000
28C	Slice Domain Config Register (DOMAIN_M4CORE)	32	RW	0000_0000
290	Slice Status Register (STAT_M4CORE)	32	W1C	0000_0000
2A0	Slice Authentication Register (AUTHEN_M7CORE)	32	RW	0000_0000
2A4	Slice Control Register (CTRL_M7CORE)	32	RW	0000_0000
2A8	Slice Setpoint Config Register (SETPOINT_M7CORE)	32	RW	0000_0000
2AC	Slice Domain Config Register (DOMAIN_M7CORE)	32	RW	0000_0000
2B0	Slice Status Register (STAT_M7CORE)	32	W1C	0000_0000
2C0	Slice Authentication Register (AUTHEN_M4DEBUG)	32	RW	0000_0000
2C4	Slice Control Register (CTRL_M4DEBUG)	32	RW	0000_0000
2C8	Slice Setpoint Config Register (SETPOINT_M4DEBUG)	32	RW	0000_0000
2CC	Slice Domain Config Register (DOMAIN_M4DEBUG)	32	RW	0000_0000
2D0	Slice Status Register (STAT_M4DEBUG)	32	W1C	0000_0000
2E0	Slice Authentication Register (AUTHEN_M7DEBUG)	32	RW	0000_0000
2E4	Slice Control Register (CTRL_M7DEBUG)	32	RW	0000_0000
2E8	Slice Setpoint Config Register (SETPOINT_M7DEBUG)	32	RW	0000_0000
2EC	Slice Domain Config Register (DOMAIN_M7DEBUG)	32	RW	0000_0000
2F0	Slice Status Register (STAT_M7DEBUG)	32	W1C	0000_0000
300	Slice Authentication Register (AUTHEN_USBPHY1)	32	RW	0000_0000
304	Slice Control Register (CTRL_USBPHY1)	32	RW	0000_0000
308	Slice Setpoint Config Register (SETPOINT_USBPHY1)	32	RW	0000_0000
30C	Slice Domain Config Register (DOMAIN_USBPHY1)	32	RW	0000_0000
310	Slice Status Register (STAT_USBPHY1)	32	W1C	0000_0000
320	Slice Authentication Register (AUTHEN_USBPHY2)	32	RW	0000_0000
324	Slice Control Register (CTRL_USBPHY2)	32	RW	0000_0000
328	Slice Setpoint Config Register (SETPOINT_USBPHY2)	32	RW	0000_0000
32C	Slice Domain Config Register (DOMAIN_USBPHY2)	32	RW	0000_0000
330	Slice Status Register (STAT_USBPHY2)	32	W1C	0000_0000

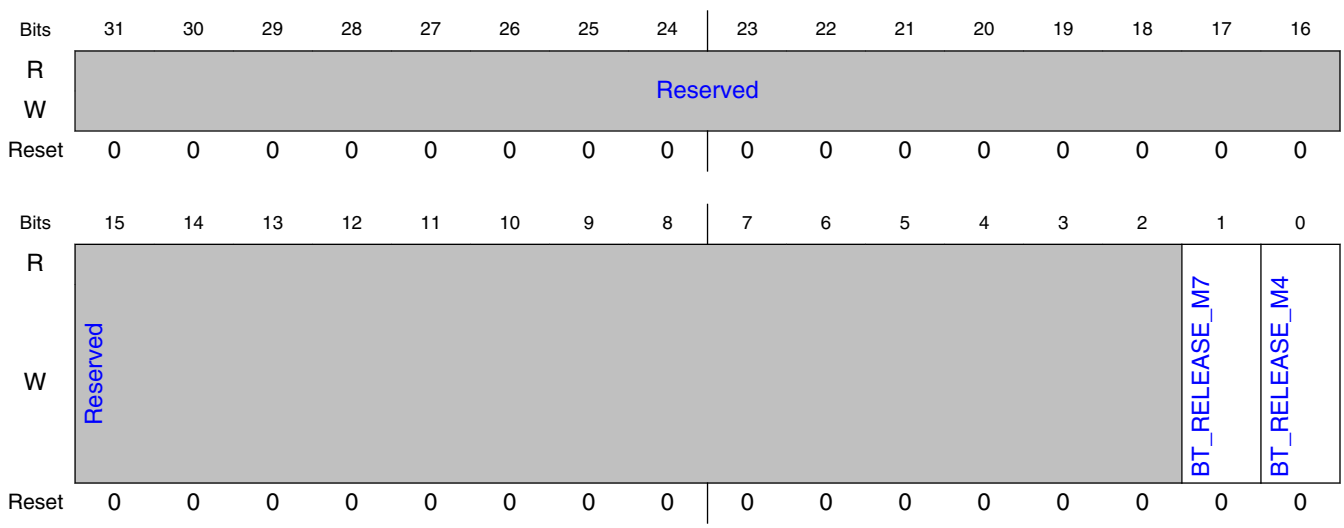
25.6.1.2 SRC Control Register (SCR)

The Reset control register (SCR), contains bits that control operation of the reset controller.

25.6.1.2.1 Offset

Register	Offset
SCR	0h

25.6.1.2.2 Diagram



25.6.1.2.3 Fields

Field	Description
31-2 —	Reserved
1 BT_RELEASE_M7	cm7 core reset will be held until boot core write this bit to 1 to release it. 0 - cm7 core reset is asserted 1 - cm7 core reset is released
0 BT_RELEASE_M4	cm4 core reset will be held until boot core write this bit to 1 to release it. 0 - cm4 core reset is asserted 1 - cm4 core reset is released

25.6.1.3 SRC Reset Mode Register (SRMR)

The Reset Mode Register (SRMR), contains bits that control reset behavior of reset sources. A reset source can reset a single core or entire SoC, depending on its corresponding mask bit value.

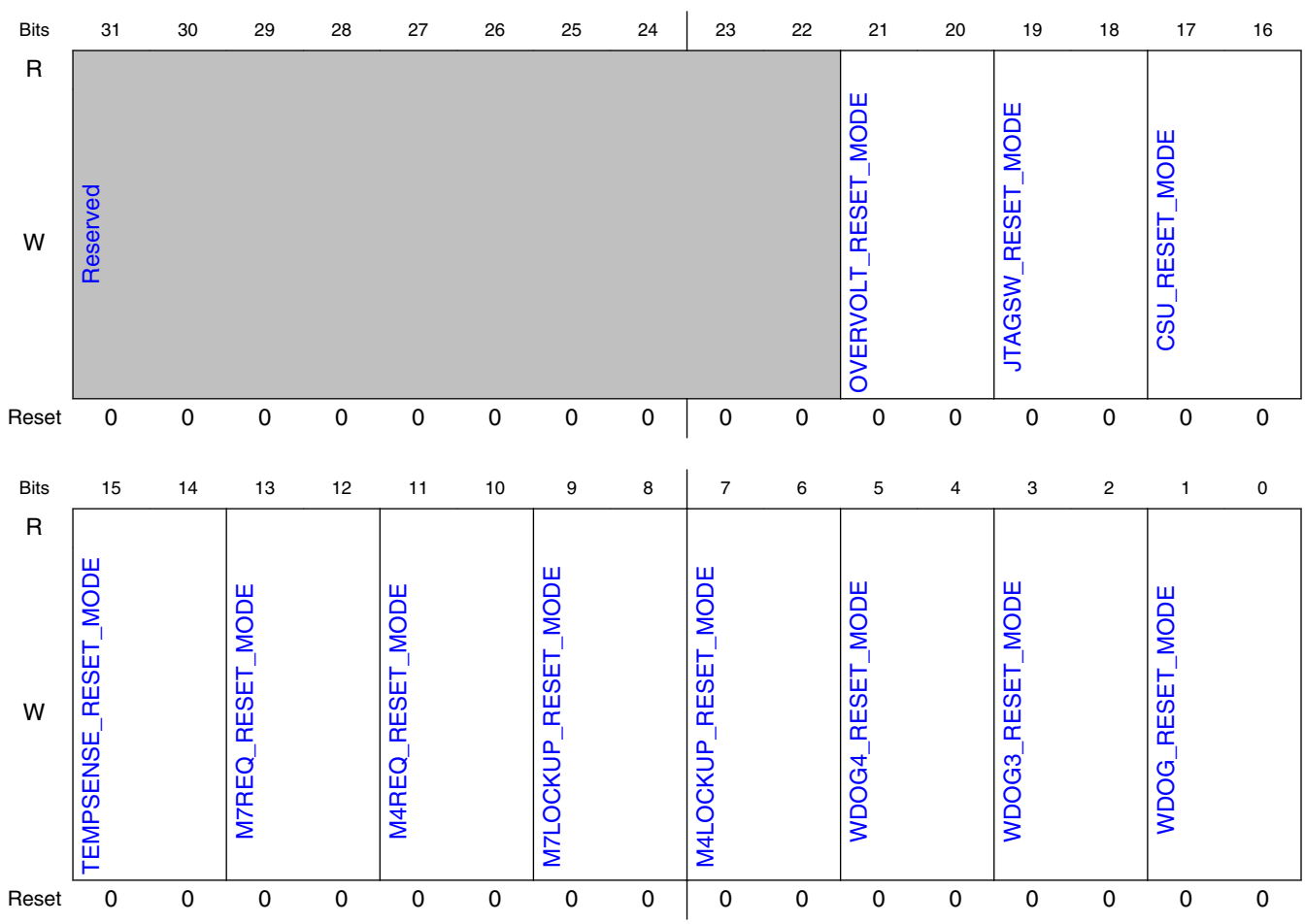
NOTE

Only valid to reset either a single core or entire system. Should be ensured by software.

25.6.1.3.1 Offset

Register	Offset
SRMR	4h

25.6.1.3.2 Diagram



25.6.1.3.3 Fields

Field	Description
31-22 —	Reserved
21-20 OVERVOLT_RESET_MODE	Overvoltage reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
19-18 JTAGSW_RESET_MODE	Jtag SW reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
17-16 CSU_RESET_MODE	CSU reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
15-14 TEMPSENSE_RESET_MODE	Tempsense reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
13-12 M7REQ_RESET_MODE	M7 request reset configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
11-10 M4REQ_RESET_MODE	M4 request reset configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
9-8 M7LOCKUP_RESET_MODE	M7 core lockup reset mode configuration 00 - reset system 01 - reserved

Table continues on the next page...

Field	Description
	10 - reserved 11 - do not reset anything
7-6 M4LOCKUP_RESET_MODE	M4 core lockup reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
5-4 WDOG4_RESET_MODE	Wdog4 reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
3-2 WDOG3_RESET_MODE	Wdog3 reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything
1-0 WDOG_RESET_MODE	Wdog reset mode configuration 00 - reset system 01 - reserved 10 - reserved 11 - do not reset anything

25.6.1.4 SRC Boot Mode Register 1 (SBMR1)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip. The reset value is configuration dependent (depending on boot/fuses/IO pads).

25.6.1.4.1 Offset

Register	Offset
SBMR1	8h

25.6.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BOOT_CFG4								BOOT_CFG3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG2								BOOT_CFG1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.6.1.4.3 Fields

Field	Description
31-24 BOOT_CFG4	Please see fusemap.
23-16 BOOT_CFG3	Please see fusemap.
15-8 BOOT_CFG2	Please see fusemap.
7-0 BOOT_CFG1	Please see fusemap.

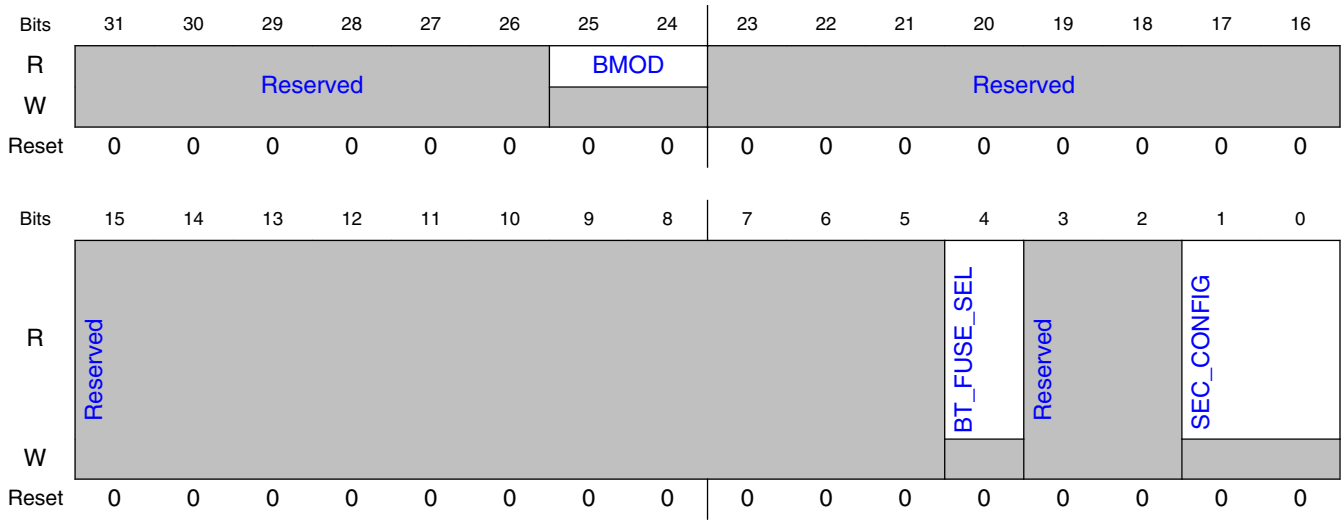
25.6.1.5 SRC Boot Mode Register 2 (SBMR2)

The Boot Mode register (SBMR), contains bits that reflect the status of Boot Mode Pins of the chip. The default values for those bits depends on the values of pins/fuses during reset sequence, hence the question mark on their default value.

25.6.1.5.1 Offset

Register	Offset
SBMR2	Ch

25.6.1.5.2 Diagram



25.6.1.5.3 Fields

Field	Description
31-26 —	Reserved
25-24 BMOD	BMOD[1:0] shows the latched state of the BOOT_MODE1 and BOOT_MODE0 signals on the rising edge of POR_B. See the Boot mode pin settings section of System Boot.
23-16 —	Reserved
15-5 —	Reserved
4 BT_FUSE_SEL	BT_FUSE_SEL shows the state of the BT_FUSE_SEL fuse. See fusemap for additional information on this fuse.1 Reset is a result of the mentioned case.
3-2 —	Reserved
1-0 SEC_CONFIG	SECONFIG[1] shows the state of the SECONFIG[1] fuse. See Fusemap for additional information on this fuse. SECONFIG[0] shows the state of the SECONFIG[0] fuse. This fuse is shown as reserved in fusemap (address 0x440[1]) because it does not have a user-relevant function.

25.6.1.6 SRC Reset Status Register (SRSR)

The SRSR is a write to one clear register which records the source of the reset events for the chip. The SRC reset status register will capture all the reset sources that have occurred. This register is reset on ipp_reset_b. This is a read-write register.

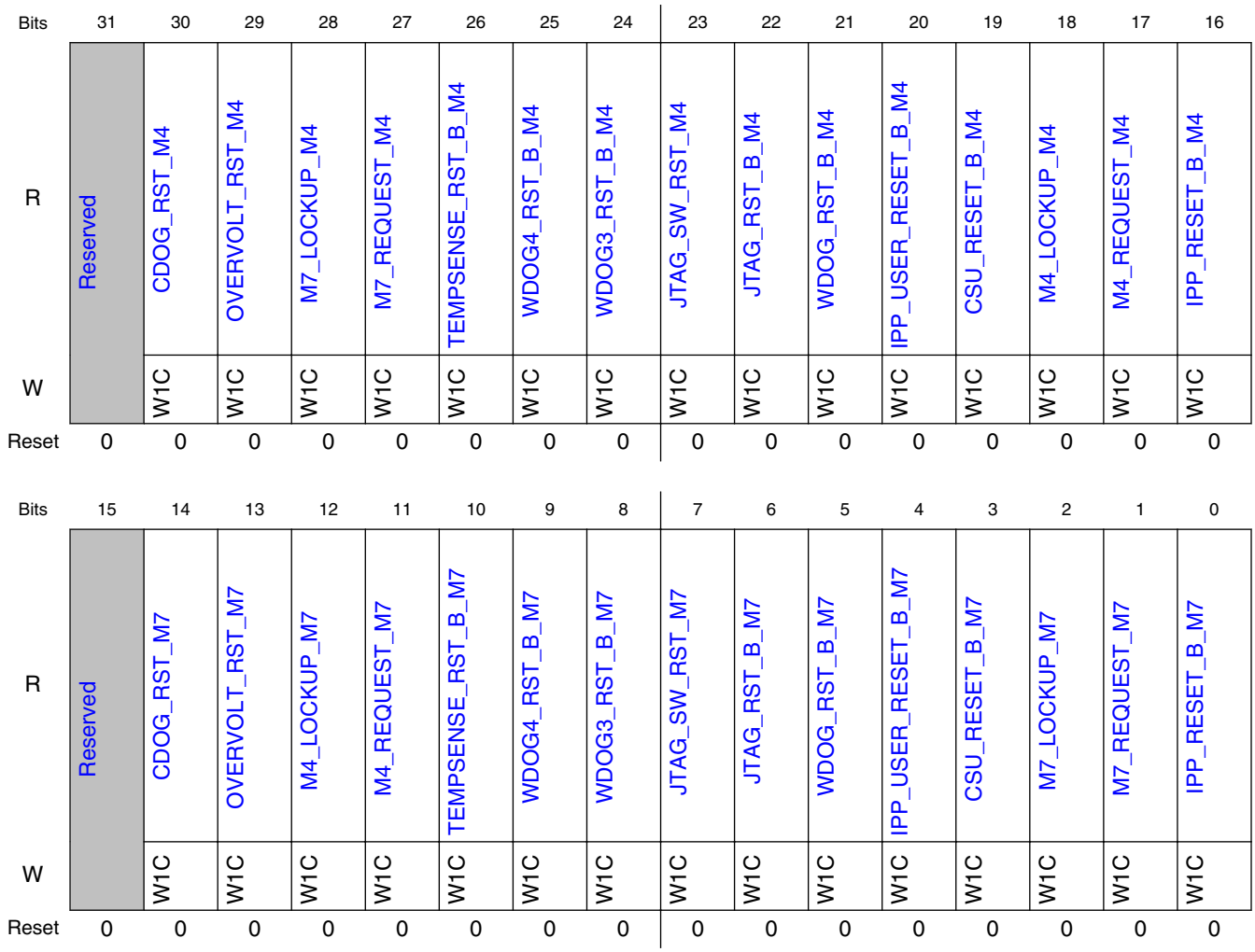
Writing zero does not have any effect. Writing one will clear the corresponding bit. When the system comes out of reset, this register will have bits set corresponding to all the reset sources that occurred during system reset. Software has to take care to clear this register by writing one after every reset that occurs so that the register will contain the information of recently occurred reset.

Since M4 and M7 core might be reset by different source, reset status register must provide each core with dedicated bits. Bits [15:0] are assigned to M7 core and [31:16] are assigned to M4 core.

25.6.1.6.1 Offset

Register	Offset
SRSR	10h

25.6.1.6.2 Diagram



25.6.1.6.3 Fields

Field	Description
31 —	Reserved
30 CDOG_RST_M4	Indicates a reset has been caused by CDOG reset. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
29 OVERVOLT_RST_M4	Indicates a reset has been caused by power suppy voltage over the highest permitted level. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
28	Indicates a reset has been caused by M7 CPU lockup or software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register of the ARM core. Software needs to set some GPR bit

Table continues on the next page...

Field	Description
M7_LOCKUP_M4	before writing SYSRESETREQ bit and use the GPR bit to distinguish if the reset is caused by SYSRESETREQ or CPU lockup. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
27 M7_REQUEST_M4	Indicates whether reset was the result of m7 reset request. 0 - Reset is not a result of m7 reset request. 1 - Reset is a result of m7 reset request.
26 TEMPSENSE_RST_B_M4	Temper Sensor software reset. Indicates whether the reset was the result of software reset from on-chip Temperature Sensor. 0 - Reset is not a result of software reset from Temperature Sensor. 1 - Reset is a result of software reset from Temperature Sensor.
25 WDOG4_RST_B_M4	IC Watchdog4 Time-out reset. Indicates whether the reset was the result of the watchdog4 time-out event. 0 - Reset is not a result of the watchdog4 time-out event. 1 - Reset is a result of the watchdog4 time-out event.
24 WDOG3_RST_B_M4	IC Watchdog3 Time-out reset. Indicates whether the reset was the result of the watchdog3 time-out event. 0 - Reset is not a result of the watchdog3 time-out event. 1 - Reset is a result of the watchdog3 time-out event.
23 JTAG_SW_RST_M4	JTAG software reset. Indicates whether the reset was the result of software reset from JTAG. 0 - Reset is not a result of software reset from JTAG. 1 - Reset is a result of software reset from JTAG.
22 JTAG_RST_B_M4	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset from JTAG. 0 - Reset is not a result of HIGH-Z reset from JTAG. 1 - Reset is a result of HIGH-Z reset from JTAG.
21 WDOG_RST_B_M4	IC Watchdog Time-out reset. Indicates whether the reset was the result of the watchdog time-out event. 0 - Reset is not a result of the watchdog time-out event. 1 - Reset is a result of the watchdog time-out event.
20 IPP_USER_RESET_B_M4	Indicates whether the reset was the result of the ipp_user_reset_b qualified reset. 0 - Reset is not a result of the ipp_user_reset_b qualified as COLD reset event. 1 - Reset is a result of the ipp_user_reset_b qualified as COLD reset event.
19 CSU_RESET_B_M4	Indicates whether the reset was the result of the csu_reset_b input. 0 - Reset is not a result of the csu_reset_b event. 1 - Reset is a result of the csu_reset_b event.
18 M4_LOCKUP_M4	Indicates a reset has been caused by M4 CPU lockup or software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register of the ARM core. Software needs to set some GPR bit before writing SYSRESETREQ bit and use the GPR bit to distinguish if the reset is caused by SYSRESETREQ or CPU lockup. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
17	Indicates whether reset was the result of m4 reset request

Table continues on the next page...

Field	Description
M4_REQUEST_M4	0 - Reset is not a result of m4 reset request. 1 - Reset is a result of m4 reset request.
16 IPP_RESET_B_M4	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 - Reset is not a result of ipp_reset_b pin. 1 - Reset is a result of ipp_reset_b pin.
15 —	Reserved
14 CDOG_RST_M7	Indicates a reset has been caused by CDOG reset. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
13 OVERVOLT_RST_M7	Indicates a reset has been caused by power suppy voltage over the highest permitted level. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
12 M4_LOCKUP_M7	Indicates a reset has been caused by M4 CPU lockup or software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register of the ARM core. Software needs to set some GPR bit before writing SYSRESETREQ bit and use the GPR bit to distinguish if the reset is caused by SYSRESETREQ or CPU lockup. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
11 M4_REQUEST_M7	Indicates whether reset was the result of m4 reset request. 0 - Reset is not a result of m4 reset request. 1 - Reset is a result of m4 reset request.
10 TEMPSENSE_RST_B_M7	Temper Sensor software reset. Indicates whether the reset was the result of software reset from on-chip Temperature Sensor. 0 - Reset is not a result of software reset from Temperature Sensor. 1 - Reset is a result of software reset from Temperature Sensor.
9 WDOG4_RST_B_M7	IC Watchdog4 Time-out reset. Indicates whether the reset was the result of the watchdog4 time-out event. 0 - Reset is not a result of the watchdog4 time-out event. 1 - Reset is a result of the watchdog4 time-out event.
8 WDOG3_RST_B_M7	IC Watchdog3 Time-out reset. Indicates whether the reset was the result of the watchdog3 time-out event. 0 - Reset is not a result of the watchdog3 time-out event. 1 - Reset is a result of the watchdog3 time-out event.
7 JTAG_SW_RST_M7	JTAG software reset. Indicates whether the reset was the result of software reset from JTAG. 0 - Reset is not a result of software reset from JTAG. 1 - Reset is a result of software reset from JTAG.
6 JTAG_RST_B_M7	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset from JTAG. 0 - Reset is not a result of HIGH-Z reset from JTAG. 1 - Reset is a result of HIGH-Z reset from JTAG.
5	IC Watchdog Time-out reset. Indicates whether the reset was the result of the watchdog time-out event.

Table continues on the next page...

Field	Description
WDOG_RST_B_M7	0 - Reset is not a result of the watchdog time-out event. 1 - Reset is a result of the watchdog time-out event.
4 IPP_USER_RESET_B_M7	Indicates whether the reset was the result of the ipp_user_reset_b qualified reset. 0 - Reset is not a result of the ipp_user_reset_b qualified as COLD reset event. 1 - Reset is a result of the ipp_user_reset_b qualified as COLD reset event.
3 CSU_RESET_B_M7	Indicates whether the reset was the result of the csu_reset_b input. 0 - Reset is not a result of the csu_reset_b event. 1 - Reset is a result of the csu_reset_b event.
2 M7_LOCKUP_M7	Indicates a reset has been caused by M7 CPU lockup or software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register of the ARM core. Software needs to set some GPR bit before writing SYSRESETREQ bit and use the GPR bit to distinguish if the reset is caused by SYSRESETREQ or CPU lockup. 0 - Reset is not a result of the mentioned case. 1 - Reset is a result of the mentioned case.
1 M7_REQUEST_M7	Indicates whether reset was the result of m7 reset request 0 - Reset is not a result of m7 reset request. 1 - Reset is a result of m7 reset request.
0 IPP_RESET_B_M7	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 - Reset is not a result of ipp_reset_b pin. 1 - Reset is a result of ipp_reset_b pin.

25.6.1.7 SRC General Purpose Register (GPR1 - GPR20)

NOTE

The following GPRs are used by ROM code and should not be used by application software: 0,1,2,3,4,9

25.6.1.7.1 Offset

For a = 1 to 20:

Register	Offset
GPRa	10h + (a × 4h)

25.6.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.6.1.7.3 Fields

Field	Description
31-0 GPR	General Purpose Register. Please verify any dedicated ROM usage before programming.

25.6.1.8 Slice Authentication Register (AUTHEN_MEGA - AUTHEN_USBPHY2)

The list of reset slice instances are reflected in the table of register array offsets. For associated reset slice descriptions, please see Reset Control.

NOTE

Each reset slice has dedicated register set to control domain access and reset behavior during Setpoint/CPU mode change. Software reset is also implemented for each slice.

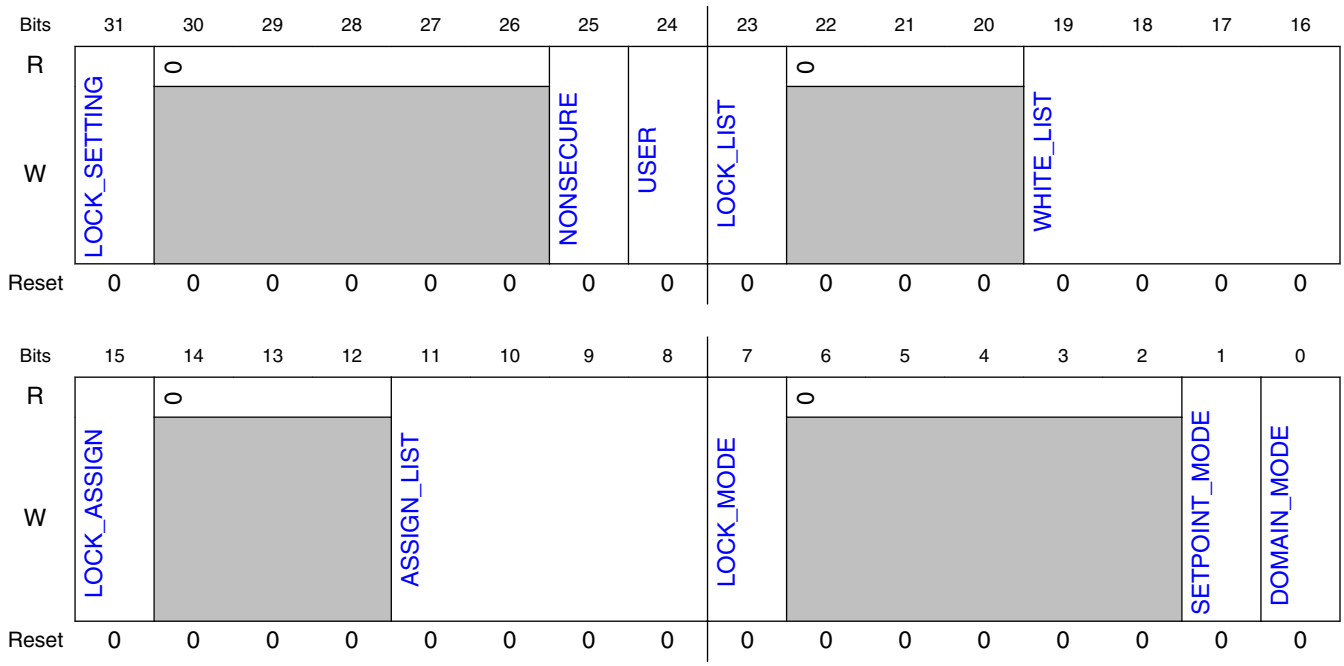
25.6.1.8.1 Offset

Register	Offset
AUTHEN_MEGA	200h
AUTHEN_DISPLAY	220h
AUTHEN_WAKEUP	240h
AUTHEN_M4CORE	280h
AUTHEN_M7CORE	2A0h

Table continues on the next page...

Register	Offset
AUTHEN_M4DEBUG	2C0h
AUTHEN_M7DEBUG	2E0h
AUTHEN_USBPHY1	300h
AUTHEN_USBPHY2	320h

25.6.1.8.2 Diagram



25.6.1.8.3 Fields

Field	Description
31 LOCK_SETTING	Lock NONSECURE and USER this field is write once and can lock value of USER and NONSECURE field
30-26 —	Reserved
25 NONSECURE	Allow non-secure mode access Allow both secure and non-secure mode to access Setpoint control registers
24 USER	Allow user mode access Allow both privilege and user mode to access Setpoint control registers
23 LOCK_LIST	White list lock this field is write once and can lock value of WHITE_LIST field

Table continues on the next page...

Field	Description
22-20 —	Reserved
19-16 WHITE_LIST	Domain ID white list When this bitfield set to 1, the corresponding domain ID can write Slice Authentication Rregister bit[31:8] and Slice Control Register. Else can only read.
15 LOCK_ASSIGN	Assign list lock this field is write once and can lock value of ASSIGN_LIST field
14-12 —	Reserved
11-8 ASSIGN_LIST	when this bitfield set to 1, reset of slice would be subject to corresponding core status transition
7 LOCK_MODE	Domain/Setpoint mode lock this field is write once and can lock value of SETPOINT_MODE and DOMAIN_MODE fields
6-2 —	Reserved
1 SETPOINT_MODE	Control whether reset slice is in Setpoint mode 0 - slice hardware reset will NOT be triggered by Setpoint transition 1 - slice hardware reset will be triggered by Setpoint transition. Do not set this bit and DOMAIN_MODE at the same time.
0 DOMAIN_MODE	Control whether reset slice is in domain mode 0 - slice hardware reset will NOT be triggered by CPU power mode transition 1 - slice hardware reset will be triggered by CPU power mode transition. Do not set this bit and SETPOINT_MODE at the same time.

25.6.1.9 Slice Control Register (CTRL_MEGA - CTRL_USBPHY2)

The list of reset slice instances are reflected in the table of register array offsets. For associated reset slice descriptions, please see Reset Control.

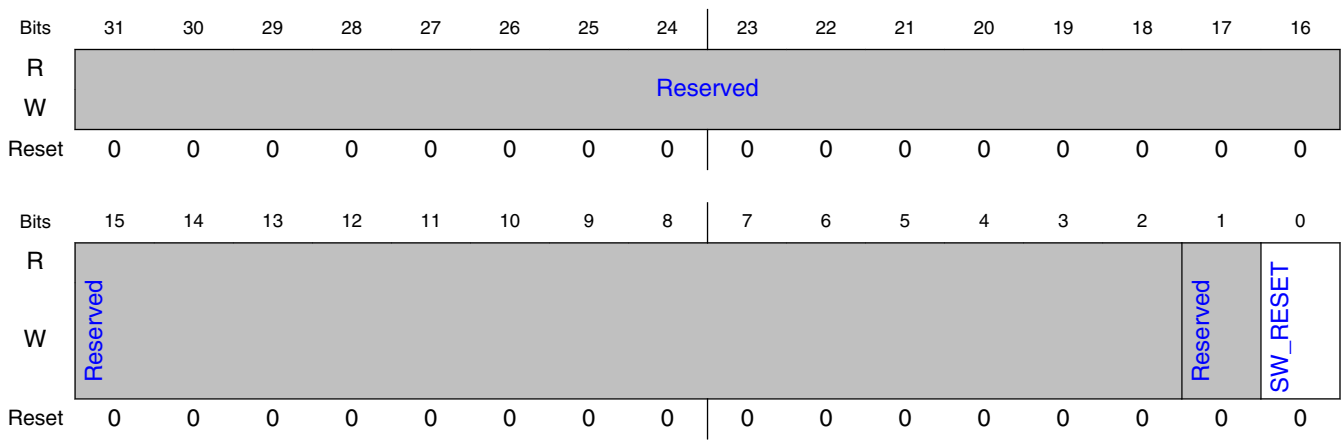
NOTE

Each reset slice has dedicated register set to control domain access and reset behavior during Setpoint/CPU mode change. Software reset is also implemented for each slice.

25.6.1.9.1 Offset

Register	Offset
CTRL_MEGA	204h
CTRL_DISPLAY	224h
CTRL_WAKEUP	244h
CTRL_M4CORE	284h
CTRL_M7CORE	2A4h
CTRL_M4DEBUG	2C4h
CTRL_M7DEBUG	2E4h
CTRL_USBPHY1	304h
CTRL_USBPHY2	324h

25.6.1.9.2 Diagram



25.6.1.9.3 Fields

Field	Description
31-2 —	Reserved
1 —	Reserved
0 SW_RESET	This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 - do not assert slice software reset 1 - assert slice software reset

25.6.1.10 Slice Setpoint Config Register (SETPOINT_MEGA - SETPOINT_USBPHY2)

The list of reset slice instances are reflected in the table of register array offsets. For associated reset slice descriptions, please see Reset Control.

NOTE

Each reset slice has dedicated register set to control domain access and reset behavior during Setpoint/CPU mode change. Software reset is also implemented for each slice.

25.6.1.10.1 Offset

Register	Offset
SETPOINT_MEGA	208h
SETPOINT_DISPLAY	228h
SETPOINT_WAKEUP	248h
SETPOINT_M4CORE	288h
SETPOINT_M7CORE	2A8h
SETPOINT_M4DEBUG	2C8h
SETPOINT_M7DEBUG	2E8h
SETPOINT_USBPHY1	308h
SETPOINT_USBPHY2	328h

25.6.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SETPOINT15	SETPOINT14	SETPOINT13	SETPOINT12	SETPOINT11	SETPOINT10	SETPOINT9	SETPOINT8	SETPOINT7	SETPOINT6	SETPOINT5	SETPOINT4	SETPOINT3	SETPOINT2	SETPOINT1	SETPOINT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.6.1.10.3 Fields

Field	Description
31-16 —	Reserved
15-0 SETPOINTn	SETPOINTn Configuration setting for Setpointn 0 - Slice reset will be de-asserted when system in Setpoint n 1 - Slice reset will be asserted when system in Setpoint n

25.6.1.11 Slice Domain Config Register (DOMAIN_MEGA - DOMAIN_USBPHY2)

The list of reset slice instances are reflected in the table of register array offsets. For associated reset slice descriptions, please see Reset Control.

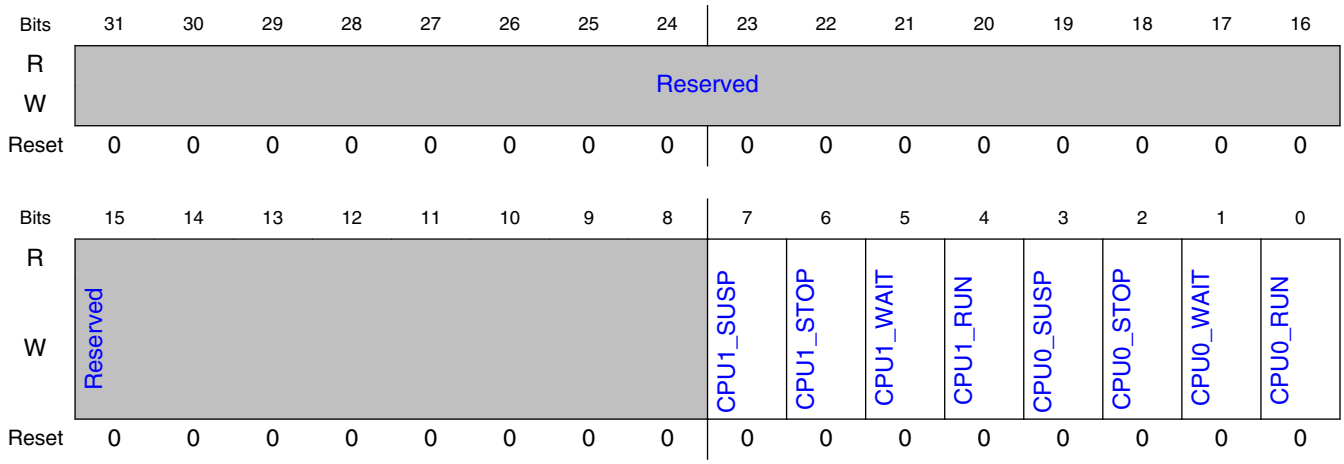
NOTE

Each reset slice has dedicated register set to control domain access and reset behavior during Setpoint/CPU mode change. Software reset is also implemented for each slice.

25.6.1.11.1 Offset

Register	Offset
DOMAIN_MEGA	20Ch
DOMAIN_DISPLAY	22Ch
DOMAIN_WAKEUP	24Ch
DOMAIN_M4CORE	28Ch
DOMAIN_M7CORE	2ACh
DOMAIN_M4DEBUG	2CCh
DOMAIN_M7DEBUG	2ECh
DOMAIN_USBPHY1	30Ch
DOMAIN_USBPHY2	32Ch

25.6.1.11.2 Diagram



25.6.1.11.3 Fields

Field	Description
31-8 —	Reserved
7 CPU1_SUSP	CPU mode setting for SUSPEND 0 - Slice reset will be de-asserted when CPU1 in SUSPEND mode 1 - Slice reset will be asserted when CPU1 in SUSPEND mode
6 CPU1_STOP	CPU mode setting for STOP 0 - Slice reset will be de-asserted when CPU1 in STOP mode 1 - Slice reset will be asserted when CPU1 in STOP mode
5 CPU1_WAIT	CPU mode setting for WAIT 0 - Slice reset will be de-asserted when CPU1 in WAIT mode 1 - Slice reset will be asserted when CPU1 in WAIT mode
4 CPU1_RUN	CPU mode setting for RUN 0 - Slice reset will be de-asserted when CPU1 in RUN mode 1 - Slice reset will be asserted when CPU1 in RUN mode
3 CPU0_SUSP	CPU mode setting for SUSPEND 0 - Slice reset will be de-asserted when CPU0 in SUSPEND mode 1 - Slice reset will be asserted when CPU0 in SUSPEND mode
2 CPU0_STOP	CPU mode setting for STOP 0 - Slice reset will be de-asserted when CPU0 in STOP mode 1 - Slice reset will be asserted when CPU0 in STOP mode
1 CPU0_WAIT	CPU mode setting for WAIT 0 - Slice reset will be de-asserted when CPU0 in WAIT mode 1 - Slice reset will be asserted when CPU0 in WAIT mode

Table continues on the next page...

Field	Description
0 CPU0_RUN	CPU mode setting for RUN 0 - Slice reset will be de-asserted when CPU0 in RUN mode 1 - Slice reset will be asserted when CPU0 in RUN mode

25.6.1.12 Slice Status Register (STAT_MEGA - STAT_USBPHY2)

The list of reset slice instances are reflected in the table of register array offsets. For associated reset slice descriptions, please see Reset Control.

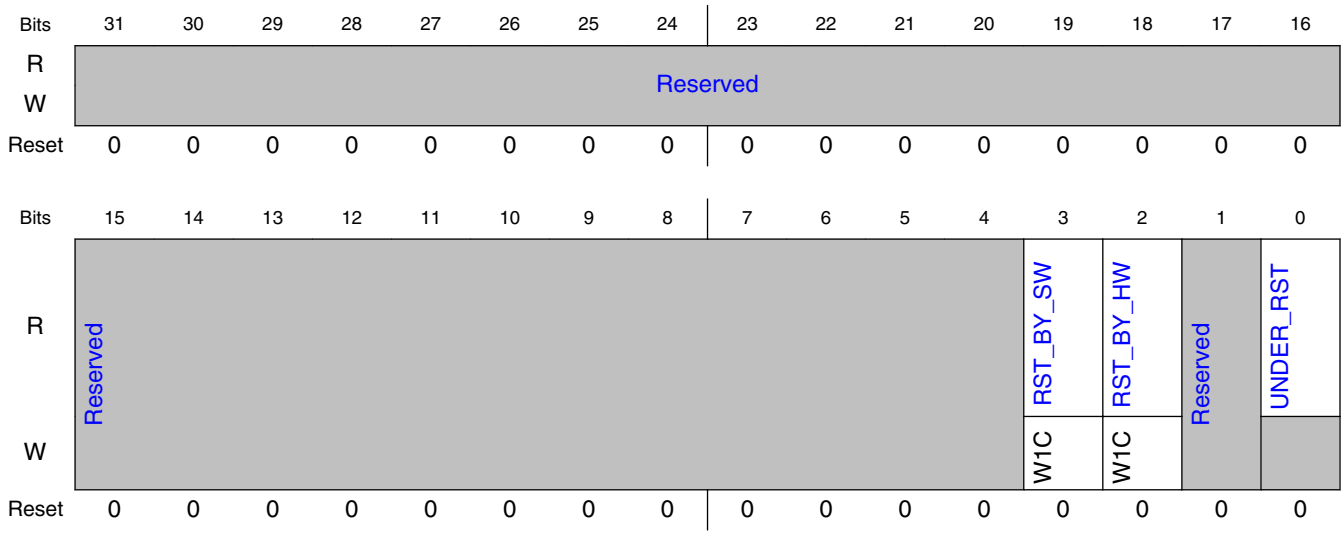
NOTE

Each reset slice has dedicated register set to control domain access and reset behavior during Setpoint/CPU mode change. Software reset is also implemented for each slice.

25.6.1.12.1 Offset

Register	Offset
STAT_MEGA	210h
STAT_DISPLAY	230h
STAT_WAKEUP	250h
STAT_M4CORE	290h
STAT_M7CORE	2B0h
STAT_M4DEBUG	2D0h
STAT_M7DEBUG	2F0h
STAT_USBPHY1	310h
STAT_USBPHY2	330h

25.6.1.12.2 Diagram



25.6.1.12.3 Fields

Field	Description
31-4 —	Reserved
3 RST_BY_SW	This bit indicate if the reset is caused by setting SW_RESET bit. 0 - the reset is not caused by software setting 1 - the reset is caused by software setting
2 RST_BY_HW	This bit indicate if the reset is caused by the power mode transfer. 0 - the reset is not caused by the power mode transfer 1 - the reset is caused by the power mode transfer
1 —	Reserved
0 UNDER_RST	This is a Read Only bit. It indicate if the reset is in process. 0 - the reset is finished 1 - the reset is in process

Chapter 26
Fusemap

26.1 Overview

This section details the various fuses and their associated configuration settings. There is one main fuse array with an associated supplementary array.

This chapter also describes various modes and the selection of the required boot devices.

NOTE

The fuses marked as “Reserved” are reserved for NXP internal (and future) use only, or are security related. Customers must not attempt to burn these, because the IC behavior may be unpredictable. The reserved fuses can be read as either '0' or '1'. Security-related fuses are described in the Security Reference Manual (SRM) for this device.

26.1.1 Bank redundancy vs Error correction code (ECC)

There are two mechanisms used by the fuse block to ensure that its values are correct. Each of these is performed on a word by word basis; however, the selected option is hardwired on a bank by bank basis.

Fuses can generally only be set from 0 to 1 and the most common failure is a 1 appearing as a 0. The simplest mechanism to verify these failures is redundancy. Redundancy is when two bits are OR'd together so that if either is set, then the output value is a 1.

In this case only 16 of the 32 bits are available (bits 15:0 are automatically copied to 31:16) and checked on read. The ECC bits 38:32 are not used in this case. The value here is that each pair of bits can be programmed at different times. This allows for multiple programming operations on a word, so bits can be configured at different times.

NOTE

In shadow registers, the inverted version of [15:0] will show in [31:16]. In the fuse array, [31:16] is the same as [15:0].

The more complex ECC mechanism is used when higher numbers of bits must be stored and configured at the same time in a group of 32. In this case, when written, the ECC must be calculated to allow an ECC correction, when read any one bit of failure will be corrected, 2 bits will be detected (and an error flag asserted) by the controller.

The majority of the main arrays use ECC, however, the first 128 words use redundancy to accommodate the monotonic array.

NOTE

The monotonic array requires redundancy, not ECC, as it's programmed individually

NOTE

As the ECC is generated for the whole word, the whole word must be setup at one time and individual bits cannot be modified independently.

26.1.2 Lock bits

There are two kinds of lock bits, write lock and read lock.

The write lock bits are used by the controller to control writing/programming of the fuse bits and the shadow registers (which hold the value read from the fuses). Each word can be write locked independently, which means that it can no longer be programmed. If the word is write locked, only read is permitted and the overwriting of the shadow registers is not permitted. If the word is not locked, then it can be read sensed and programmed and the shadow registers can be overwritten (to allow emulation of programming without actually blowing fuses). The supplemental array of fuses are always override locked.

Read lock bits are a specific set of fuses that give you the ability to block bus access to specific words. Read locks are on parts of the array consisting of multiple words. The bits for these are stored in the supplementary array section in the last word.

For fuse words using redundancy mode, the word is write locked by setting the OCOTP CTRL[WORDLOCK] bit when programming that fuse word. The write lock will apply to the entire word and cannot be used for partial locking. Fuse words using ECC mode can only be programmed one time, so they are automatically write locked after they are programmed. Each fuse word has write lock bits.

26.2 Boot Fusemap

This section describes the various modes and the selection of the required boot devices.

A separate map is given for each and every boot device. The device select is specified by the BOOT_CFG1[7:4] fuses.

Table 26-1. Boot device select

Boot Device	BOOT_CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]
FlexSPI (Serial NOR)	0	0	0	0
SD	0	1	x	x
MMC/eMMC	1	0	x	x
SEMC (NAND)	0	0	1	x
FlexSPI (Serial NAND)	1	1	x	x

NOTE

The fuses marked as “Reserved” are reserved for NXP internal (and future) use only. Customers **must not** attempt to burn these, because the IC behavior may be unpredictable. The reserved fuses can be read as either '0' or '1'. Security-related fuses are described in the Security Reference Manual (SRM) for this device.

Table 26-2. FlexSPI (Serial NOR) boot fusemap

Address	7	6	5	4	3	2	1	0
0x940 [7:0] (BOOT_CFG1)	0	0	0	0	FLASH_PROBE_TYPE 0 - QuadSPI NOR 1 - MXIC Octal 2 -Micron Octal 3 - Adesto Octal		Reserved	FLASH_AUTO_PROBE_EN 0 - Not enabled 1 - Enabled
0x940 [15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	FLEXSPI_INSTANCE 0 - FLEXSPI1 1 - FLEXSPI2	xSPI_FLASH_TYPE 0 - Boot with default 0x03 Read Enabled 1 - Reserved 2 - HyperFLASH 1V8 3 - HyperFLASH 3V0 4 - MXIC Octal Read 5 - Micron Octal Read		
0x940 [23:16]	Reserved							

Table continues on the next page...

Table 26-2. FlexSPI (Serial NOR) boot fusemap (continued)

Address	7	6	5	4	3	2	1	0
0x940 [31:24]	Reserved							
0x950 [31:0] (BOOT_CFG3 and BOOT_CFG4)	Reserved							ROM preloading 0-Disabled 1-Enabled
0x9A0 [7:0]	Reserved							
0x9A0 [15:8]	Reserved				FLEXSPI_DQS_PIN_SEL 0 - Primary DQS pin 1 - Secondary DQS Pin	FLEXSPI_PIN_GROUP_SEL 0 - Primary Group 1 - Secondary Group	FLASH_CONNECTION_SEL 0 - Connected via PORTA 1 - Parallel Mode 2 - Connected via PORTB	
0x9A0 [23:16]	Reserved							
0x9A0 [31:24]	Reserved							
0xC80 [7:0]	RESET_PIN_EN 0 - Disabled 1 - Enabled	JEDEC_HW_RESET_EN 0-Disabled 1-Enabled	RESET_PIN_SEL 0 - GPIO4[3] 1 - GPIO2[8]	HOLD_TIME 0 - 500us 1 - 1ms 2 - 3ms 3 - 10ms		xSPI_FLASH_FREQ 0 - 100MHz 1 - 120MHz 2 - 133MHz 3 - 166MHz 5 - 80MHz 6 - 60MHz		
0xC80 [15:8]	xSPI_FLASH_IMG_SIZE 0 - specified by the xSPI_FLASH_SEC_IMG_OFFSET 1, ..., 12 - 1MB, ..., 12MB 13 - 256KB 14 - 512KB 15 - 768KB				xSPI_FLASH_DUMMY_CYCLE 0 - Auto-probed Dummy cycles Others - Customer specified dummy cycles			
0xC80 [23:16]	xSPI_FLASH_SEC_IMG_OFFSET fuse value * 256KB							
0xC80 [31:24]	Reserved							

Table 26-3. SD boot fusemap

Address	7	6	5	4	3	2	1	0
0x940 [7:0] (BOOT_CFG1)	0	1	SD/SDXC Speed 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104		SD Power Cycle Enable '0' - No power cycle '1' - Enabled via USDHC_RST pad	SD Loopback Clock Source Sel (for SDR50 and SDR104 only) '0' - Through SD pad '1' - Direct	Port Select 0 - eSDHC1 1 - eSDHC2	Reserved
0x940 [15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Bus Width 0 - 1-bit 1 - 4-bit	SD2_VOLTAGE_SELECTION 0 - 3.3V 1 - 1.8V
0x940 [23:16]	Reserved							
0x940 [31:24]	Reserved							
0x950 [31:0] (BOOT_CFG3 and BOOT_CFG4)	Reserved							
0x970 [7:0]	SD_PWR_CYCLE_SEL '00' - 20ms '01' - 10ms '10' - 5ms '11' - 2.5ms		PWR_STABLE_CYCLE_SEL 0 - 5 ms 1 - 2.5ms	USDHC_CMD_IOMUX_SION_BIT_DISABLE 0 - Enable 1 - Disable	Reserved	SD1_RST_POLARITY_SEL 0 - Reset active low 1 - Reset active high	SD2_RST_POLARITY_SEL 0 - Reset active low 1 - Reset active high	Reserved
0x970 [15:8]	SDMMC_MFG_MODE_DISABLE 0 - Allowed 1 - Disabled	Reserved			USB_SDP_DISABLE 0 - Allowed 1 - Disabled	DLL_OVERRIDE 0 - DLL Slave Mode for SD (delay 10, step 2) 1 - DLL Override Mode for SD	SD_DLL_DLY_CALIB_STEP 00 - 1 01 - 2 10 - 4 11 - 6	
0x970 [23:16]	Reserved							
0x970 [31:24]	Reserved							

Table 26-4. MMC/eMMC boot fusemap

Address	7	6	5	4	3	2	1	0
0x940 [7:0] (BOOT_CFG1)	1	0	SD/MMC Speed 0 - Normal 1- High	Fast Boot Acknowledge Enable 0 - Boot Ack Disabled 1 - Boot Ack Enabled	SD Power Cycle Enable '0' - No power cycle '1' - Enabled via USDHC_RST pad	SD Loopback Clock Source Sel (for SDR50 and SDR104 only) '0' - Through SD pad '1' - Direct	Port Select 0 - eSDHC1 1 - eSDHC2	Fast Boot 0 - Regular 1 - Fast Boot
0x940 [15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	Reserved	Bus Width 00 - 4-bit 01 - 8-bit 10 - 4-bit DDR (MMC 4.4) 11 - 8-bit DDR (MMC 4.4)		SD2 VOLTAGE SELECTION 0 - 3.3V 1 - 1.8V
0x940 [23:16]	Reserved							
0x940 [31:24]	Reserved							
0x950 [31:0] (BOOT_CFG3 and BOOT_CFG4)	Reserved							
0x970 [7:0] BOOT_PARAM1	MMC_PWR_CYCLE_SEL '00' - 20ms '01' - 10ms '10' - 5ms '11' - 2.5ms		PWR_STABLE_CYCLE_SEL 0 - 5 ms 1 - 2.5ms	USDHC_CMD_IOMUX_SION_BIT_DISABLE 0 - Enable 1 - Disable	SD1_VOLTAGE_SEL 0 - 3.3V 1 - 1.8V	SD1_RST_POLARITY_SEL 0 - Reset active low 1 - Reset active high	SD2_RST_POLARITY_SEL 0 - Reset active low 1 - Reset active high	eMMC 5.0 - RESET TO PRE-IDLE STATE 0 - Reset to Idle state 1 - No Idle state
0x970 [15:8] BOOT_PARAM1	SDMMC_MFG_MODE_DISABLE 0 – Allowed 1 – Disabled	Reserved						
0x970 [23:16]	Reserved							
0x970 [31:24]	Reserved							

Table 26-5. SEMC (NAND) boot fusemap

Address	7	6	5	4	3	2	1	0
0x940 [7:0] (BOOT_CFG1)	0	0	1	BOOT_SEARCH_STRIDE Search Stride for FCB and DBBT Search strides in terms of page 0000 - 64 Other: Value = 2^(BOOT_SEARCH_STRIDE)				BOOT_SEARCH_COUNT 0 - 1 1 - 2
0x940 [15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	Reserved	SEMC Access Command 0 - IPG 1 - AXI	SEMC EDO Mode 0 - EDO Mode 1 - Non-EDO mode	ONFI compliant 0 – Yes, ONFI 1 – No, spec
0x940 [23:16]	Reserved							
0x940 [31:24]	Reserved							
0x950 [31:0] (BOOT_CFG3 and BOOT_CFG4)	Reserved							
0xC80 [7:0]	SEMC Clock Freq 0 - 133MHz (2nd Max) 1 - 166MHz (Max)	Ready Check Type 0 - SR 1 - R/B#	RDY# Polarity 0 - Low Active 1 - High Active	ECC_ALG_SEL Disable 0 - Device ECC selected 1 - SW ECC selected	I/O Port Size 0 - 8bit 1 - 16bit	Reserved	Reserved	Reserved
0xC80 [15:8]	Reserved	Status Cmd Type 0 - Common 1 - Enhanced	COL_ADDRESS_WIDTH 000 - 12bits 001 - 09bits 010 - 10bits 011 - 11bits 100 - 13bits 101 - 14bits 110 - 15bits 111 - 16bits			Row_Column_address_mode 00x - 5byte - CA2RA3 010 - 4byte - CA2RA2 011 - 3byte - CA2RA1 10x - 4byte - CA1RA3 110 - 3byte - CA1RA2 111 - 2byte - CA1RA1		
0xC80 [23:16]	Reserved	Reserved	PCS_SEL 000 -CSX0 001 - CSX1 010 - CSX2 011 - CSX3			Pages In Block 000 - 128 001 - 8 010 - 16 011 - 32 100 - 64		

Table continues on the next page...

Table 26-5. SEMC (NAND) boot fusemap (continued)

Address	7	6	5	4	3	2	1	0
							101 - 256 110 - 512 111 - 1024	
0xC80 [31:24]	Reserved	Reserved	Reserved	Reserved	ONFI Timing Mode 000 - Mode0 - 10MHz 001 - Mode1 - 20MHz 010 - Mode2 - 28MHz 011 - Mode3 - 33MHz 100 - Mode4 - 40MHz 101 - Mode5 - 50MHz 11x - Fastest Mode			Device ECC initial status 0 - Enabled 1 - Disabled

Table 26-6. FlexSPI (Serial NAND) boot fusemap

Address	7	6	5	4	3	2	1	0
0x940 [7:0] (BOOT_CFG1)	1	1	SAFE_FREQ Default safe communication frequency 0 – High Speed (50MHz) 1 – Low Speed (30MHZ)	COL_ADDRESS_WIDTH 0 – 12 bits 1 – 13 bits	SPI NAND HOLD TIME Wait hold time before access SPI NAND Flash device 00 - 0 us 01 - 500us 10 - 1ms 11 - 3ms		BOOT_SEARCH_STRIDE Search Stride for FCB and DBBT (in terms of pages) 0 - 64 1 - 128 2 - 256 3 - 32	
0x940 [15:8] (BOOT_CFG2)	Reserved	Reserved	Reserved	Reserved	FLEXSPI_INSTANCE 0 - FLEXSPI1 1 - FLEXSPI2	CS_INTERVAL CS de-asserted interval between two commands 0 – 100ns 1 – 200ns 2 – 400ns 3 – 50ns		BOOT_SEARCH_COUNT 0 - 1 1 - 2
0x940 [23:16]	Reserved							
0x940 [31:24]	Reserved							
0x950 [31:0]	Reserved							

Table continues on the next page...

Table 26-6. FlexSPI (Serial NAND) boot fusemap (continued)

Address	7	6	5	4	3	2	1	0
(BOOT_CFG3 and BOOT_CFG4)								
0xC80 [7:0]	SPI NAND BOOT - Override busy offset 0- Use default busy bit offset 0 1 - Override default busy bit offset using busy bit offset	Reserved	PAGE_RD_TIME 0 - 75us Others - Fuse value * 10us					
0xC80 [15:8]	Bypass Read Status command 0 - Determine wheter the FLASH busy according to the FLASH status register via Read Status command 1 - Assume FLASH is idle after waiting the interval specified by the "Page Read Time" Fuse field	Bypass ECC Read 0 - ECC Read is enabled 1 - ECC read is bypassed	SPI NAND BOOT - Busy Bit offset Busy bit offset = Fuse value					
0xC80 [23:16]	Page Read Command							
0xC80 [31:24]	Cache Read Command							

The following table applies for all boot devices.

Table 26-7. Boot parameter fusemap

Address	7	6	5	4	3	2	1	0
0x840 [7:0]	Reserved				XECC_ENABLE 0 - Disabled 1 - Enabled	MECC_ENABLE 0 - Disabled 1 - Enabled	Reserved	
0x840 [15:8]	FLEXRAMECC_ENABLE 0 - Disabled 1 - Enabled	Reserved						
0x970 [7:0] BOOT_PARAM1	Reserved							
0x970 [15:8] BOOT_PARAM1	Reserved	XMC_CHK_EN 0 - XMC Integrity check is not enabled 1 - XMC Integrity Check is enabled	Reserved					
0x970 [23:16]	Reserved							
0x970 [31:24]	Reserved							
0x980 [31:0]	Reserved							
0x990 [7:0] BOOT_PARAM3	Reserved		BOOT_FAIL_IND_EN 0 - Disabled 1 - Enabled	Boot Failure Indicator Pin Select[4:0] 00000 - GPIO1.IO00 00001 - GPIO1.IO01 ... 11111 - GPIO1.IO31 Please refer to the Chip IO section for pins which GPIO1.IOxx will be muxed to by default.				
0x990 [15:8] BOOT_PARAM3	Override FLEXSPI/ SEMC/ SDEMMC Pad Settings (using PAD_SETTINGS value) 0 - Disable	SDMMC_DLL_DLY[6:0] Delay target for SD/eMMC DLL, it is applied to override mode target delay depends on DLL Override fuse bit value. DELAY_CELL_NUM (FlexSPI NOR) "000" - DLL Override feature is disabled "001-111" - DLL Override feature is enabled Please refer to OVRDVAL in the FLEXSPI chapter for more details.						

Table continues on the next page...

Table 26-7. Boot parameter fusemap (continued)

Address	7	6	5	4	3	2	1	0
	1 - Enable							
0x990 [23:16]	Reserved							
0x990 [31:24]	Reserved							
0x9A0 [7:0] BOOT_PARAM4	I_CACHE_DISABLE 0 = cache enabled 1 = cache disabled	D_CACHE_DISABLE 0 = cache enabled 1 = cache disabled	LPB_BOOT Divide (Core / Bus) based on Boot frequencies '00' - Div by 1 '01' - Div by 2 '10' - Div by 4 '11' - Div by 8		BOOT_FREQ M7: 0 - 400 MHz 1 - 700 MHz M4: 0 - 200 MHz 1 - 240 MHz	OSC_REF_FREQ 0 - 24 MHz 1 - 19.2 MHz	UART_SDP_DISABLE 0 = enable 1 = disable	SDP_DISABLE 0 = enable 1 = disable
0x9A0 [15:8] BOOT_PARAM4	WDOG_ENABLE 0 = WDOG1 is not enabled 1 = WDOG1 is enabled	FORCE_INTERNAL_BOOT 0 - Boot mode is determined by the BOOT_MODE pins 1 - Forced Internal boot mode, the value of BOOT_MODE pins is ignored	Reserved		FLEXSPI_DQS_PIN_SEL 0 - Primary DQS pin 1 - Secondary DQS Pin	FLEXSPI_PIN_GROUP_SEL 0 - Primary Group 1 - Secondary Group	FLASH_CONNECTION_SEL 0 - Connected via PORTA 1 - Parallel Mode 2 - Connected via PORTB	
0x9A0 [23:16]	Reserved							
0x9A0 [31:24]	Reserved							
0x9B0 [7:0]	Reserved		WDOG_B_PIN_EN 0 - Dsiabled 1 - Enabled	WDOG_B_PIN_SEL 0 - GPIO_DISP_B2_00 1 - GPIO_DISP_B2_15 2 - GPIO_AD_04 3 - GPIO_EMC_B2_18	WDOG Timeout Select 000 - 64s 001 - 32s 010 - 16s 011 - 8s			
0xC70 [7:0]	Reserved							
0xC70 [15:8]	FLEXSPI/SEMC/SD/EMMC PAD_SETTINGS [1]: DSE/PDRV. This value is used to overwrite the DSE/PDRV bit for pad setting.							
0xC70 [23:16]	Reserved	Reserved	Default_FlexRAM_Part[5:0] See FlexRAM Partition Table for details.					

Table continues on the next page...

Table 26-7. Boot parameter fusemap (continued)

Address	7	6	5	4	3	2	1	0
0xC70 [31:24]	Reserved		Reserved	SPI_MEM_SPEED 0 - 20MHz 1 - 10MHz 2 - 5MHz 3 - 2MHz		SPI INSTANCE 0 - LPSPI1 1 - LPSPI2 2 - LPSPI3 3 - LPSPI4		RECOVERY_BOOT_EN 0 - Disabled 1 - Enabled
0xC90 - 0x18F0	Reserved							

26.3 Fusemap Descriptions Table

This section describes the chip fusemap descriptions.

Table 26-8. Fusemap Descriptions

Fuse Address	Fuse Name	# of Fuses	Fuse Function	Setting	Read Locked by	ECC/RED
0x800 - 0x840[1]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x840[2]	MECC_ENABLE	1	Enable memory ECC	0 - disabled 1 - enabled	N/A	RED
0x840[3]	XECC_ENABLE	1	Enable external memory ECC	0 - disabled 1 - enabled	N/A	RED
0x840[4] - 0x840[14]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x840[15]	FLEXRAMECC_ENABLE	1	Enable FlexRAM ECC	0 - disabled 1 - enabled	N/A	RED
0x840[16] - 0x860	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x870[0]	M7_DEBUG_DISABLE	1	Disable ability to debug M7	0 - enabled 1 - disabled	N/A	RED
0x870[1]	M4_DEBUG_DISABLE	1	Disable ability to debug M4	0 - enabled 1 - disabled	N/A	RED
0x870[3:2]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x870[4]	FBB_DISABLE	1	Disable FBB **NOTE:** The user should check the value of this fuse bit, then correctly configure the	0 - FBB is required for overdrive mode operation 1 - FBB is not required for overdrive mode operation	N/A	RED

Table continues on the next page...

Table 26-8. Fusemap Descriptions (continued)

Fuse Address	Fuse Name	# of Fuses	Fuse Function	Setting	Read Locked by	ECC/RED
			FBB_M7_ENABLE_SP_REGISTER to enable or disable the FBB feature for a specific Set Point.			
0x870[5] - 0x890[9]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x890[10]	GP1_RLOCK	1	Read lock for GP1	0 - Unlock (Reads enabled) 1 - Lock (Reads disabled)	N/A	RED
0x890[11]	GP2_RLOCK	1	Read lock for GP2	0 - Unlock (Reads enabled) 1 - Lock (Reads disabled)	N/A	RED
0x890[12]	GP3_RLOCK	1	Read lock for GP3	0 - Unlock (Reads enabled) 1 - Lock (Reads disabled)	N/A	RED
0x890[13]	GP4_RLOCK	1	Read lock for GP4	0 - Unlock (Reads enabled) 1 - Lock (Reads disabled)	N/A	RED
0x890[14]	GP5_RLOCK	1	Read lock for GP5	0 - Unlock (Reads enabled) 1 - Lock (Reads disabled)	N/A	RED
0x890[15] - 0x8F0[31:16]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x900[31:0] - 0x910[10:0]	JTAG_CHALL[42:0]/ UNIQUE_ID[42:0]	43	JTAG CHALLENGE/ Unique ID Burned by NXP	-	N/A	ECC
0x910[15:11]	JTAG_CHALL[47:43] / UNIQUE_ID[47:43]	5	JTAG CHALLENGE/ Unique ID Burned by NXP	-	N/A	ECC
0x910[23:16]	JTAG_CHALL[55:48] / UNIQUE_ID[55:48]	8	JTAG CHALLENGE/ Unique ID Burned by NXP	-	N/A	ECC
0x910[31:24]	JTAG_CHALL[63:56] / UNIQUE_ID[63:56]	8	JTAG CHALLENGE/ Unique ID Burned by NXP	-	N/A	ECC
0x920[23:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x920[27:24]	SI_REV[3:0]	4	Silicon Revision number.	-	N/A	ECC
0x920[31:28]	Reserved	4	Reserved	Reserved	Reserved	Reserved
0x930[31:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x940[7:0]	BOOT_CFG1[7:0]	8	Refer to Boot Fusemap	-	N/A	RED
0x940[15:8]	BOOT_CFG2[7:0]	8	Refer to Boot Fusemap	-	N/A	RED
0x940[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved

Table continues on the next page...

Table 26-8. Fusemap Descriptions (continued)

Fuse Address	Fuse Name	# of Fuses	Fuse Function	Setting	Read Locked by	ECC/RED
0x950[7:0]	BOOT_CFG3[7:0]	8	Refer to Boot Fusemap	-	N/A	RED
0x950[15:8]	BOOT_CFG4[7:0]	8	Refer to Boot Fusemap	-	N/A	RED
0x950[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0x960[3:0]	Reserved	4	Reserved	Reserved	Reserved	Reserved
0x960[4]	BT_FUSE_SEL	1	Determines, whether using fuses for boot configuration, or GPIO /Serial downloader	If boot_mode="10" (internal boot) 0=Boot mode configuration is taken from GPIOs. 1=Boot mode configuration is taken from fuses. If boot_mode="00" (boot from fuses) 0 - Boot using Serial Loader (USB) 1- Boot mode configuration is taken from fuses.	N/A	RED
0x960[5]	Reserved	1	Reserved	Reserved	Reserved	Reserved
0x960[11:6]	Rreserved	1	Reserved	Reserved	Reserved	Reserved
0x960[12]	BT_CORE_SEL	1	Boot Core Selection. Only applied to multi-core part.	0 - boot from M7 1 - boot from M4	N/A	RED
0x960[31:13]	Reserved	19	Reserved	Reserved	Reserved	Reserved
0x970[15:0]	BOOT_PARAM1	16	Refer to Boot Fusemap	-	N/A	RED
0x970[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0x980[15:0]	BOOT_PARAM2	16	Refer to Boot Fusemap	-	N/A	RED
0x980[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0x990[15:0]	BOOT_PARAM3	16	Refer to Boot Fusemap	-	N/A	RED
0x990[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0x9A0[15:0]	BOOT_PARAM4	16	Refer to Boot Fusemap	-	N/A	RED
0x9A0[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0x9B0[15:0]	BOOT_PARAM5	16	Refer to Boot Fusemap	-	N/A	RED
0x9B0[31:16] - 0xA20[15:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0xA20[27:16]	TMPSNS_TEMP_VAL	12	The TMPSNS value for ts25c	-	N/A	RED

Table continues on the next page...

Table 26-8. Fusemap Descriptions (continued)

Fuse Address	Fuse Name	# of Fuses	Fuse Function	Setting	Read Locked by	ECC/RED
0xA20[31:28] - 0xA70[31:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0xA90[15:0] - 0xA80[31:0]	MAC1_ADDR[47:0]	48	Ethernet 1 MAC Address	-	N/A	ECC
0xA90[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0xAB0[15:0] - 0xAA0[31:0]	MAC2_ADDR[47:0]	48	Ethernet 2 MAC Address	-	N/A	ECC
0xAB0[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0xAD0[15:0] - 0xAC0[31:0]	MAC3_ADDR[47:0]	48	Ethernet 3 MAC Address	-	N/A	ECC
0xAD0[31:16]	Reserved	16	Reserved	Reserved	Reserved	Reserved
0xAE0[15:0]	USB_VID[15:0]	16	USB VID	-	N/A	ECC
0xAE0[31:16]	USB_PID[15:0]	16	USB PID	-	N/A	ECC
0xAF0[31:0]	Reserved	32	Reserved	Reserved	Reserved	Reserved
0xB00[31:0] - 0xC70[7:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0xC70[15:8]	BOOT_CONFIG_MISC1[15:8]	8	Refer to Boot Fusemap	-	N/A	ECC
0xC70[21:16]	Default_FlexRAM_Part[5:0]	6	Default FlexRAM RAM bank partioning	See FlexRAM Partition Table	N/A	ECC
0xC70[31:22]	BOOT_CONFIG_MISC1[31:22]	10	Refer to Boot Fusemap	-	N/A	ECC
0xC80[31:0]	BOOT_CONFIG_MISC2[31:0]	32	Refer to Boot Fusemap	-	N/A	ECC
0xC90[31:0]	BOOT_CONFIG_MISC3[31:0]	32	Refer to Boot Fusemap	-	N/A	ECC
0xCA0[31:0]	BOOT_CONFIG_MISC4[31:0]	32	Refer to Boot Fusemap	-	N/A	ECC
0xCB0[31:0] - 0x12F0[31:0]	Reserved	-	Reserved	Reserved	Reserved	Reserved
0x1300[511:0]	GP1[511:0]	512	General Purpose fuse register #1	-	GP1_RLOCK	ECC
0x1400[511:0]	GP2[511:0]	512	General Purpose fuse register #2	-	GP2_RLOCK	ECC
0x1500[511:0]	GP3[511:0]	512	General Purpose fuse register #3	-	GP3_RLOCK	ECC
0x1600[511:0]	GP4[511:0]	512	General Purpose fuse register #4	-	GP4_RLOCK	ECC
0x1700[511:0]	GP5[511:0]	512	General Purpose fuse register #5	-	GP5_RLOCK	ECC
0x1800[511:0]	BOOT_CONFIG_MISC5[511:0]	512	Refer to Boot Fusemap	-	-	ECC

26.4 FlexRAM Partition Table

This section describes the default FlexRAM partitioning table.

eFuse Default_FlexRAM_Part[5:0]	DTCM SIZE	ITCM SIZE	OCRAM SIZE
6'b000000	256KB	256KB	0KB
6'b000001	320KB	192KB	0KB
6'b000010	384KB	128KB	0KB
6'b000011	448KB	64KB	0KB
6'b000100	512KB	0KB	0KB
6'b000101	192KB	320KB	0KB
6'b000110	128KB	384KB	0KB
6'b000111	64KB	448KB	0KB
6'b001000	0KB	512KB	0KB
6'b001001	256KB	192KB	64KB
6'b001010	320KB	128KB	64KB
6'b001011	384KB	64KB	64KB
6'b001100	448KB	0KB	64KB
6'b001101	192KB	256KB	64KB
6'b001110	128KB	320KB	64KB
6'b001111	64KB	384KB	64KB
6'b010000	0KB	448KB	64KB
6'b010001	192KB	192KB	128KB
6'b010010	256KB	128KB	128KB
6'b010011	320KB	64KB	128KB
6'b010100	384KB	0KB	128KB
6'b010101	128KB	256KB	128KB
6'b010110	64KB	320KB	128KB
6'b010111	0KB	384KB	128KB
6'b011000	192KB	128KB	192KB
6'b011001	256KB	64KB	192KB
6'b011010	320KB	0KB	192KB
6'b011011	128KB	192KB	192KB
6'b011100	64KB	256KB	192KB
6'b011101	0KB	320KB	192KB
6'b011110	128KB	128KB	256KB
6'b011111	192KB	64KB	256KB
6'b100000	256KB	0KB	256KB
6'b100001	64KB	192KB	256KB

Table continues on the next page...

eFuse Default_FlexRAM_Part[5:0]	DTCM SIZE	ITCM SIZE	OCRAM SIZE
6'b100010	0KB	256KB	256KB
6'b100011	128KB	64KB	320KB
6'b100100	192KB	0KB	320KB
6'b100101	64KB	128KB	320KB
6'b100110	0KB	192KB	320KB
6'b100111	64KB	64KB	384KB
6'b101000	128KB	0KB	384KB
6'b101001	0KB	128KB	384KB
6'b101010	64KB	0KB	448KB
6'b101011	0KB	64KB	448KB
6'b101100	0KB	0KB	512KB

Chapter 27
On-Chip OTP Controller (OCOTP_CTRL)

27.1 Chip-specific OCOTP_CTRL information

Table 27-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

Security-related fuses are shown as reserved here. For details, refer to the Security Reference Manual.

27.2 Overview

This section contains information describing the On-Chip OTP controller (OCOTP_CTRL) along with details about the block functionality and implementation.

In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

27.2.1 Block Diagram

The system level block diagram for OCOTP is provided below.

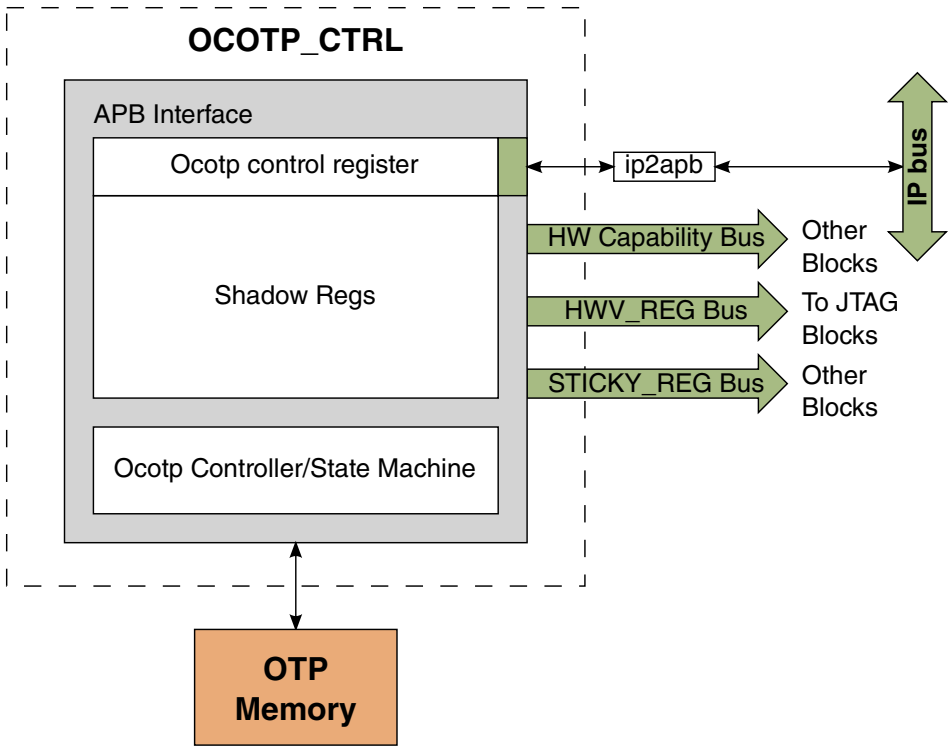


Figure 27-1. OCOTP System Level Block Diagram

27.2.2 Features

The OCOTP provides the following features :

- 32-bit word restricted program and read to 8 kbit of eFuse.
- Loading and housing of eFuse content into shadow registers
- Memory-mapped (restricted) access to shadow registers
- Generation of STICKY_REG which consists of sticky register bits
- Provides program-protect and read-protect of eFuse
- Provides read protection of shadow register
- Supports Error-correcting code (ECC) mode programming and reading for all the user fuse words
- Supports redundancy mode programming and reading for all the supplementary fuse words

27.3 Functional Description

27.3.1 Operations

The IP bus interface of the OCOTP has the following functions.

- Configure control registers for programming and reading all the fuse words
- Read shadow registers

OCOTP configuration for programs and reads are performed on 32-bit words for software (SW) convenience. For writes, the 32-bit word reflects the "write-mask". Bit fields with 0 will not be programmed and bit fields with 1 will be programmed.

NOTE

For fuse words that use ECC, the fuse word can only be programmed one time, otherwise it will report single-bit error correction (SEC) / double-bit error detection (DED) issues in read or reload. Refer to the fusemap chapter to identifiy which fuse words use ECC for error correction.

NOTE

For fuse words that use redundant bits, the fuse word can be programmed more than one time (not the same fuse bit). Refer to the fusemap chapter to identifiy which fuse words use redundant bits for error correction.

The full 8 Kbits of fuses are implemented in one physical OTP memory block. The 8 kbit is divided into 32 banks by function, where each bank has 8 fuse words (8 x 32-bit = 256 bits per bank).

27.3.1.1 Shadow Register Reload

All fuse words in efuse box are shadowed. This means that at reset, fuse values are read and then copied into memory-mapped shadow registers. If fuses are subsequently programmed, the shadow registers should be reloaded to keep them coherent with the fuse bank arrays.

The "reload shadows" feature allows the user to force a reload of the shadow registers without having to reset the device. To force a reload, complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a new access can be requested.
2. Set the CTRL[RELOAD_SHADOWS] bit. OCOTP will read all the fuses one by one and put it into corresponding shadow register.
3. Wait for CTRL[BUSY] and CTRL[RELOAD_SHADOWS] to be cleared by the controller.
4. Check the values of OUT_STATUS[SEC_RELOAD] and OUT_STATUS[DED_RELOAD].
   - If OUT_STATUS[SEC_RELOAD] is set, it indicates at least one single bit correction happened when reloading.
   - If OUT_STATUS[DED_RELOAD] is set, it indicates at least one double-bit error happened when reloading. That is, one or more fuse words are not correct when reading from eFuse.

The controller will automatically clear the CTRL[RELOAD_SHADOWS] bit after the successful completion of the operation.

27.3.1.2 Fuse and Shadow Register Read

All shadow registers are always readable through the IPS bus except some secret key regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA.

In addition, CTRL[ERROR] will be set. It must be cleared by software before any new write, read, or reload access can be issued. Subsequent reads to unlocked shadow registers will still work successfully.

To read fuse words directly from the fusebox instead of the shadow registers, complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a read access can be requested.
2. Write the requested fuse address to CTRL[ADDR].
3. Set READ_CTRL[READ_FUSE] to start the read operation.
4. Wait for the controller to clear CTRL[BUSY]. A read request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition, CTRL[ERROR] will be set. It must be cleared by software before any new access can be issued.

5. Read READ_FUSE_DATA to get fuse word value. READ_FUSE_DATA will be 0xBADABADA when CTRL[ERROR] is set.
6. After reading is completed, it needs to check OUT_STATUS[DED] value to confirm if read fuse value is correct. If it is 1, read fuse value is wrong, otherwise it is correct.

27.3.1.3 Fuse Writes

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank complete the following steps:

1. Check that CTRL[BUSY] and CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read, or reload must be completed before a write access can be requested.
2. Write the requested address to CTRL[ADDR] and program the unlock code into CTRL[WR_UNLOCK]. The unlock code must be programmed for each write access. The unlock code is documented in the register description. Both the unlock code and address can be written in the same operation.
3. When programming fuse word, OCOTP_CTRL will auto select ECC or redundancy mode to program the fuse word according to fuse map definition. In ECC mode, the 32 fuse bits in one word can only be written one time. In redundancy mode, the word can be written more than one time as long as they are different fuse bits. For the fuse word which needs be programmed with ECC mode, it is recommended to set CTRL[WORDLOCK] with unlock code and address in same operation.
4. Write the data to the DATA register. This will automatically set CTRL[BUSY] and clear CTRL[WR_UNLOCK]. Bit fields with 1's will result in that OTP bit being programmed. Bit fields with 0's will be ignored. At the same time that the write is accepted, the controller makes an internal copy of CTRL[ADDR] which cannot be updated until the next write sequence is initiated. This copy guarantees that erroneous writes to CTRL[ADDR] will not affect an active write operation. During the write operation, DATA cannot be modified. If the requested fuse word is locked, the programming request will not be performed. Checking the appropriate LOCKEDn register before attempting to program is recommended
5. Wait for the controller to clear CTRL[BUSY]. A write request to a protected or locked region will result in no OTP access and no setting of CTRL[BUSY]. In addition CTRL[ERROR] will be set. It must be cleared by software before any new write access can be issued.
6. Check the values of OUT_STATUS[PROGFAIL] and OUT_STATUS[LOCKED]. If either of these two bits is set, it indicates that programming has failed.

It should be noted that write latencies to OTP are numbers of 10 micro-seconds per word. The write latency is based on the number of "1" bits being written. For example, to program half the fuse bits in one 32-bt word requires 10 us x 16.

27.3.1.4 Bank redundancy vs ECC

There are two mechanisms used by the fuse block to ensure that its values are correct. Each of these is performed on a word by word basis, however, the selected option is hardwired on a bank by bank basis.

Fuses can generally only be set from 0 to 1 and the most common failure is a 1 appearing as a 0. The simplest mechanism to verify these failures is redundancy. Redundancy is when two bits are OR'd together so that if either is set, then the output value is a 1.

In this case only 16 of the 32 bits are available (bits 15:0 are automatically copied to 31:16) and checked on read. The value here is that each pair of bits can be programmed at different times. This allows for multiple programming operations on a word, so bits can be configured at different times.

NOTE

In shadow registers, the inverted version of [15:0] will show in [31:16]. In the fuse array, [31:16] is the same as [15:0].

The more complex ECC mechanism is used when higher numbers of bits must be stored and configured at the same time in a group of 32. When written, the ECC must be calculated to allow an ECC correction. When read, any one bit of failure will be corrected, 2 bits will be detected (and an error flag asserted) by the controller.

NOTE

As the ECC is generated for the whole word, the whole word must be setup at one time and individual bits cannot be modified independently.

27.3.1.5 Error Conditions

The CTRL[ERROR] bit will be set under the following conditions:

- A read is performed to a shadow register which has been read locked.
- A program is performed to a non-existent fuse word
- A read is performed to a fuse word which has been read locked.
- Overlapping access - a read, write, or reload is attempted while a read, write, or reload is already in progress.

27.3.1.6 Write Postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations (direct reads, shadow reloads, or other writes) following a write must wait at least 2 µs after CTRL[BUSY] clears. The delay guarantees programming voltages on-chip reach a steady state when exiting a write sequence.

A recommended software sequence to meet the postamble requirements is as follows:

1. Issue the write and poll for CTRL[BUSY].
2. After CTRL[BUSY] is clear, wait 2 µs.
3. Perform the next OTP operation.

27.3.1.7 Power control of OTP memory

For every fuse operation, the controller will automatically enable power to OTP memory. It is SW responsibility to shut off power to OTP memory in idle status to save power.

1. After system power-up, SW should write 1 to PDN[PDN] to shut off power to the OTP array.
2. After fuse program/read triggered by SW are completed, SW should write 1 to PDN[PDN] to shut off power to the OTP array unless additional fuse operations are needed.

27.3.2 Fuse Shadow Memory Footprint

The OTP memory footprint is shown in the following figure. The registers are grouped by lock region. Their names correspond to the fusemap names.

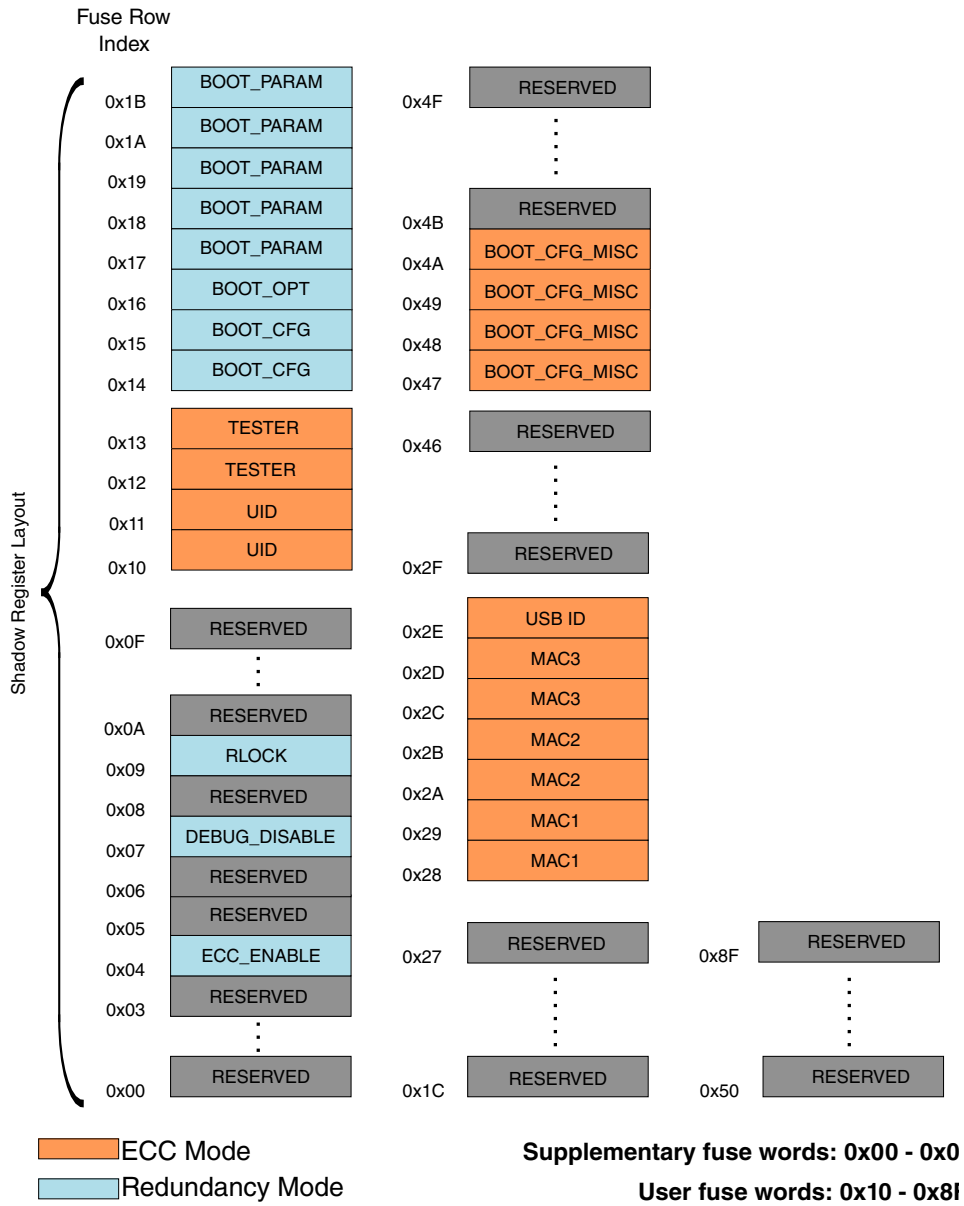


Figure 27-2. OTP Memory Footprint

NOTE

The device includes additional fuses that are not readable through shadow registers. Fuse row indexes 0xB0-0x110, must be read directly from the fusebox. Refer to Fuse and Shadow Register Read for details.

27.3.3 Resets

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time CTRL[BUSY] is set.

27.3.4 Clocks

The table found here describes the clock sources for OCOTP. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 27-2. OCOTP Clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock

27.4 External Signals

There are no external signals pinned out for OCOTP.

27.5 Fuse Map

See the Fusemap chapter of this reference manual for more information.

27.6 Memory Map/Register Definition

The OCOTP Memory Map/Register Definition can be found here.

NOTE

Writing or reading an unimplemented register address in the OCOTP Controller will not send a bus error and read data will be 0.

27.6.1 register descriptions

27.6.1.1 OCOTP memory map

OCOTP base address: 40CA_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	OTP Controller Control and Status Register (CTRL)	32	Table 27-2	0000_0000h
4h	OTP Controller Control and Status Register (CTRL_SET)	32	Table 27-2	0000_0000h
8h	OTP Controller Control and Status Register (CTRL_CLR)	32	Table 27-2	0000_0000h
Ch	OTP Controller Control and Status Register (CTRL_TOG)	32	Table 27-2	0000_0000h
10h	OTP Controller PDN Register (PDN)	32	Table 27-2	0000_0000h
20h	OTP Controller Write Data Register (DATA)	32	RW	0000_0000h
30h	OTP Controller Read Control Register (READ_CTRL)	32	Table 27-2	0000_0000h
90h	8K OTP Memory STATUS Register (OUT_STATUS)	32	Table 27-2	0000_0000h
94h	8K OTP Memory STATUS Register (OUT_STATUS_SET)	32	Table 27-2	0000_0000h
98h	8K OTP Memory STATUS Register (OUT_STATUS_CLR)	32	Table 27-2	0000_0000h
9Ch	8K OTP Memory STATUS Register (OUT_STATUS_TOG)	32	Table 27-2	0000_0000h
B0h	OTP Controller Version Register (VERSION)	32	RO	0A00_0000h
100h	OTP Controller Read Data 0 Register (READ_FUSE_DATA0)	32	RW	0000_0000h
110h	OTP Controller Read Data 1 Register (READ_FUSE_DATA1)	32	RW	0000_0000h
120h	OTP Controller Read Data 2 Register (READ_FUSE_DATA2)	32	RW	0000_0000h
130h	OTP Controller Read Data 3 Register (READ_FUSE_DATA3)	32	RW	0000_0000h
140h	SW_LOCK Register (SW_LOCK)	32	RW	0000_0000h
150h	BIT_LOCK Register (BIT_LOCK)	32	RW	0000_0000h
600h	OTP Controller Program Locked Status 0 Register (LOCKED0)	32	Table 27-2	0000_0000h
610h	OTP Controller Program Locked Status 1 Register (LOCKED1)	32	RO	0000_0000h
800h	Value of fuse word 0 (FUSE0)	32	RO	0000_0000h
810h	Value of fuse word 1 (FUSE1)	32	RO	0000_0000h
820h	Value of fuse word 2 (FUSE2)	32	RO	0000_0000h
830h	Value of fuse word 3 (FUSE3)	32	RO	0000_0000h
840h	Value of fuse word 4 (FUSE4)	32	RO	0000_0000h
850h	Value of fuse word 5 (FUSE5)	32	RO	0000_0000h
860h	Value of fuse word 6 (FUSE6)	32	RO	0000_0000h
870h	Value of fuse word 7 (FUSE7)	32	RO	0000_0000h
880h	Value of fuse word 8 (FUSE8)	32	RO	0000_0000h
890h	Value of fuse word 9 (FUSE9)	32	RO	0000_0000h
8A0h	Value of fuse word 10 (FUSE10)	32	RO	0000_0000h
8B0h	Value of fuse word 11 (FUSE11)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8C0h	Value of fuse word 12 (FUSE12)	32	RO	0000_0000h
8D0h	Value of fuse word 13 (FUSE13)	32	RO	0000_0000h
8E0h	Value of fuse word 14 (FUSE14)	32	RO	0000_0000h
8F0h	Value of fuse word 15 (FUSE15)	32	RO	0000_0000h
900h	Value of fuse word 16 (FUSE16)	32	RO	0000_0000h
910h	Value of fuse word 17 (FUSE17)	32	RO	0000_0000h
920h	Value of fuse word 18 (FUSE18)	32	RO	0000_0000h
930h	Value of fuse word 19 (FUSE19)	32	RO	0000_0000h
940h	Value of fuse word 20 (FUSE20)	32	RO	0000_0000h
950h	Value of fuse word 21 (FUSE21)	32	RO	0000_0000h
960h	Value of fuse word 22 (FUSE22)	32	RO	0000_0000h
970h	Value of fuse word 23 (FUSE23)	32	RO	0000_0000h
980h	Value of fuse word 24 (FUSE24)	32	RO	0000_0000h
990h	Value of fuse word 25 (FUSE25)	32	RO	0000_0000h
9A0h	Value of fuse word 26 (FUSE26)	32	RO	0000_0000h
9B0h	Value of fuse word 27 (FUSE27)	32	RO	0000_0000h
9C0h	Value of fuse word 28 (FUSE28)	32	RO	0000_0000h
9D0h	Value of fuse word 29 (FUSE29)	32	RO	0000_0000h
9E0h	Value of fuse word 30 (FUSE30)	32	RO	0000_0000h
9F0h	Value of fuse word 31 (FUSE31)	32	RO	0000_0000h
A00h	Value of fuse word 32 (FUSE32)	32	RO	0000_0000h
A10h	Value of fuse word 33 (FUSE33)	32	RO	0000_0000h
A20h	Value of fuse word 34 (FUSE34)	32	RO	0000_0000h
A30h	Value of fuse word 35 (FUSE35)	32	RO	0000_0000h
A40h	Value of fuse word 36 (FUSE36)	32	RO	0000_0000h
A50h	Value of fuse word 37 (FUSE37)	32	RO	0000_0000h
A60h	Value of fuse word 38 (FUSE38)	32	RO	0000_0000h
A70h	Value of fuse word 39 (FUSE39)	32	RO	0000_0000h
A80h	Value of fuse word 40 (FUSE40)	32	RO	0000_0000h
A90h	Value of fuse word 41 (FUSE41)	32	RO	0000_0000h
AA0h	Value of fuse word 42 (FUSE42)	32	RO	0000_0000h
AB0h	Value of fuse word 43 (FUSE43)	32	RO	0000_0000h
AC0h	Value of fuse word 44 (FUSE44)	32	RO	0000_0000h
AD0h	Value of fuse word 45 (FUSE45)	32	RO	0000_0000h
AE0h	Value of fuse word 46 (FUSE46)	32	RO	0000_0000h
AF0h	Value of fuse word 47 (FUSE47)	32	RO	0000_0000h
B00h	Value of fuse word 48 (FUSE48)	32	RO	0000_0000h
B10h	Value of fuse word 49 (FUSE49)	32	RO	0000_0000h
B20h	Value of fuse word 50 (FUSE50)	32	RO	0000_0000h
B30h	Value of fuse word 51 (FUSE51)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B40h	Value of fuse word 52 (FUSE52)	32	RO	0000_0000h
B50h	Value of fuse word 53 (FUSE53)	32	RO	0000_0000h
B60h	Value of fuse word 54 (FUSE54)	32	RO	0000_0000h
B70h	Value of fuse word 55 (FUSE55)	32	RO	0000_0000h
B80h	Value of fuse word 56 (FUSE56)	32	RO	0000_0000h
B90h	Value of fuse word 57 (FUSE57)	32	RO	0000_0000h
BA0h	Value of fuse word 58 (FUSE58)	32	RO	0000_0000h
BB0h	Value of fuse word 59 (FUSE59)	32	RO	0000_0000h
BC0h	Value of fuse word 60 (FUSE60)	32	RO	0000_0000h
BD0h	Value of fuse word 61 (FUSE61)	32	RO	0000_0000h
BE0h	Value of fuse word 62 (FUSE62)	32	RO	0000_0000h
BF0h	Value of fuse word 63 (FUSE63)	32	RO	0000_0000h
C00h	Value of fuse word 64 (FUSE64)	32	RO	0000_0000h
C10h	Value of fuse word 65 (FUSE65)	32	RO	0000_0000h
C20h	Value of fuse word 66 (FUSE66)	32	RO	0000_0000h
C30h	Value of fuse word 67 (FUSE67)	32	RO	0000_0000h
C40h	Value of fuse word 68 (FUSE68)	32	RO	0000_0000h
C50h	Value of fuse word 69 (FUSE69)	32	RO	0000_0000h
C60h	Value of fuse word 70 (FUSE70)	32	RO	0000_0000h
C70h	Value of fuse word 71 (FUSE71)	32	RO	0000_0000h
C80h	Value of fuse word 72 (FUSE72)	32	RO	0000_0000h
C90h	Value of fuse word 73 (FUSE73)	32	RO	0000_0000h
CA0h	Value of fuse word 74 (FUSE74)	32	RO	0000_0000h
CB0h	Value of fuse word 75 (FUSE75)	32	RO	0000_0000h
CC0h	Value of fuse word 76 (FUSE76)	32	RO	0000_0000h
CD0h	Value of fuse word 77 (FUSE77)	32	RO	0000_0000h
CE0h	Value of fuse word 78 (FUSE78)	32	RO	0000_0000h
CF0h	Value of fuse word 79 (FUSE79)	32	RO	0000_0000h
D00h	Value of fuse word 80 (FUSE80)	32	RO	0000_0000h
D10h	Value of fuse word 81 (FUSE81)	32	RO	0000_0000h
D20h	Value of fuse word 82 (FUSE82)	32	RO	0000_0000h
D30h	Value of fuse word 83 (FUSE83)	32	RO	0000_0000h
D40h	Value of fuse word 84 (FUSE84)	32	RO	0000_0000h
D50h	Value of fuse word 85 (FUSE85)	32	RO	0000_0000h
D60h	Value of fuse word 86 (FUSE86)	32	RO	0000_0000h
D70h	Value of fuse word 87 (FUSE87)	32	RO	0000_0000h
D80h	Value of fuse word 88 (FUSE88)	32	RO	0000_0000h
D90h	Value of fuse word 89 (FUSE89)	32	RO	0000_0000h
DA0h	Value of fuse word 90 (FUSE90)	32	RO	0000_0000h
DB0h	Value of fuse word 91 (FUSE91)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
DC0h	Value of fuse word 92 (FUSE92)	32	RO	0000_0000h
DD0h	Value of fuse word 93 (FUSE93)	32	RO	0000_0000h
DE0h	Value of fuse word 94 (FUSE94)	32	RO	0000_0000h
DF0h	Value of fuse word 95 (FUSE95)	32	RO	0000_0000h
E00h	Value of fuse word 96 (FUSE96)	32	RO	0000_0000h
E10h	Value of fuse word 97 (FUSE97)	32	RO	0000_0000h
E20h	Value of fuse word 98 (FUSE98)	32	RO	0000_0000h
E30h	Value of fuse word 99 (FUSE99)	32	RO	0000_0000h
E40h	Value of fuse word 100 (FUSE100)	32	RO	0000_0000h
E50h	Value of fuse word 101 (FUSE101)	32	RO	0000_0000h
E60h	Value of fuse word 102 (FUSE102)	32	RO	0000_0000h
E70h	Value of fuse word 103 (FUSE103)	32	RO	0000_0000h
E80h	Value of fuse word 104 (FUSE104)	32	RO	0000_0000h
E90h	Value of fuse word 105 (FUSE105)	32	RO	0000_0000h
EA0h	Value of fuse word 106 (FUSE106)	32	RO	0000_0000h
EB0h	Value of fuse word 107 (FUSE107)	32	RO	0000_0000h
EC0h	Value of fuse word 108 (FUSE108)	32	RO	0000_0000h
ED0h	Value of fuse word 109 (FUSE109)	32	RO	0000_0000h
EE0h	Value of fuse word 110 (FUSE110)	32	RO	0000_0000h
EF0h	Value of fuse word 111 (FUSE111)	32	RO	0000_0000h
F00h	Value of fuse word 112 (FUSE112)	32	RO	0000_0000h
F10h	Value of fuse word 113 (FUSE113)	32	RO	0000_0000h
F20h	Value of fuse word 114 (FUSE114)	32	RO	0000_0000h
F30h	Value of fuse word 115 (FUSE115)	32	RO	0000_0000h
F40h	Value of fuse word 116 (FUSE116)	32	RO	0000_0000h
F50h	Value of fuse word 117 (FUSE117)	32	RO	0000_0000h
F60h	Value of fuse word 118 (FUSE118)	32	RO	0000_0000h
F70h	Value of fuse word 119 (FUSE119)	32	RO	0000_0000h
F80h	Value of fuse word 120 (FUSE120)	32	RO	0000_0000h
F90h	Value of fuse word 121 (FUSE121)	32	RO	0000_0000h
FA0h	Value of fuse word 122 (FUSE122)	32	RO	0000_0000h
FB0h	Value of fuse word 123 (FUSE123)	32	RO	0000_0000h
FC0h	Value of fuse word 124 (FUSE124)	32	RO	0000_0000h
FD0h	Value of fuse word 125 (FUSE125)	32	RO	0000_0000h
FE0h	Value of fuse word 126 (FUSE126)	32	RO	0000_0000h
FF0h	Value of fuse word 127 (FUSE127)	32	RO	0000_0000h
1000h	Value of fuse word 128 (FUSE128)	32	RO	0000_0000h
1010h	Value of fuse word 129 (FUSE129)	32	RO	0000_0000h
1020h	Value of fuse word 130 (FUSE130)	32	RO	0000_0000h
1030h	Value of fuse word 131 (FUSE131)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1040h	Value of fuse word 132 (FUSE132)	32	RO	0000_0000h
1050h	Value of fuse word 133 (FUSE133)	32	RO	0000_0000h
1060h	Value of fuse word 134 (FUSE134)	32	RO	0000_0000h
1070h	Value of fuse word 135 (FUSE135)	32	RO	0000_0000h
1080h	Value of fuse word 136 (FUSE136)	32	RO	0000_0000h
1090h	Value of fuse word 137 (FUSE137)	32	RO	0000_0000h
10A0h	Value of fuse word 138 (FUSE138)	32	RO	0000_0000h
10B0h	Value of fuse word 139 (FUSE139)	32	RO	0000_0000h
10C0h	Value of fuse word 140 (FUSE140)	32	RO	0000_0000h
10D0h	Value of fuse word 141 (FUSE141)	32	RO	0000_0000h
10E0h	Value of fuse word 142 (FUSE142)	32	RO	0000_0000h
10F0h	Value of fuse word 143 (FUSE143)	32	RO	0000_0000h

27.6.1.2 OTP Controller Control and Status Register (CTRL)

27.6.1.2.1 Offset

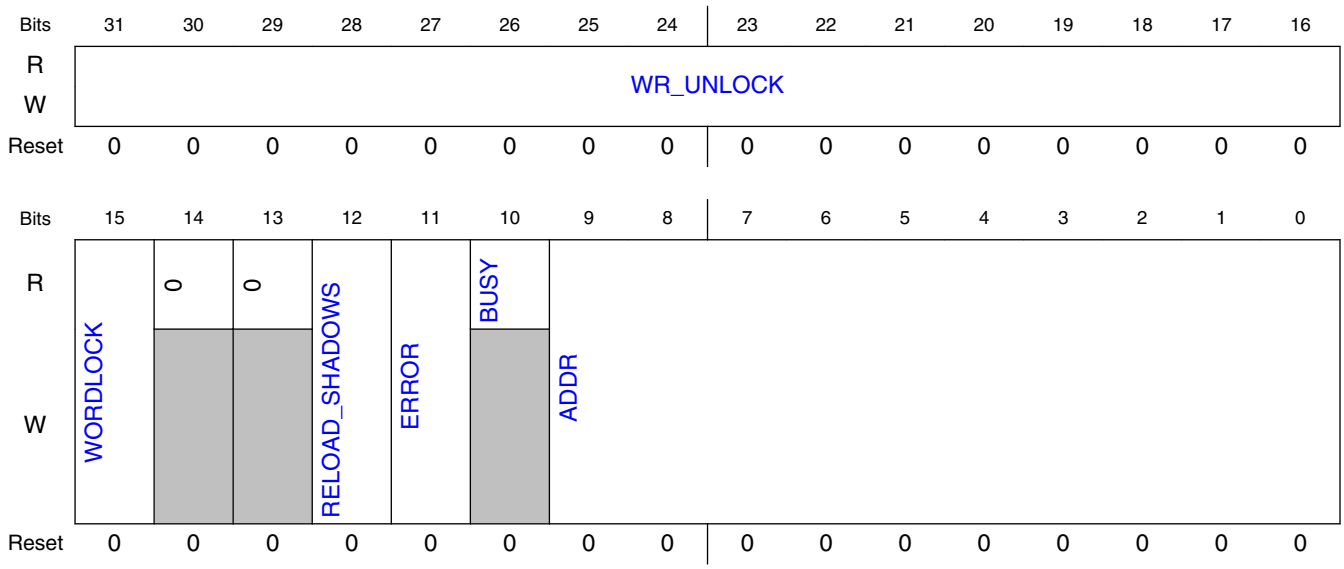
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	OTP Controller Control and Status Register
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

27.6.1.2.2 Function

The OCOTP Control and Status Register provides the necessary software interface for performing read and write operations to the On-Chip OTP. The control fields such as WR_UNLOCK, ADDR, BUSY, and ERROR may be used in conjunction with the DATA register to perform write operations. Read operations to the On-Chip OTP involve ADDR, BUSY, and ERROR bit fields and the READ_CTRL register. The read value is saved in the READ_FUSE_DATA register.

27.6.1.2.3 Diagram



27.6.1.2.4 Fields

Field	Function
31-16 WR_UNLOCK	Write unlock Write 0x3E77 to enable OTP write accesses. **NOTE:** This register must be unlocked on a write-by-write basis (a write is initiated when DATA is written), so the UNLOCK bit field must contain the correct key value during all writes to DATA, otherwise, a write shall not be initiated. This field is automatically cleared after a successful write completion (clearing of BUSY). Values not listed are reserved. 0000_0000_0000_0000b - OTP write access is locked. 0011_1110_0111_0111b - OTP write access is unlocked.
15 WORDLOCK	Lock fuse word Set this bit to 1 to lock fuse word whose programming mode is redundancy mode. **NOTE:** This bit has no effect on fuse words using ECC mode. ECC fuse words are automatically locked when they are programmed. 0b - No change to LOCK bit when programming a word using redundancy 1b - LOCK bit for fuse word will be set after successfully programming a word using redundancy
14 —	Reserved
13 —	Reserved
12	Reload Shadow Registers

Table continues on the next page...

Field	Function
RELOAD_SHADOWS	This bit can be set to force the re-loading of the shadow registers. This operation will automatically set BUSY. After the shadow registers have been re-loaded, BUSY and RELOAD_SHADOWS are automatically cleared by the controller. 0b - Do not force shadow register re-load. 1b - Force shadow register re-load. This bit is cleared automatically after shadow registers are re-loaded.
11 ERROR	Locked Region Access Error This bit is set by the controller when an access to a locked region (OTP or shadow register) is requested. It must be cleared before any further access can be performed. This bit can only be set by the controller. Reset this bit by writing a one to the CTRL_CLR[ERROR] bit and not by writing to the HW_OCOTP_CTRL register. 0b - No error. 1b - Error - access to a locked region requested.
10 BUSY	OTP controller status bit This bit is automatically set by the controller when a write or read access to the OTP is started. When set, no new write access or read access to OTP (including RELOAD_SHADOWS) can be performed. This bit is cleared by the controller when an access is completed. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller until the HW/SW and LOCK registers are successfully copied, after which time it is automatically cleared by the controller. 0b - No write or read access to OTP started. 1b - Write or read access to OTP started.
9-0 ADDR	OTP write and read access address register Specifies one of the 272 word address locations. There are 16 supplementary fuse words and 256 user fuse words. The ADDR value to be specified can be calculated as follows: ((Address that is to be programmed/read) - 0x800) >> 4 For example, to program or read address 0x960, ADDR value should be set to 0x16 ((0x960 - 0x800) >> 4). If a valid access is accepted by the controller, the controller makes an internal copy of this value. This internal copy will not be updated until the access is complete. Address values not specified below are invalid. 00_0000_0000b-00_0000_1111b - Address of one of the 16 supplementary fuse words in OTP memory. 00_0001_0000b-01_0000_1111b - Address of one of the 256 user fuse words in OTP memory.

27.6.1.3 OTP Controller PDN Register (PDN)

27.6.1.3.1 Offset

Register	Offset
PDN	10h

27.6.1.3.2 Function

This register specifies power control for the OTP fuse array.

27.6.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															PDN
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.3.4 Fields

Field	Function
31-1 —	Reserved
0 PDN	PDN value This bit indicates the PDN value of the OTP memory. If 1 is written to this bit, PDN value will be 0. Writing 0 to this bit will have no effect. **NOTE:** SW should write 1 to this bit to shut off power to the OTP memory, after system power up. At the beginning of every fuse operation, HW will auto enable power to OTP memory. After every fuse operation, SW should write 1 to this bit to shut off power to the OTP memory and avoid unnecessary power wasting. 0b - OTP memory is not powered 1b - OTP memory is powered

27.6.1.4 OTP Controller Write Data Register (DATA)

27.6.1.4.1 Offset

Register	Offset
DATA	20h

27.6.1.4.2 Function

This register is used in conjunction with the CTRL register to perform one-time writes to the OTP. See Fuse Writes for operating details.

27.6.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.4.4 Fields

Field	Function
31-0 DATA	Data Used to initiate a write to OTP. See Fuse Writes for operating details.

27.6.1.5 OTP Controller Read Control Register (READ_CTRL)

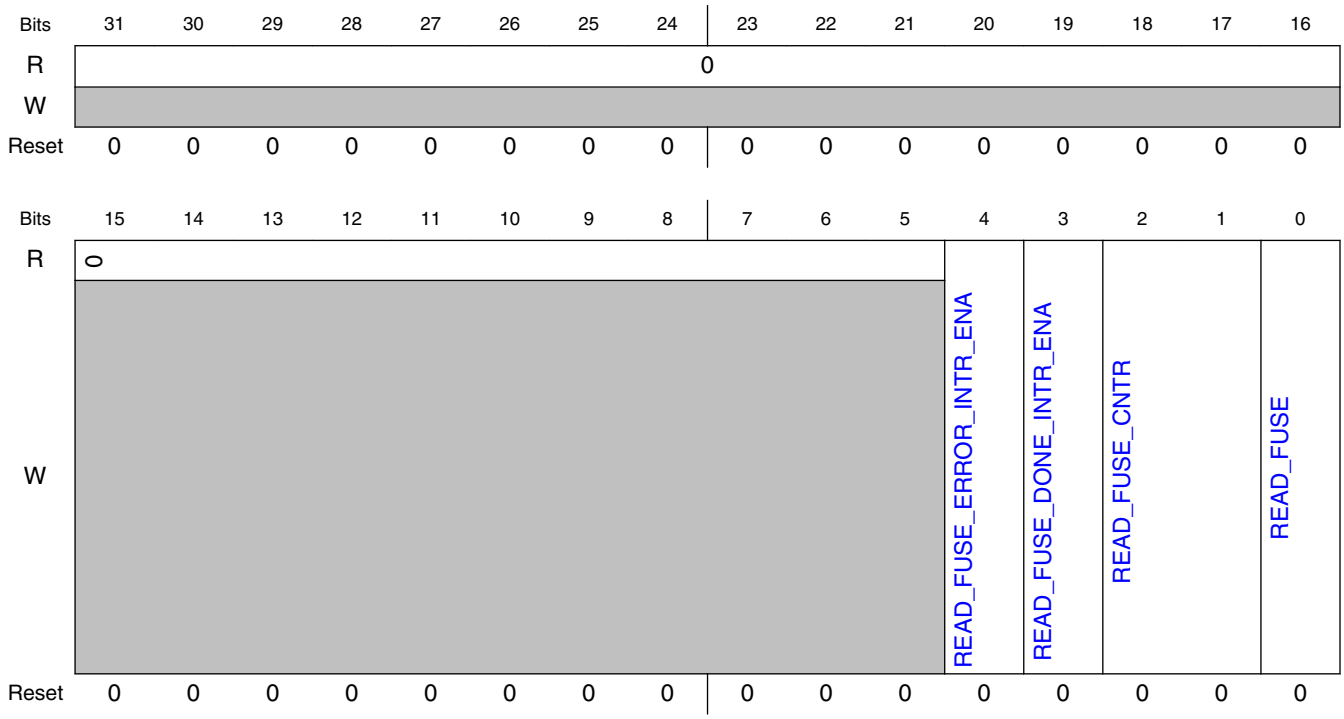
27.6.1.5.1 Offset

Register	Offset
READ_CTRL	30h

27.6.1.5.2 Function

This register is used in conjunction with the CTRL register to perform one time reads to the OTP. See Fuse and Shadow Register Read for operating details.

27.6.1.5.3 Diagram



27.6.1.5.4 Fields

Field	Function
31-5 —	Reserved
4 READ_FUSE_ERROR_INTR_ENA	Enable read-error interrupt 0b - Disable 1b - Enable
3 READ_FUSE_DONE_INTR_ENA	Enable read-done interrupt 0b - Disable 1b - Enable
2-1 READ_FUSE_CNTR	Number of words to read. 00b - 1 word 01b - 2 words 10b - 3 words 11b - 4 words
0 READ_FUSE	Read Fuse Used to initiate a read from OTP. See Fuse and Shadow Register Read for operating details. 0b - Do not initiate a read from OTP 1b - Initiate a read from OTP

27.6.1.6 8K OTP Memory STATUS Register (OUT_STATUS)

27.6.1.6.1 Offset

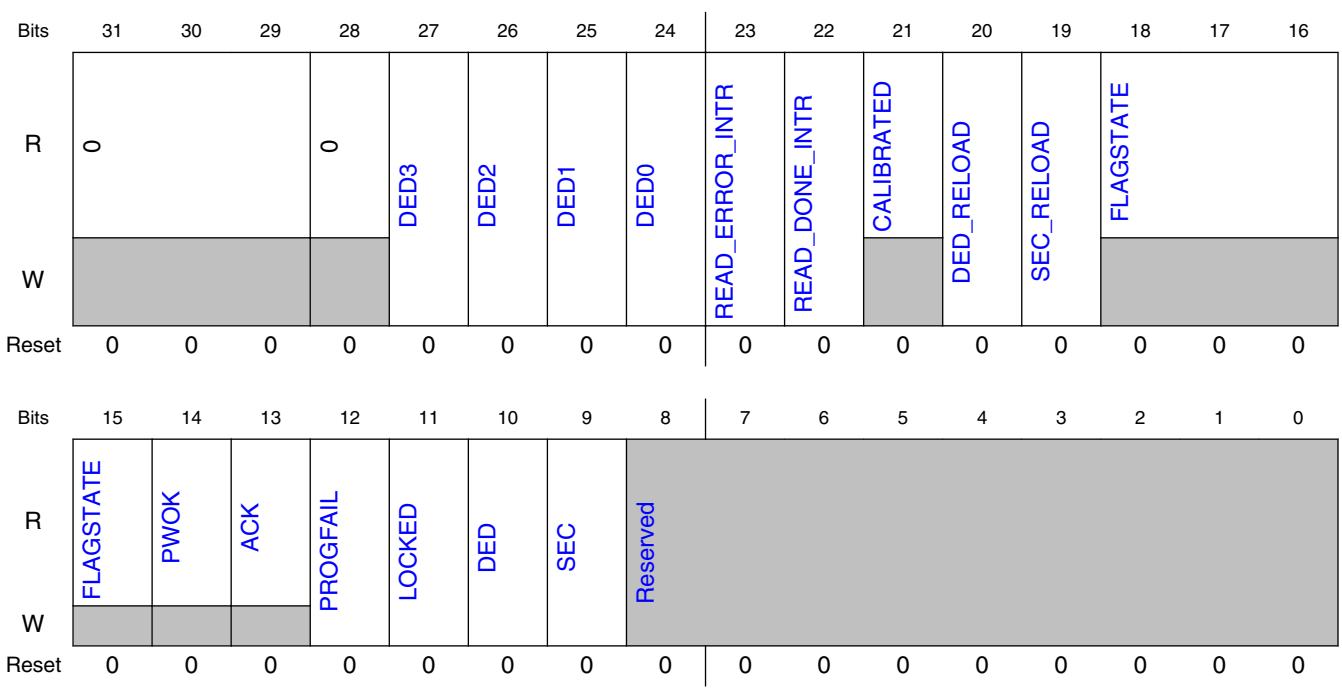
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
OUT_STATUS	90h	8K OTP Memory STATUS Register
OUT_STATUS_SET	94h	Writing 1 to a bit in this register ensures that the corresponding bit in OUT_STATUS is 1
OUT_STATUS_CLR	98h	Writing 1 to a bit in this register ensures that the corresponding bit in OUT_STATUS is 0
OUT_STATUS_TOG	9Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in OUT_STATUS

27.6.1.6.2 Function

This register stores output signals of OTP memory for programming and reading OTP memory.

27.6.1.6.3 Diagram



27.6.1.6.4 Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27 DED3	Double error detect Indicates double error has been detected (no correction) when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
26 DED2	Double error detect Indicates double error has been detected (no correction) when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
25 DED1	Double error detect Indicates double error has been detected (no correction) when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
24 DED0	Double error detect Indicates double error has been detected (no correction) when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
23 READ_ERROR_INTR	Fuse read error indicates that error occurred during the read fuse operation. **NOTE:** When READ_ERROR_INTR is set to 1, it means read operation finished with an error, while READ_DONE_INTR would keep this bit 0. 0b - Read operation finished with out any error 1b - Read operation finished with an error
22 READ_DONE_INTR	Read fuse done Indicates read fuse operation was done successfully (no error occurred)
21 CALIBRATED	Calibrated status Indicates OTP memory calibrated status.
20 DED_RELOAD	Indicates double error detection occured on reload Indicates double error detection ever happens in reset reload or reload by setting CTRL[RELOAD_SHADOWS] bit. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
19 SEC_RELOAD	Indicates single error correction occured on reload Indicates single error correction ever happens in reset reload or reload by setting CTRL[RELOAD_SHADOWS] bit. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
18-15 FLAGSTATE	Flag state Indicates OTP memory current state.
14	Power OK

Table continues on the next page...

Field	Function
PWOK	Indicates that power vdd and vdd1 of OTP memory are in operating range.
13 ACK	Acknowledge Indicates operation acknowledgement from OTP memory
12 PROGFAIL	Programming failed Indicates that programming failed. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
11 LOCKED	Word Locked Indicates if a word is locked or not when programming or reading. If this bit is asserted when programming one fuse word, it means that the current programming operation has failed since the fuse word is already locked. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
10 DED	Double error detect Indicates that 2 errors have been detected (no correction) when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
9 SEC	Single Error Correct Indicates that one error has been corrected in data or ECC syndrome when reading one fuse word. This bit can only be set by the controller. Reset this bit by writing a one to the SCT clear address space and not by a general write.
8-0 —	Reserved

27.6.1.7 OTP Controller Version Register (VERSION)

27.6.1.7.1 Offset

Register	Offset
VERSION	B0h

27.6.1.7.2 Function

This register indicates the RTL version in use.

27.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.7.4 Fields

Field	Function
31-24	Major RTL Version
MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23-16	Minor RTL Version
MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
15-0	RTL Version Stepping
STEP	Fixed read-only value reflecting the stepping of the RTL version.

27.6.1.8 OTP Controller Read Data 0 Register (READ_FUSE_DATA0)

27.6.1.8.1 Offset

Register	Offset
READ_FUSE_DATA0	100h

27.6.1.8.2 Function

Read data from fuse word 0.

27.6.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.8.4 Fields

Field	Function
31-0 DATA	Data The data read from the fuse word 0. See Fuse and Shadow Register Read for operating details.

27.6.1.9 OTP Controller Read Data 1 Register (READ_FUSE_DATA1)

27.6.1.9.1 Offset

Register	Offset
READ_FUSE_DATA1	110h

27.6.1.9.2 Function

Read data from fuse word 1.

27.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.9.4 Fields

Field	Function
31-0 DATA	Data The data read from the fuse word 1. See Fuse and Shadow Register Read for operating details.

27.6.1.10 OTP Controller Read Data 2 Register (READ_FUSE_DATA2)

27.6.1.10.1 Offset

Register	Offset
READ_FUSE_DATA2	120h

27.6.1.10.2 Function

Read data from fuse word 2.

27.6.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.10.4 Fields

Field	Function
31-0 DATA	Data The data read from the fuse word 2. See Fuse and Shadow Register Read for operating details.

27.6.1.11 OTP Controller Read Data 3 Register (READ_FUSE_DATA3)

27.6.1.11.1 Offset

Register	Offset
READ_FUSE_DATA3	130h

27.6.1.11.2 Function

Read data from fuse word 3.

27.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.11.4 Fields

Field	Function
31-0 DATA	Data The data read from the fuse word 3. See Fuse and Shadow Register Read for operating details.

27.6.1.12 SW_LOCK Register (SW_LOCK)

27.6.1.12.1 Offset

Register	Offset
SW_LOCK	140h

27.6.1.12.2 Function

The SW_LOCK register contains lock information, which has the same function as the RLOCK fuse words (supplementary fuse words 8 (0x880) and 9 (0x890)) in fuse memory. Refer to the Fusemap chapter for information on the RLOCK fuse words.

27.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_LOCK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_LOCK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.12.4 Fields

Field	Function
31-0 SW_LOCK	This register contains lock information, which has the same function as the RLOCK fuse words (supplementary fuse words 8 (0x880) and 9 (0x890)) in fuse memory. Refer to the Fusemap chapter for information on the function of RLOCK fuse words. Each bit is ORed with RLOCK to control it's corresponding region. The reset value of this register is 32'h0. After reset is de-asserted, '1' can only be written to each bit of this register once. The value of the bit cannot go back to '0' until the next reset is asserted.

27.6.1.13 BIT_LOCK Register (BIT_LOCK)

27.6.1.13.1 Offset

Register	Offset
BIT_LOCK	150h

27.6.1.13.2 Function

Write lock for supplementary fuse word 13 (0x8D0) and it's shadow register.

27.6.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BIT_LOCK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BIT_LOCK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.13.4 Fields

Field	Function
31-0 BIT_LOCK	Each bit controls the corresponding bit in supplementary fuse word 13 and its shadow register. If one bit is 1, the corresponding bit cannot be written. The reset value of this register is 32'h0. After reset is de-asserted, '1' can only be written to each bit of this register once. The value of the bit cannot go back to '0' until the next reset is asserted.

27.6.1.14 OTP Controller Program Locked Status 0 Register (LOCKED0)

27.6.1.14.1 Offset

Register	Offset
LOCKED0	600h

27.6.1.14.2 Function

This register stores program locked status for fuse words 0-15

27.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKED															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.14.4 Fields

Field	Function
31-16 —	Reserved
15-0 LOCKED	Stores program locked status for fuse words 0-15.

27.6.1.15 OTP Controller Program Locked Status 1 Register (LOCKED1)

27.6.1.15.1 Offset

Register	Offset
LOCKED1	610h

27.6.1.15.2 Function

This register stores program locked status for fuse words 16-47

27.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCKED															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKED															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.15.4 Fields

Field	Function
31-0 LOCKED	Stores program locked status for fuse words 16-47

27.6.1.16 Value of fuse word a (FUSE0 - FUSE143)

27.6.1.16.1 Offset

For a = 0 to 143:

Register	Offset
FUSEa	800h + (a × 10h)

27.6.1.16.2 Function

Shadowed memory mapped access to the fuse word.

27.6.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BITS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BITS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.6.1.16.4 Fields

Field	Function
31-0 BITS	Reflects value of the fuse word

Chapter 28
External Memory Controllers

28.1 Overview

This chip has these external memory interfaces and controllers:

- Smart External Memory Controller (SEMC)
- eSD/eMMC/SDIO Interface
- Flexible Serial Peripheral Interface (FlexSPI)

28.2 eMMC/eSD/SDIO

This chip has two Ultra Secured Digital Host Controller (uSDHC) modules for SD/eMMC interface. It provides the interface between the host system and the SD/SDIO/MMC cards.

The key features include:

- Support SD/SDIO standard, up to version 3.0
- Support MMC standard, up to version 5.0
- Support 3.3V and 1.8V operation.

28.3 Flexible Serial Peripheral Interface

There are two Flexible Serial Peripheral Interfaces - FlexSPI1 and FlexSPI2. Each FlexSPI host controller supports two SPI ports - A and B. Each port supports up two chip select outputs. Each port of FlexSPI2 supports Single/Dual/Quad/Octal mode data transfer (1/2/4/8 bidirectional data lines), while each port of FlexSPI1 supports Single/Dual/Quad mode data transfer (1/2/4 bidirectional data lines).

FlexSPI1 uses 4-bit bus with ports A and B, while FlexSPI2 uses 8-bit bus with ports A and B.

The key features includes:

- Support both SDR mode and DDR mode
- Support Individual/Parallel mode
- HyperBus device (HyperFlash/HyperRAM)

Chapter 29
Smart External Memory Controller (SEMC)

29.1 Chip-specific SEMC information

Table 29-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

29.2 Overview

The SEMC is a multi-standard memory controller optimized for both high-performance and low pin-count. It can support multiple external memories in the same application with shared address and data pins. The interfaces supported include SDRAM, NOR Flash, SRAM, NAND Flash, and 8080 display interface.

29.2.1 Block Diagram

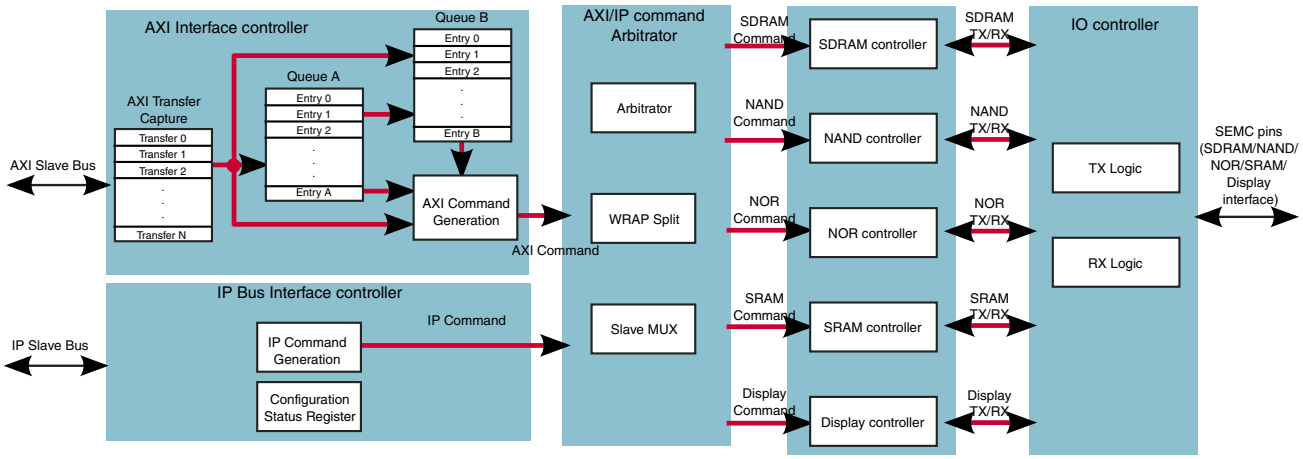


Figure 29-1. SEMC Block Diagram

29.2.2 Features

The SEMC includes the following features:

- Memory space:
  - Supports 12 memory regions:
    - Region #0 for SDRAM CS0 device
    - Region #1 for SDRAM CS1 device
    - Region #2 for SDRAM CS2 device
    - Region #3 for SDRAM CS3 device
    - Region #4 and #8 for NAND device (IP command and AXI command use different region to avoid large region address range on AXI interface)
    - Region #5 for NOR device
    - Region #6 for SRAM CS0 device
    - Region #7 for 8080 display frame buffer
    - Region #9 for SRAM CS1 device
    - Region #10 for SRAM CS2 device
    - Region #11 for SRAM CS3 device
  - Each region is configurable for base address, memory size and valid bit
- AXI slave interface
  - Supports 16 outstanding transfers (total including both reads and writes)
    - Supports 8 outstanding write transfers
    - Supports 16 outstanding read transfers
  - Multi-device access pattern optimization with Queue A

        - 8 entries command reordering in Queue A
        - Adjustable arbitration scheme based on QoS priority, latency and efficiency
    - SDRAM access pattern optimization with Queue B
        - 8 entries command reordering in Queue B
        - Adjustable arbitration scheme based on QoS priority, latency and efficiency
    - Memory-mapped AXI interface for read and write access
    - Supports 32/16/8 bit access
- IP Bus interface
    - Internal configuration register access
    - Trigger device access such as device initialization/configuration/read/write (IP command)
- SDRAM interface
    - Supports 8/16/32 bit modes
    - Supports 4 Chip Select (CS) and each CS up to 1024Mb
- NAND Flash interface
    - Supports 8/16 bit modes
    - No hardware ECC
    - Supports device which is capable of cache read/write operations
    - Supports page based read/write operation
    - Supports 32 bit transfer size for AXI write operation
    - Up to 512GB
- NOR Flash interface
    - Supports 8/16 bit modes
    - Supports ADMUX, AADM and Non-ADMUX modes
    - NOR Flash write access is supported only by IP command
    - Up to 4096Mb memory size

NOTE

- For 16 bit devices, up to 4096Mb memory size
- For 8 bit devices, up to 2048Mb memory size

- SRAM interface
    - Supports SRAM and Pseudo SRAM
    - Supports 8/16 bit modes
    - Supports ADMUX, AADM and Non-ADMUX modes
    - Up to 4 Chip Select (CS)
    - Up to 4096Mb memory size

NOTE

- For 16 bit devices, up to 4096Mb memory size
- For 8 bit devices, up to 2048Mb memory size

- 8080 display interface

    - Supports 8/16 bit modes
    - No tearing effect support
- Misc
    - Dynamic I/O sharing which enables multiple external memory device in parallel
    - Up to 7 Chip Select (CS) signals
    - Supports Address Latch Enable (ALE) for SRAM/NOR device

29.3 Functional Description

The following sections describe functional details of the SEMC module.

29.3.1 Operations

The following sections describe operations of the SEMC module.

29.3.1.1 Modes of Operation

The SEMC supports the following operation modes:

- Module Disable mode

  In module disable mode, the internal clock is gated for low power, but access to configuration and status registers is available.

  This mode is entered by setting MCR[MDIS], and exited by clearing MCR[MDIS]. Software should poll STS0[IDLE] to make sure there are no pending transactions before entering module disable mode.

- Stop mode

  When the system requires the SEMC to enter stop mode, the SEMC waits for all transactions to complete (STS0[IDLE]=1) and returns acknowledge handshake to system. The system can gate off the SEMC clock after the stop acknowledge handshake is returned. There is no clock gating within the SEMC module.

  The SEMC exits this mode immediately when the system stop mode request is negated.

- Normal mode

Access to external devices and internal registers are available in normal mode. The SEMC clock is not gated internally.

29.3.1.2 Device Access by AXI Command

Devices can be accessed by AXI bus access directly. AXI command is the device access sequence triggered by AXI bus access. The device access sequence for AXI commands is described in the following sections.

NOTE

- The AXI bus memory map for each device is defined by Base Registers.
- There is no software configuration or polling needed for AXI command.
- AXI command is the first choice to access device memory. IP command is suitable for other access, such as device register/OTP space access, device initialization, etc.
- IP command can be used to access memory like AXI command, but it is not recommended if AXI command is available. IP command access is low efficiency.

29.3.1.2.1 Bus Master Control Registers (BMCRn) Configuration

BMCRn can be used to balance external memory access efficiency, urgency and latency based on the requirements of a specific application.

29.3.1.2.1.1 BMCR0 Register Configuration

For Queue A controlled by BMCR0, the arbitration logic adopts a weight based algorithm that assigns each command in the reordering queue with a final score.

The arbitration logic calculates a SCORE for each command based on the formula below, where the weighting factor for each parameter is controlled by the BMCR0 configuration. The command with the highest SCORE is served first.

SCORE = QOS*WQOS + AGE*WAGE/4 + WSH + WRWS

1. QOS stands for AxQOS of AXI bus, and WQOS is the weight factor of QOS.
2. AGE stands for the wait period for each command in queue, and WAGE is the weight factor of AGE.
3. WSH stands for weight of slave hit without read/write switch scenario.
4. WRWS stands for weight of slave hit with read/write switch scenario.

Recommend to set BMCR0 with 0x0 for applications that require restrict sequence of transactions, such as stack is allocated in SDRAM. The write and read transactions to the same address might be reordered if transactions in the queue is reordered, and it might crash the routine. WQOS can be non-zero value if needed, meanwhile WAGE must be non-zero value as well. In any case, WBR, WRWS, and WPH should be 0x0.

29.3.1.2.1.2 BMCR1 Register Configuration

For Queue B controlled by BMCR1, the arbitration logic adopts a weight based algorithm that assigns each command in the reordering queue with a final score.

The arbitration logic calculates SCORE for each SDRAM command based on the formula below. The command with the highest SCORE is served first.

SCORE = QOS*WQOS + AGE*WAGE/4 + WPH + WBR + WRWS

1. QOS stands for AxQOS of AXI bus, and WQOS is the weight factor of QOS.
2. AGE stands for the wait period for each command in queue, and WAGE is the weight factor of AGE.
3. WPH stands for weight of page hit scenario.
4. WBR stands for weight of bank rotation/switch scenario.
5. WRWS stands for weight of slave hit without read/write switch scenario.

Recommend to set BMCR1 with 0x0 for applications that require restrict sequence of transactions, such as stack is allocated in SDRAM. The write and read transactions to the same address might be reordered if transactions in the queue is reordered, and it might crash the routine. WQOS can be non-zero value if needed, meanwhile WAGE must be non-zero value as well. Anyway, WRWS, WBR and WPH should be 0x0.

29.3.1.3 Device Access by IP Command

Devices can be accessed (initialized/configured/read/written) by IP bus access also. IP command is the device access sequence triggered by IP bus access. The device access sequence for IP command is described in the following sections.

Device access can be triggered by IP command using the following steps:

1. Set IPCR0/IPCR1/IPCR2/IPTXDAT registers for device address, data size, write mask and write data.
2. Set IPCMD register with valid command code (IPCMD[CMD]) and command key (IPCMD[KEY]).
3. Wait for IP command done (INTR[IPCMDDONE] sets).
4. Read the device data from IPRXDAT register. This is for read access only.

NOTE

- If device address or command code setting is not valid, an IP command error interrupt is generated (INTR[IPCMDERR]) and no device access occurs.
- If another IP command is triggered when the previous IP command has not finished yet, an IP command error interrupt is generated.
- IPCR0/IPCR1/IPCR2/IPTXDAT registers are protected when an IP command is in progress. Any write access to these registers is ignored until the IP command completes.
- The IPCR0 slave address field must be configured even if the memory does not require an address for the selected command (such as WRITE ENABLE command). The SEMC determines the slave device by decoding the device address.
- The slave device is limited to NAND when IPCMD[KEY] is 0x5AA5.

29.3.1.4 SDRAM Controller Operations

This section describes the operations of SDRAM controller.

29.3.1.4.1 SDRAM Address Multiplexing

SDRAMs use a multiplexed address split into rows, columns, and banks. Figure 29-2 shows how the bits of the linear address sent to the SEMC are split up into the row, column, and bank addresses sent to the SDRAM memory device.

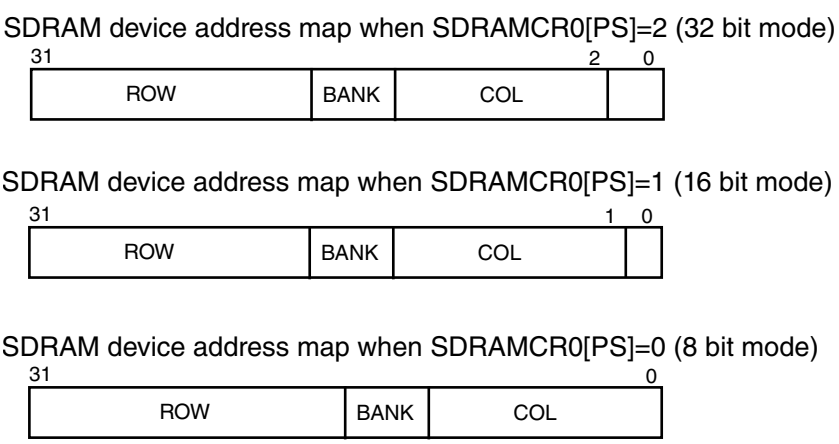


Figure 29-2. SDRAM Device Address Multiplexing

NOTE

- SDRAM address consists of row, bank and column address bits.
- Column address bits offset is bit 0 if SDRAMCR0[PS]=0, bit 1 if SDRAMCR0[PS]=1, bit 2 if SDRAMCR0[PS]=2. Column address bit width is determined by SDRAMCR0[COL] and SDRAMCR0[COL8].
- Bank address bits offset is determined by SDRAMCR0[PS], SDRAMCR0[COL] and SDRAMCR0[COL8]. Bank address bit width is determined by SDRAMCR0[BANK2].
- Row address bits offset is determined by SDRAMCR0[PS], SDRAMCR0[COL], SDRAMCR0[COL8] and SDRAMCR0[BANK2]. Row address bit width is determined by external SDRAM device.
- SDRAM device base address on SoC is configured by BR0, BR1, BR2 and BR3 registers.

29.3.1.4.2 SDRAM Access by IP Command

Table 29-2 lists the SDRAM access types that are triggered by IP commands.

Table 29-2. SDRAM IP Command

Command	Code	Comment
EXTENDED MODE REGISTER SET	0x5	Writes the extended mode register inside the SDRAM memory
DEEP POWER DOWN	0x6	Puts the SDRAM into deep power down mode
READ	0x8	Reads from the SDRAM memory
WRITE	0x9	Writes to the SDRAM memory
MODE REGISTER SET	0xA	Writes the mode register inside the SDRAM memory
ACTIVE	0xB	Sends a new bank and row address to the memory
AUTO REFRESH	0xC	Auto refresh
SELF REFRESH	0xD	Puts the SDRAM into low power data retention mode. The memory cannot be accessed while in self refresh mode.
PRECHARGE	0xE	Closes the currently active row for the specified bank
PRECHARGE ALL	0xF	Closes the active rows for all banks

29.3.1.4.2.1 READ

When an IP command is triggered with command code - READ, the SEMC will send a READ command to the SDRAM device.

Figure 29-3 shows the sequence for IP command - READ.

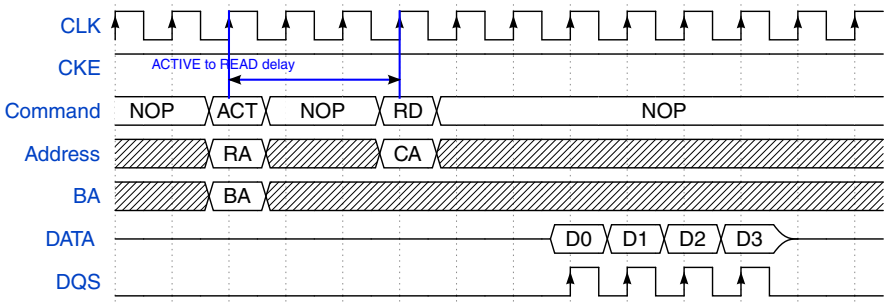


Figure 29-3. IP Command - SDRAM READ

NOTE

- It is supported but not recommended to send READ command with IP command trigger. The SEMC should read SDRAM device by AXI command.
- Read address is from IPCR0 register, read data is sent to IPRXDAT register.
- For this example:
  - ACTIVE to READ/WRITE delay is 3 (SDRAMCR1[ACT2RW]=2)

29.3.1.4.2.2 WRITE

When an IP command is triggered with command code - WRITE, the SEMC will send a WRITE command to the SDRAM device.

Figure 29-4 shows the sequence for IP command - WRITE.

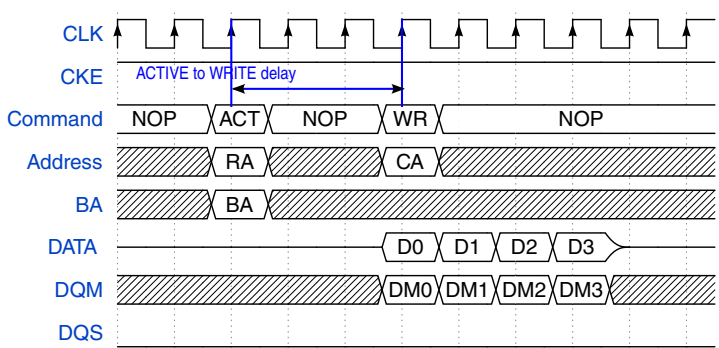


Figure 29-4. IP Command - SDRAM WRITE

NOTE

- It is supported but not recommended to send WRITE command with IP command trigger. The SEMC should write SDRAM device by AXI command.

- Write address is from IPCR0 register, write data is from IPTXDAT register, write mask is from IPCR2 register.
- For this example:
  - ACTIVE to READ/WRITE delay is 3 (SDRAMCR1[ACT2RW]=2)

29.3.1.4.2.3 MODE REGISTER SET

When an IP command is triggered with command code - MODE REGISTER SET, the SEMC will send a MODE REGISTER SET command to the SDRAM device.

Figure 29-5 shows the sequence for IP command - MODE REGISTER SET.

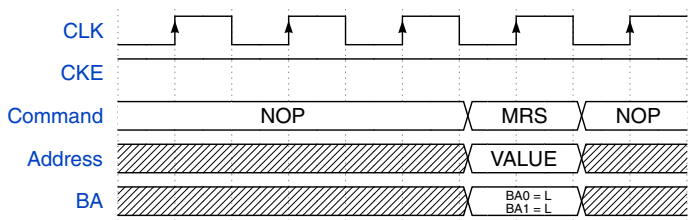


Figure 29-5. IP Command - SDRAM MODE REGISTER SET

NOTE

- MODE REGISTER SET command triggered by IP command should be used for SDRAM device initialization only.
- The "Value" of Address comes from IPTXDAT[12:0] register.

29.3.1.4.2.4 EXTENDED MODE REGISTER SET

When an IP command is triggered with command code - EXTENDED MODE REGISTER SET, the SEMC will send an EXTENDED MODE REGISTER SET command to the SDRAM device.

Figure 29-6 shows the sequence for IP command - EXTENDED MODE REGISTER SET.

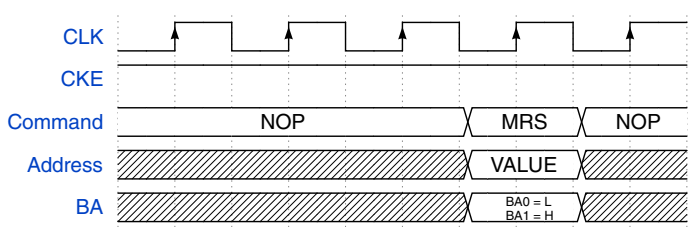


Figure 29-6. IP Command - EXTENDED SDRAM MODE REGISTER SET

NOTE

- EXTENDED MODE REGISTER SET command triggered by IP command should be used for SDRAM device initialization only.
- The "Value" of Address comes from IPTXDAT[12:0] register.

29.3.1.4.2.5 DEEP POWER DOWN

When an IP command is triggered with command code - DEEP POWER DOWN, the SEMC will send a DEEP POWER DOWN command to the SDRAM device.

Figure 29-7 shows the sequence for IP command - DEEP POWER DOWN.

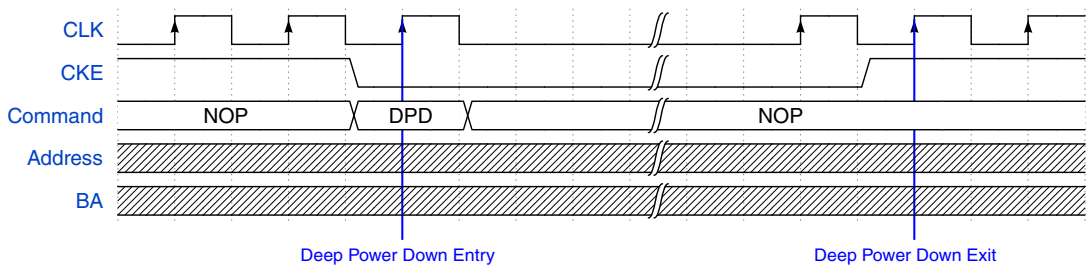


Figure 29-7. IP Command - DEEP POWER DOWN

NOTE

- The SEMC sends PRECHARGE ALL command to close any opened page on device before sending DEEP POWER DOWN command. There is no PRECHARGE ALL command if no pages on the selected device are open.
- Deep Power Mode is exited by any IP or AXI command.

29.3.1.4.2.6 ACTIVE

When an IP command is triggered with command code - ACTIVE, the SEMC will send an ACTIVE command to the SDRAM device.

Figure 29-8 shows the sequence for IP command - ACTIVE.

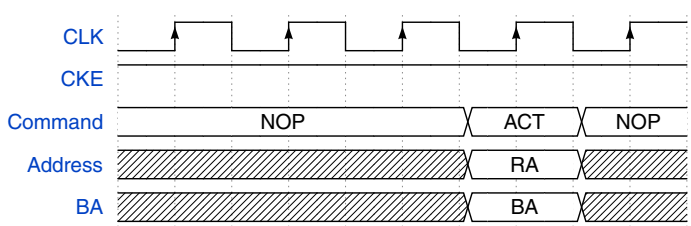


Figure 29-8. IP Command - SDRAM ACTIVE

NOTE

- It is supported but not recommended to send ACTIVE command with an IP command trigger. The SEMC can handle page management for READ/WRITE/AUTO REFRESH/SELF REFRESH etc.

29.3.1.4.2.7 AUTO REFRESH

When an IP command is triggered with command code - AUTO REFRESH, the SEMC will send an AUTO REFRESH command to the SDRAM device.

Figure 29-9 shows the sequence for IP command - AUTO REFRESH.

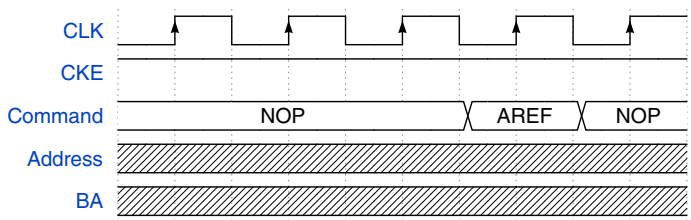


Figure 29-9. IP Command - SDRAM AUTO REFRESH

NOTE

- There is only one AUTO REFRESH command sent when triggered by IP command and only one device is accessed.
- The SEMC sends PRECHARGE ALL command to close any opened page on device before sending AUTO REFRESH command. There is no PRECHARGE ALL command if no pages on the selected device are open.
- AUTO REFRESH command triggered by IP command should be used for SDRAM device initialization only. For entering auto refresh during normal operation, set SDRAMCR3[REN] to 1 and the SEMC will send AUTO REFRESH command automatically.

29.3.1.4.2.8 SELF REFRESH

When a SELF REFRESH IP command is requested for any SDRAM device, the SEMC will send a SELF REFRESH command to all SDRAM devices. After the SELF REFRESH command is completed, all SDRAM devices should be in self refresh mode and the SEMC keeps this mode on SDRAM interface (SEMC_CKE=0 and SEMC_CLK=0).

The SEMC brings all SDRAM devices out of self refresh mode when there is any new IP command or AXI command triggered to an SDRAM device.

Figure 29-10 shows the sequence for self refresh entering and exiting by AXI command.

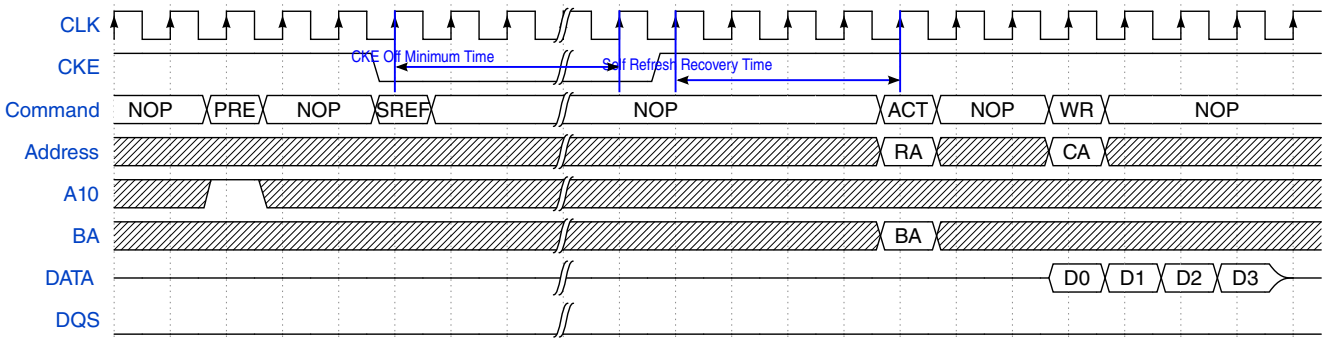


Figure 29-10. IP Command - SDRAM SELF REFRESH

NOTE

- The SEMC sends PRECHARGE ALL command to close any opened page on device before entering self refresh mode. There is no PRECHARGE ALL command if no page is opened at any device.
- SELF REFRESH command sends to all devices.
- It is not recommended to put device into self refresh mode by IP command triggering although it is supported. Recommend to put device into self refresh mode by the SEMC STOP mode instead.
- Disable the SEMC auto refresh enable bit (SDRAMCR3[REN]=0) before entering self refresh mode, otherwise the SEMC may bring devices out of self refresh mode.
- For this example:
  - Self refresh recovery time is 4 clock cycles (SDRAMCR2[SRRC]=3)
  - CKE off minimum time is 4 clock cycles (SDRAMCR1[CKEOFF]=3)
  - The SEMC brings SDRAM out of self refresh mode when an AXI write command is triggered.

29.3.1.4.2.9 PRECHARGE

When an IP command is triggered with command code - PRECHARGE, The SEMC will send a PRECHARGE command to the SDRAM device.

Figure 29-11 shows the sequence for IP command - PRECHARGE.

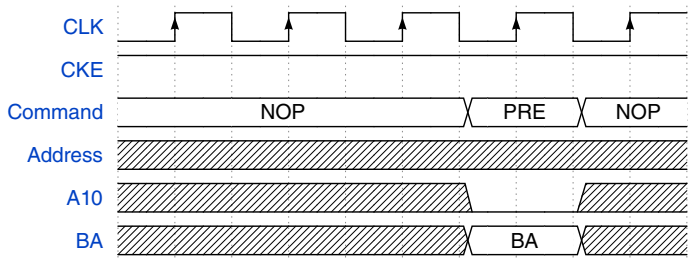


Figure 29-11. IP Command - SDRAM PRECHARGE

NOTE

- It is not recommended to send PRECHARGE command triggered by IP command although it is supported. The SEMC can handle page management for READ/WRITE/ AUTO REFRESH/SELF REFRESH etc.

29.3.1.4.2.10 PRECHARGE ALL

When an IP command is triggered with command code - PRECHARGE ALL, the SEMC will send a PRECHARGE ALL command to the SDRAM device.

Figure 29-12 shows the sequence for IP command - PRECHARGE ALL.

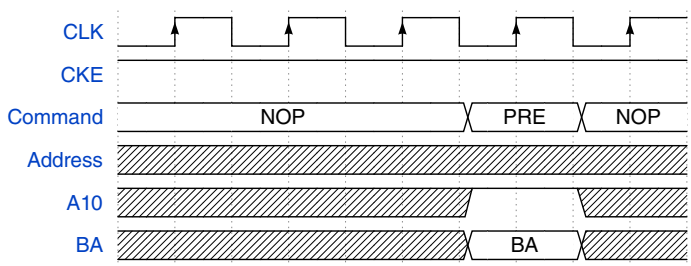


Figure 29-12. IP Command - SDRAM PRECHARGE ALL

NOTE

- It is not recommended to send PRECHARGE ALL command triggered by IP command although it is supported. The SEMC can handle page management for READ/WRITE/AUTO REFRESH/SELF REFRESH etc.
- PRECHARGE ALL command triggered by IP command should be used for SDRAM device initialization only.

29.3.1.4.2.11 Combination of IP Commands

Figure 29-13 shows a combination sequence of IP commands, that includes AUTO REFRESH, ACTIVE, WRITE and PRECHARGE IP commands.

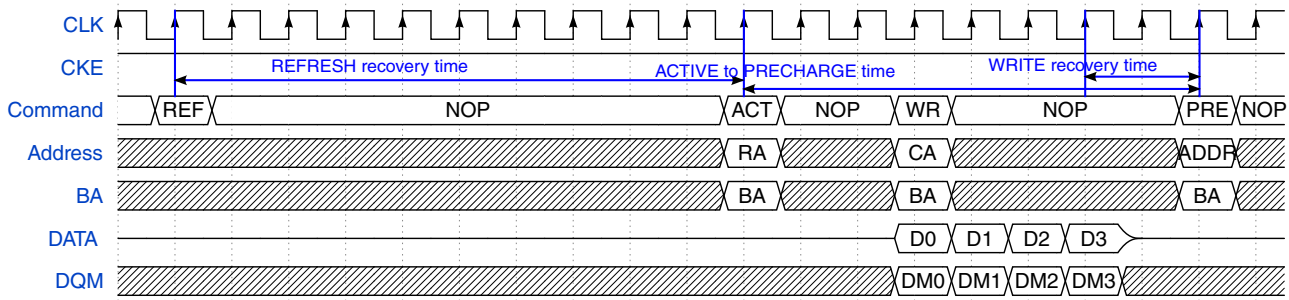


Figure 29-13. IP Command - AUTO REFRESH, ACTIVE, WRITE and PRECHARGE

NOTE

- For this example:
  - REFRESH recovery time is 10 clock cycles (SDRAMCR1[RFRC]=9)
  - WRITE recovery time is 2 clock cycles (SDRAMCR1[WRC]=1)
  - ACTIVE to PRECHARGE minimum time is 6 clock cycles (SDRAMCR1[ACT2PRE]=5). Current ACTIVE to PRECHARGE time is 8 clock cycles by the limitation of WRITE burst beats and recovery time.

29.3.1.4.3 SDRAM Device Access by AXI Command

SDRAM device read/write access can be triggered by AXI Command. For AXI read access, the SEMC will send READ commands to SDRAM device automatically and return data on AXI bus. For AXI write access, the SEMC will save AXI bus write data and send WRITE commands to the SDRAM device automatically.

29.3.1.4.3.1 Read Access

Figure 29-14 shows an SDRAM read triggered by an AXI command.

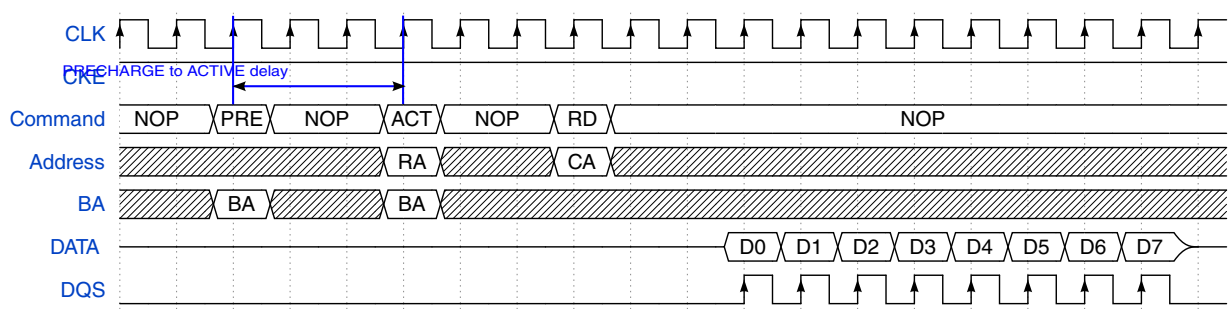


Figure 29-14. AXI Command - SDRAM READ

NOTE

- The SEMC opens/closes pages automatically for current AXI command.
- For this example:
  - CAS latency is 3 (SDRAMCR0[CL]=3)
  - Burst length is 8 (SDRAMCR0[BL]=3)
  - PRECHARGE to ACTIVE delay is 3 clock cycles (SDRAMCR1[PRE2ACT]=2)

29.3.1.4.3.2 Write Access

Figure 29-15 shows an SDRAM write triggered by an AXI command.

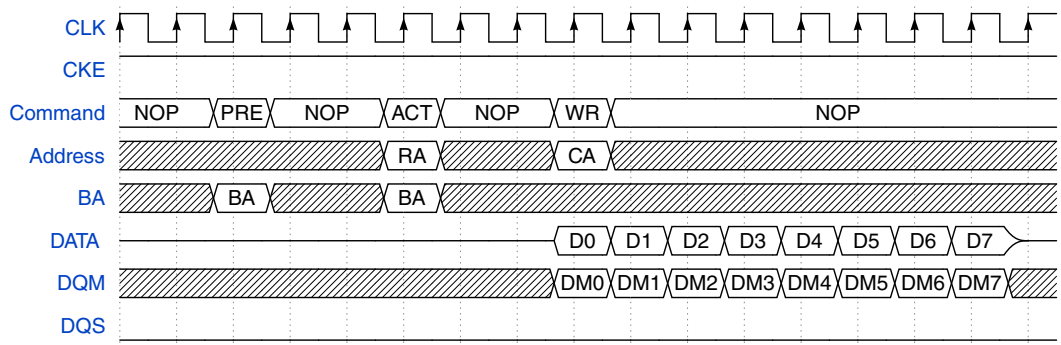


Figure 29-15. AXI Command - SDRAM WRITE

NOTE

- The SEMC opens/closes pages automatically for current AXI command.
- SDRAM device burst length is 8 in this example. The SEMC may send multiple WRITE commands if AXI burst size is larger than SDRAM burst length.

29.3.1.4.3.3 Pipelined Access

Figure 29-16 shows an SDRAM pipelined read access that is triggered by AXI bus read access.

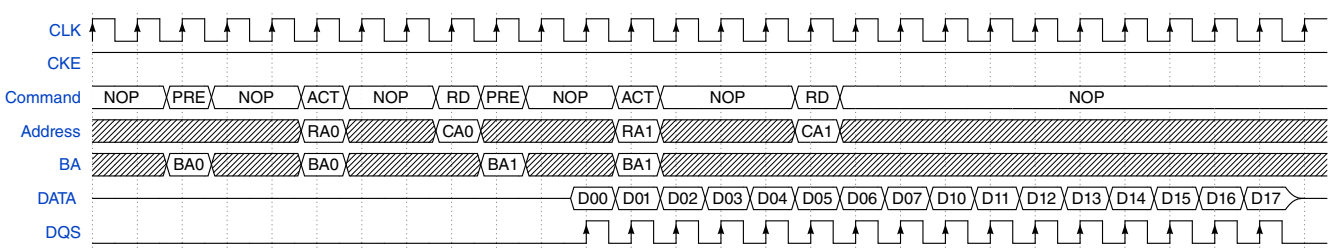


Figure 29-16. AXI Command - Pipelined SDRAM Read

NOTE

- The SEMC opens/closes pages automatically for next AXI command during SDRAM Read burst.
- Data D0x is from CA0, and data D1x is from CA1. This naming rule is also applied to the following SDRAM figures.

Figure 29-17 shows the SDRAM device pipelined write access that triggered by AXI bus write access.

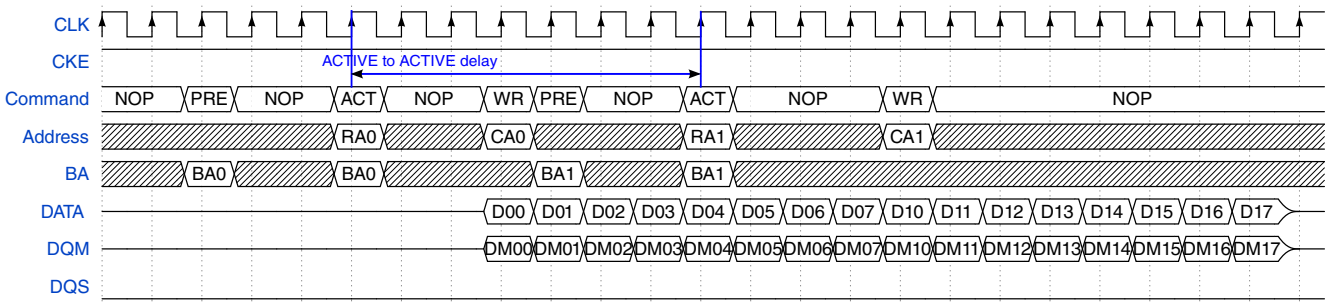


Figure 29-17. AXI Command - Pipelined SDRAM Write

NOTE

- The SEMC opens/closes pages automatically for next AXI command during SDRAM Write burst.

29.3.1.4.3.4 SDRAM Back-to-Back Access

This section describes the SDRAM back-to-back access.

29.3.1.4.3.4.1 Read-to-Read Access

Figure 29-18 shows SDRAM read-to-read access timing when CL=1.

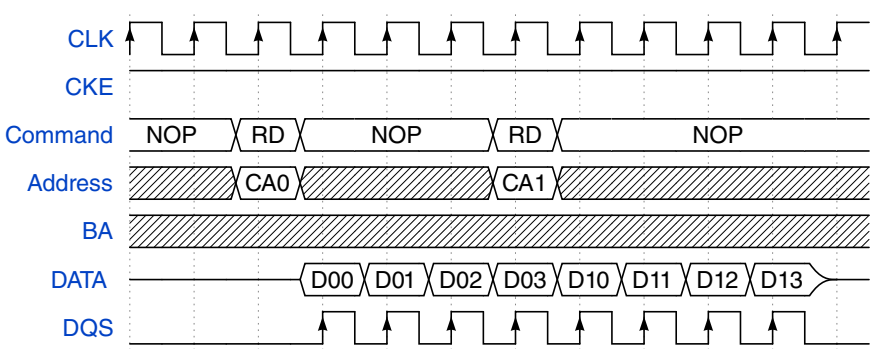


Figure 29-18. SDRAM Read-to-Read Access (CL=1)

Figure 29-19 shows SDRAM read-to-read access timing when CL=2.

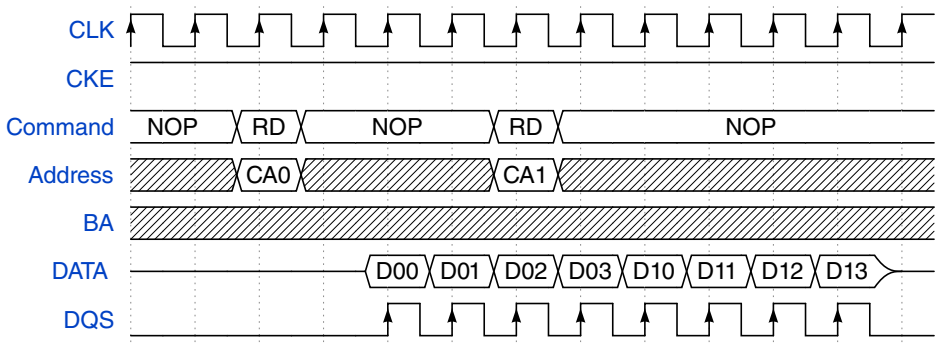


Figure 29-19. SDRAM Read-to-Read Access (CL=2)

Figure 29-20 shows SDRAM read-to-read access timing when CL=3.

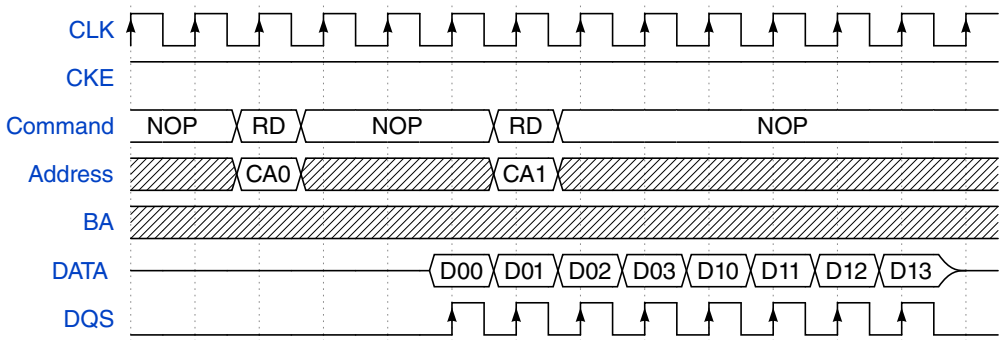


Figure 29-20. SDRAM Read-to-Read Access (CL=3)

29.3.1.4.3.4.2 Read-to-Write Access

Figure 29-21 shows SDRAM read-to-write access timing when CL=1.

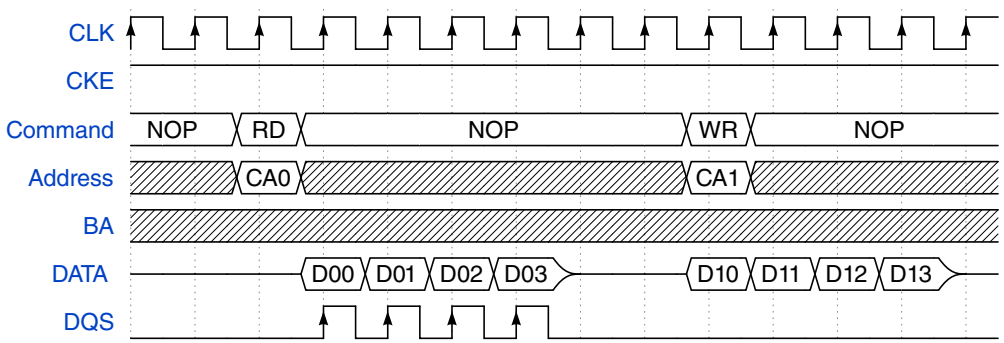


Figure 29-21. SDRAM Read-to-Write Access (CL=1)

Figure 29-22 shows SDRAM read-to-write access timing when CL=2.

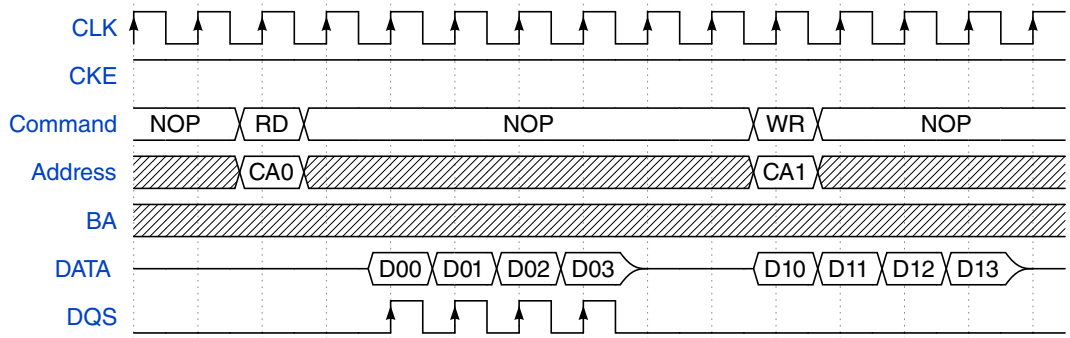


Figure 29-22. SDRAM Read-to-Write Access (CL=2)

Figure 29-23 shows SDRAM read-to-write access timing when CL=3.

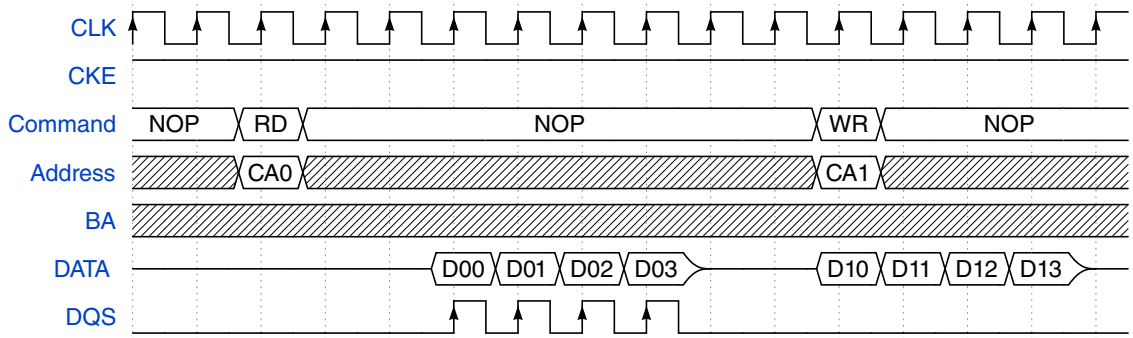


Figure 29-23. SDRAM Read-to-Write Access (CL=3)

29.3.1.4.3.4.3 Write-to-Write Access

Figure 29-24 shows SDRAM write-to-write access timing.

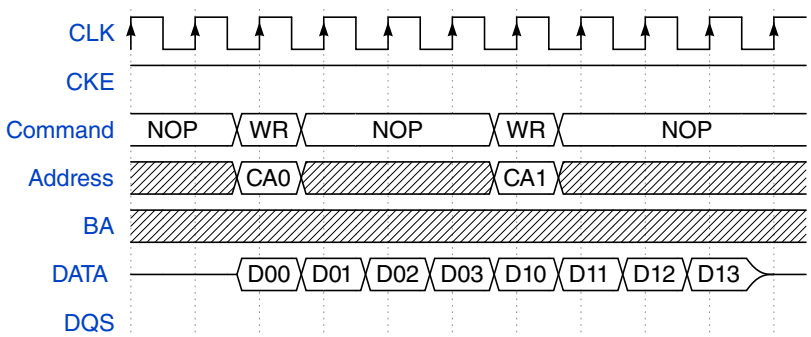


Figure 29-24. SDRAM Write-to-Write Access

29.3.1.4.3.4.4 Write-to-Read Access

Figure 29-25 shows SDRAM write-to-read access timing when CL=1.

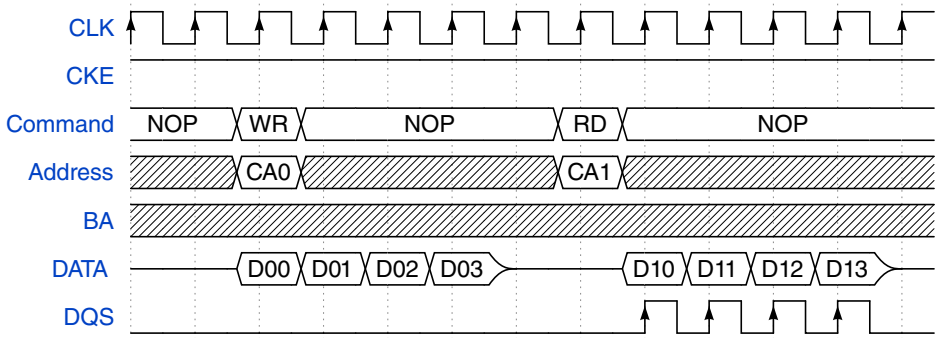


Figure 29-25. SDRAM Write-to-Read Access (CL=1)

Figure 29-26 shows SDRAM write-to-read access timing when CL=2.

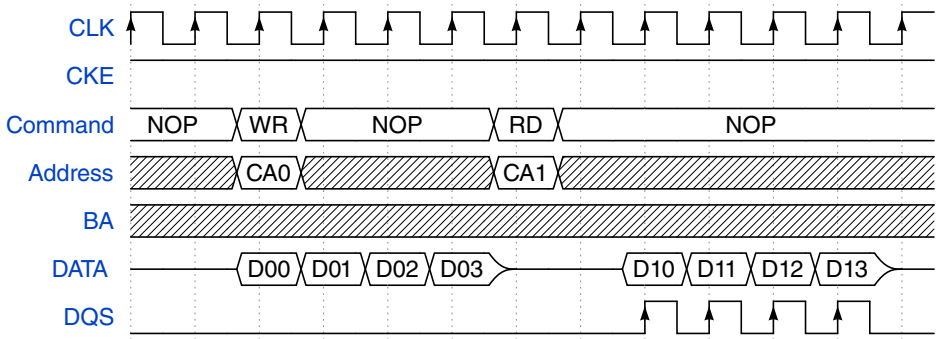


Figure 29-26. SDRAM Write-to-Read Access (CL=2)

Figure 29-27 shows SDRAM write-to-read access timing when CL=3.

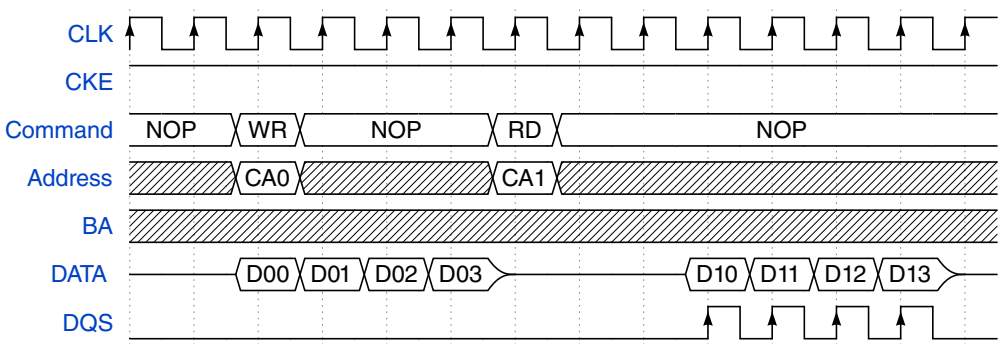


Figure 29-27. SDRAM Write-to-Read Access (CL=3)

29.3.1.4.4 Refresh Command - SDRAM AUTO REFRESH

When SDRAM refresh is enabled (SDRAMCR3[REN]=1), the SEMC will send an AUTO REFRESH command to all SDRAM devices whenever the refresh timer expires. Multiple AUTO REFRESH commands can be sent if refresh burst is not set to one (SDRAMCR3[REBL]!=0).

The refresh timer's count cycle is controlled by UT, RT and PRESCALE bits in SDRAMCR3.

Figure 29-28 shows the sequence of AUTO REFRESH Command.

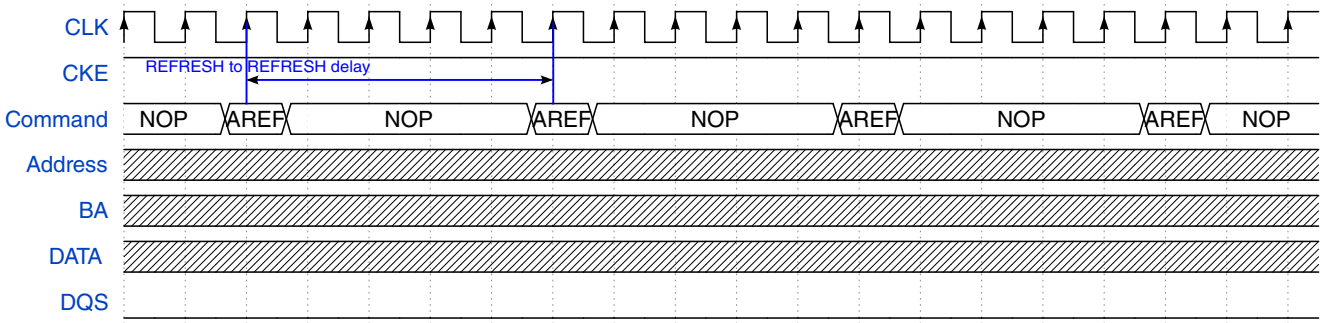


Figure 29-28. Refresh Command - SDRAM AUTO REFRESH

NOTE

- AUTO REFRESH command is sent to all devices.
- For this example:
  - Refresh to refresh delay is 5 clock cycles (SDRAMCR2[REF2REF]=4)
  - Refresh burst length is 4 (SDRAMCR3[REBL]=3)

29.3.1.4.5 SDRAM Low Power Feature

This section describes the low power feature that is implemented for SDRAM.

29.3.1.4.5.1 SDRAM Stop Mode

The SEMC puts SDRAM to self refresh mode automatically when the system enters STOP mode. The SEMC brings SDRAM out of self refresh mode when the system exits STOP mode.

Figure 29-29 shows the sequence of STOP mode entering and exiting.

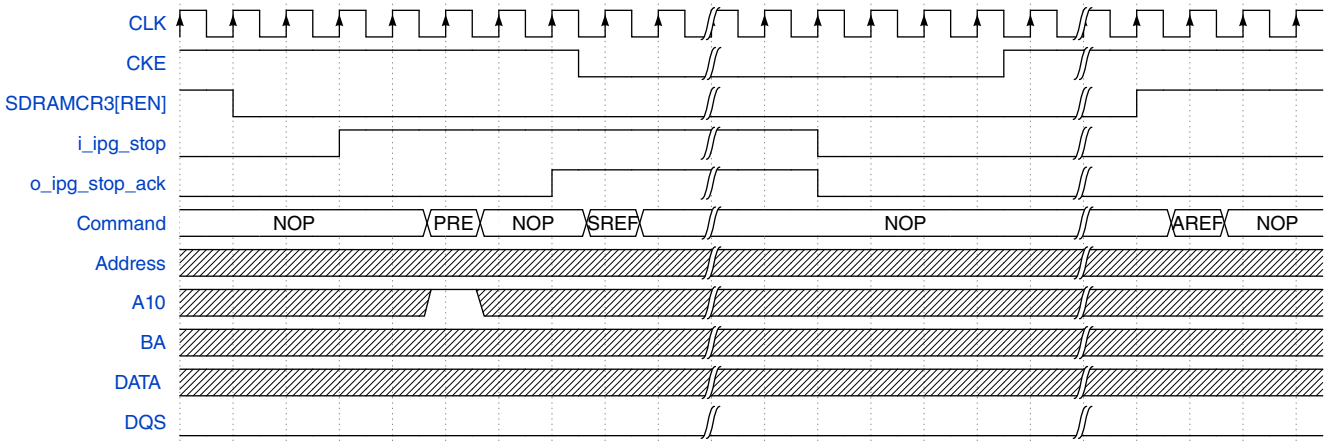


Figure 29-29. SDRAM Stop Mode Enter and Exit

NOTE

- The SEMC sends PRECHARGE ALL command to close any opened page on device before entering self refresh mode. There is no PRECHARGE ALL command if no page is opened at any device.
- SELF REFRESH command is sent to all devices.
- Disable the SEMC auto refresh bit (SDRAMCR3[REN]=0) before entering STOP mode, otherwise the SEMC may bring devices out of self refresh mode.

29.3.1.4.5.2 SDRAM Bus Idle

When there are no pending IP, AXI, or refresh commands, SDRAM interface will in the idle state. When the SDRAM interface bus idle timer expires, the SEMC closes all opened pages on all SDRAM devices. Refer to SDRAMCR2[ITO] for more information.

29.3.1.4.6 SDRAM Page Management

This section describes the SDRAM page management.

The SEMC will save row addresses for up to 8 opened pages: 4 pages for CS0/CS2 (4 banks), 4 pages for CS1/CS3 (4 banks). Device CS0 and CS2 pages are never opened at the same time. Before the SEMC opens pages on CS2, it closes all opened pages on CS0. This is similar for CS1/CS3.

The SEMC closes all pages on all CS by issuing a PRECHARGE ALL command in following cases:

1. SELF REFRESH triggered by IP command
2. SELF REFRESH triggered by stop mode entry

3. The idle time on AXI bus and SDRAM interface is longer than SDRAM idle timeout time (SDRAMCR2[ITO]).

NOTE

The SEMC will not send out PRECHARGE ALL command if there are no pages opened on any bank/CS.

The SEMC will close all pages on one CS in following cases:

1. PRECHARGE ALL command triggered by IP command
2. AUTO REFRESH command triggered by IP command
3. ACTIVE/WRITE/READ command triggered by IP command and CS miss
4. WRITE/READ command triggered by current AXI command and CS miss
5. WRITE/READ command triggered by next AXI command (not accepted yet), CS miss and not same CS as current AXI command

NOTE

One example for CS miss: current IP command access CS2 but there is page opened on CS0 already.

The SEMC will close one page on one CS in following cases:

1. PRECHARGE command triggered by IP command and this bank is opened
2. ACTIVE/READ/WRITE command triggered by IP command, this bank is opened and page missed.
3. WRITE/READ command triggered by current AXI command, this bank is opened and page missed.
4. WRITE/READ command triggered by next AXI command (not accepted yet), this bank is opened, page missed and different bank accessed with current AXI command.

NOTE

One example for page miss: current bank is opened but the opened page row address is different with current SDRAM command access.

The SEMC will open one page on one CS in following cases:

1. ACTIVE/READ/WRITE command triggered by IP command, this bank is not opened.
2. WRITE/READ command triggered by current AXI command, this bank is not opened.
3. WRITE/READ command triggered by next AXI command (not accepted yet), this bank is not opened and different bank accessed with current AXI command.

29.3.1.5 NAND Flash Controller Operations

This section describes the operations of NAND Flash controller.

29.3.1.5.1 NAND Flash Address Multiplexing

The NAND Flash controller uses a multiplexed address split into rows and columns. Figure 29-30 shows how the bits of the linear address sent to the SEMC are split up into the row and column addresses sent to the NAND Flash memory device.

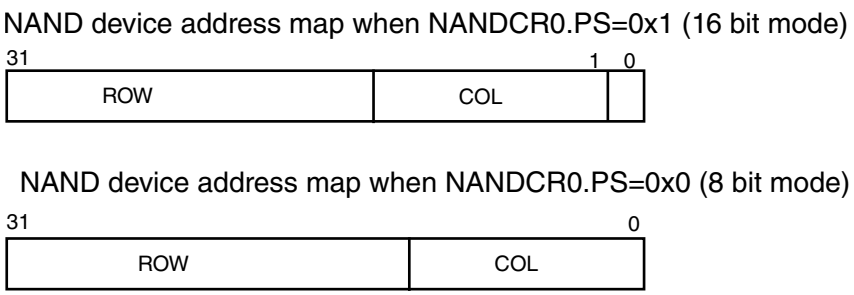


Figure 29-30. NAND Flash Address Multiplexing

NOTE

- NAND address bits consist of row and column address bits.
- Column address bits offset is bit 0 if NANDCR0[PS]=0, the offset is bit 1 if NANDCR0[PS]=1. Column address bit width is determined by NANDCR0[COL].
- Column bit number should cover all page size. For example, column bit number should be 14 if NAND Flash page size is 8192 + 448 (main + spare) bytes.
- Row address bits offset is determined by NANDCR0[PS] and NANDCR[COL].
- NAND device base address on SoC is configured by BR4 and BR8 registers.

29.3.1.5.2 NAND Flash Access Address Type

Table 29-3 lists the NAND address types that are triggered by IP commands.

Table 29-3. NAND Address Types

Address Type	IPCMD[7:4]
Five Bytes Row/Column Address	0x0
One Byte Column Address	0x1
Two Bytes Column Address	0x2
One Byte Row Address	0x3
Two Bytes Row Address	0x4
Three Bytes Row Address	0x5

For IP command, address type is determined by IPCMD register. For AXI command, address type is fixed to "Two Bytes Column Address" because AXI access is limited in opened NAND page.

29.3.1.5.2.1 Five Bytes Row/Column Address

Figure 29-31 shows the Five Bytes Row/Column Address type access timing.

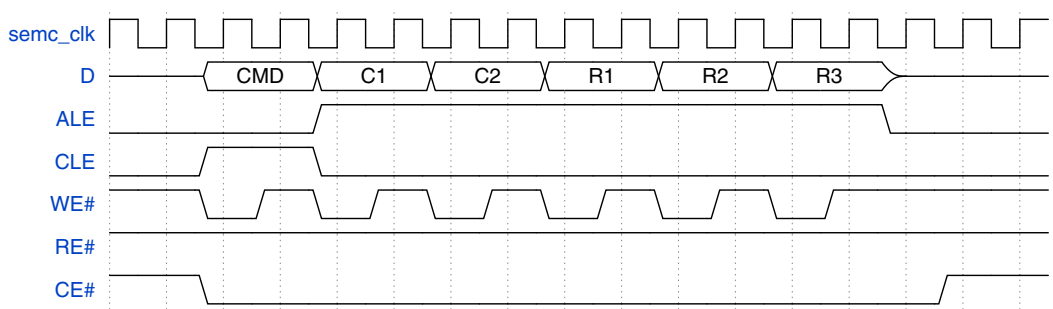


Figure 29-31. NAND Flash Access Address Type - Five Bytes Row/Column Address

There are some option bits to support different NAND address types in the Five Bytes Row/Column Address type. Table 29-4 lists all the combinations of address types.

Table 29-4. Address Bytes Order for five Bytes Row/Column Address Mode

NANDCR3[NDOPT3]	NANDCR3[NDOPT2]	NANDCR3[NDOPT1]	Address byte order
0	0	0	C1/C2/R1/R2/R3
0	0	1	C1/R1/R2/R3
0	1	0	C1/C2/R1/R2
0	1	1	C1/R1/R2
1	-	0	C1/C2/R1
1	-	1	C1/R1

29.3.1.5.2.2 One Byte Column Address

Figure 29-32 shows the One Byte Column Address type access timing.

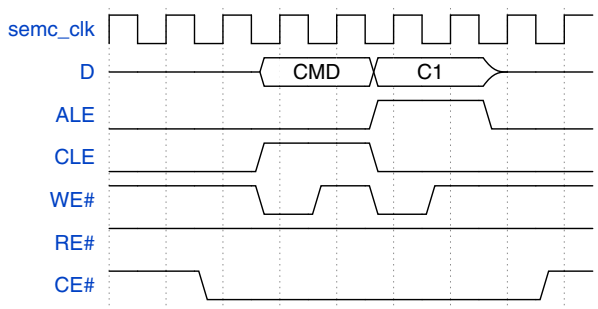


Figure 29-32. NAND Flash Access Address Type - One Byte Column Address

29.3.1.5.2.3 Two Bytes Column Address

Figure 29-33 shows the Two Bytes Column Address type access timing.

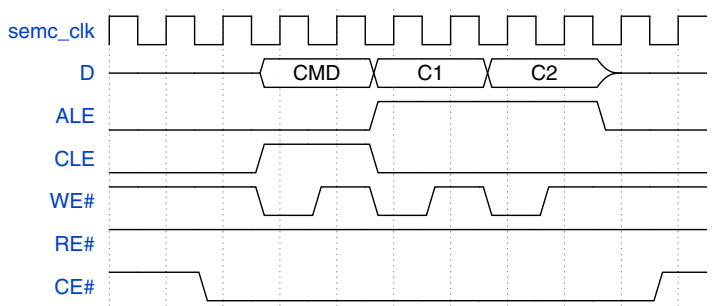


Figure 29-33. NAND Flash Access Address Type - Two Bytes Column Address

29.3.1.5.2.4 One Byte Row Address

Figure 29-34 shows the One Byte Row Address type access timing.

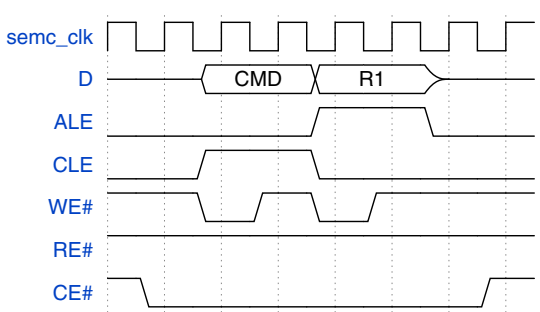


Figure 29-34. NAND Flash Access Address Type - One Byte Row Address

29.3.1.5.2.5 Two Bytes Row Address

Figure 29-35 shows the Two Bytes Row Address type access timing.

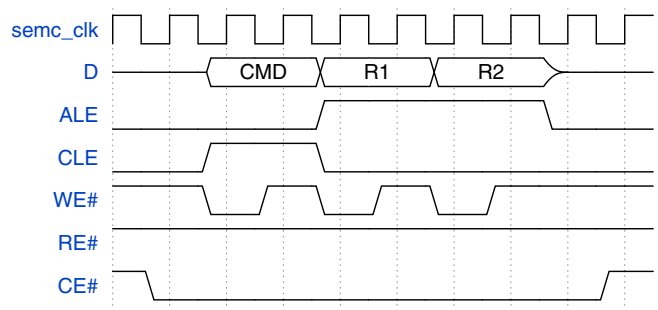


Figure 29-35. NAND Flash Access Address Type - Two Bytes Row Address

29.3.1.5.2.6 Three Bytes Row Address

Figure 29-36 shows the Three Bytes Row Address type access timing.

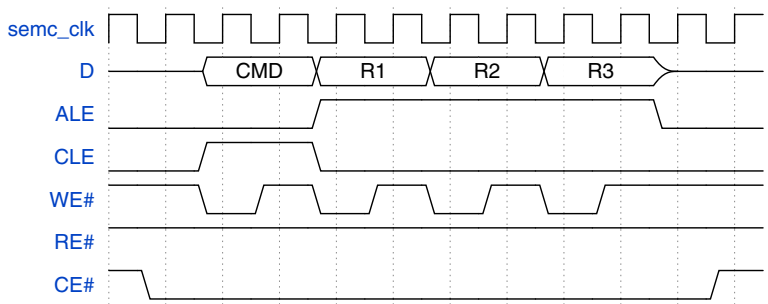


Figure 29-36. NAND Flash Access Address Type - Three Bytes Row Address

29.3.1.5.3 NAND Flash Access Command Type

NAND Flash access support following command types:

1. Command Phase
2. Command + Wait Phase
3. Command + Address Phase
4. Command + Address + Wait Phase
5. Command + Address + Read Phase
6. Command + Address + Write Phase (for IP command)
7. Command + Write Phase
8. Command + Read Phase
9. Write Phase
10. Read Phase
11. Command + Address + Command + Read Phase (for AXI command - Read)
12. Command + Address + Write Phase (for AXI command - Write)

29.3.1.5.3.1 Command Phase

Figure 29-37 shows the NAND Flash Command Phase access timing.

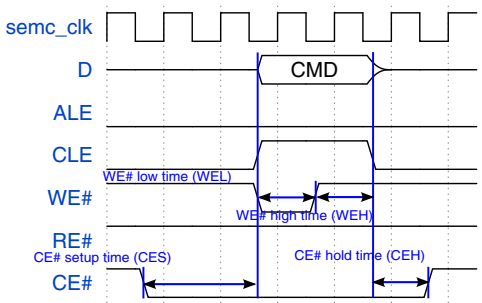


Figure 29-37. NAND Flash Access Command Type - Command Phase

NOTE

- semc_clk is the SEMC functional clock.
- For this example:
  - CE# setup time is 2 clock cycles (NANDCR1[CES]=1)
  - WE# low time is 1 clock cycle (NANDCR1[WEL]=0)
  - WE# high time is 1 clock cycle (NANDCR1[WEH]=0)
  - CE# hold time is 1 clock cycle (NANDCR1[CEH]=0)

29.3.1.5.3.2 Command + Wait Phase

Figure 29-38 shows NAND Flash Command + Wait Phase access timing

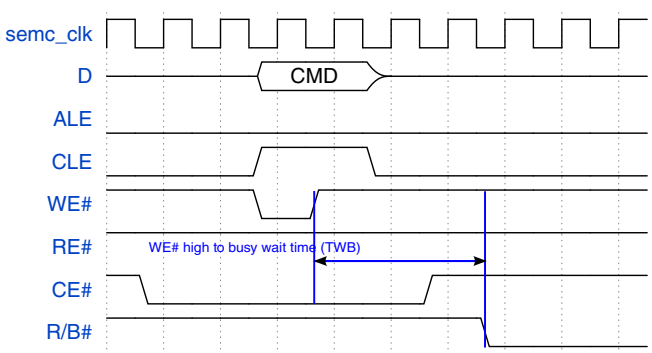


Figure 29-38. NAND Flash Access Command Type - Command + Wait Phase

NOTE

- For this example:
  - WE# high to busy time is 3 clock cycles (NANDCR1[TWB]=2)

29.3.1.5.3.3 Command + Address Phase

Figure 29-39 shows the NAND Flash Command + Address Phase access timing.

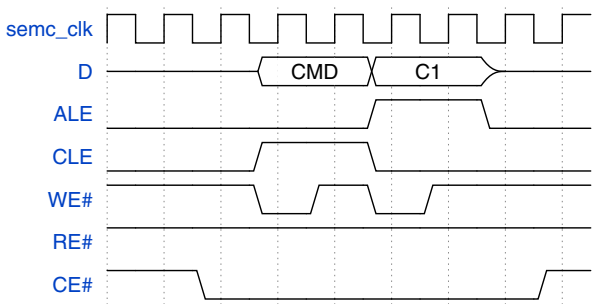


Figure 29-39. NAND Flash Access Command Type - Command + Address Phase

29.3.1.5.3.4 Command + Address + Wait Phase

Figure 29-40 shows NAND Flash Command + Address + Wait Phase access timing.

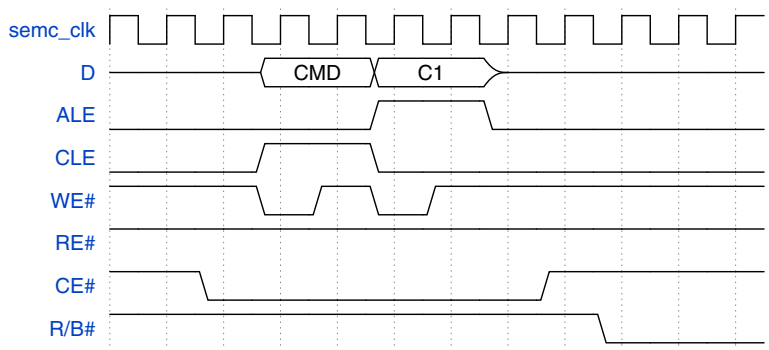


Figure 29-40. NAND Flash Access Command Type - Command + Address + Wait Phase

29.3.1.5.3.5 Command + Address + Read Phase

Figure 29-41 shows NAND Flash Command + Address + Read Phase access timing.

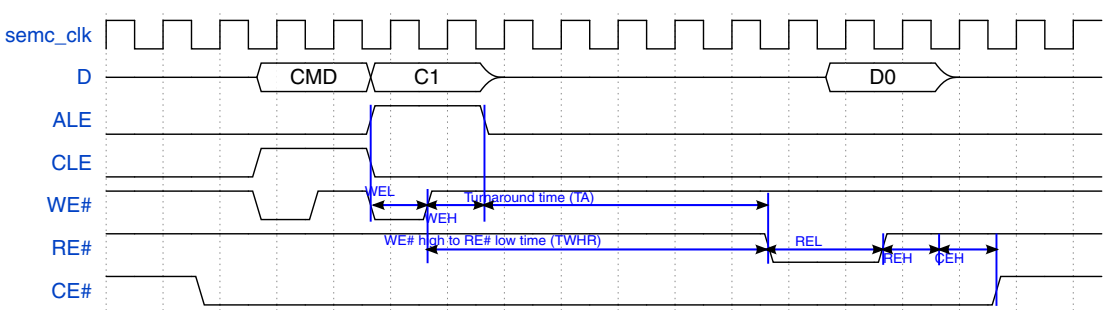


Figure 29-41. NAND Flash Access Command Type - Command + Address + Read Phase

NOTE

- For this example:
  - WE# low time is 1 clock cycle (NANDCR1[WEL]=0)
  - WE# high time is 1 clock cycle (NANDCR1[WEH]=0)
  - RE# low time is 2 clock cycles (NANDCR1[REL]=1)
  - RE# high time is 1 clock cycle (NANDCR1[REH]=0)
  - Turnaround time is 5 clock cycles (NANDCR1[TA]=4)
  - WE# high to RE# low time is 6 clock cycles (NANDCR2[TWHR]=5)
  - CE# hold time is 1 clock cycle (NANDCR1[CEH]=0)

29.3.1.5.3.6 Command + Address + Write Phase (for IP Command)

Figure 29-42 shows NAND Flash Command + Address + Write Phase access timing.

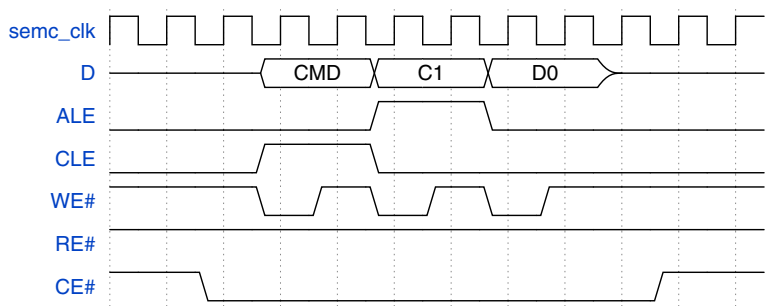


Figure 29-42. NAND Flash Access Command Type - Command + Address + Write Phase

29.3.1.5.3.7 Command + Read Phase

Figure 29-43 shows NAND Flash Command + Read Phase access timing.

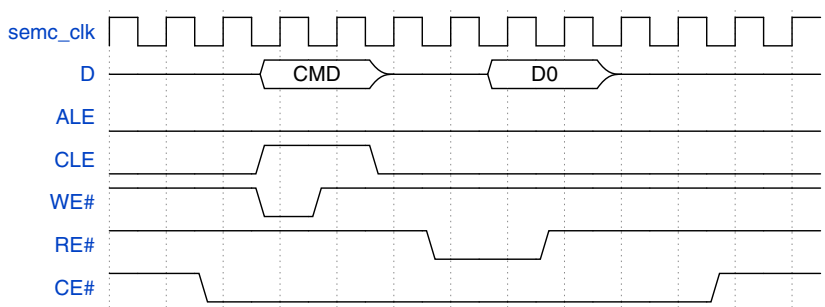


Figure 29-43. NAND Flash Access Command Type - Command + Read Phase

29.3.1.5.3.8 Command + Write Phase

Figure 29-44 shows NAND Flash Command + Write Phase access timing.

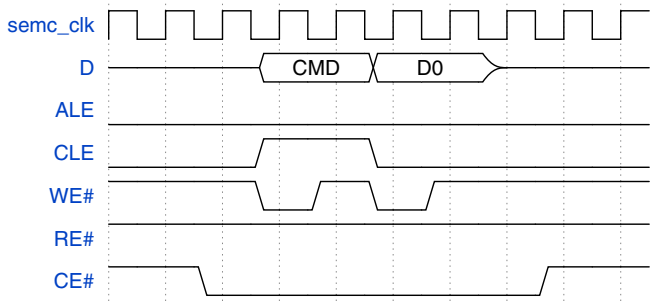


Figure 29-44. NAND Flash Access Command Type - Command + Write Phase

29.3.1.5.3.9 Read Phase

Figure 29-45 shows NAND Flash Read Phase access timing.

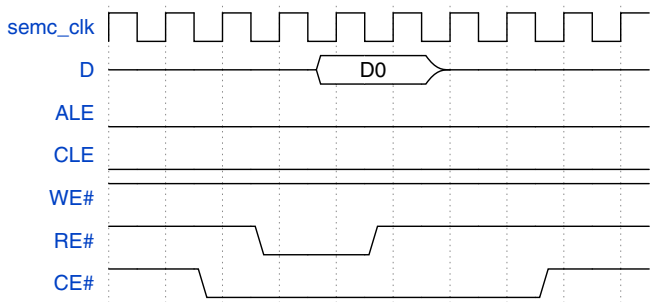


Figure 29-45. NAND Flash Access Command Type - Read Phase

29.3.1.5.3.10 Write Phase

Figure 29-46 shows the timing of NAND Flash Write Phase access.

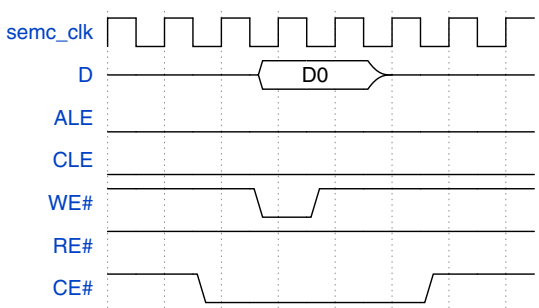


Figure 29-46. NAND Flash Access Command type - Write Phase

29.3.1.5.3.11 Command + Address + Command + Read Phase

Figure 29-47 shows the NAND Flash Command + Address + Command + Read Phase access timing.

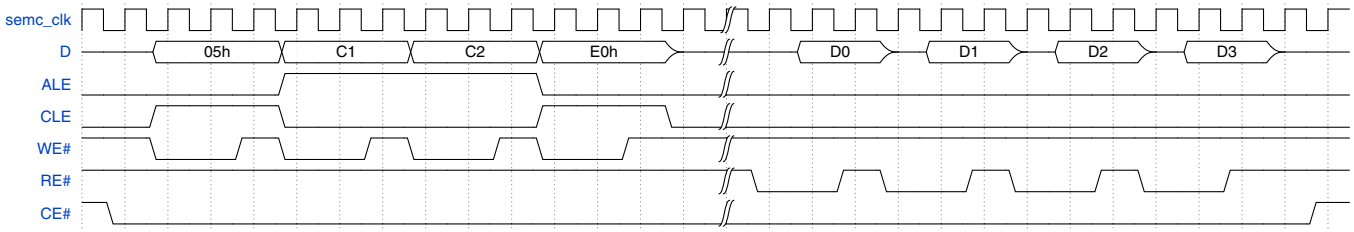


Figure 29-47. NAND Flash Access Command Type - Command + Address + Command + Read Phase

NOTE

- This command type is for AXI command only
- The command code in this command type is fixed value (Change Column Address) to support ONFI NAND device. The first byte command code is 05h, and second byte command code is E0h.
- The address type in this command type is fixed to two byte column address.

29.3.1.5.3.12 Command + Address + Write Phase (for AXI command)

Figure 29-48 shows the NAND Flash Command + Address + Write Phase access timing.

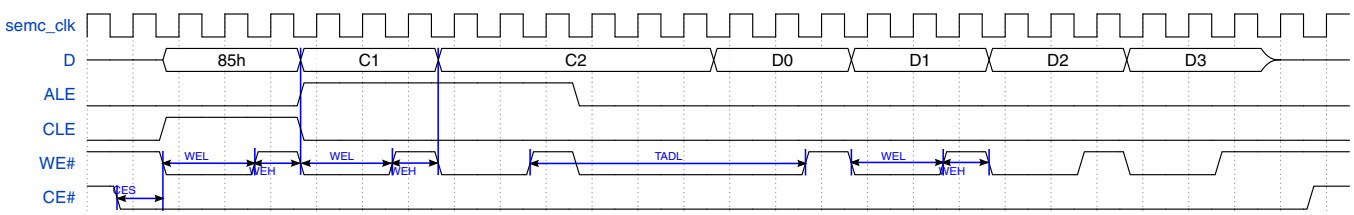


Figure 29-48. NAND Flash Access Command Type - Command + Address + Write Phase

NOTE

- This command type is for AXI command only
- The command code in this command type is fixed value (85h - Change Write Column Address) to support ONFI NAND device.
- The address type in this command type is fixed to two byte column address.
- For this example:
  - WE# low time is 2 clock cycles (NANDCR1[WEL]=1)
  - WE# high time is 1 clock cycle (NANDCR1[WEH]=0)
  - Address cycle to data loading time is 6 clock cycles (NANDCR2[TADL]=5)

29.3.1.5.4 NAND Flash Read Access in EDO and non-EDO Mode

NAND Flash support both EDO and non-EDO modes. Figure 29-49 shows the access timing of EDO mode.

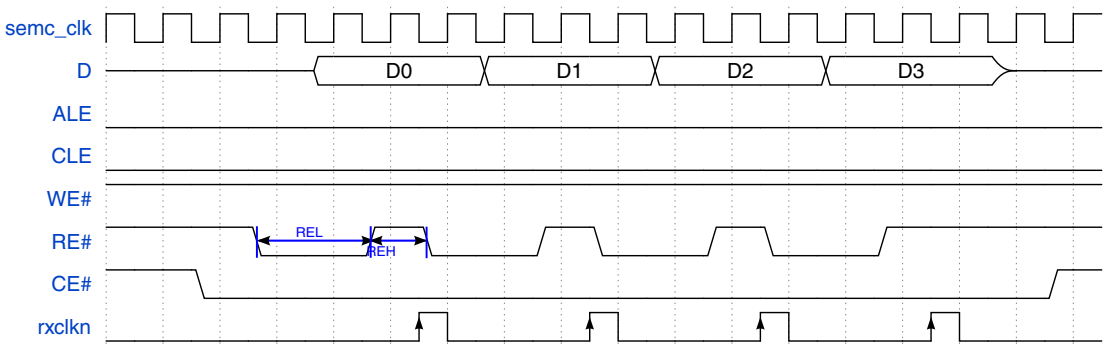


Figure 29-49. NAND Flash Read Access in EDO mode

NOTE

- rxclkn is the SEMC internal clock to sample read data. It is close to falling edge of RE#.
- For this example:
  - RE# low time is 2 clock cycles (NANDCR1[REL]=1)
  - RE# high time is 1 clock cycle (NANDCR1[REH]=0)

Figure 29-50 shows the access timing of non-EDO mode.

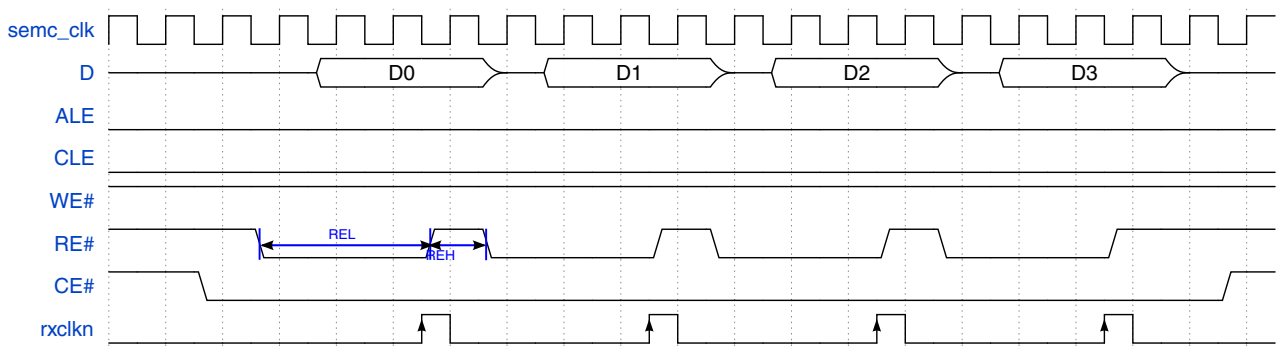


Figure 29-50. NAND Flash Read Access in non-EDO mode

NOTE

- rxclkn is the SEMC internal clock to sample read data. It is close to rising edge of RE#.
- For this example:
  - RE# low time is 3 clock cycles (NANDCR1[REL]=2)
  - RE# high time is 1 clock cycle (NANDCR1[REH]=0)

29.3.1.5.5 NAND Flash Access in SYNC Mode

NAND Flash supports SYNC mode, the following chapters describes the access timings.

29.3.1.5.5.1 NAND Command + Address + Command + Read Operation in SYNC Mode

Figure 29-51 shows NAND Command + Address + Command + Read operation in SYNC mode.

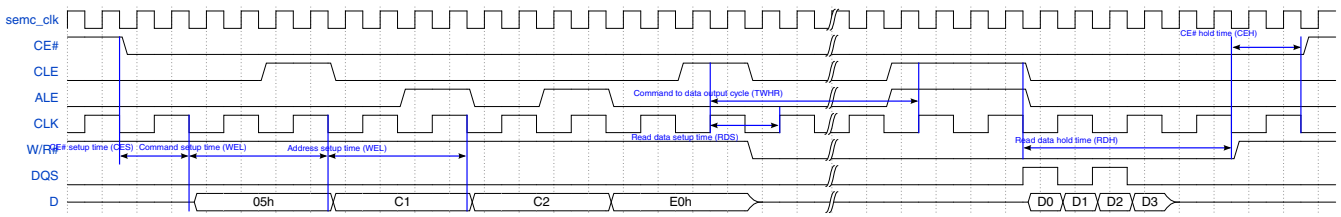


Figure 29-51. NAND Command + Address + Command + Read in SYNC Mode

NOTE

- semc_clk is the SEMC internal functional clock. CLK is the synchronous clock to latch command and address cycles. The frequency of CLK is half of semc_clk. For example, if semc_clk frequency is 200 MHz, the CLK frequency is 100 MHz.
- For this example:
  - CE# setup time is 2 clock cycles (NANDCR1[CES]=1)
  - Command/Address setup time is 4 clock cycles (NANDCR1[WEL]=2)
  - NANDCR3[RDS] and NANDCR2[TWHR] control the timing from command to data read phase. In general, NANDCR3[RDS] controls the minimum delay before data read phase. In this figure, NANDCR3[RDS]=0.
  - Read data hold time is 6 clock cycles (NANDCR3[RDH]=4)
  - CE# hold time is 2 clock cycles (NANDCR1[CEH]=1)

29.3.1.5.5.2 NAND Command + Address + Write Operation in SYNC Mode

Figure 29-52 shows NAND Command + Address + Write operation in SYNC mode.

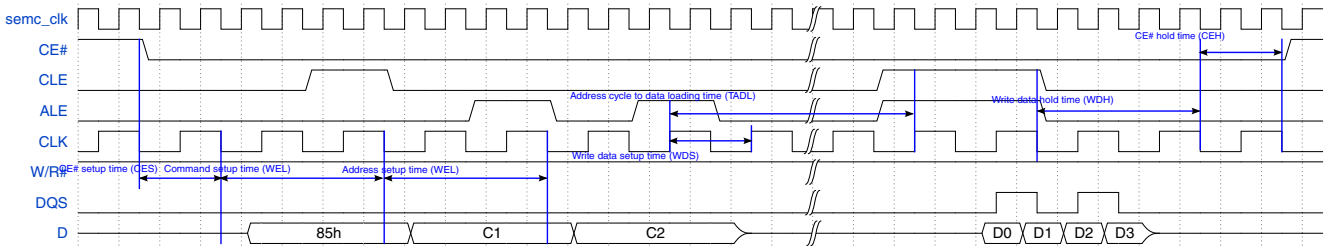


Figure 29-52. NAND Command + Address + Write in SYNC Mode

NOTE

- For this example:
  - CE# setup time is 2 clock cycles (NANDCR1[CES]=1)
  - Command/Address setup time is 4 clock cycles (NANDCR1[WEL]=2)
  - NANDCR3[WDS] and NANDCR2[TADL] control the timing from command to data write phase. In general, NANDCR3[WDS] controls the minimum delay before data write phase. In this figure, NANDCR3[WDS]=0.
  - Write data hold time is 4 clock cycles (NANDCR3[WDH]=2)
  - CE# hold time is 2 clock cycles (NANDCR1[CEH]=1)

29.3.1.6 NOR Flash Controller Operations

This section describes the operations of NOR Flash controller.

29.3.1.6.1 NOR Flash Address Multiplexing

NOR Flash use a multiplexed address split into rows and columns. Figure 29-53 shows how the bits of the linear address sent to the SEMC are split up into the row and column addresses sent to the NOR Flash memory device.

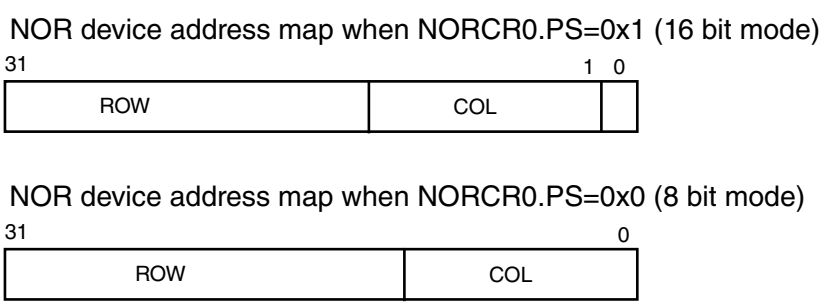


Figure 29-53. NOR Flash Address Multiplexing

NOTE

- NOR address bits consist of row and column address bits.
- Column address bits offset is bit 0 if NORCR0[PS]=0 and bit 1 if NORCR0[PS]=1. Column address bit width is determined by NORCR0[COL].
- Row address bits offset is determined by NORCR0[PS] and NORCR0[COL].
- NOR device base address on SoC is configured by BR5 register.
- Write data should be 16 bit aligned when connecting to 16 bit NOR device.

29.3.1.6.2 NOR Flash Read Operation in ASYNC Mode

NOR Flash controller supports following address modes, which setting by NORCR0[AM].

Figure 29-54 shows the ASYNC NOR Flash read in Non-ADMUX address mode.

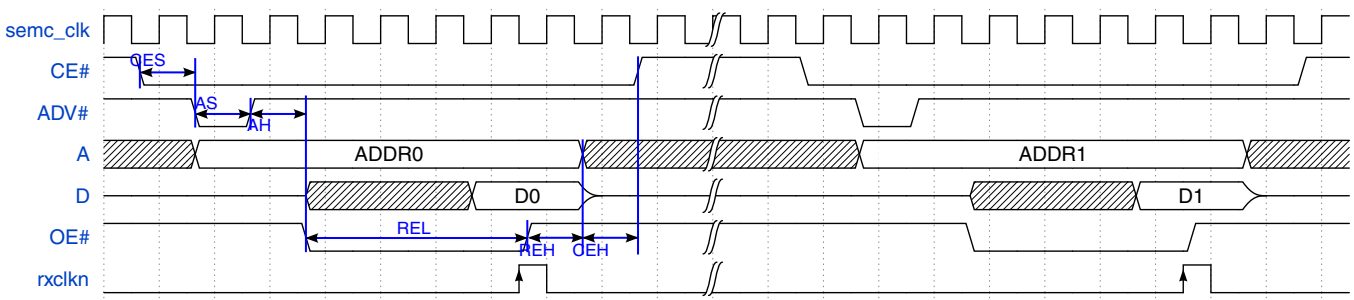


Figure 29-54. ASYNC NOR Flash Read in Non-ADMUX Address Mode

NOTE

- semc_clk and rxclkn are the internal signals. semc_clk is the SEMC functional clock. rxclkn is the clock to sample data from NOR Flash.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 1 clock cycle (NORCR1[AS]=0)
  - Address hold time is 1 clock cycle (NORCR1[AH]=0)
  - OE# low time is 4 clock cycles (NORCR1[REL]=3)
  - OE# high time is 1 clock cycle (NORCR1[REH]=0)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-55 shows the ASYNC NOR Flash read in ADMUX address mode.

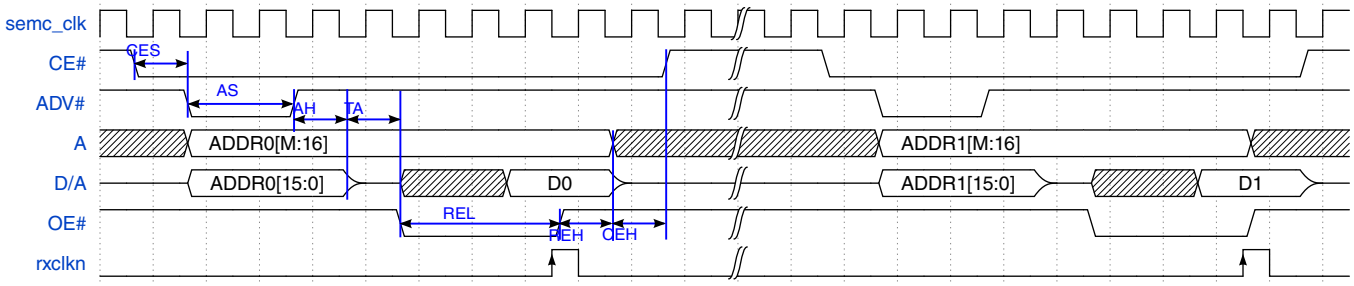


Figure 29-55. ASYNC NOR Flash Read in ADMUX Address Mode

NOTE

- semc_clk and rxclkn are the internal signals. semc_clk is the SEMC functional clock. rxclkn is the clock to sample data from NOR Flash.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 2 clock cycles (NORCR1[AS]=1)
  - Address hold time is 1 clock cycle (NORCR1[AH]=0)
  - Turnaround time is 1 clock cycle (NORCR2[TA]=0)
  - OE# low time is 3 clock cycles (NORCR1[REL]=2)
  - OE# high time is 1 clock cycle (NORCR1[REH]=0)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-56 shows the ASYNC NOR Flash read in AADM address mode.

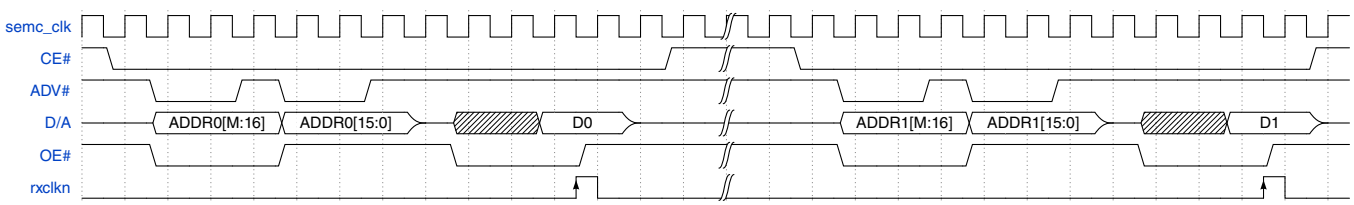


Figure 29-56. ASYNC NOR Flash Read in AADM Address Mode

29.3.1.6.3 NOR Flash Write Operation in ASYNC Mode

Figure 29-57 shows the ASYNC NOR Flash write in Non-ADMUX address mode.

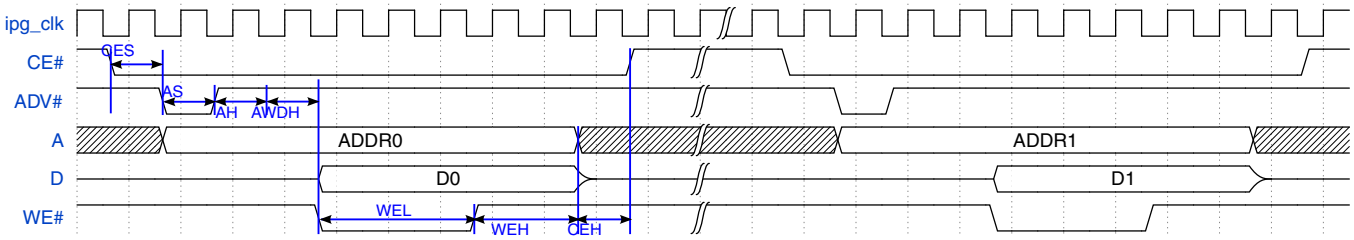


Figure 29-57. ASYNC NOR Flash Write in Non-ADMUX Address Mode

NOTE

- semc_clk is the SEMC functional clock.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 1 clock cycle (NORCR1[AS]=0)
  - Address hold time is 1 clock cycle (NORCR1[AH]=0)
  - Address to write data hold time is 1 clock cycle (NORCR2[AWDH]=1)
  - WE# low time is 3 clock cycles (NORCR1[WEL]=2)
  - WE# high time is 2 clock cycles (NORCR1[WEH]=1)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-58 shows the ASYNC NOR Flash write in ADMUX address mode.

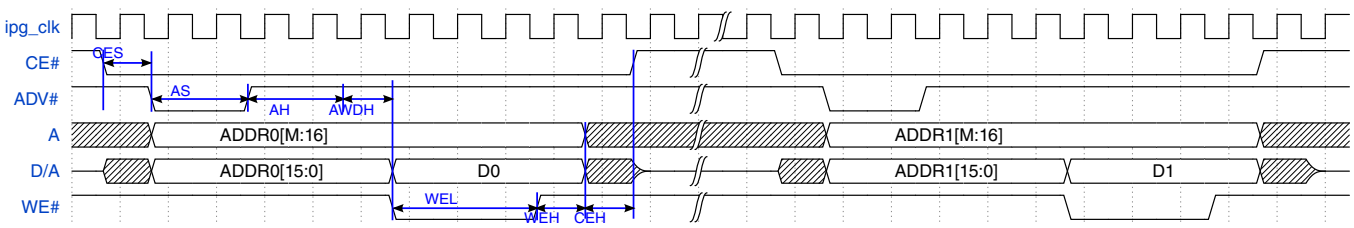


Figure 29-58. ASYNC NOR Flash Write in ADMUX Address Mode

NOTE

- semc_clk is the SEMC functional clock.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 2 clock cycles (NORCR1[AS]=1)
  - Address hold time is 2 clock cycles (NORCR1[AH]=1)
  - Address to write data hold time is 1 clock cycle (NORCR2[AWDH]=1)
  - WE# low time is 3 clock cycles (NORCR1[WEL]=2)
  - WE# high time is 1 clock cycle (NORCR1[WEH]=0)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-59 shows the ASYNC NOR Flash write in AADM address mode.

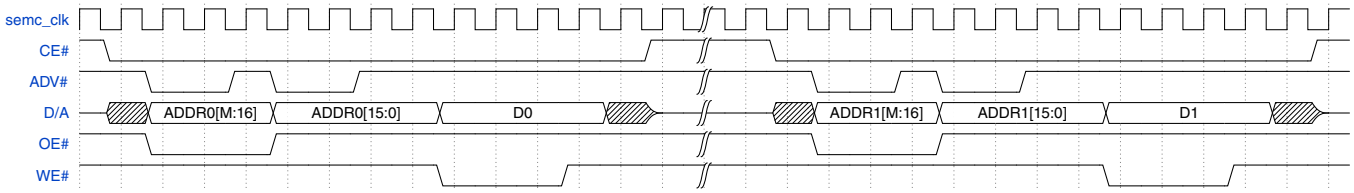


Figure 29-59. ASYNC NOR Flash Write in AADM Address Mode

29.3.1.6.4 NOR Flash Read Operation in SYNC Mode

Figure 29-60 shows the SYNC NOR Flash read in Non-ADMUX address mode.

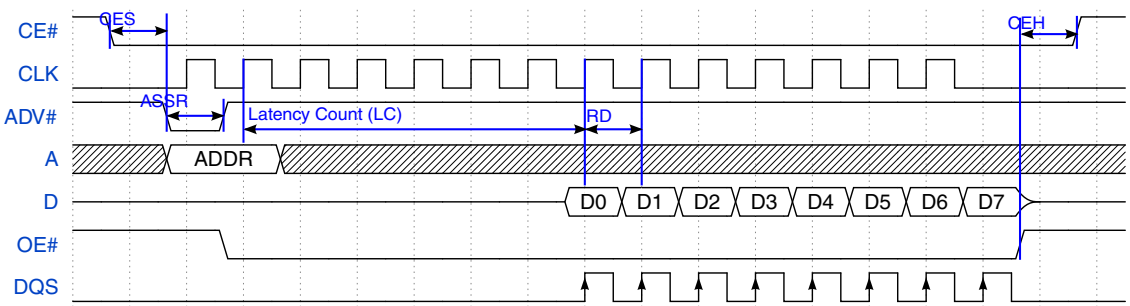


Figure 29-60. SYNC Flash Read in Non-ADMUX Address Mode

NOTE

- DQS is the clock to sample data from NOR Flash. It generally loops back from DQS pad.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 1 clock cycle (NORCR3[ASSR]=0)
  - Latency count is 6 clock cycles (NORCR2[LC]=6)
  - Read time is 1 clock cycle (NORCR2[RD]=0)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-61 shows the SYNC NOR Flash read in ADMUX address mode.

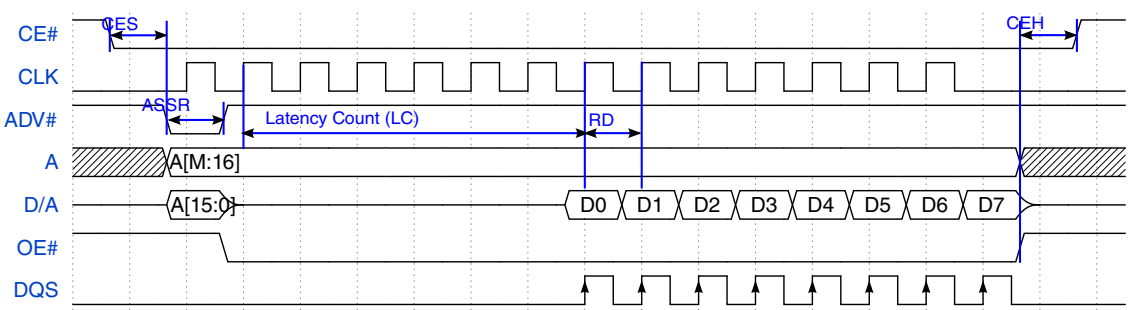


Figure 29-61. SYNC Flash Read in ADMUX Address Mode

NOTE

- DQS is the clock to sample data from NOR Flash. It generally loops back from DQS pad.
- For this example:
  - CE# setup time is 1 clock cycle (NORCR1[CES]=0)
  - Address setup time is 1 clock cycle (NORCR3[ASSR]=0)
  - Latency count is 6 clock cycles (NORCR2[LC]=6)
  - Read time is 1 clock cycle (NORCR2[RD]=0)
  - CE# hold time is 1 clock cycle (NORCR1[CEH]=0)

Figure 29-62 shows the SYNC NOR Flash read in AADM address mode.

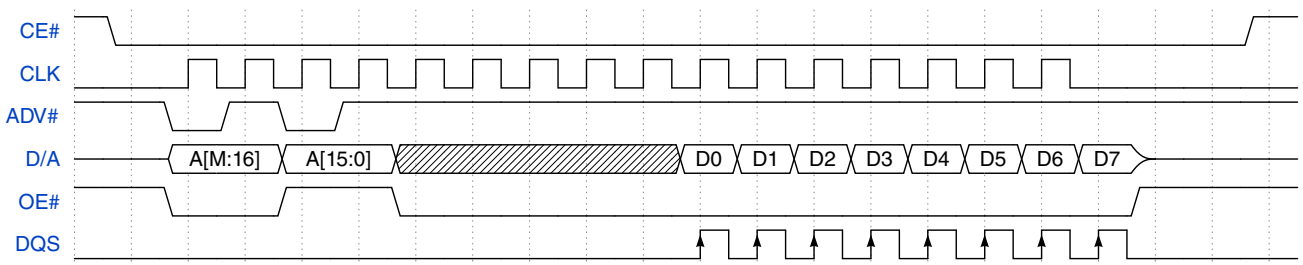


Figure 29-62. SYNC Flash Read in AADM Address Mode

29.3.1.6.5 NOR Flash Write Operation in SYNC Mode

Flash write operation in SYNC mode is same as asynchronous.

29.3.1.7 SRAM Controller Operations

This section describes the operations of SRAM controller.

29.3.1.7.1 SRAM Address Multiplexing

SRAM use a multiplexed address split into rows and columns. Figure 29-63 shows how the bits of the linear address sent to the SEMC are split up into the row and column addresses sent to the SRAM memory device.

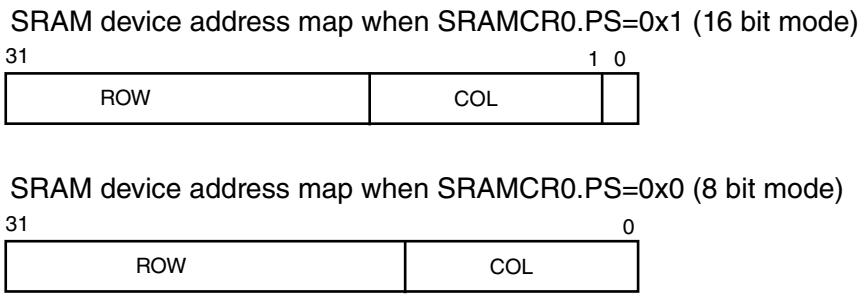


Figure 29-63. SRAM Address Multiplexing

NOTE

- SRAM address bits consist of row and column address bits.
- Column address bits offset is bit 0 if SRAMCR0[PS]=0, offset is bit 1 if SRAMCR0[PS]=1. Column address bit width is determined by SRAMCR0[COL].
- Row address bits offset is determined by SRAMCR0[PS] and SRAMCR0[COL].
- SRAM device base address on SoC is configured by BR6, BR9, BR10 and BR11 registers.

29.3.1.7.2 SRAM Read Operation in ASYNC Mode

Figure 29-64 shows the ASYNC SRAM read in Non-ADMUX address mode.

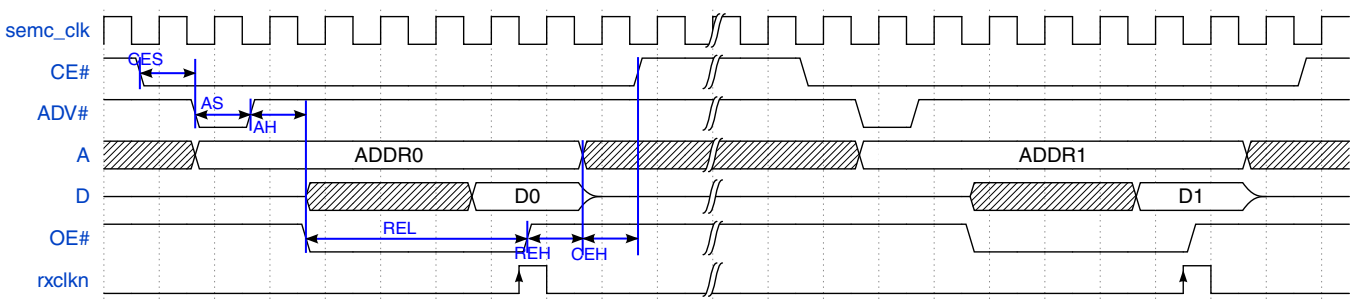


Figure 29-64. ASYNC SRAM Read in Non-ADMUX Address Mode

NOTE

- semc_clk and rxclkn are the internal signals. semc_clk is the SEMC functional clock. rxclkn is the clock to sample data from SRAM.
- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)

- Address hold time is 1 clock cycle (SRAMCR1[AH]=0)
- OE# low time is 4 clock cycles (SRAMCR1[REL]=3)
- OE# high time is 1 clock cycle (SRAMCR1[REH]=0)
- CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-65 shows the ASYNC SRAM read in ADMUX address mode.

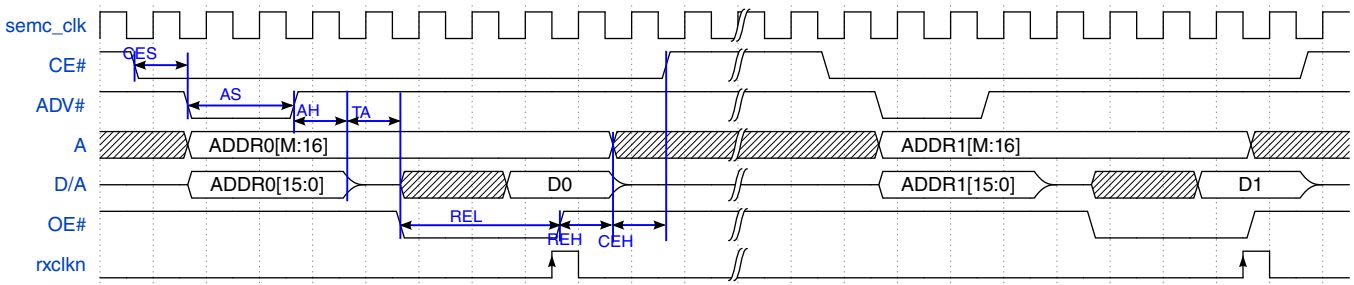


Figure 29-65. ASYNC SRAM Read in ADMUX Address Mode

NOTE

- semc_clk and rxclkn are the internal signals. semc_clk is the SEMC functional clock. rxclkn is the clock to sample data from SRAM.
- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 2 clock cycles (SRAMCR1[AS]=1)
  - Address hold time is 1 clock cycle (SRAMCR1[AH]=0)
  - Turnaround time is 1 clock cycle (SRAMCR2[TA]=0)
  - OE# low time is 3 clock cycles (SRAMCR1[REL]=2)
  - OE# high time is 1 clock cycle (SRAMCR1[REH]=0)
  - CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-66 shows the ASYNC SRAM read in AADM address mode.

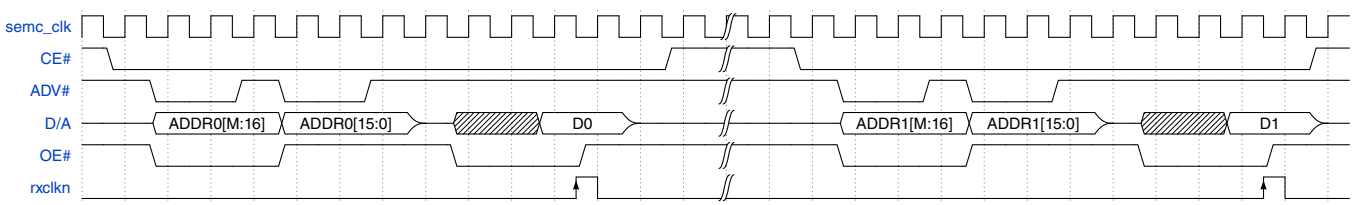


Figure 29-66. ASYNC SRAM Read in AADM Address Mode

29.3.1.7.3 SRAM Write Operation in ASYNC Mode

Figure 29-67 shows the ASYNC SRAM write in Non-ADMUX address mode.

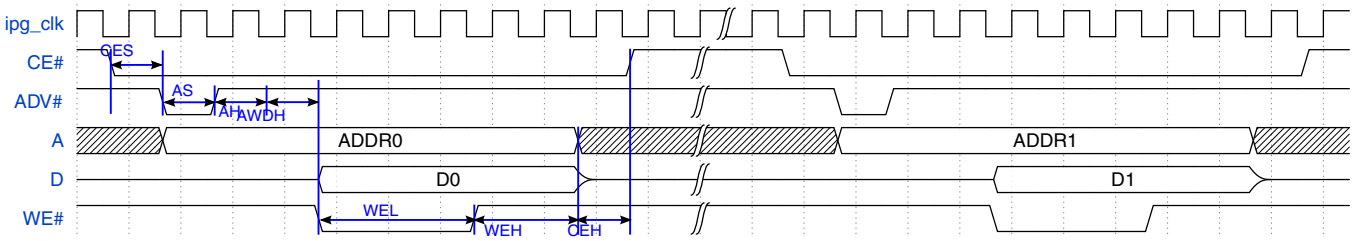


Figure 29-67. ASYNC SRAM Write in Non-ADMUX Address Mode

NOTE

- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)
  - Address hold time is 1 clock cycle (SRAMCR1[AH]=0)
  - Address to write data hold time is 1 clock cycle (SRAMCR2[AWDH]=1)
  - WE# low time is 3 clock cycles (SRAMCR1[WEL]=2)
  - WE# high time is 2 clock cycles (SRAMCR1[WEH]=1)
  - CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-68 shows the ASYNC SRAM write in ADMUX address mode.

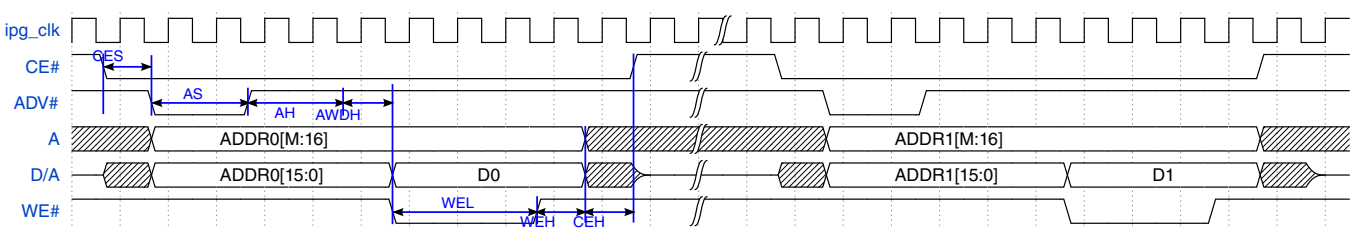


Figure 29-68. ASYNC SRAM Write in ADMUX Address Mode

NOTE

- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 2 clock cycles (SRAMCR1[AS]=1)
  - Address hold time is 2 clock cycles (SRAMCR1[AH]=1)
  - Address to write data hold time is 1 clock cycle (SRAMCR2[AWDH]=1)

- WE# low time is 3 clock cycles (SRAMCR1[WEL]=2)
- WE# high time is 1 clock cycle (SRAMCR1[WEH]=0)
- CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-69 shows the ASYNC SRAM write in AADM address mode.

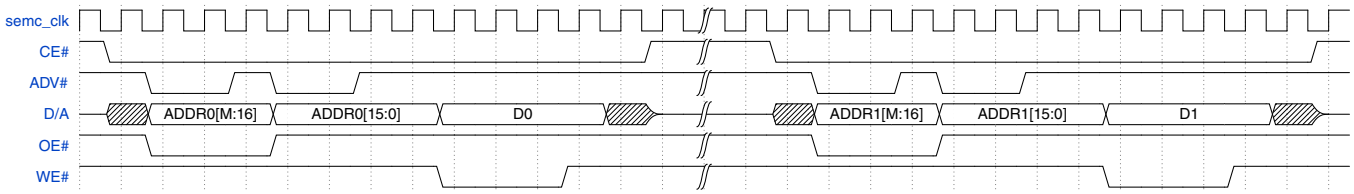


Figure 29-69. ASYNC SRAM Write in AADM Address Mode

29.3.1.7.4 SRAM Register Operation in ASYNC Mode

Figure 29-70 shows the ASYNC SRAM register access in ADMUX address mode.

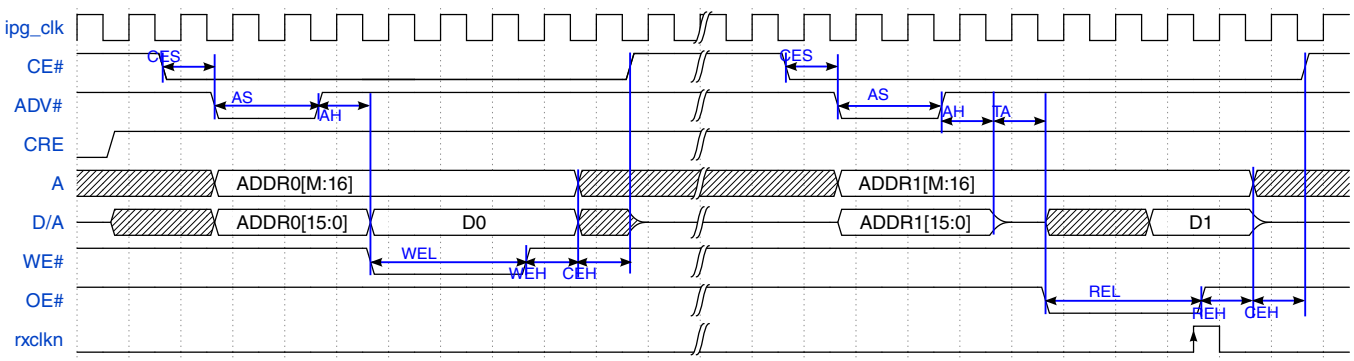


Figure 29-70. ASYNC SRAM Register Access in ADMUX Address Mode

NOTE

- For this example:
  - First operation is SRAM register write access. ADDR0[M:0] contains the value for writing to SRAM register. The value of D0 can be ignored.
  - Second operation is SRAM register read access. ADDR1[M:0] contains the value for choosing of SRAM register. D1 is the value that read from SRAM register.
  - The value of ADDR0/1 comes from IPCR0 register.

- If BR6[BA]=0x98000 and IPCR0=0x9810883E, ADDR0/1 is 0x8441F when port size is 16bit mode. The address is divided by 2 in 16bit mode.
- If BR6[BA]=0x98000 and IPCR0=0x9810883E, ADDR0/1 is 0x10883E when port size is 8bit mode.

- CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
- Address setup time is 2 clock cycles (SRAMCR1[AS]=1)
- Address hold time is 1 clock cycle (SRAMCR1[AH]=0)
- WE# low time is 3 clock cycles (SRAMCR1[WEL]=2)
- WE# high time is 1 clock cycle (SRAMCR1[WEH]=0)
- CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)
- Turnaround time is 1 clock cycle (SRAMCR2[TA]=0)
- OE# low time is 3 clock cycles (SRAMCR1[REL]=2)
- OE# high time is 1 clock cycle (SRAMCR1[REH]=0)

29.3.1.7.5 SRAM Read Operation in ASYNC Mode with Wait Pin

SRAM read operation in ASYNC mode with wait pin is indicated in following diagrams:

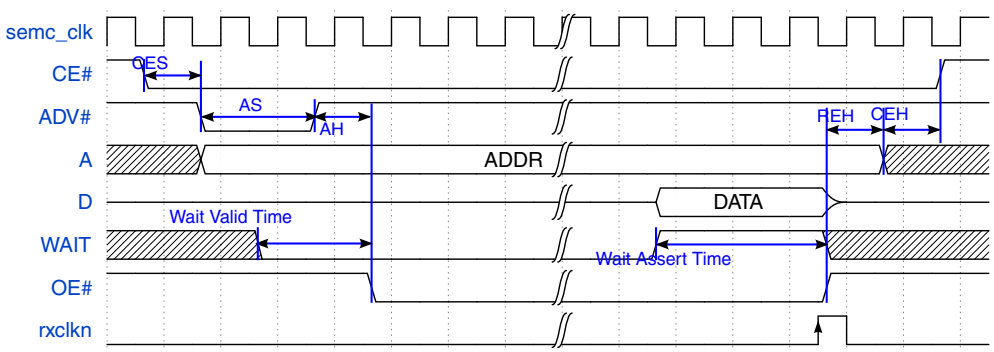


Figure 29-71. SRAM Read in ASYNC Mode (Non-ADMUX) with Wait Pin

NOTE

1. Wait Valid Time must be no less than 2 semc_clk cycles.
2. Wait and DATA must be valid till OE# high.
3. For this example:
   - SRAMCR0[WAITEN]=1
   - SRAMCR0[WAITSP]=1
   - MCR[WPOL0]=0

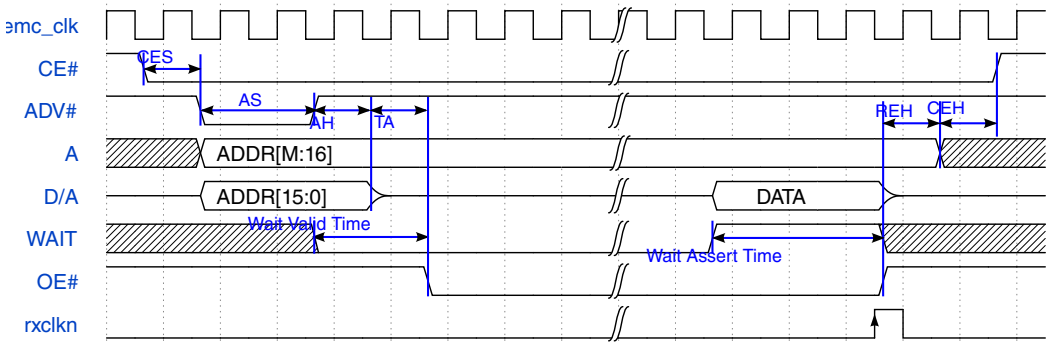


Figure 29-72. SRAM Read in ASYNC Mode (ADMUX) with Wait Pin

NOTE

1. Wait Valid Time must be no less than 2 semc_clk cycles.
2. Wait and DATA must be valid till OE# high.
3. For this example:
   - SRAMCR0[WAITEN]=1
   - SRAMCR0[WAITSP]=1
   - MCR[WPOL0]=0

29.3.1.7.6 SRAM Write Operation in ASYNC Mode with Wait Pin

SRAM write operation in ASYNC mode is indicated in following diagrams:

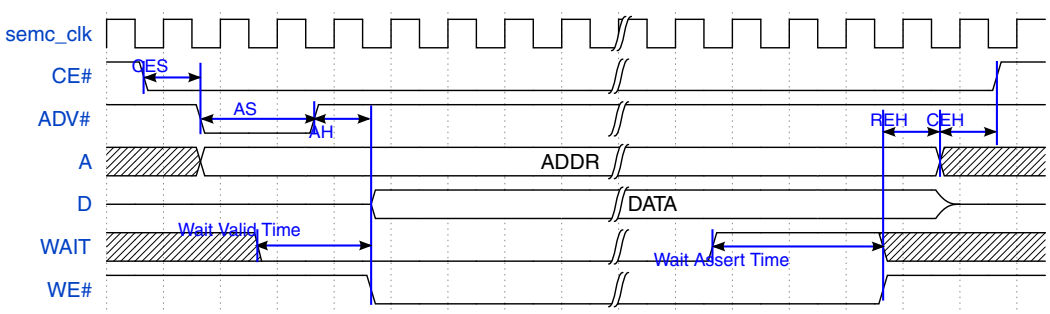


Figure 29-73. SRAM Write in ASYNC Mode (Non-ADMUX) with Wait Pin

NOTE

1. Wait Valid Time must be no less than 2 semc_clk cycles.
2. Wait must be valid till WE# high.
3. For this example:
   - SRAMCR0[WAITEN]=1
   - SRAMCR0[WAITSP]=1
   - MCR[WPOL0]=0

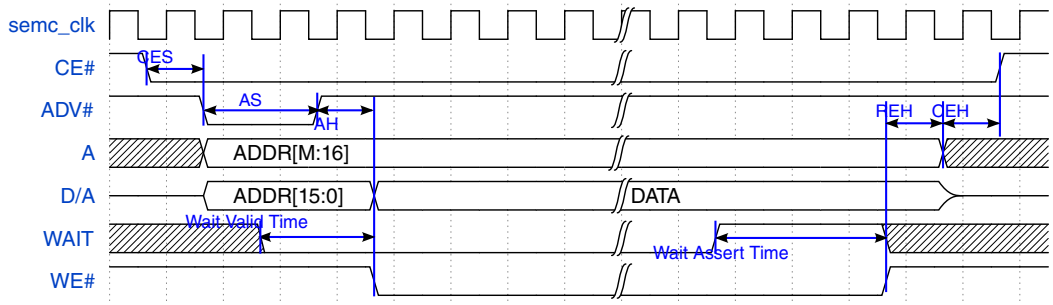


Figure 29-74. SRAM Write in ASYNC Mode (ADMUX) with Wait Pin

NOTE

1. Wait Valid Time must be no less than 2 semc_clk cycles.
2. Wait must be valid till WE# high.
3. For this example:
   - SRAMCR0[WAITEN]=1
   - SRAMCR0[WAITSP]=1
   - MCR[WPOL0]=0

29.3.1.7.7 SRAM Read Operation in SYNC Mode

Figure 29-75 shows the SYNC SRAM read in Non-ADMUX address mode.

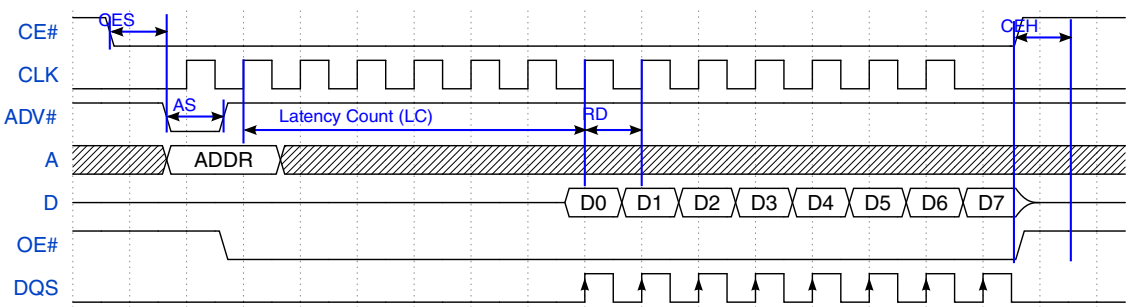


Figure 29-75. SYNC SRAM Read in Non-ADMUX Address Mode

NOTE

- DQS is the clock to sample data from SRAM. It generally loops back from DQS pad.
- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)
  - Latency count is 6 clock cycles (SRAMCR2[LC]=6)
  - Read time is 1 clock cycle (SRAMCR2[RD]=0)
  - CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-76 shows the SYNC SRAM read in ADMUX address mode.

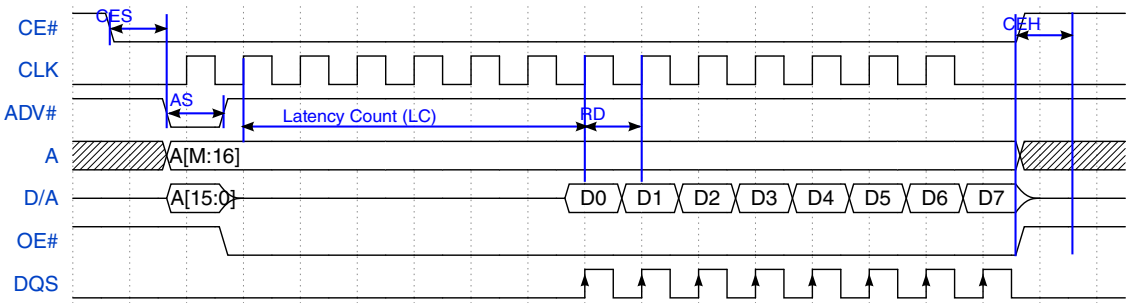


Figure 29-76. SYNC SRAM Read in ADMUX Address Mode

NOTE

- DQS is the clock to sample data from SRAM. It generally loops back from DQS pad.
- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)
  - Latency count is 6 clock cycles (SRAMCR2[LC]=6)
  - Read time is 1 clock cycle (SRAMCR2[RD]=0)
  - CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

29.3.1.7.8 SRAM Write Operation in SYNC Mode

Figure 29-77 shows the SYNC SRAM write in Non-ADMUX address mode.

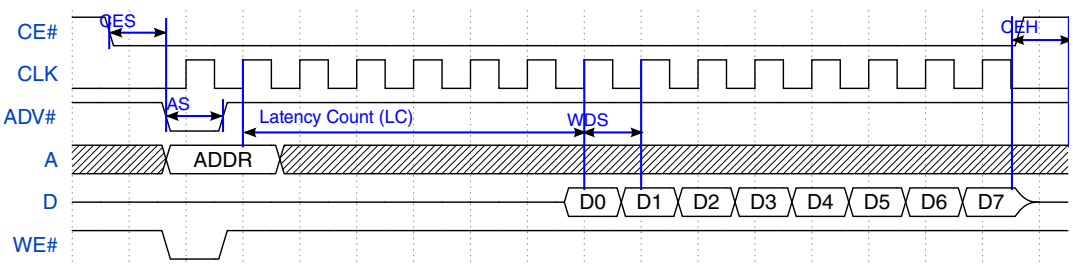


Figure 29-77. SRAM Write in SYNC Mode (Non-ADMUX)

NOTE

- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)
  - Latency count is 6 clock cycles (SRAMCR2[LC]=6)

- Write data setup time is 1 clock cycle (SRAMCR2[WDS]=0)
- CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

Figure 29-78 shows the SYNC SRAM write in ADMUX address mode.

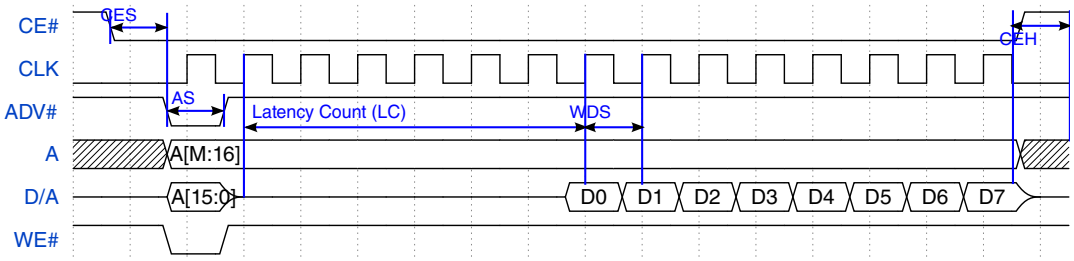


Figure 29-78. SRAM Write in SYNC Mode (ADMUX)

NOTE

- For this example:
  - CE# setup time is 1 clock cycle (SRAMCR1[CES]=0)
  - Address setup time is 1 clock cycle (SRAMCR1[AS]=0)
  - Latency count is 6 clock cycles (SRAMCR2[LC]=6)
  - Write data setup time is 1 clock cycle (SRAMCR2[WDS]=0)
  - CE# hold time is 1 clock cycle (SRAMCR1[CEH]=0)

29.3.1.8 Display Bus Interface Controller Operations

This section describes the operations of DBI controller.

DCX signal is controlled by write address. Write access to DBI is handled as DBI COMMAND WRITE transaction if address is higher than 0x10000 (DCX low). Otherwise, it is handled as DBI DATA WRITE transaction (DCX high).

For example, if BR7[BA]=0x9C000:

- An AXI write to address 0x9C010000 is treated as DBI COMMAND WRITE transaction.
- An AXI write to address 0x9C000000 is treated as DBI DATA WRITE transaction.

Figure 29-79 shows DBI write operation.

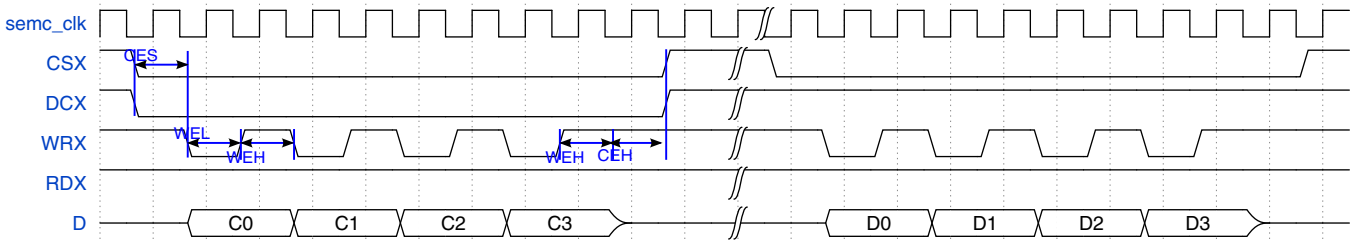


Figure 29-79. DBI Write Operation

NOTE

- DBI address is used to determine DBI COMMAND or DATA WRITE.
- DBI bus never send out address information for COMMAND and DATA WRITE.
- AXI WRAP write access to DBI is supported.
- DBI bus interface does not support WRITE mask feature.
- All write strobe in AXI write access to DBI must be valid. If AXI write strobe is invalid, the SEMC sends write data 0x00 instead of original write data.
- When DBI bus is 16 bit (DBICR0[PS]=1), AXI or IP command access start address to DBI bus must be 16 bit aligned. Otherwise the write data to DBI may be wrong. There is no bus error response or internal error status generated for this case.
- For this example:
  - CSX setup time is 1 clock cycle (DBICR1[CES]=0)
  - WRX low time is 1 clock cycle (DBICR1[WEL]=0)
  - WRX high time is 1 clock cycle (DBICR1[WEH]=0)
  - CSX hold time is 1 clock cycle (DBICR1[CEH]=0)

Figure 29-80 shows DBI read operation.

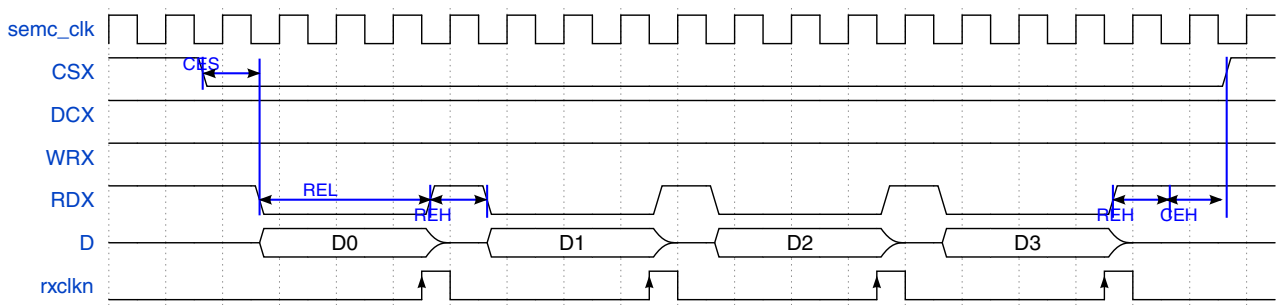


Figure 29-80. DBI Read Operation

NOTE

- Read access to DBI is always handled as DBI DATA READ transaction.
- DBI bus never sent out address information for DATA READ command.
- AXI WRAP read access to DBI is not supported except when the first beat address offset is zero in the burst.
- When DBI bus is 16 bit (DBICR0[PS]=1), AXI command or IP command access start address to DBI bus must be 16 bit aligned. Otherwise the read data is unknown. There is no bus error response or internal error status generated for this case.
- For this example:
  - CSX setup time is 1 clock cycle (DBICR1[CES]=0)
  - RDX low time is 3 clock cycles (DBICR1[REL]=2)
  - RDX high time is 1 clock cycle (DBICR1[REH]=0)
  - CSX hold time is 1 clock cycle (DBICR1[CEH]=0)

29.3.2 Clocks

There is one clock used for the SEMC AXI bus, IP bus, internal logic and external interface. It is shown as semc_clk in SEMC timing figures.

29.3.3 Reset

There is a hardware reset for the SEMC, which reset all the logic inside it. When set MCR[SWRST], there will be a software reset to reset all logic in the SEMC except configuration registers.

29.3.4 Interrupts

There is an interrupt output from the SEMC. It indicates the interrupt source that defined in INTR register. The interrupt source is enabled by INTEN register.

29.4 External Signals

Table 29-5 lists the SEMC module signals.

Table 29-5. SEMC Signals

Signal	Direction	Description
SEMC_DATA[31-00]	I/O	Data/Address Bits The SEMC has several controllers, and they share pins. SEMC_DATA works as data signal in SDRAM/NAND/8080 applications and works as data or address signal in NOR/SRAM applications. For detail pin mux information, refer to Pin Mux.
SEMC_DM[3-0]	O	Write Data Mask Bits
SEMC_ADDR[12-00]	O	Address Bits SEMC_ADDR08 can work as chip select pin in NAND/NOR/SRAM/ 8080 applications.
SEMC_BA[1-0]	O	Bank Address Bits SEMC_BA0 can work as WAIT in SRAM application. SEMC_BA1 can work as ALE/ADV#/DCX in NAND/NOR/SRAM/8080 applications.
SEMC_CAS	O	SDRAM Column Address Strobe
SEMC_RAS	O	SDRAM Row Address Strobe
SEMC_WE	O	SDRAM Write Enable
SEMC_CKE	O	SDRAM Clock enable
SEMC_CLK	O	SDRAM Clock
SEMC_CS0	O	SDRAM Chip Select 0
SEMC_CSX[3-0]	O	Configurable Chip Select bits
SEMC_RDY	I/O	Ready input pin for NAND It can also work as chip select pin in SDRAM/NOR/SRAM/8080 applications.
SEMC_DQS	I/O	DQS SEMC controller provides internal dummy read strobe for read data from SDRAM/NOR/SRAM. Higher read frequency can be achieved by looping back this dummy read strobe from pad. This pin can be floated or put some cap loads on board level to compensate DATA/CLK pins load.
SEMC_DQS4	I/O	DQS Data read strobe for NAND Flash.
SEMC_CLKX0	O	Configurable clock 0 for NOR/SRAM
SEMC_CLKX1	O	Configurable clock 1 for NOR/SRAM

29.4.1 Pin Mux

Table 29-6 lists the pin mux of the SEMC module.

Table 29-6. SEMC Pin Mux Overview

Signal	SDRAM	NAND	NOR (ADMUX 16bit)	SRAM (ADMUX 16bit)	DBI	Comment
SEMC_DATA00	D0	D0	D0/A0	D0/A0	D0	Hardware MUX The SEMC determines the pin function according to its internal state
SEMC_DATA01	D1	D1	D1/A1	D1/A1	D1	
SEMC_DATA02	D2	D2	D2/A2	D2/A2	D2	
SEMC_DATA03	D3	D3	D3/A3	D3/A3	D3	
SEMC_DATA04	D4	D4	D4/A4	D4/A4	D4	
SEMC_DATA05	D5	D5	D5/A5	D5/A5	D5	
SEMC_DATA06	D6	D6	D6/A6	D6/A6	D6	
SEMC_DATA07	D7	D7	D7/A7	D7/A7	D7	
SEMC_DATA08	D8	D8	D8/A8	D8/A8	D8	
SEMC_DATA09	D9	D9	D9/A9	D9/A9	D9	
SEMC_DATA10	D10	D10	D10/A10	D10/A10	D10	
SEMC_DATA11	D11	D11	D11/A11	D11/A11	D11	
SEMC_DATA12	D12	D12	D12/A12	D12/A12	D12	
SEMC_DATA13	D13	D13	D13/A13	D13/A13	D13	
SEMC_DATA14	D14	D14	D14/A14	D14/A14	D14	
SEMC_DATA15	D15	D15	D15/A15	D15/A15	D15	
SEMC_DATA16	D16	-	A24	A24	-	
SEMC_DATA17	D17	-	A25	A25	-	
SEMC_DATA18	D18	-	A26	A26	-	
SEMC_DATA19	D19	-	A27	A27	-	
SEMC_DATA20	D20	-	-	-	-	
SEMC_DATA21	D21	-	-	-	-	
SEMC_DATA22	D22	-	-	-	-	
SEMC_DATA23	D23	-	-	-	-	
SEMC_DATA24	D24	-	-	-	-	
SEMC_DATA25	D25	-	-	-	-	
SEMC_DATA26	D26	-	-	-	-	
SEMC_DATA27	D27	-	-	-	-	
SEMC_DATA28	D28	-	-	-	-	
SEMC_DATA29	D29	-	-	-	-	
SEMC_DATA30	D30	-	-	-	-	
SEMC_DATA31	D31	-	-	-	-	
SEMC_DM3	DQM3	-	-	-	-	
SEMC_DM2	DQM2	-	-	-	-	
SEMC_DM1	DQM1	-	-	UB#	-	

Table continues on the next page...

Table 29-6. SEMC Pin Mux Overview (continued)

Signal	SDRAM	NAND	NOR (ADMUX 16bit)	SRAM (ADMUX 16bit)	DBI	Comment
SEMC_DM0	DQM0	-	-	LB#	-	
SEMC_ADDR00	A0	-	A16	A16	-	
SEMC_ADDR01	A1	-	A17	A17	-	
SEMC_ADDR02	A2	-	A18	A18	-	
SEMC_ADDR03	A3	-	A19	A19	-	
SEMC_ADDR04	A4	-	A20	A20	-	
SEMC_ADDR05	A5	-	A21	A21	-	
SEMC_ADDR06	A6	-	A22	A22	-	
SEMC_ADDR07	A7	-	A23	A23	-	
SEMC_ADDR08	A8	CE#	CE#/A24	CE#/A24	CSX	Configured by IOCR[MUX_A8]
SEMC_ADDR09	A9	CLE	-	CRE	-	Hardware MUX
SEMC_ADDR10	A10	-	-	-	-	The SEMC determines the pin function according to its internal state
SEMC_ADDR11	A11	WE#	WE#	WE#	WRX	SEMC_ADDR11 works as CLK signal in NAND synchronous mode
SEMC_ADDR12	A12	RE#	OE#	OE#	RDX	SEMC_ADDR12 works as W/R# signal in NAND synchronous mode
SEMC_BA0	BA0	-	-	WAIT	-	Hardware MUX
SEMC_BA1	BA1	ALE	ADV#	ADV#	DCX	The SEMC determines the pin function according to its internal state
SEMC_CAS	CAS#	-	-	-	-	
SEMC_RAS	RAS#	-	-	-	-	
SEMC_WE	WE#	-	-	-	-	
SEMC_CKE	CKE	-	-	-	-	
SEMC_CLK	CLK	-	-	-	-	
SEMC_CS0	CS0	-	-	-	-	
SEMC_CSX0	CS	CE#	CE#	CE#	CSX	Configured by IOCR[MUX_CSX0]
SEMC_CSX1	CS	CE#	CE#	CE#	CSX	Configured by IOCR[MUX_CSX1]
SEMC_CSX2	CS	CE#	CE#	CE#	CSX	Configured by IOCR[MUX_CSX2]
SEMC_CSX3	CS	CE#	CE#	CE#	CSX	Configured by IOCR[MUX_CSX3]
SEMC_RDY	CS	R/B#	CE#	CE#	CSX	Configured by IOCR[MUX_RDY]
SEMC_DQS	DQS	-	DQS	DQS	-	Hardware MUX
SEMC_DQS4	-	DQS	-	-	-	Hardware MUX
SEMC_CLKX0	-	-	CLK	CLK	-	Configured by IOCR[MUX_CLKX0]
SEMC_CLKX1	-	-	CLK	CLK	-	Configured by IOCR[MUX_CLKX1]

NOTE

- IOCR configuration should be static. Dynamic remapping of pin muxing controlled by IOCR is not supported.
- NOR and SRAM pins are configured as 16 bit ADMUX mode in above table. It is the recommended configuration.

Table 29-7 lists the different configurations for NOR and SRAM pin mux of the SEMC module.

Table 29-7. SEMC Pin Mux for NOR and SRAM

Signal	NOR (Non-ADMUX)	NOR (ADMUX 8bit)	NOR (ADMUX 16bit)	NOR (AADM 16bit)	SRAM (Non-ADMUX)	SRAM (ADMUX 8bit)	SRAM (ADMUX 16bit)	SRAM (AADM 16bit)
SEMC_DATA00	D0	D0/A0	D0/A0	D0/A0	D0	D0/A0	D0/A0	D0/A0
SEMC_DATA01	D1	D1/A1	D1/A1	D1/A1	D1	D1/A1	D1/A1	D1/A1
SEMC_DATA02	D2	D2/A2	D2/A2	D2/A2	D2	D2/A2	D2/A2	D2/A2
SEMC_DATA03	D3	D3/A3	D3/A3	D3/A3	D3	D3/A3	D3/A3	D3/A3
SEMC_DATA04	D4	D4/A4	D4/A4	D4/A4	D4	D4/A4	D4/A4	D4/A4
SEMC_DATA05	D5	D5/A5	D5/A5	D5/A5	D5	D5/A5	D5/A5	D5/A5
SEMC_DATA06	D6	D6/A6	D6/A6	D6/A6	D6	D6/A6	D6/A6	D6/A6
SEMC_DATA07	D7	D7/A7	D7/A7	D7/A7	D7	D7/A7	D7/A7	D7/A7
SEMC_DATA08	D8	-	D8/A8	D8/A8	D8	-	D8/A8	D8/A8
SEMC_DATA09	D9	-	D9/A9	D9/A9	D9	-	D9/A9	D9/A9
SEMC_DATA10	D10	-	D10/A10	D10/A10	D10	-	D10/A10	D10/A10
SEMC_DATA11	D11	-	D11/A11	D11/A11	D11	-	D11/A11	D11/A11
SEMC_DATA12	D12	-	D12/A12	D12/A12	D12	-	D12/A12	D12/A12
SEMC_DATA13	D13	-	D13/A13	D13/A13	D13	-	D13/A13	D13/A13
SEMC_DATA14	D14	-	D14/A14	D14/A14	D14	-	D14/A14	D14/A14
SEMC_DATA15	D15	-	D15/A15	D15/A15	D15	-	D15/A15	D15/A15
SEMC_DATA16	A8	A16	A24	-	A8	A16	A24	-
SEMC_DATA17	A9	A17	A25	-	A9	A17	A25	-
SEMC_DATA18	A10	A18	A26	-	A10	A18	A26	-
SEMC_DATA19	A11	A19	A27	-	A11	A19	A27	-
SEMC_DATA20	A12	A20	-	-	A12	A20	-	-
SEMC_DATA21	A13	A21	-	-	A13	A21	-	-
SEMC_DATA22	A14	A22	-	-	A14	A22	-	-
SEMC_DATA23	A15	A23	-	-	A15	A23	-	-
SEMC_DATA24	A16	A24	-	-	A16	A24	-	-
SEMC_DATA25	A17	A25	-	-	A17	A25	-	-
SEMC_DATA26	A18	A26	-	-	A18	A26	-	-
SEMC_DATA27	A19	A27	-	-	A19	A27	-	-
SEMC_DATA28	A20	-	-	-	A20	-	-	-
SEMC_DATA29	A21	-	-	-	A21	-	-	-

Table continues on the next page...

Table 29-7. SEMC Pin Mux for NOR and SRAM (continued)

Signal	NOR (Non-ADMUX)	NOR (ADMUX 8bit)	NOR (ADMUX 16bit)	NOR (AADM 16bit)	SRAM (Non-ADMUX)	SRAM (ADMUX 8bit)	SRAM (ADMUX 16bit)	SRAM (AADM 16bit)
SEMC_DATA30	A22	-	-	-	A22	-	-	-
SEMC_DATA31	A23	-	-	-	A23	-	-	-
SEMC_DM3	-	-	-	-	-	-	-	-
SEMC_DM2	-	-	-	-	-	-	-	-
SEMC_DM1	-	-	-	-	UB#	-	UB#	UB#
SEMC_DM0	-	-	-	-	LB#	-	LB#	LB#
SEMC_ADDR00	A0	A8	A16	-	A0	A8	A16	-
SEMC_ADDR01	A1	A9	A17	-	A1	A9	A17	-
SEMC_ADDR02	A2	A10	A18	-	A2	A10	A18	-
SEMC_ADDR03	A3	A11	A19	-	A3	A11	A19	-
SEMC_ADDR04	A4	A12	A20	-	A4	A12	A20	-
SEMC_ADDR05	A5	A13	A21	-	A5	A13	A21	-
SEMC_ADDR06	A6	A14	A22	-	A6	A14	A22	-
SEMC_ADDR07	A7	A15	A23	-	A7	A15	A23	-
SEMC_ADDR08	CE#	CE#	CE#	CE#	CE#	CE#	CE#	CE#
SEMC_ADDR09	-	-	-	-	CRE	CRE	CRE	CRE
SEMC_ADDR10	-	-	-	-	-	-	-	-
SEMC_ADDR11	WE#	WE#	WE#	WE#	WE#	WE#	WE#	WE#
SEMC_ADDR12	OE#	OE#	OE#	OE#	OE#	OE#	OE#	OE#
SEMC_BA0	-	-	-	-	WAIT	WAIT	WAIT	WAIT
SEMC_BA1	ADV#	ADV#	ADV#	ADV#	ADV#	ADV#	ADV#	ADV#
SEMC_CAS	-	-	-	-	-	-	-	-
SEMC_RAS	-	-	-	-	-	-	-	-
SEMC_WE	-	-	-	-	-	-	-	-
SEMC_CKE	-	-	-	-	-	-	-	-
SEMC_CLK	-	-	-	-	-	-	-	-
SEMC_CS0	-	-	-	-	-	-	-	-
SEMC_CSX0	CE#/A24	CE#	CE#	CE#	CE#/A24	CE#	CE#	CE#
SEMC_CSX1	CE#/A25	CE#	CE#	CE#	CE#/A25	CE#	CE#	CE#
SEMC_CSX2	CE#/A26	CE#	CE#	CE#	CE#/A26	CE#	CE#	CE#
SEMC_CSX3	CE#/A27	CE#	CE#	CE#	CE#/A27	CE#	CE#	CE#
SEMC_RDY	CE#/A27	CE#	CE#	CE#	CE#/A27	CE#	CE#	CE#
SEMC_DQS	DQS	DQS	DQS	DQS	DQS	DQS	DQS	DQS
SEMC_DQS4	-	-	-	-	-	-	-	-
SEMC_CLKX0	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK
SEMC_CLKX1	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK

29.5 Initialization

To initialize the block:

- Ensure the clock for the SEMC is enabled.
- Configure corresponding pin MUX for the SEMC external signals.
- Configure registers of the SEMC properly, according to the applications.
- Configure the Module Control Register and clear MCR[MDIS] bit to enable the SEMC module.
- Reset SEMC optionally by setting the MCR[SWRST] bit.

29.6 Application Information

This section describes applications supported by the SEMC module.

29.6.1 SDRAM Application

This section provides the example sequences for SDRAM device.

- Configure MCR[DQSMD] bit to select the read clock source. Suggest to set it with 0x1 to reach high clock frequency.
- Configure IOCR register to choose chip select pins.
- Configure BMCR0/1 registers to adjust AXI bus access efficiency scheme.
- Configure Base Register 0/1/2/3 with base address, memory size and valid information.
- Configure INTEN and INTR registers if need to generate interrupt.
- Configure SDRAM Control Registers with valid settings that matches SDRAM device.
- Initialize SDRAM device by IP command registers (IPCR0/1/2, IPCMD and IPTXDAT).
- Write and read operations can be triggered by AXI or IP command. However, AXI command is suggested.

29.6.2 NAND Application

This section provides the example sequences for NAND device.

- Configure IOCR register to choose chip select pins.
- Configure BMCR0/1 registers to adjust AXI bus access efficiency scheme.
- Configure Base Register 4/8 with base address, memory size and valid information.
- Configure INTEN and INTR registers if need to generate interrupt.
- Configure NAND Control Registers with valid settings that matches NAND device.
- Configure DLL Control Register for synchronous mode.
- Initialize NAND device by IP command registers (IPCR0/1/2, IPCMD and IPTXDAT).
- For programming operation, need to open page by IP command. Then you can write data to the NAND buffer by IP or AXI bus.
- For read operation, need to open page by IP command. Then you can read data from the NAND buffer by IP or AXI bus.
- For erase operation, you should do it by IP command.

29.6.3 NOR Application

This section provides the example sequences for NOR device.

- Configure MCR[DQSMD] bit to select the read clock source for synchronous mode. Suggest to set it with 0x1 to reach high clock frequency.
- Configure IOCR register to choose chip select pins.
- Configure BMCR0/1 registers to adjust AXI bus access efficiency scheme.
- Configure Base Register 5 with base address, memory size and valid information.
- Configure INTEN and INTR registers if need to generate interrupt.
- Configure NOR Control Registers with valid settings that matches NOR device.
- Initialize NOR device by IP command registers (IPCR0/1/2, IPCMD and IPTXDAT) if needed.
- Programming operation can be triggered by IP command only.
- Read operation can be triggered by AXI or IP command.

29.6.4 SRAM Application

This section provides the example sequences for SRAM device.

- Configure MCR[DQSMD] bit to select the read clock source for synchronous mode. Suggest to set it with 0x1 to reach high clock frequency.
- Configure IOCR register to choose chip select pins.
- Configure BMCR0/1 registers to adjust AXI bus access efficiency scheme.

- Configure Base Register 6/9/10/11 with base address, memory size and valid information.
- Configure INTEN and INTR registers if need to generate interrupt.
- Configure SRAM Control Registers with valid settings that matches SRAM device.
- Initialize SRAM device by IP command registers (IPCR0/1/2, IPCMD and IPTXDAT) if needed.
- Write and read operations can be triggered by AXI or IP command.

29.6.5 DBI Application

This section provides the example sequences for Display Bus Interface.

- Configure IOCR register to choose chip select pins.
- Configure BMCR0/1 registers to adjust AXI bus access efficiency scheme.
- Configure Base Register 7 with base address, memory size and valid information.
- Configure INTEN and INTR registers if need to generate interrupt.
- Configure DBI Control Registers with valid settings that matches DBI device.
- Write and read operations can be triggered by AXI or IP command.

29.7 Memory Map and Register Definition

This section includes the SEMC module memory map and detailed descriptions of all registers.

29.7.1 SEMC register descriptions

29.7.1.1 SEMC memory map

SEMC base address: 400D_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Control Register (MCR)	32	RW	1000_0002h
4h	IO MUX Control Register (IOCR)	32	RW	0000_0000h
8h	Bus (AXI) Master Control Register 0 (BMCR0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Bus (AXI) Master Control Register 1 (BMCR1)	32	RW	0000_0000h
10h - 30h	Base Register n (BR0 - BR8)	32	RW	0000_0000h
34h	DLL Control Register (DLLCR)	32	RW	0000_0100h
38h	Interrupt Enable Register (INTEN)	32	RW	0000_0000h
3Ch	Interrupt Register (INTR)	32	RW	0000_0000h
40h	SDRAM Control Register 0 (SDRAMCR0)	32	RW	0000_0C26h
44h	SDRAM Control Register 1 (SDRAMCR1)	32	RW	0099_4934h
48h	SDRAM Control Register 2 (SDRAMCR2)	32	RW	8000_0EEEh
4Ch	SDRAM Control Register 3 (SDRAMCR3)	32	RW	4080_8000h
50h	NAND Control Register 0 (NANDCR0)	32	RW	0000_0000h
54h	NAND Control Register 1 (NANDCR1)	32	RW	0000_0000h
58h	NAND Control Register 2 (NANDCR2)	32	RW	0001_0410h
5Ch	NAND Control Register 3 (NANDCR3)	32	RW	0000_0000h
60h	NOR Control Register 0 (NORCR0)	32	RW	0000_0000h
64h	NOR Control Register 1 (NORCR1)	32	RW	0000_0000h
68h	NOR Control Register 2 (NORCR2)	32	RW	0000_0000h
6Ch	NOR Control Register 3 (NORCR3)	32	RW	0000_0000h
70h	SRAM Control Register 0 (SRAMCR0)	32	RW	0000_0000h
74h	SRAM Control Register 1 (SRAMCR1)	32	RW	0000_0000h
78h	SRAM Control Register 2 (SRAMCR2)	32	RW	0000_0000h
7Ch	SRAM Control Register 3 (SRAMCR3)	32	RW	0000_0000h
80h	DBI-B Control Register 0 (DBICR0)	32	RW	0000_0000h
84h	DBI-B Control Register 1 (DBICR1)	32	RW	0000_0000h
88h	DBI-B Control Register 2 (DBICR2)	32	RW	0000_0000h
90h	IP Command Control Register 0 (IPCR0)	32	RW	0000_0000h
94h	IP Command Control Register 1 (IPCR1)	32	RW	0000_0000h
98h	IP Command Control Register 2 (IPCR2)	32	RW	0000_0000h
9Ch	IP Command Register (IPCMD)	32	RW	0000_0000h
A0h	TX DATA Register (IPTXDAT)	32	RW	0000_0000h
B0h	RX DATA Register (IPRXDAT)	32	R	0000_0000h
C0h	Status Register 0 (STS0)	32	R	0000_0001h
C4h	Status Register 1 (STS1)	32	R	0000_0000h
C8h	Status Register 2 (STS2)	32	R	0000_0000h
CCh	Status Register 3 (STS3)	32	R	0000_0000h
D0h	Status Register 4 (STS4)	32	R	0000_0000h
D4h	Status Register 5 (STS5)	32	R	0000_0000h
D8h	Status Register 6 (STS6)	32	R	0000_0000h
DCh	Status Register 7 (STS7)	32	R	0000_0000h
E0h	Status Register 8 (STS8)	32	R	0000_0000h
E4h	Status Register 9 (STS9)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
E8h	Status Register 10 (STS10)	32	R	0000_0000h
ECh	Status Register 11 (STS11)	32	R	0000_0000h
F0h	Status Register 12 (STS12)	32	R	0000_0000h
F4h	Status Register 13 (STS13)	32	R	0000_0100h
F8h	Status Register 14 (STS14)	32	R	0000_0000h
FCh	Status Register 15 (STS15)	32	R	0000_0000h
100h	Base Register 9 (BR9)	32	RW	A000_0018h
104h	Base Register 10 (BR10)	32	RW	A400_0018h
108h	Base Register 11 (BR11)	32	RW	A800_0018h
120h	SRAM Control Register 4 (SRAMCR4)	32	RW	0000_0000h
124h	SRAM Control Register 5 (SRAMCR5)	32	RW	0000_0000h
128h	SRAM Control Register 6 (SRAMCR6)	32	RW	0000_0000h
150h	Delay Chain Control Register (DCCR)	32	RW	0000_0000h

29.7.1.2 Module Control Register (MCR)

29.7.1.2.1 Offset

Register	Offset
MCR	0h

29.7.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			BTO					CTO							
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								WPOL1	WPOL0	Reserved			DQSMD	MDIS	SWRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

29.7.1.2.3 Fields

Field	Function
31-29 —	Reserved
28-24 BTO	Bus timeout cycles The SEMC monitors AXI transactions in the queue. When it exceeds the timeout cycles, an AXIBUSERR interrupt is generated. The AXI Bus timeout is 255*2^{BTO} clock cycles. 0_0000b - 255*1 0_0001b - 255*2 1_1111b - 255*2^31
23-16 CTO	Command Execution timeout cycles When Command Execution time exceeds the timeout cycles, an IPCMDERR or AXICMDERR interrupt is generated. When CTO is set to zero, timeout cycles is 256*1024 cycle. Otherwise timeout cycles is CTO*1024 cycle.
15-8 —	Reserved
7 WPOL1	R/B# polarity for NAND device This bit configures the expected polarity of R/B# signal from NAND device. 0b - R/B# polarity is not changed. 1b - R/B# polarity is inverted.
6 WPOL0	WAIT/RDY polarity for SRAM/NOR This bit configures the expected polarity of WAIT/RDY signal from SRAM and NOR devices. 0b - WAIT/RDY polarity is not changed. 1b - WAIT/RDY polarity is inverted.
5-3 —	Reserved
2 DQSMD	DQS (read strobe) mode DQSMD controls the read clock source of SDRAM, NOR and SRAM in synchronous mode. When it set, the read clock is loopbacked from DQS pad, it benefits the read path timing, and can get high read clock frequency. Delay Chain Control Register (DCCR) register can slightly adjust the DQS timing if DQS is faster than read data in sample phase. 0b - Dummy read strobe loopbacked internally 1b - Dummy read strobe loopbacked from DQS pad
1 MDIS	Module Disable When this bit set, the SEMC internal clock is stopped, and no function is available except register access. 0b - Module enabled 1b - Module disabled
0 SWRST	Software Reset Reset all internal logic in the SEMC except configuration registers. It is an auto clear bit. 0b - No reset 1b - Reset

29.7.1.3 IO MUX Control Register (IOCR)

29.7.1.3.1 Offset

Register	Offset
IOCR	4h

29.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		CLKX1_AO	CLKX0_AO	MUX_CLKX1		MUX_CLKX0		MUX_RDY				MUX_CSX3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MUX_CSX2				MUX_CSX1				MUX_CSX0				MUX_A8			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.3.3 Fields

Field	Function
31-30 —	Reserved
29 CLKX1_AO	SEMC_CLKX1 Always On This field controls the clock gating of SEMC_CLKX1 signal. 0b - SEMC_CLKX1 is controlled by MUX_CLKX1 1b - SEMC_CLKX1 is always on
28 CLKX0_AO	SEMC_CLKX0 Always On This field controls the clock gating of SEMC_CLKX0 signal. 0b - SEMC_CLKX0 is controlled by MUX_CLKX0 1b - SEMC_CLKX0 is always on
27-26 MUX_CLKX1	SEMC_CLKX1 function selection This field controls the clock source of SEMC_CLKX1 signal. 00b - Keep low 01b - NOR clock 10b - SRAM clock 11b - NOR and SRAM clock, suitable for Multi-Chip Product package
25-24 MUX_CLKX0	SEMC_CLKX0 function selection This field controls the clock source of SEMC_CLKX0 signal. 00b - Keep low 01b - NOR clock

Table continues on the next page...

Field	Function
	10b - SRAM clock 11b - NOR and SRAM clock, suitable for Multi-Chip Product package
23-20 MUX_RDY	SEMC_RDY function selection This field controls the function of SEMC_RDY signal. 0000b - NAND R/B# input 0001b - SDRAM CS1 0010b - SDRAM CS2 0011b - SDRAM CS3 0100b - NOR/SRAM Address bit 27 (A27) in Non-ADMUX mode 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX 1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - NOR/SRAM Address bit 27 in Non-ADMUX mode
19-16 MUX_CSX3	SEMC_CSX3 output selection This field controls the function of SEMC_CSX3 signal. 0000b - NOR/SRAM Address bit 27 (A27) in Non-ADMUX mode 0001b - SDRAM CS1 0010b - SDRAM CS2 0011b - SDRAM CS3 0100b - NAND CE# 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX 1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - NOR/SRAM Address bit 27 (A27) in Non-ADMUX mode
15-12 MUX_CSX2	SEMC_CSX2 output selection This field controls the function of SEMC_CSX2 signal. 0000b - NOR/SRAM Address bit 26 (A26) in Non-ADMUX mode 0001b - SDRAM CS1 0010b - SDRAM CS2 0011b - SDRAM CS3 0100b - NAND CE# 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX 1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - NOR/SRAM Address bit 26 (A26) in Non-ADMUX mode
11-8 MUX_CSX1	SEMC_CSX1 output selection This field controls the function of SEMC_CSX1 signal. 0000b - NOR/SRAM Address bit 25 (A25) in Non-ADMUX mode 0001b - SDRAM CS1 0010b - SDRAM CS2 0011b - SDRAM CS3 0100b - NAND CE# 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX

Table continues on the next page...

Field	Function
	1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - NOR/SRAM Address bit 25 (A25) in Non-ADMUX mode
7-4 MUX_CSX0	SEMC_CSX0 output selection This field controls the function of SEMC_CSX0 signal. 0000b - NOR/SRAM Address bit 24 (A24) in Non-ADMUX mode 0001b - SDRAM CS1 0010b - SDRAM CS2 0011b - SDRAM CS3 0100b - NAND CE# 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX 1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - NOR/SRAM Address bit 24 (A24) in Non-ADMUX mode
3-0 MUX_A8	SEMC_ADDR08 output selection This field controls the function of SEMC_ADDR08 signal. 0000b-0011b - SDRAM Address bit 8 (A8) or NOR/SRAM Address bit 24 (A24) in ADMUX 16bit mode 0100b - NAND CE# 0101b - NOR CE# 0110b - SRAM CE# 0 0111b - DBI CSX 1000b - SRAM CE# 1 1001b - SRAM CE# 2 1010b - SRAM CE# 3 1011b-1111b - SDRAM Address bit 8 (A8) or NOR/SRAM Address bit 24 (A24) in ADMUX 16bit mode

29.7.1.4 Bus (AXI) Master Control Register 0 (BMCR0)

29.7.1.4.1 Offset

Register	Offset
BMCR0	8h

29.7.1.4.2 Function

Queue A weight settings for AXI bus access to SDRAM, NAND, NOR, SRAM and DBI slaves.

NOTE

Please refer to BMCR0 Register Configuration

29.7.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								WRWS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WSH								WAGE				WQOS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.4.4 Fields

Field	Function
31-24 —	Reserved
23-16 WRWS	Weight of slave hit with Read/Write Switch This weight score is valid when the queue command's slave is same as current executing command with read/write operation switch.
15-8 WSH	Weight of Slave Hit without read/write switch This weight score is valid when queue command's slave is the same as current executing command without read/write operation switch.
7-4 WAGE	Weight of AGE Each command in queue has an age score to indicate its wait period. The age score is multiplied by WAGE to get the weight score. Recommend to use non-zero value.
3-0 WQOS	Weight of QOS AXI bus access has AxQOS signal set, which is used as a priority indicator for the associated write or read transaction. A higher value indicates a higher priority transaction. AxQOS is multiplied by WQOS to get the weight score.

29.7.1.5 Bus (AXI) Master Control Register 1 (BMCR1)

29.7.1.5.1 Offset

Register	Offset
BMCR1	Ch

29.7.1.5.2 Function

Queue B weight settings for AXI bus access to SDRAM slave.

NOTE

Please refer to BMCR1 Register Configuration

29.7.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WBR								WRWS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WPH								WAGE				WQOS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.5.4 Fields

Field	Function
31-24 WBR	Weight of Bank Rotation This weight score is valid when queue command's bank is not same as current executing command.
23-16 WRWS	Weight of slave hit without Read/Write Switch This weight score is valid when queue command's read/write operation is same as current executing command.
15-8 WPH	Weight of Page Hit This weight score is valid when queue command's page hits current executing command.
7-4 WAGE	Weight of AGE Each command in queue has an age signal to indicate its wait period. It is multiplied by WAGE to get the weight score. Recommend to use non-zero value.
3-0 WQOS	Weight of QOS AXI bus access has AxQOS signal set, which is used as a priority indicator for the associated write or read transaction. A higher value indicates a higher priority transaction. AxQOS is multiplied by WQOS to get the weight score.

29.7.1.6 Base Register n (BR0 - BR8)

29.7.1.6.1 Offset

For n = 0 to 8:

Register	Offset
BRn	10h + (n × 4h)

29.7.1.6.2 Function

Table 29-8 shows the detail information about the Base Registers.

Table 29-8. Base Register Information

Register	Reset Value	Purpose	Comments
BR0	0x8000001D	For SDRAM CS0	For AXI and IP commands
BR1	0x8400001C	For SDRAM CS1	For AXI and IP commands
BR2	0x8800001C	For SDRAM CS2	For AXI and IP commands
BR3	0x8C00001C	For SDRAM CS3	For AXI and IP commands
BR4	0x9E00001A	For NAND	For AXI command only
BR5	0x90000018	For NOR	For AXI and IP commands
BR6	0x98000018	For SRAM CS0	For AXI and IP commands
BR7	0x9C00001A	For DBI-B	For AXI and IP commands
BR8	0x00000026	For NAND	For IP command only

NOTE

Base Register 9 (BR9), Base Register 10 (BR10) and Base Register 11 (BR11) show Base Registers for other SRAM devices.

29.7.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BA				Reserved						MS					VLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.6.4 Fields

Field	Function
31-12 BA	Base Address This field determines high position 20 bits of SoC level base address. The low position 12 bits of the address are all zero. The base address must be within the SEMC range defined for the device memory map.
11-6 —	Reserved
5-1 MS	Memory size It defines the size of the memory. Memory size must be aligned with base address. Memory regions should be configured to avoid overlapping areas. Overlapping regions will cause bus errors. 0_0000b - 4KB 0_0001b - 8KB 0_0010b - 16KB 0_0011b - 32KB 0_0100b - 64KB 0_0101b - 128KB 0_0110b - 256KB 0_0111b - 512KB 0_1000b - 1MB 0_1001b - 2MB 0_1010b - 4MB 0_1011b - 8MB 0_1100b - 16MB 0_1101b - 32MB 0_1110b - 64MB 0_1111b - 128MB 1_0000b - 256MB 1_0001b - 512MB 1_0010b - 1GB 1_0011b - 2GB 1_0100b-1_1111b - 4GB
0 VLD	Valid This bit controls whether the memory is valid or not. 0b - The memory is invalid, can not be accessed. 1b - The memory is valid, can be accessed.

29.7.1.7 DLL Control Register (DLLCR)

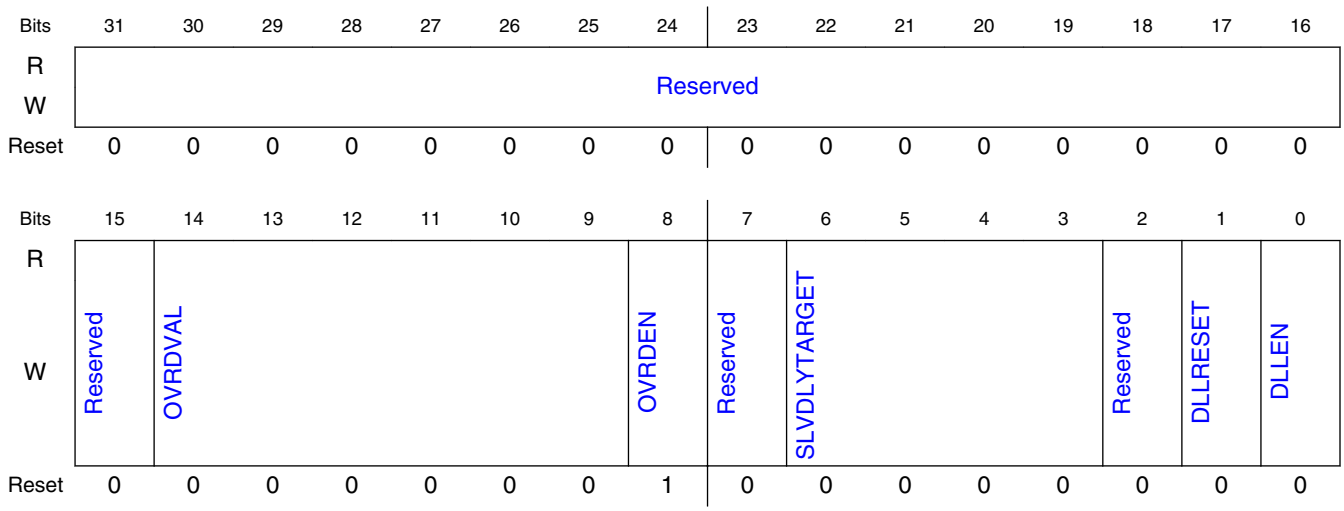
29.7.1.7.1 Offset

Register	Offset
DLLCR	34h

29.7.1.7.2 Function

This register provides the configuration fields for sample clock from DLL. It is for NAND only.

29.7.1.7.3 Diagram



29.7.1.7.4 Fields

Field	Function
31-15 —	Reserved
14-9 OVRDVAL	Override Value When OVRDEN is set, the delay cell number is OVRDVAL+1 for slave clock delay line.
8 OVRDEN	Override Enable Slave clock delay line delay cell number selection override enable. 0b - The delay cell number is not overridden. 1b - The delay cell number is overridden.
7 —	Reserved
6-3 SLVDLYTARGET	Delay Target for Slave The delay target for slave delay line is: ((SLVDLYTARGET+1) * 1/32 * clock cycle of reference clock (ipgclock).
2 —	Reserved
1 DLLRESET	DLL Reset Software can force a reset on DLL by setting this field to 0x1. This causes the DLL to lose lock and re-calibrate to detect an ref_clock half period phase shift. The reset action is edge triggered, so software need to clear this bit after setting this bit (no delay limitation).

Table continues on the next page...

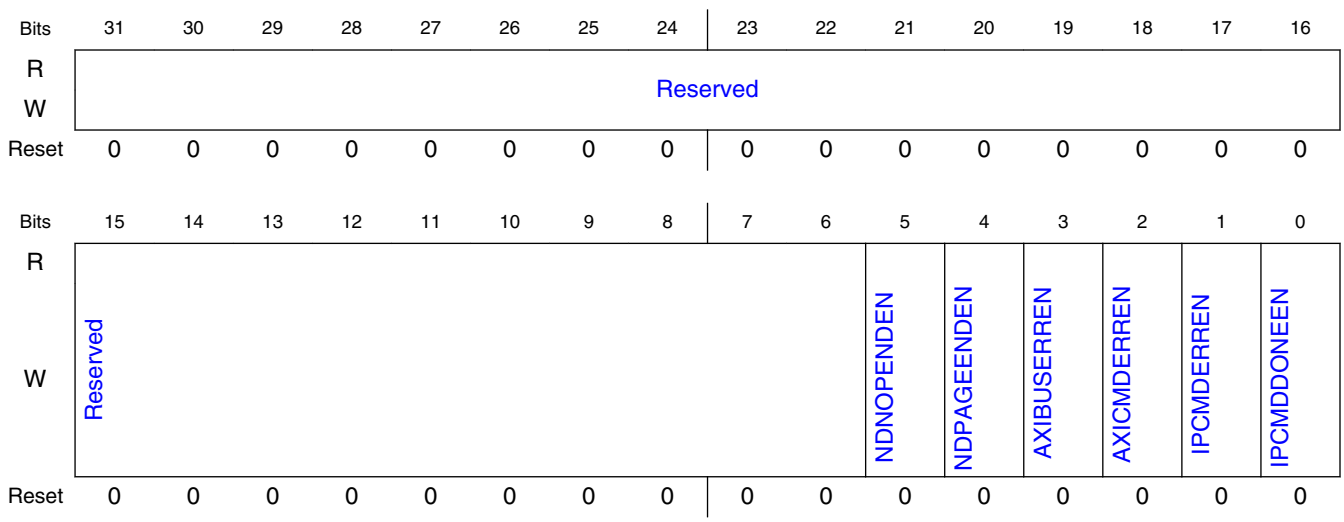
Field	Function
	0b - DLL is not reset. 1b - DLL is reset.
0 DLLEN	DLL calibration enable When this bit is cleared, DLL calibration is disabled and the delay cell number in slave delay line is always 1. Please note that SLV delay line is overridden when OVRDEN bit is set and this bit field setting is ignored. 0b - DLL calibration is disabled. 1b - DLL calibration is enabled.

29.7.1.8 Interrupt Enable Register (INTEN)

29.7.1.8.1 Offset

Register	Offset
INTEN	38h

29.7.1.8.2 Diagram



29.7.1.8.3 Fields

Field	Function
31-6 —	Reserved

Table continues on the next page...

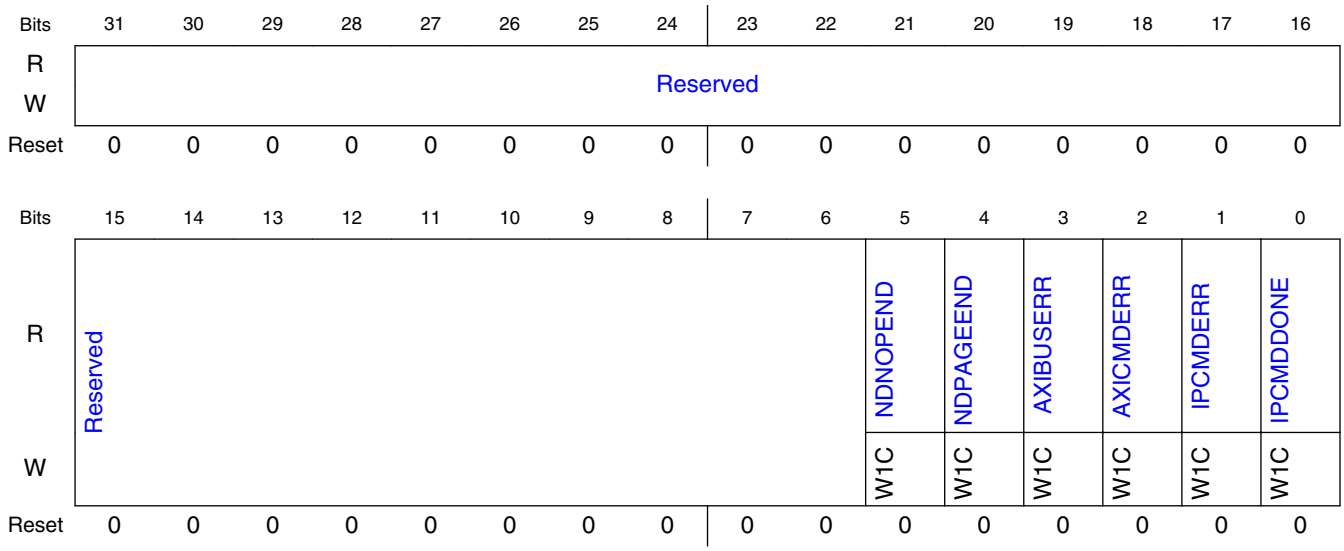
Field	Function
5 NDNOPENDEN	NAND no pending AXI access interrupt enable This bit enable/disable the NDNOPEND interrupt generation. 0b - Interrupt is disabled 1b - Interrupt is enabled
4 NDPAGEENDEN	NAND page end interrupt enable This bit enable/disable the NDPAGEEND interrupt generation. 0b - Interrupt is disabled 1b - Interrupt is enabled
3 AXIBUSERREN	AXI bus error interrupt enable 0b - Interrupt is disabled 1b - Interrupt is enabled
2 AXICMDERREN	AXI command error interrupt enable 0b - Interrupt is disabled 1b - Interrupt is enabled
1 IPCMDERREN	IP command error interrupt enable 0b - Interrupt is disabled 1b - Interrupt is enabled
0 IPCMDDONEEN	IP command done interrupt enable 0b - Interrupt is disabled 1b - Interrupt is enabled

29.7.1.9 Interrupt Register (INTR)

29.7.1.9.1 Offset

Register	Offset
INTR	3Ch

29.7.1.9.2 Diagram



29.7.1.9.3 Fields

Field	Function
31-6 —	Reserved
5 NDNOPEND	NAND no pending AXI write transaction interrupt This interrupt is generated when all pending AXI write transactions to NAND are finished. 0b - At least one NAND AXI write transaction is pending or no NAND write transaction is sent to the queue. 1b - All NAND AXI write pending transactions are finished.
4 NDPAGEEND	NAND page end interrupt This interrupt is generated when the last address of one page in NAND device is written by AXI command. 0b - The last address of main space in the NAND is not written by AXI command. 1b - The last address of main space in the NAND is written by AXI command.
3 AXIBUSERR	AXI bus error interrupt AXI Bus error interrupt is generated in following cases: • AXI address is invalid • AXI write to NOR Flash • AXI 8-bit write to 16-bit NAND Flash • AXI 8-bit or 16-bit WRAP write/read 0b - No AXI bus error. 1b - AXI bus error occurs.
2 AXICMDERR	AXI command error interrupt AXI command error interrupt is generated when AXI command execution times out. 0b - No AXI command error. 1b - AXI command error occurs.
1	IP command error done interrupt

Table continues on the next page...

Field	Function
IPCMDERR	IP command error interrupt is generated in following cases: • IP Command Address target invalid device space • IP Command Code unsupported • IP Command triggered when previous command not finished yet • IP Command Execution timeout 0b - No IP command error. 1b - IP command error occurs.
0 IPCMDDONE	IP command normal done interrupt It monitors whether IP command is done or not. 0b - IP command is not done. 1b - IP command is done.

29.7.1.10 SDRAM Control Register 0 (SDRAMCR0)

29.7.1.10.1 Offset

Register	Offset
SDRAMCR0	40h

29.7.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	BANK2	Reserved		CL		COL		COL8	BL			Reserved		PS	
Reset	0	0	0	0	1	1	0	0	0	0	1	0	0	1	1	0

29.7.1.10.3 Fields

Field	Function
31-15 —	Reserved

Table continues on the next page...

Field	Function
14 BANK2	2 Bank selection bit This bit configures the bank numbers. 0b - SDRAM device has 4 banks. 1b - SDRAM device has 2 banks.
13-12 —	Reserved
11-10 CL	CAS Latency The CAS latency (CL) is the delay, in clock cycles, between the registration of a READ command and the availability of the output data. 00b - 1 01b - 1 10b - 2 11b - 3
9-8 COL	Column address bit number 00b - 12 01b - 11 10b - 10 11b - 9
7 COL8	Column 8 selection This bit works with COL field to determine the column address bit number. 0b - Column address bit number is decided by COL field. 1b - Column address bit number is 8. COL field is ignored.
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 8 101b - 8 110b - 8 111b - 8
3-2 —	Reserved
1-0 PS	Port Size The port size must be aligned with SDRAM data width. 00b - 8bit 01b - 16bit 10b - 32bit 11b - Reserved

29.7.1.11 SDRAM Control Register 1 (SDRAMCR1)

29.7.1.11.1 Offset

Register	Offset
SDRAMCR1	44h

29.7.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ACT2PRE				CKEOFF			
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WRC			RFRC					ACT2RW				PRE2ACT			
Reset	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0

29.7.1.11.3 Fields

Field	Function
31-24 —	Reserved
23-20 ACT2PRE	ACTIVE to PRECHARGE minimum time The SEMC promises ACT2PRE+1 clock cycles delay between ACTIVE command to PRECHARGE/ PRECHARGE ALL command. It helps to meet tRAS timing requirement by SDRAM device.
19-16 CKEOFF	CKE off minimum time The SEMC promises at least CKEOFF+1 clock cycles in self refresh mode. It helps to meet the minimum period request to remain in self refresh mode.
15-13 WRC	WRITE recovery time The SEMC promises WRC+1 clock cycles delay between WRITE command to PRECHARGE/ PRECHARGE ALL command. It helps to meet tWR timing requirement by SDRAM device.
12-8 RFRC	REFRESH recovery time The SEMC promises RFRC+1 clock cycles delay between AUTO REFRESH command to ACTIVE command. It helps to meet tRFC timing requirement by SDRAM device.
7-4 ACT2RW	ACTIVE to READ/WRITE delay The SEMC promises ACT2RW+1 clock cycles delay between ACTIVE command to READ/WRITE command. It helps to meet tRCD timing requirement by SDRAM device.
3-0 PRE2ACT	PRECHARGE to ACTIVE/REFRESH command wait time The SEMC promises PRE2ACT+1 clock cycles delay between PRECHARGE/PRECHARGE ALL command to ACTIVE/REFRESH command. It helps to meet tRP timing requirement by SDRAM device.

29.7.1.12 SDRAM Control Register 2 (SDRAMCR2)

29.7.1.12.1 Offset

Register	Offset
SDRAMCR2	48h

29.7.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ITO								ACT2ACT							
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REF2REF								SRRC							
Reset	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0

29.7.1.12.3 Fields

Field	Function
31-24 ITO	SDRAM idle timeout The SEMC closes all opened pages if the SDRAM idle time lasts more than idle timeout period. SDRAM is considered idle when there is no AXI Bus transfer and no SDRAM command pending. 0000_0000b - IDLE timeout period is 256*Prescale period. 0000_0001b-1111_1111b - IDLE timeout period is ITO*Prescale period.
23-16 ACT2ACT	ACTIVE to ACTIVE delay The SEMC promises ACT2ACT+1 clock cycles delay between ACTIVE command to ACTIVE command. It helps to meet tRRD timing requirement by SDRAM device.
15-8 REF2REF	REFRESH to REFRESH delay The SEMC promises REF2REF+1 clock cycles delay between REFRESH command to REFRESH command. It helps to meet tRFC timing requirement by SDRAM device.
7-0 SRRC	SELF REFRESH recovery time The SEMC promises SRRC+1 clock cycles delay between SELF REFRESH command to any subsequent command. SDRAM device may have limitation on self refresh recovery time. It help to meet the wait time from exiting self refresh mode to sending new command to device.

29.7.1.13 SDRAM Control Register 3 (SDRAMCR3)

29.7.1.13.1 Offset

Register	Offset
SDRAMCR3	4Ch

29.7.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UT								RT							
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRESCALE								Reserved				REBL			REN
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.13.3 Fields

Field	Function
31-24 UT	Urgent refresh threshold Urgent refresh threshold is used to handle urgent refresh request. Urgent refresh timer starts to count when normal refresh request is asserted but not handled by the controller yet. When the urgent refresh timer count up to the threshold, urgent refresh request is sent to the controller. The priority of urgent refresh is higher than any pending AXI or IP command to SDRAM, so it can be handled by the controller at the first time. **NOTE:** Refresh request is considered as urgent refresh request when urgent threshold is no less than refresh period. 0000_0000b - 256*(Prescaler period) 0000_0001b-1111_1111b - UT*(Prescaler period)
23-16 RT	Refresh timer period Refresh timer period is the threshold for SDRAM normal refresh request. The normal refresh request is sent to the controller when the refresh timer count up to it. The priority of normal refresh request is lower than any pending AXI or IP command to SDRAM. 0000_0000b - (256+1)*(Prescaler period) 0000_0001b-1111_1111b - (RT+1)*(Prescaler period)
15-8 PRESCALE	Prescaler period Prescaler period is used as a clock cycle count unit for refresh and bus idle timer period.

Table continues on the next page...

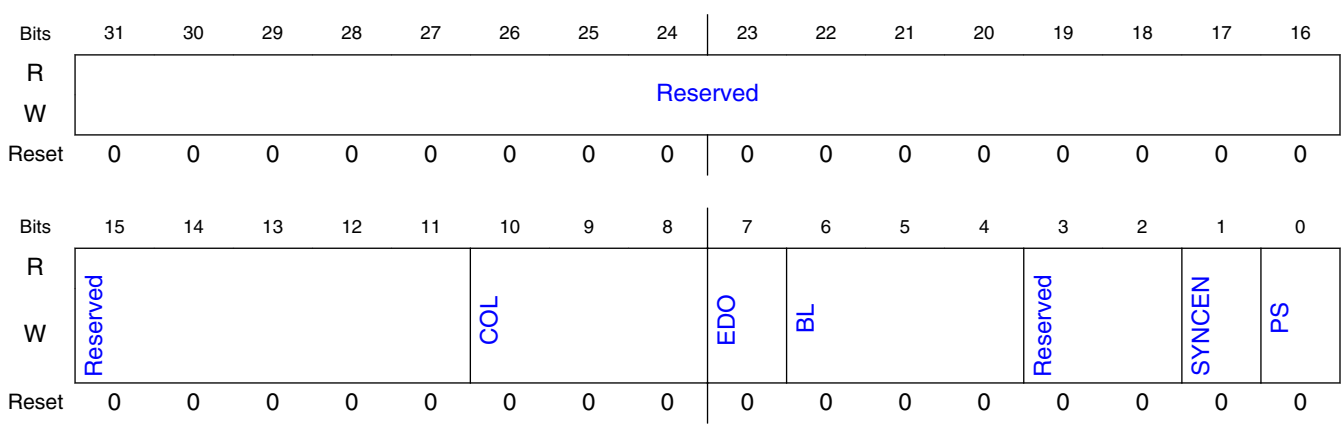
Field	Function
	0000_0000b - (256*16+1) clock cycles 0000_0001b-1111_1111b - (PRESCALE*16+1) clock cycles
7-4 —	Reserved
3-1 REBL	Refresh burst length The SEMC can send multiple AUTO REFRESH commands in one burst when REBL is set to non-zero. The number of AUTO REFRESH command in one burst is listed below. 000b - 1 001b - 2 010b - 3 011b - 4 100b - 5 101b - 6 110b - 7 111b - 8
0 REN	Refresh enable The SEMC sends AUTO REFRESH command automatically when REN bit is set. The AUTO REFRESH command interval is decided by UT, RT and PRESCALE bits. 0b - The SEMC does not send AUTO REFRESH command automatically 1b - The SEMC sends AUTO REFRESH command automatically

29.7.1.14 NAND Control Register 0 (NANDCR0)

29.7.1.14.1 Offset

Register	Offset
NANDCR0	50h

29.7.1.14.2 Diagram



29.7.1.14.3 Fields

Field	Function
31-11 —	Reserved
10-8 COL	Column address bit number 000b - 16 001b - 15 010b - 14 011b - 13 100b - 12 101b - 11 110b - 10 111b - 9
7 EDO	EDO mode enabled It defines the read access timing in asynchronous mode. 0b - EDO mode disabled 1b - EDO mode enabled
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 64
3-2 —	Reserved
1 SYNCEN	Synchronous Mode Enable It defines the access mode. 0b - Asynchronous mode is enabled. 1b - Synchronous mode is enabled.
0 PS	Port Size The port size must be aligned with NAND Flash data width. 0b - 8bit 1b - 16bit

29.7.1.15 NAND Control Register 1 (NANDCR1)

29.7.1.15.1 Offset

Register	Offset
NANDCR1	54h

29.7.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CEITV				TA				REH				REL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WEH				WEL				CEH				CES			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.15.3 Fields

Field	Function
31-28	CE# interval time
CEITV	CE# interval minimum time is CEITV+1 clock cycles.
27-24	Turnaround time
TA	Turnaround time is TA+1 clock cycles. Both the SEMC and Device do not drive Data Lines to avoid bus confliction. This field applied to ASYNC mode only.
23-20	RE# high time
REH	RE# high time is REH+1 clock cycles. This field applied to ASYNC mode only.
19-16	RE# low time
REL	RE# low time is REL+1 clock cycles. This field applied to ASYNC mode only.
15-12	WE# high time
WEH	WE# high time is WEH+1 clock cycles in ASYNC mode. It is used as command and address hold time in SYNC mode. WEH must be even number in SYNC mode, and command/address hold time is WEH clock cycles.
11-8	WE# low time
WEL	WE# low time is WEL+1 clock cycles in ASYNC mode. It is used as command and address setup time in SYNC mode. WEL must be even number in SYNC mode, and command/address setup time is WEL+2 clock cycles.
7-4	CE# hold time
CEH	CE# hold minimum time is CEH+1 clock cycles. CEH must be odd number in SYNC mode. CE# hold time may be larger than this time to wait all read data sampled.
3-0	CE# setup time

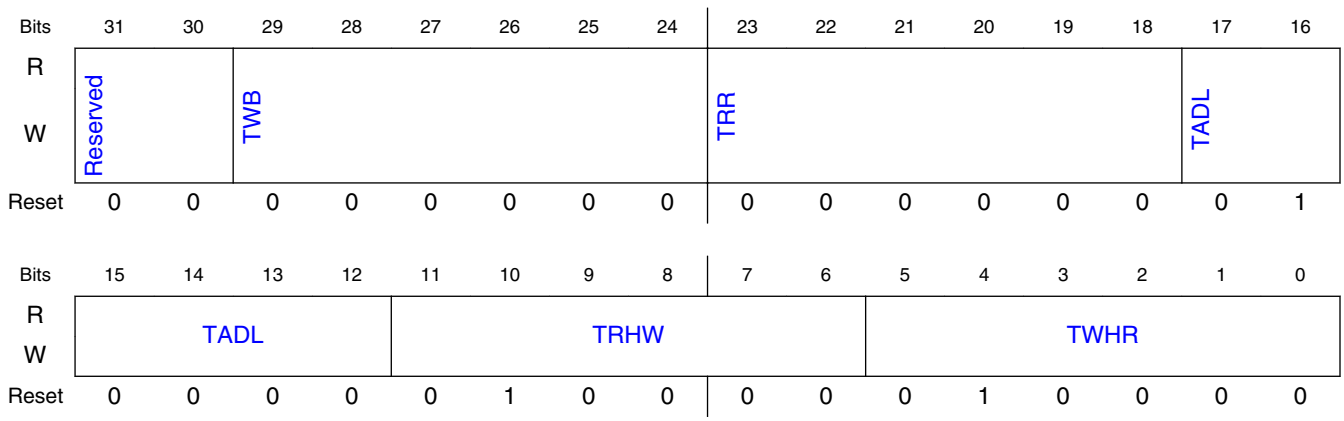
Field	Function
CES	CE# setup time is CES+1 clock cycles. CES must be odd number in SYNC mode.

29.7.1.16 NAND Control Register 2 (NANDCR2)

29.7.1.16.1 Offset

Register	Offset
NANDCR2	58h

29.7.1.16.2 Diagram



29.7.1.16.3 Fields

Field	Function
31-30 —	Reserved
29-24 TWB	WE# high to busy time WE# high to busy time is TWB+1 clock cycles
23-18 TRR	Ready to RE# low time Ready to RE# low minimum time is TRR+1 clock cycles. Actual time may be 1 or 2 clock cycles more than this because of synchronizer logic. This field applied to ASYNC mode only.
17-12 TADL	Address cycle to data loading time Address to data loading time is TADL+1 clock cycles. Timing for TADL begins in the address cycle on the final rising edge of WE#, and ends with the first rising edge of WE# for data input.

Table continues on the next page...

Field	Function
11-6 TRHW	RE# high to WE# low time RE# high to WE# low time is TRHW+1 clock cycles
5-0 TWHR	WE# high to RE# low time WE# high to RE# low time is TWHR+1 clock cycles in ASYNC mode. TWHR represents command, address or data input cycle to data output clock cycles in SYNC mode. The delay is TWHR+2 clock cycles and TWHR must be even.

29.7.1.17 NAND Control Register 3 (NANDCR3)

29.7.1.17.1 Offset

Register	Offset
NANDCR3	5Ch

29.7.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WDH				WDS				RDH				RDS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												CLE	NDOPT3	NDOPT2	NDOPT1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.17.3 Fields

Field	Function
31-28 WDH	Write Data Hold time Write data hold time is WDH+2 clock cycles, and WDH must be even number. This field applied to SYNC mode only.
27-24 WDS	Write Data Setup time Write data setup time is WDS+2 clock cycles, and WDS must be even number. This field applied to SYNC mode only.

Table continues on the next page...

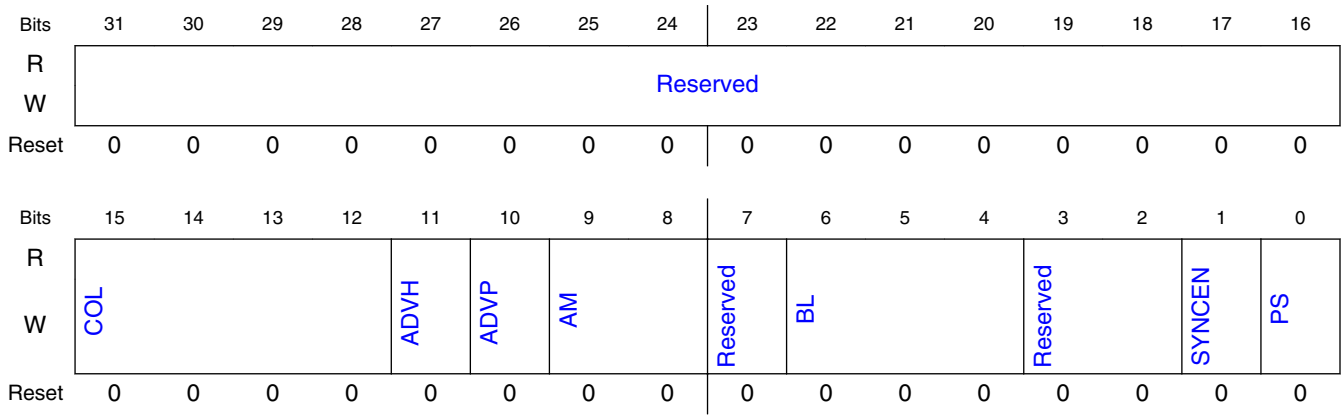
Field	Function
23-20 RDH	Read Data Hold time Read data hold time is RDH+2 clock cycles, and RDH must be even number. This field applied to SYNC mode only.
19-16 RDS	Read Data Setup time Read data setup time is RDS+2 clock cycles, and RDS must be even number. It provides the minimum timing control before data output phase. This field applied to SYNC mode only.
15-4 —	Reserved
3 CLE	NAND CLE Option This bit is intended to add a CLE valid clock cycle before WE# valid clock cycle for ASYNC mode.
2 NDOPT3	NAND option bit 3 This bit is intended to bypass row address byte #1 for address mode 0x0 - Column and Row address(5 Byte-CA0/CA1/RA0/RA1/RA2).
1 NDOPT2	NAND option bit 2 This bit is intended to bypass row address byte #2 for address mode 0x0 - Column and Row address(5 Byte-CA0/CA1/RA0/RA1/RA2).
0 NDOPT1	NAND option bit 1 This bit is intended to bypass column address byte #1 for address mode 0x0 - Column and Row address(5 Byte-CA0/CA1/RA0/RA1/RA2). **NOTE:** Please refer to NAND Flash Access Address Type

29.7.1.18 NOR Control Register 0 (NORCR0)

29.7.1.18.1 Offset

Register	Offset
NORCR0	60h

29.7.1.18.2 Diagram



29.7.1.18.3 Fields

Field	Function
31-16 —	Reserved
15-12 COL	Column Address bit width 0000b - 12 Bits 0001b - 11 Bits 0010b - 10 Bits 0011b - 9 Bits 0100b - 8 Bits 0101b - 7 Bits 0110b - 6 Bits 0111b - 5 Bits 1000b - 4 Bits 1001b - 3 Bits 1010b - 2 Bits 1011b - 12 Bits 1100b - 12 Bits 1101b - 12 Bits 1110b - 12 Bits 1111b - 12 Bits
11 ADVH	ADV# level control during address hold state 0b - ADV# is high during address hold state. 1b - ADV# is low during address hold state.
10 ADVP	ADV# Polarity 0b - ADV# is active low. 1b - ADV# is active high.
9-8 AM	Address Mode It defines the address mode. 00b - Address/Data MUX mode (ADMUX) 01b - Advanced Address/Data MUX mode (AADM) 10b - Address/Data non-MUX mode (Non-ADMUX) 11b - Address/Data non-MUX mode (Non-ADMUX)
7	Reserved

Table continues on the next page...

Field	Function
—	
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 64
3-2 —	Reserved
1 SYNCEN	Synchronous Mode Enable It defines the access mode. 0b - Asynchronous mode is enabled. 1b - Synchronous mode is enabled. Only fixed latency mode is supported.
0 PS	Port Size The port size must be aligned with NOR Flash data width. 0b - 8bit 1b - 16bit

29.7.1.19 NOR Control Register 1 (NORCR1)

29.7.1.19.1 Offset

Register	Offset
NORCR1	64h

29.7.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REH				REL				WEH				WEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AH				AS				CEH				CES			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.19.3 Fields

Field	Function
31-28	RE high time
REH	RE high time is REH+1 clock cycles for ASYNC mode.
27-24	RE low time
REL	RE low time is REL+1 clock cycles for ASYNC mode.
23-20	WE high time
WEH	WE high time is WEH+1 clock cycles for ASYNC mode.
19-16	WE low time
WEL	WE low time is WEL+1 clock cycles for ASYNC mode.
15-12	Address hold time
AH	Address hold time is AH+1 clock cycles for ASYNC mode.
11-8	Address setup time
AS	Address setup time is AS+1 clock cycles
7-4	CE hold time
CEH	CE Hold time is CEH+1 clock cycles This field determines the minimum hold time. Actual hold time may be longer in order to wait all read data sampled.
3-0	CE setup time
CES	CE setup time is CES+1 clock cycles.

29.7.1.20 NOR Control Register 2 (NORCR2)

29.7.1.20.1 Offset

Register	Offset
NORCR2	68h

29.7.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDH				CEITV				RD				LC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AWDH				TA				Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.20.3 Fields

Field	Function
31-28 RDH	Read hold time Read hold time is RDH clock cycles. This field applied to SYNC mode only.
27-24 CEITV	CE# interval time CE# interval minimum time is CEITV+1 clock cycles.
23-20 RD	Read time Read time is RD+1 clock cycles. This field applied to SYNC mode only.
19-16 LC	Latency count Latency count is LC clock cycles. This field applied to SYNC mode only.
15-12 AWDH	Address to write data hold time Address to write data hold time is AWDH clock cycles. This field applied to ASYNC mode only.
11-8 TA	Turnaround time Turnaround time is TA+1 clock cycles. Both the SEMC and Device do not drive Data Lines to avoid bus confliction. This field applied to ASYNC mode only.
7-0 —	Reserved

29.7.1.21 NOR Control Register 3 (NORCR3)

29.7.1.21.1 Offset

Register	Offset
NORCR3	6Ch

29.7.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AHSR				ASSR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.21.3 Fields

Field	Function
31-8 —	Reserved
7-4 AHSR	Address hold time for SYNC read Address hold time is AHSR clock cycles for SYNC read.
3-0 ASSR	Address setup time for SYNC read Address setup time ASSR+1 clock cycles for SYNC read.

29.7.1.22 SRAM Control Register 0 (SRAMCR0)

29.7.1.22.1 Offset

Register	Offset
SRAMCR0	70h

29.7.1.22.2 Function

This register configures SRAM device 0

29.7.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COL				ADVH	ADVP	AM		Reserved	BL			WAITSP	WAITEN	SYNCEN	PS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.22.4 Fields

Field	Function
31-16 —	Reserved
15-12 COL	Column Address bit width 0000b - 12 Bits 0001b - 11 Bits 0010b - 10 Bits 0011b - 9 Bits 0100b - 8 Bits 0101b - 7 Bits 0110b - 6 Bits 0111b - 5 Bits 1000b - 4 Bits 1001b - 3 Bits 1010b - 2 Bits 1011b - 12 Bits 1100b - 12 Bits 1101b - 12 Bits 1110b - 12 Bits 1111b - 12 Bits
11 ADVH	ADV# level control during address hold state 0b - ADV# is high during address hold state. 1b - ADV# is low during address hold state.
10 ADVP	ADV# polarity 0b - ADV# is active low. 1b - ADV# is active high.
9-8 AM	Address Mode It defines the address mode. 00b - Address/Data MUX mode (ADMUX) 01b - Advanced Address/Data MUX mode (AADM) 10b - Address/Data non-MUX mode (Non-ADMUX) 11b - Address/Data non-MUX mode (Non-ADMUX)
7	Reserved

Table continues on the next page...

Field	Function
—	
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 64
3 WAITSP	Wait Sample Wait pin is sampled by internal clock if this bit is set. It is ignored when WAITEN bit clear. It should be set for ASYNC mode when WAITEN set. 0b - Wait pin is directly used by the SEMC. 1b - Wait pin is sampled by internal clock before it is used.
2 WAITEN	Wait Enable The SEMC monitors wait pin if this bit set. It is for ASYNC mode only. 0b - The SEMC does not monitor wait pin. 1b - The SEMC monitors wait pin. The SEMC does not transfer/receive data when wait pin is asserted.
1 SYNCEN	Synchronous Mode Enable It defines the access mode. 0b - Asynchronous mode is enabled. 1b - Synchronous mode is enabled. Only fixed latency mode is supported.
0 PS	Port Size The port size must be aligned with SRAM data width. 0b - 8bit 1b - 16bit

29.7.1.23 SRAM Control Register 1 (SRAMCR1)

29.7.1.23.1 Offset

Register	Offset
SRAMCR1	74h

29.7.1.23.2 Function

This register configures SRAM device 0

29.7.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REH				REL				WEH				WEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AH				AS				CEH				CES			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.23.4 Fields

Field	Function
31-28 REH	RE high time RE high time is REH+1 clock cycles. This field applied to ASYNC mode only.
27-24 REL	RE low time RE low time is REL+1 clock cycles. This field applied to ASYNC mode only.
23-20 WEH	WE high time WE high time is WEH+1 clock cycles. This field applied to ASYNC mode only.
19-16 WEL	WE low time WE low time is WEL+1 clock cycles. This field applied to ASYNC mode only.
15-12 AH	Address hold time Address hold time is AH+1 clock cycles for ASYNC mode. Address hold time is AH clock cycles for SYNC mode.
11-8 AS	Address setup time Address setup time is AS+1 clock cycles
7-4 CEH	CE hold time CE Hold time is CEH+1 clock cycles This field determines the minimum hold time. Actual hold time may be larger in order to wait all read data sampled.
3-0 CES	CE setup time CE setup time is CES+1 clock cycles.

29.7.1.24 SRAM Control Register 2 (SRAMCR2)

29.7.1.24.1 Offset

Register	Offset
SRAMCR2	78h

29.7.1.24.2 Function

This register configures SRAM device 0

29.7.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDH				CEITV				RD				LC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AWDH				TA				WDH				WDS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.24.4 Fields

Field	Function
31-28	Read hold time
RDH	Read hold time is RDH clock cycles. This field applied to SYNC mode only.
27-24	CE# interval time
CEITV	CE# interval minimum time is CEITV+1 clock cycles.
23-20	Read time
RD	Read time is RD+1 clock cycles. This field applied to SYNC mode only.
19-16	Latency count
LC	Latency count is LC clock cycles. This field applied to SYNC mode only.
15-12	Address to write data hold time
AWDH	Address to write data hold time is AWDH clock cycles. This field applied to ASYNC mode only.
11-8	Turnaround time
TA	Turnaround time is TA+1 clock cycles. Both the SEMC and Device do not drive Data Lines to avoid bus confliction. This field applied to ASYNC mode only.
7-4	Write Data hold time
WDH	Write data hold time is WDH clock cycles. This field applied to SYNC mode only.
3-0	Write Data setup time

Field	Function
WDS	Write data setup time is WDS+1 clock cycles. This field applied to SYNC mode only.

29.7.1.25 SRAM Control Register 3 (SRAMCR3)

29.7.1.25.1 Offset

Register	Offset
SRAMCR3	7Ch

29.7.1.25.2 Function

This register configures SRAM device 0

29.7.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.25.4 Fields

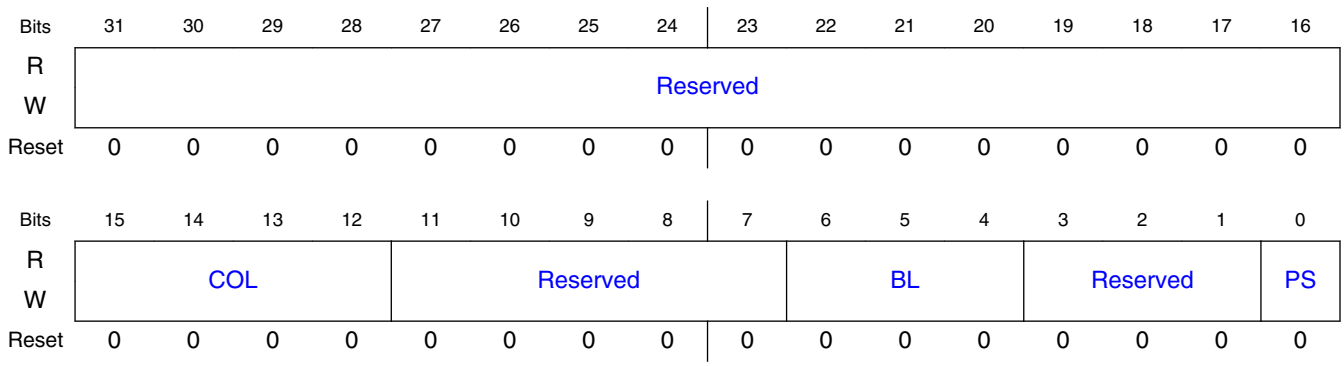
Field	Function
31-0 —	Reserved

29.7.1.26 DBI-B Control Register 0 (DBICR0)

29.7.1.26.1 Offset

Register	Offset
DBICR0	80h

29.7.1.26.2 Diagram



29.7.1.26.3 Fields

Field	Function
31-16 —	Reserved
15-12 COL	Column Address bit width 0000b - 12 Bits 0001b - 11 Bits 0010b - 10 Bits 0011b - 9 Bits 0100b - 8 Bits 0101b - 7 Bits 0110b - 6 Bits 0111b - 5 Bits 1000b - 4 Bits 1001b - 3 Bits 1010b - 2 Bits 1011b - 12 Bits 1100b - 12 Bits 1101b - 12 Bits 1110b - 12 Bits 1111b - 12 Bits
11-7 —	Reserved
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1

Table continues on the next page...

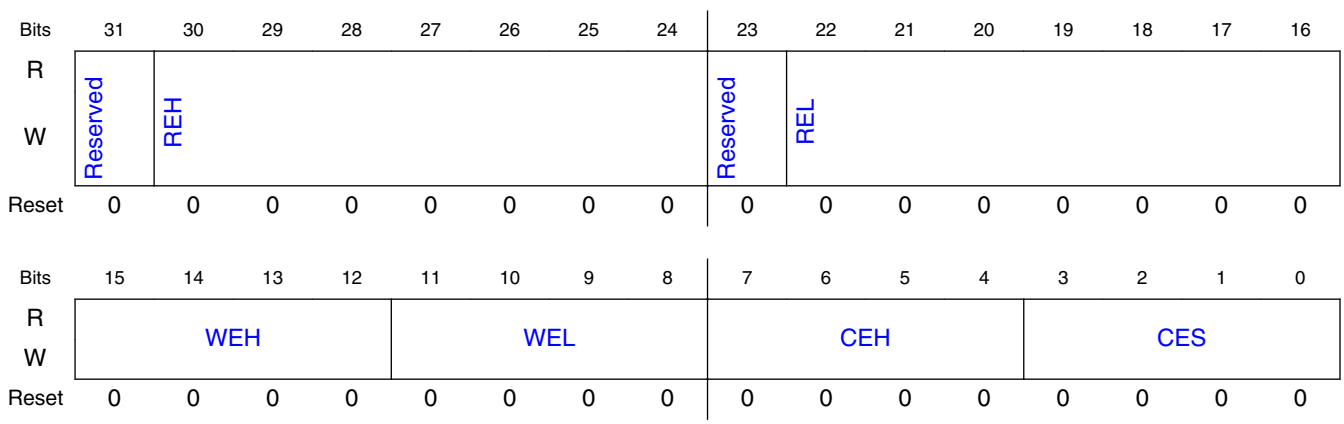
Field	Function
	001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 64
3-1 —	Reserved
0 PS	Port Size The port size must be aligned with DBI data width. 0b - 8bit 1b - 16bit

29.7.1.27 DBI-B Control Register 1 (DBICR1)

29.7.1.27.1 Offset

Register	Offset
DBICR1	84h

29.7.1.27.2 Diagram



29.7.1.27.3 Fields

Field	Function
31	Reserved

Table continues on the next page...

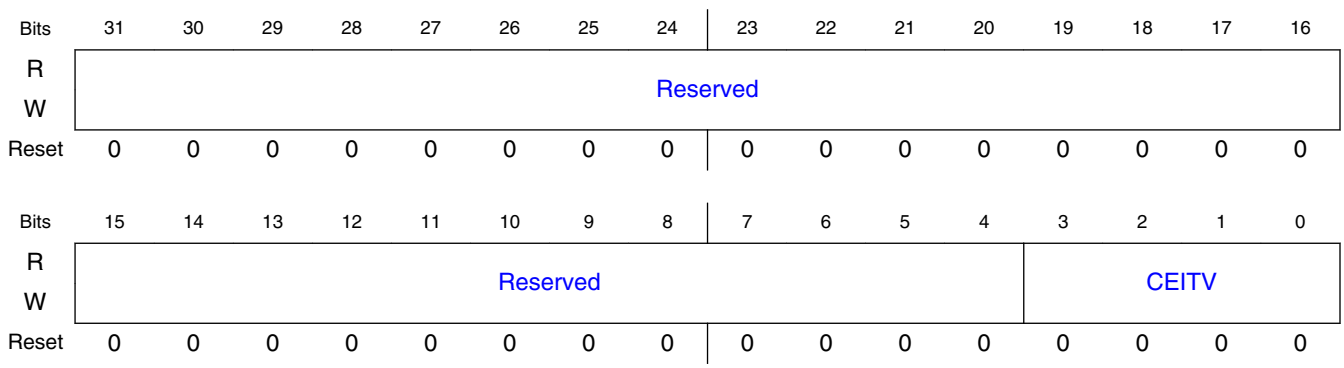
Field	Function
—	
30-24 REH	RDX High Time RDX High time is REH+1 clock cycles
23 —	Reserved
22-16 REL	RDX Low Time RDX Low time is REL+1 clock cycles
15-12 WEH	WRX High Time WRX High time is WEH+1 clock cycles
11-8 WEL	WRX Low Time WRX Low time is WEL+1 clock cycles
7-4 CEH	CSX Hold Time CSX Hold minimum time is CEH+1 clock cycles. This field determines the minimum hold time. Actual hold time may be larger in order to wait all read data sampled.
3-0 CES	CSX Setup Time CSX Setup time is CES+1 clock cycles

29.7.1.28 DBI-B Control Register 2 (DBICR2)

29.7.1.28.1 Offset

Register	Offset
DBICR2	88h

29.7.1.28.2 Diagram



29.7.1.28.3 Fields

Field	Function
31-4 —	Reserved
3-0 CEITV	CSX interval time CSX interval minimum time is CEITV+1 clock cycles.

29.7.1.29 IP Command Control Register 0 (IPCR0)

29.7.1.29.1 Offset

Register	Offset
IPCR0	90h

29.7.1.29.2 Function

Slave address

29.7.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.29.4 Fields

Field	Function
31-0 SA	Slave address The slave address is used for any command from IP bus.

29.7.1.30 IP Command Control Register 1 (IPCR1)

29.7.1.30.1 Offset

Register	Offset
IPCR1	94h

29.7.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_EXT_ADDR								Reserved					DATSZ		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.30.3 Fields

Field	Function
31-16 —	Reserved
15-8 NAND_EXT_ADDR	NAND Extended Address NAND IP command address is extended to 40-bit when IPCMD's KEY is 0x5AA5. High 8-bit is from NAND_EXT_ADDR, and low 32-bit is from IPCR0.
7-3 —	Reserved
2-0 DATSZ	Data Size in Byte It controls the data size of any write/read command. When IP command is not a write/read operation, DATSZ field can be ignored. 000b - 4 001b - 1 010b - 2 011b - 3 100b - 4 101b - 4 110b - 4 111b - 4

29.7.1.31 IP Command Control Register 2 (IPCR2)

29.7.1.31.1 Offset

Register	Offset
IPCR2	98h

29.7.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												BM3	BM2	BM1	BM0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.31.3 Fields

Field	Function
31-4 —	Reserved
3 BM3	Byte Mask for Byte 3 (IPTXDAT bit 31:24) 0b - Byte is unmasked 1b - Byte is masked
2 BM2	Byte Mask for Byte 2 (IPTXDAT bit 23:16) 0b - Byte is unmasked 1b - Byte is masked
1 BM1	Byte Mask for Byte 1 (IPTXDAT bit 15:8) 0b - Byte is unmasked 1b - Byte is masked
0 BM0	Byte Mask for Byte 0 (IPTXDAT bit 7:0) 0b - Byte is unmasked 1b - Byte is masked

29.7.1.32 IP Command Register (IPCMD)

29.7.1.32.1 Offset

Register	Offset
IPCMD	9Ch

29.7.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	KEY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	CMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.32.3 Fields

Field	Function
31-16 KEY	This field should be written with 0xA55A when trigging an IP command for all device types. The memory device is selected by the settings of IPCR0 and Base Registers. This field should be written with 0x5AA5 when trigging an IP command for NAND only. The address is extended to 40-bit with the use of NAND_EXT_ADDR. NAND IP base address is from 0x0 and memory size is not limited by BR8[MS].
15-0 CMD	SDRAM Commands: • 0x5: Extended Mode Register Set • 0x6: Deep Power Down • 0x8: Read • 0x9: Write • 0xA: Mode Register Set • 0xB: Active • 0xC: Auto Refresh • 0xD: Self Refresh • 0xE: Precharge • 0xF: Precharge All • Others: Reserved **NOTE:** Self Refresh is sent to all SDRAM devices because they share the same SEMC_CLK pin. NAND Commands: Bit 15-8 (Command Code)

Field	Function
	Bit 7-4 (Address mode): • 0x0: Column and Row address(5 Byte-CA0/CA1/RA0/RA1/RA2) • 0x1: Column address only (1 Byte-CA0) • 0x2: Column address only (2 Byte-CA0/CA1) • 0x3: Row address only (1 Byte-RA0) • 0x4: Row address only (2 Byte-RA0/RA1) • 0x5: Row address only (3 Byte-RA0/RA1/RA2) • Others: Reserved Bit 3-0 (Command mode): • 0x0: Command(0x05)-Address-Command-Read (Reserved for AXI Read) • 0x1: Command(0x85)-Address-Write (Reserved for AXI Write) • 0x2: Command • 0x3: Command-Hold • 0x4: Command-Address • 0x5: Command-Address-Hold • 0x6: Command-Address-Read • 0x7: Command-Address-Write • 0x8: Command-Read • 0x9: Command-Write • 0xA: Read • 0xB: Write • Others: Reserved NOR Commands: • 0x2: Read • 0x3: Write • Others: Reserved SRAM Commands: • 0x2: Memory Array Read • 0x3: Memory Array Write • 0x4: Memory Register Read • 0x5: Memory Register Write • Others: Reserved DBI_B Commands: • 0x2: Read • 0x3: Write • Others: Reserved

29.7.1.33 TX DATA Register (IPTXDAT)

29.7.1.33.1 Offset

Register	Offset
IPTXDAT	A0h

29.7.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.33.3 Fields

Field	Function
31-0 DAT	Data value to use for an IP write command. Bytes to use are determined based on IPCR2 configuration.

29.7.1.34 RX DATA Register (IPRXDAT)

29.7.1.34.1 Offset

Register	Offset
IPRXDAT	B0h

29.7.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DAT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DAT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.34.3 Fields

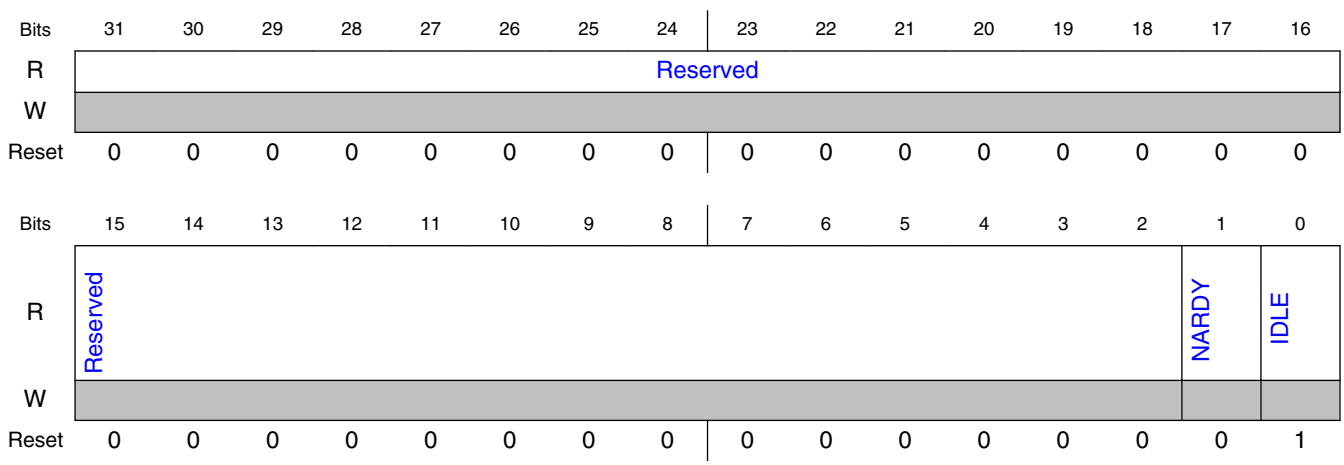
Field	Function
31-0 DAT	Data returned by device for an IP read command.

29.7.1.35 Status Register 0 (STS0)

29.7.1.35.1 Offset

Register	Offset
STS0	C0h

29.7.1.35.2 Diagram



29.7.1.35.3 Fields

Field	Function
31-2 —	Reserved
1 NARDY	Indicating NAND device Ready/WAIT# pin level. 0b - NAND device is not ready 1b - NAND device is ready
0	Indicating whether the SEMC is in idle state.

Field	Function
IDLE	When IDLE=1, the SEMC is in idle state. There is no pending AXI command in internal queue and no pending device access.

29.7.1.36 Status Register 1 (STS1)

29.7.1.36.1 Offset

Register	Offset
STS1	C4h

29.7.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.36.3 Fields

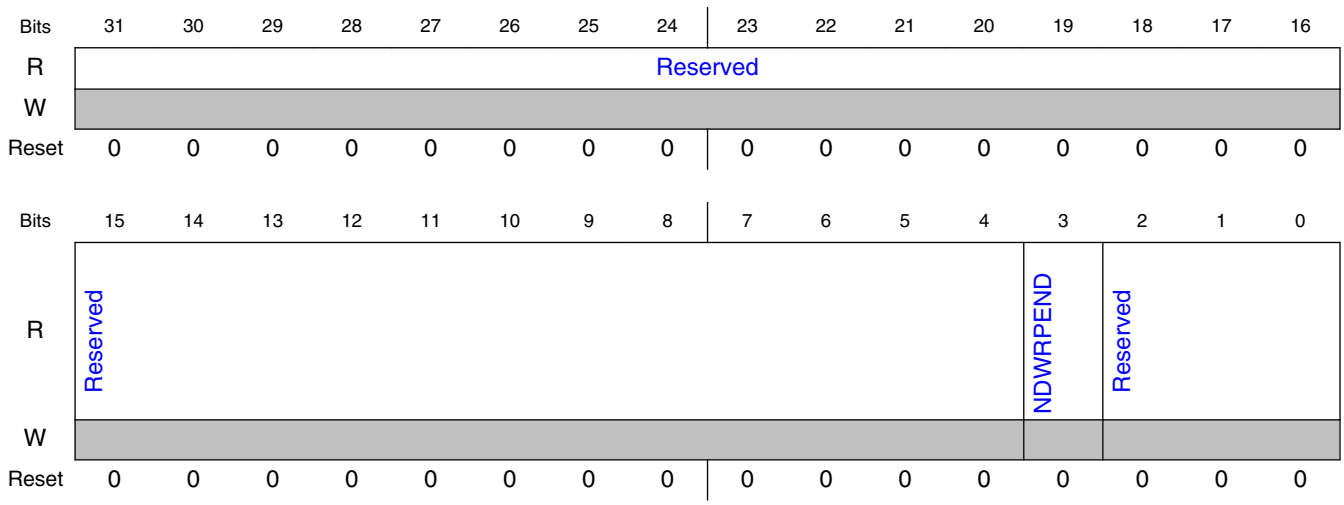
Field	Function
31-0 —	Reserved

29.7.1.37 Status Register 2 (STS2)

29.7.1.37.1 Offset

Register	Offset
STS2	C8h

29.7.1.37.2 Diagram



29.7.1.37.3 Fields

Field	Function
31-4 —	Reserved
3 NDWRPEND	This field indicating whether there is pending AXI command (write) to NAND device. 0b - No pending 1b - Pending
2-0 —	Reserved

29.7.1.38 Status Register 3 (STS3)

29.7.1.38.1 Offset

Register	Offset
STS3	CCh

29.7.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.38.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.39 Status Register 4 (STS4)

29.7.1.39.1 Offset

Register	Offset
STS4	D0h

29.7.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.39.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.40 Status Register 5 (STS5)

29.7.1.40.1 Offset

Register	Offset
STS5	D4h

29.7.1.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.40.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.41 Status Register 6 (STS6)

29.7.1.41.1 Offset

Register	Offset
STS6	D8h

29.7.1.41.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.41.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.42 Status Register 7 (STS7)

29.7.1.42.1 Offset

Register	Offset
STS7	DCh

29.7.1.42.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.42.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.43 Status Register 8 (STS8)

29.7.1.43.1 Offset

Register	Offset
STS8	E0h

29.7.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.43.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.44 Status Register 9 (STS9)

29.7.1.44.1 Offset

Register	Offset
STS9	E4h

29.7.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.44.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.45 Status Register 10 (STS10)

29.7.1.45.1 Offset

Register	Offset
STS10	E8h

29.7.1.45.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.45.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.46 Status Register 11 (STS11)

29.7.1.46.1 Offset

Register	Offset
STS11	ECh

29.7.1.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.46.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.47 Status Register 12 (STS12)

29.7.1.47.1 Offset

Register	Offset
STS12	F0h

29.7.1.47.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NDADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NDADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.47.3 Fields

Field	Function
31-0 NDADDR	This field indicating the last write address (AXI command) to NAND device (without base address in SEMC_BR4).

29.7.1.48 Status Register 13 (STS13)

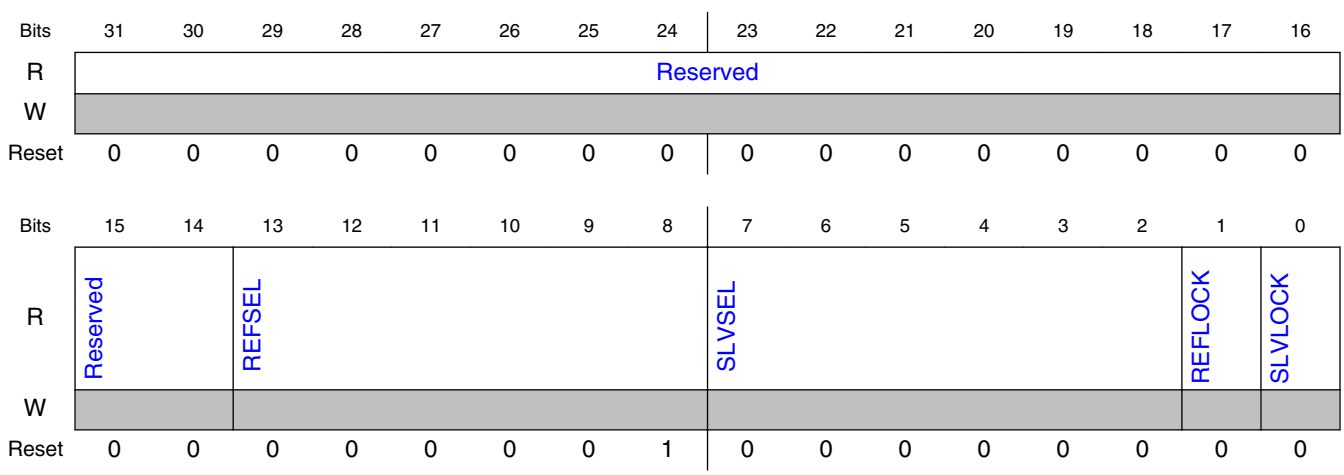
29.7.1.48.1 Offset

Register	Offset
STS13	F4h

29.7.1.48.2 Function

This register indicates the status of sample clock DLLs for synchronous NAND read access.

29.7.1.48.3 Diagram



29.7.1.48.4 Fields

Field	Function
31-14	Reserved

Table continues on the next page...

Field	Function
—	
13-8 REFSEL	Sample clock reference delay line delay cell number selection.
7-2 SLVSEL	Sample clock slave delay line delay cell number selection.
1 REFLOCK	Sample clock reference delay line locked. 0b - Reference delay line is not locked. 1b - Reference delay line is locked.
0 SLVLOCK	Sample clock slave delay line locked. 0b - Slave delay line is not locked. 1b - Slave delay line is locked.

29.7.1.49 Status Register 14 (STS14)

29.7.1.49.1 Offset

Register	Offset
STS14	F8h

29.7.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.49.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.50 Status Register 15 (STS15)

29.7.1.50.1 Offset

Register	Offset
STS15	FCh

29.7.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.50.3 Fields

Field	Function
31-0 —	Reserved

29.7.1.51 Base Register 9 (BR9)

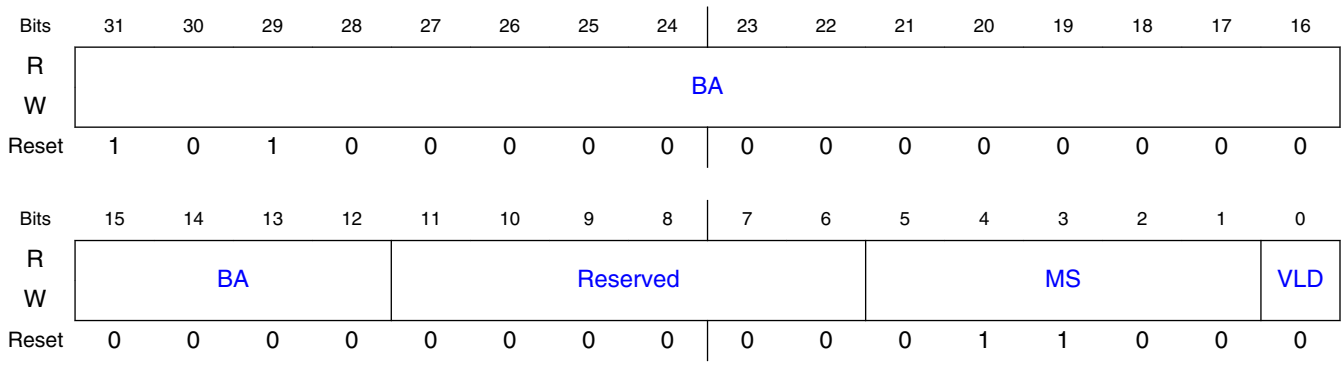
29.7.1.51.1 Offset

Register	Offset
BR9	100h

29.7.1.51.2 Function

This is the address base register for SRAM device 1. It is applied to both AXI and IP commands.

29.7.1.51.3 Diagram



29.7.1.51.4 Fields

Field	Function
31-12 BA	Base Address This field determines high position 20 bits of SoC level Base Address. The low position 12 bits of the address are all zero. The base address must be within the SEMC range defined for the device memory map.
11-6 —	Reserved
5-1 MS	Memory size It defines the size of the memory. Memory size must be aligned with base address. Memory regions should be configured to avoid overlapping areas. Overlapping regions will cause bus errors. 0_0000b - 4KB 0_0001b - 8KB 0_0010b - 16KB 0_0011b - 32KB 0_0100b - 64KB 0_0101b - 128KB 0_0110b - 256KB 0_0111b - 512KB 0_1000b - 1MB 0_1001b - 2MB 0_1010b - 4MB 0_1011b - 8MB 0_1100b - 16MB 0_1101b - 32MB 0_1110b - 64MB 0_1111b - 128MB 1_0000b - 256MB 1_0001b - 512MB 1_0010b - 1GB

Table continues on the next page...

Field	Function
	1_0011b - 2GB 1_0100b-1_1111b - 4GB
0 VLD	Valid This bit controls whether the memory is valid or not. 0b - The memory is invalid, can not be accessed. 1b - The memory is valid, can be accessed.

29.7.1.52 Base Register 10 (BR10)

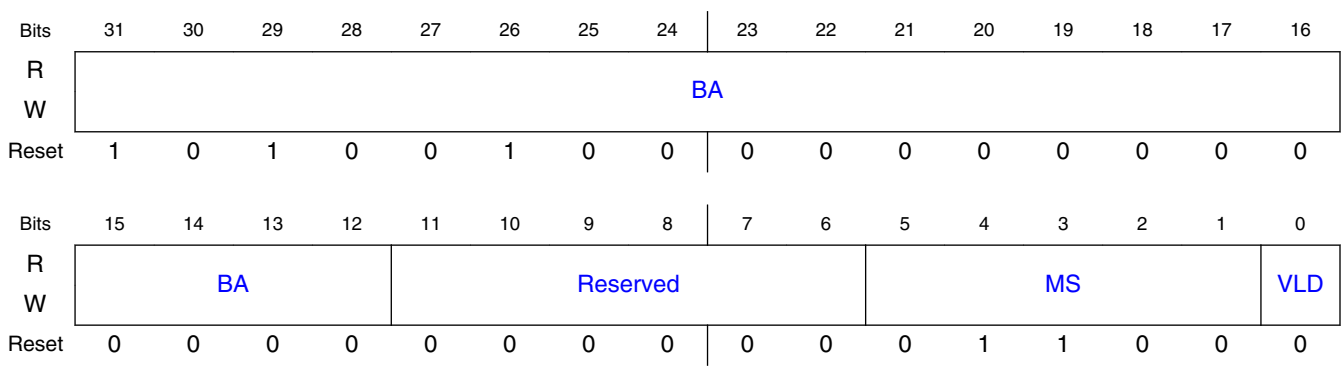
29.7.1.52.1 Offset

Register	Offset
BR10	104h

29.7.1.52.2 Function

This is the address base register for SRAM device 2. It is applied to both AXI and IP commands.

29.7.1.52.3 Diagram



29.7.1.52.4 Fields

Field	Function
31-12 BA	Base Address

Table continues on the next page...

Field	Function
	This field determines high position 20 bits of SoC level Base Address. The low position 12 bits of the address are all zero. The base address must be within the SEMC range defined for the device memory map.
11-6 —	Reserved
5-1 MS	Memory size It defines the size of the memory. Memory size must be aligned with base address. Memory regions should be configured to avoid overlapping areas. Overlapping regions will cause bus errors. 0_0000b - 4KB 0_0001b - 8KB 0_0010b - 16KB 0_0011b - 32KB 0_0100b - 64KB 0_0101b - 128KB 0_0110b - 256KB 0_0111b - 512KB 0_1000b - 1MB 0_1001b - 2MB 0_1010b - 4MB 0_1011b - 8MB 0_1100b - 16MB 0_1101b - 32MB 0_1110b - 64MB 0_1111b - 128MB 1_0000b - 256MB 1_0001b - 512MB 1_0010b - 1GB 1_0011b - 2GB 1_0100b-1_1111b - 4GB
0 VLD	Valid This bit controls whether the memory is valid or not. 0b - The memory is invalid, can not be accessed. 1b - The memory is valid, can be accessed.

29.7.1.53 Base Register 11 (BR11)

29.7.1.53.1 Offset

Register	Offset
BR11	108h

29.7.1.53.2 Function

This is the address base register for SRAM device 3. It is applied to both AXI and IP commands.

29.7.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BA															
Reset	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BA				Reserved						MS					VLD
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

29.7.1.53.4 Fields

Field	Function
31-12 BA	Base Address This field determines high position 20 bits of SoC level Base Address. The low position 12 bits of the address are all zero. The base address must be within the SEMC range defined for the device memory map.
11-6 —	Reserved
5-1 MS	Memory size It defines the size of the memory. Memory size must be aligned with base address. Memory regions should be configured to avoid overlapping areas. Overlapping regions will cause bus errors. 0_0000b - 4KB 0_0001b - 8KB 0_0010b - 16KB 0_0011b - 32KB 0_0100b - 64KB 0_0101b - 128KB 0_0110b - 256KB 0_0111b - 512KB 0_1000b - 1MB 0_1001b - 2MB 0_1010b - 4MB 0_1011b - 8MB 0_1100b - 16MB 0_1101b - 32MB 0_1110b - 64MB 0_1111b - 128MB 1_0000b - 256MB 1_0001b - 512MB 1_0010b - 1GB 1_0011b - 2GB 1_0100b-1_1111b - 4GB
0 VLD	Valid This bit controls whether the memory is valid or not.

Field	Function
	0b - The memory is invalid, can not be accessed. 1b - The memory is valid, can be accessed.

29.7.1.54 SRAM Control Register 4 (SRAMCR4)

29.7.1.54.1 Offset

Register	Offset
SRAMCR4	120h

29.7.1.54.2 Function

This register configures SRAM device 1 - 3

29.7.1.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COL				ADVH	ADVP	AM		Reserved	BL			WAITSP	WAITEN	SYNCEN	PS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.54.4 Fields

Field	Function
31-16 —	Reserved
15-12 COL	Column Address bit width 0000b - 12 Bits 0001b - 11 Bits 0010b - 10 Bits 0011b - 9 Bits

Table continues on the next page...

Field	Function
	0100b - 8 Bits 0101b - 7 Bits 0110b - 6 Bits 0111b - 5 Bits 1000b - 4 Bits 1001b - 3 Bits 1010b - 2 Bits 1011b - 12 Bits 1100b - 12 Bits 1101b - 12 Bits 1110b - 12 Bits 1111b - 12 Bits
11 ADVH	ADV# level control during address hold state 0b - ADV# is high during address hold state. 1b - ADV# is low during address hold state.
10 ADVP	ADV# polarity 0b - ADV# is active low. 1b - ADV# is active high.
9-8 AM	Address Mode It defines the address mode. 00b - Address/Data MUX mode (ADMUX) 01b - Advanced Address/Data MUX mode (AADM) 10b - Address/Data non-MUX mode (Non-ADMUX) 11b - Address/Data non-MUX mode (Non-ADMUX)
7 —	Reserved
6-4 BL	Burst Length The burst length determines the number of column locations that can be accessed for a given READ or WRITE command. 000b - 1 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 64
3 WAITSP	Wait Sample Wait pin is sampled by internal clock if this bit is set. It is ignored when WAITEN bit clear. It should be set for ASYNC mode. 0b - Wait pin is directly used by the SEMC. 1b - Wait pin is sampled by internal clock before it is used.
2 WAITEN	Wait Enable The SEMC monitors wait pin if this bit set. It is for ASYNC mode only. 0b - The SEMC does not monitor wait pin. 1b - The SEMC monitors wait pin. The SEMC does not transfer/receive data when wait pin is asserted.
1 SYNCEN	Synchronous Mode Enable It defines the access mode. 0b - Asynchronous mode is enabled. 1b - Synchronous mode is enabled. Only fixed latency mode is supported.

Table continues on the next page...

Field	Function
0 PS	Port Size The port size must be aligned with SRAM data width. 0b - 8bit 1b - 16bit

29.7.1.55 SRAM Control Register 5 (SRAMCR5)

29.7.1.55.1 Offset

Register	Offset
SRAMCR5	124h

29.7.1.55.2 Function

This register configures SRAM device 1 - 3

29.7.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REH				REL				WEH				WEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AH				AS				CEH				CES			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.55.4 Fields

Field	Function
31-28	RE high time
REH	RE high time is REH+1 clock cycles. This field applied to ASYNC mode only.
27-24	RE low time
REL	RE low time is REL+1 clock cycles. This field applied to ASYNC mode only.
23-20	WE high time

Table continues on the next page...

Field	Function
WEH	WE high time is WEH+1 clock cycles. This field applied to ASYNC mode only.
19-16 WEL	WE low time WE low time is WEL+1 clock cycles. This field applied to ASYNC mode only.
15-12 AH	Address hold time Address hold time is AH+1 clock cycles for ASYNC mode. Address hold time is AH clock cycles for SYNC mode.
11-8 AS	Address setup time Address setup time is AS+1 clock cycles
7-4 CEH	CE hold time CE Hold time is CEH+1 clock cycles This field determines the minimum hold time. Actual hold time may be larger in order to wait all read data sampled.
3-0 CES	CE setup time CE setup time is CES+1 clock cycles.

29.7.1.56 SRAM Control Register 6 (SRAMCR6)

29.7.1.56.1 Offset

Register	Offset
SRAMCR6	128h

29.7.1.56.2 Function

This register configures SRAM device 1 - 3

29.7.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDH				CEITV				RD				LC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AWDH				TA				WDH				WDS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.56.4 Fields

Field	Function
31-28	Read hold time
RDH	Read hold time is RDH clock cycles. This field applied to SYNC mode only.
27-24	CE# interval time
CEITV	CE# interval minimum time is CEITV+1 cycles.
23-20	Read time
RD	Read time is RD+1 clock cycles. This field applied to SYNC mode only.
19-16	Latency count
LC	Latency count is LC clock cycles. This field applied to SYNC mode only.
15-12	Address to write data hold time
AWDH	Address to write data hold time is AWDH clock cycles. This field applied to ASYNC mode only.
11-8	Turnaround time
TA	Turnaround time is TA+1 clock cycles. Both the SEMC and Device do not drive Data Lines to avoid bus confliction. This field applied to ASYNC mode only.
7-4	Write Data hold time
WDH	Write data hold time is WDH clock cycles. This field applied to SYNC mode only.
3-0	Write Data setup time
WDS	Write data setup time is WDS+1 clock cycles. This field applied to SYNC mode only.

29.7.1.57 Delay Chain Control Register (DCCR)

29.7.1.57.1 Offset

Register	Offset
DCCR	150h

29.7.1.57.2 Function

This register provides the configuration fields for sample clock DQS in synchronous mode. It adds delays on DQS clock to compensate timings while DQS is faster than read data. It is for synchronous SDRAM, SRAM and NOR read access.

29.7.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		SRAMXVAL					SRAMXEN	Reserved		SRAM0VAL					SRAM0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NORVAL					NOREN	Reserved		SDRAMVAL					SDRAMEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.7.1.57.4 Fields

Field	Function
31-30 —	Reserved
29-25 SRAMXVAL	Clock delay line delay cell number selection value for SRAM device 1-3. When SRAMXEN is set, the delay cell number in delay chain is SRAMXVAL+1.
24 SRAMXEN	Delay chain insertion enable for SRAM device 1-3. 0b - Delay chain is not inserted. 1b - Delay chain is inserted.
23-22 —	Reserved
21-17 SRAM0VAL	Clock delay line delay cell number selection value for SRAM device 0. When SRAM0EN is set, the delay cell number in delay chain is SRAM0VAL+1.
16 SRAM0EN	Delay chain insertion enable for SRAM device 0. 0b - Delay chain is not inserted. 1b - Delay chain is inserted.
15-14 —	Reserved
13-9 NORVAL	Clock delay line delay cell number selection value for NOR device. When NOREN is set, the delay cell number in delay chain is NORVAL+1.
8 NOREN	Delay chain insertion enable for NOR device. 0b - Delay chain is not inserted. 1b - Delay chain is inserted.
7-6 —	Reserved

Table continues on the next page...

Field	Function
5-1 SDRAMVAL	Clock delay line delay cell number selection value for SDRAM device. When SDRAMEN is set, the delay cell number in delay chain is SDRAMVAL+1.
0 SDRAMEN	Delay chain insertion enable for SRAM device. 0b - Delay chain is not inserted. 1b - Delay chain is inserted.

Chapter 30
FlexSPI Controller (FLEXSPI)

30.1 Chip-specific FlexSPI information

Table 30-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

- The soft reset bit (MCR_SWRESET) should be set, to enable the inverted clock, after setting the SCKBDIFFOPT bit.
- For Octal mode support, refer to Table 30-4.
- ARM AXIM access to FlexSPI is converted to AHB command access of FlexSPI, on this device.

30.1.1 AHB Memory Map Address Ranges

The table below shows the address range calculations for the various instances of FlexSPI:

Table 30-2. Memory Map Address Range Calculations

Description	FlexSPI1	FlexSPI2
address range is mapped for AHB read access to IP RX FIFO:	- 0x2FC0_0000 – 0x2FC0_00FF	- 0x7FC0_0000 – 0x7FC0_00FF
address range is mapped for AHB write access to IP TX FIFO:	- 0x2F80_0000 – 0x2F80_00FF	- 0x7F80_0000 – 0x7F80_00FF
AHB Memory Map for Serial Flash memory access	- 0x3000_0000	- 0x6000_0000
Total AHB RX Buffer size:	- 4 KBytes	- 4 KBytes
AHB TX Buffer size:	- 64 Bytes	- 64 Bytes
IP Bus memory space access:	- 0x2FC0_0000 – 0x2FC0_00FF	- 0x7FC0_0000 – 0x7FC0_00FF
AHB bus memory space access:	- 0x3000_0000 – 0x3FFF_FFFF	- 0x6000_0000 – 0x6FFF_FFFF

30.1.2 Master ID allocation

To enable FlexSPI to view different IDs from different masters, follow the steps shown below:

- Set AHBCR[HMSTRIDREMAP] =1 to enable master ID mapping
- Configure HMSTRxCR[MSTRID] using the table below (The “x” should be masked by the mask value. Other bits can be selectively masked to group some masters into one AHB buffer region):

Table 30-3. Master IDs

Module	Master ID
CM7	xxxx xxxx xxxx x000
eDMA	xxxx_xxxx_xxxx_x101
CM4	xxxx_xxxx_xxxx_x100
EDMA_LPSR	xxxx_xxxx_xxxx_x100
CAAM	xxxx_xxxx_x0000_011
ENET_1G TX	xxxx_xxxx_x0001_011
ENET_1G RX	xxxx_xxxx_x0010_011
USB	xxxx_xxxx_x0100_011
ENET	xxxx_xxxx_x0011_011
USDHC1	xxxx_xxxx_x0101_011

Table continues on the next page...

Table 30-3. Master IDs (continued)

Module	Master ID
USDHC2	xxxx_xxxx_x0110_011
ENET QOS	xxxx_xxxx_x1000_011
GPU	xxxx_xxxx_xx000_001
PXP	xxxx_xxxx_xx001_001
ELCDIF	xxxx_xxxx_xx001_010
LCDIFv2	xxxx_xxxx_xx010_010
CSI	xxxx_xxxx_xx011_010

NOTE

- After reset, the FlexSPI sees all masters IDs as zero

30.1.3 FlexSPI connection modes

The table below shows details about the Parallel, Individual, and Combination modes for Flash memory.

Table 30-4. Relation between Modes and Bus size for FlexSPI instances

Mode	Instance	Effective Bus Size	Data Signals	Function	Devices Attached to Bus	Chip Selects
INDIVIDUAL	FLEXSPI1	4 bit	A_DATA0 - A_DATA3	Single QSPI chip support (1b, 2b,4b)	Up to 2	A_SS0_B, A_SS1_B
INDIVIDUAL	FLEXSPI1	4 bit	B_DATA0 - B_DATA3	Single QSPI chip support (1b, 2b,4b)	Up to 2	B_SS0_B, B_SS1_B
INDIVIDUAL	FLEXSPI2	4 bit	A_DATA0 - A_DATA3	Single QSPI chip support (1b, 2b,4b)	Up to 2	A_SS0_B, A_SS1_B
INDIVIDUAL	FLEXSPI2	4 bit	B_DATA0 - B_DATA3	Single QSPI chip support (1b, 2b,4b)	Up to 2	B_SS0_B, B_SS1_B
INDIVIDUAL	FLEXSPI2	8 bit	A_DATA0 - A_DATA7	Single Octal/ Hyperbus/ Xccela chip support (1x8b)	Up to 2	A_SS0_B, A_SS1_B
INDIVIDUAL	FLEXSPI2	8 bit	B_DATA0 - B_DATA7	Single Octal/ Hyperbus/ Xccela chip support (1x8b)	Up to 2	B_SS0_B, B_SS1_B

Table continues on the next page...

Table 30-4. Relation between Modes and Bus size for FlexSPI instances (continued)

COMBINE	FLEXSPI1	8 bit	A_DATA0 - A_DATA3 | B_DATA0 - B_DATA3	Single Octal/ Hyperbus/ Xccela chip support (1x8b)	Up to 2	A_SS0_B, A_SS1_B
PARALLEL	FLEXSPI1	8 bit	A_DATA0 - A_DATA3 | B_DATA0 - B_DATA3	Two QSPI chips in parallel (2x4b)	2 parallel chips x 2 = up to 4	(A_SS0_B | B_SS0_B), (A_SS1_B | B_SS1_B)
PARALLEL	FLEXSPI2	8 bit	A_DATA0 - A_DATA3 | B_DATA0 - B_DATA3	Two QSPI chips in parallel (2x4b)	2 parallel chips x 2 = up to 4	(A_SS0_B | B_SS0_B), (A_SS1_B | B_SS1_B)

30.1.4 FlexSPI memory details

The table below shows the FlexSPI memory details

Table 30-5. FlexSPI memory details

Start address	End address	Size(kB)	Description
3000_0000	3FFF_FFFF	256MB	FlexSPI1/ FlexSPI1 cipher text
6000_0000	7F7F_FFFF	504MB	FlexSPI2/ FlexSPI2 ciphertext

30.2 Overview

FlexSPI supports two SPI channels and up to four external devices. Each channel supports Single/Dual/Quad/Octal mode data transfer (1/2/4/8 bidirectional data lines).

The FlexSPI configuration depends on the chip configuration. See the system-level section for boot information and pinmux for chip-specific information regarding the modes and number of devices supported.

FlexSPI supports communication with both serial flash memory and serial RAM devices. While many of the descriptions, registers, and fields specifically reference flash memory, almost all information can also be applied to serial RAM. Flash memory is used as an example in the tables and figures in this chapter.

NOTE

Terminology in this chapter has been updated to align with JEDEC standard *EXpanded Serial Peripheral Interface (xSPI) for Non Volatile Memory Devices, Version 1.0*.

Table 30-6. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

30.2.1 Block diagram

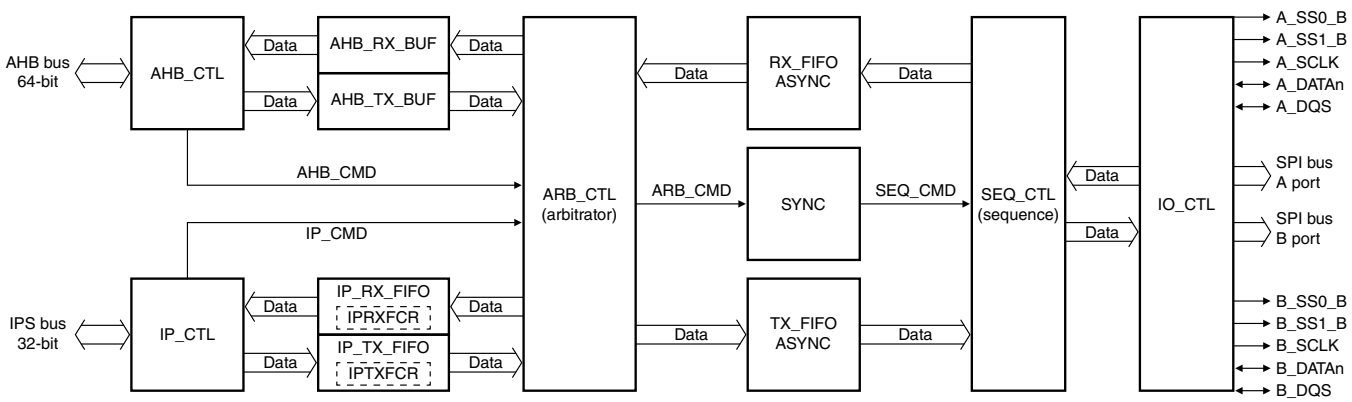


Figure 30-1. Block diagram

30.2.2 Features

FlexSPI supports:

- Flexible sequence engine (lookup table) to support various vendor devices
  - Serial NOR flash memory and other devices with SPI protocol similar to serial NOR flash memory
  - Serial NAND flash memory
  - HyperBus devices (HyperFlash/HyperRAM)

    - XCCELA devices
    - FPGA devices
- Flash memory access modes
    - Single, Dual, Quad, Octal mode
    - Single Data Transfer Rate (SDR) and Double Data Transfer Rate (DDR) mode
    - Individual/Parallel mode
- Sampling clock mode
    - Internal dummy read strobe looped back internally
    - Internal dummy read strobe looped back from pad
    - Flash-memory-provided read strobe
- Memory mapped read and write access by AHB bus
    - AHB receive buffer implemented to reduce read latency. See the chip-specific section for AHB buffer size.
    - 16 AHB controllers with programmable priority for read access from each
    - 8 flexible and configurable buffers in AHB receive buffer
    - AHB transmit buffer implemented to buffer all write data from one AHB burst. See the chip-specific section for AHB buffer size.

Note

All AHB controllers share this AHB transmit buffer. There is no AHB controller number limitation for write access.

- Software-triggered flash memory read and write access by IP bus
    - IP receive FIFO implemented to buffer all read data from external devices. FIFO size: 256 bytes
    - IP transmit FIFO implemented to buffer all write data to external devices. FIFO size: 256 bytes
    - DMA support to read IP receive FIFO
    - DMA support to fill IP transmit FIFO
    - SCLK stops when IP receive FIFO is full during reading flash memory data
    - SCLK stops when IP transmit FIFO is empty during writing flash memory data

30.3 Functional description

30.3.1 Flash connection

There are two FlexSPI interface ports (port A and port B). Each port supports up to two flash devices by providing two chip select outputs.

NOTE

FlexSPI configuration depends on the chip configuration. See the chip-specific FlexSPI information regarding the number of devices supported.

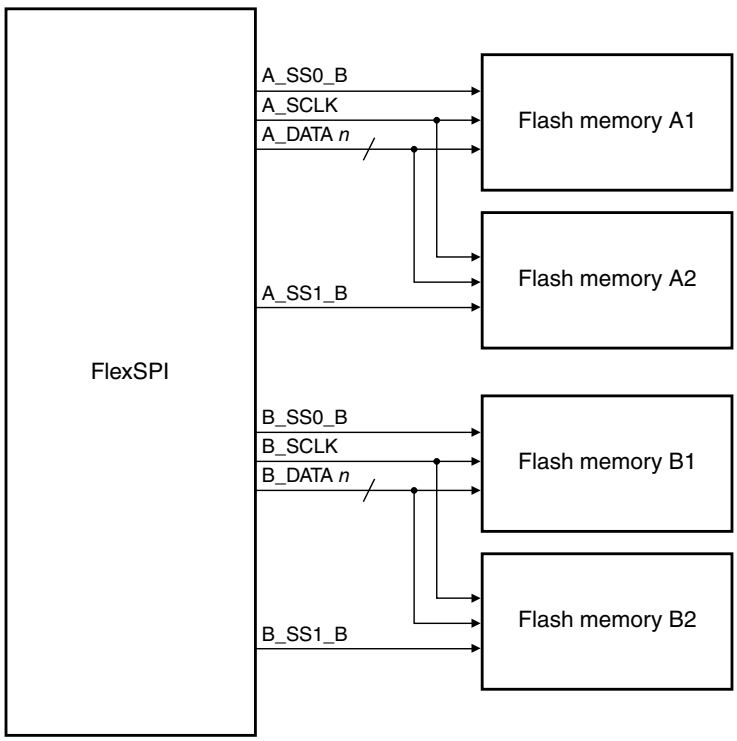


Figure 30-2. FlexSPI connection diagram with four devices

NOTE

- Flash A1 and Flash A2 can be two flash chip packages or two flash dies in the same package. There is no difference to FlexSPI. This statement is also true for Flash B1 and Flash B2.
- Flash A1 and Flash B1 can be accessed in parallel, using parallel mode. FlexSPI merges or splits the flash read and program data automatically. This statement is also true for Flash A2 and Flash B2.
- In parallel mode, Flash A1 and Flash A2 cannot be accessed at the same time. This statement is also true for Flash B1 and Flash B2.
- In individual mode, flash devices cannot be accessed at the same time. But these four devices can be accessed separately.

There is a combination mode (MCR0[COMBINATIONEN] = 1) used to provide octal flash memory support by combining Port A (A_DATA[3:0]) and Port B (B_DATA[3:0]) together. Normal mode can also support Octal mode by using all eight data lines from Port A or Port B.Figure 30-3 shows the connection diagram for combination mode.

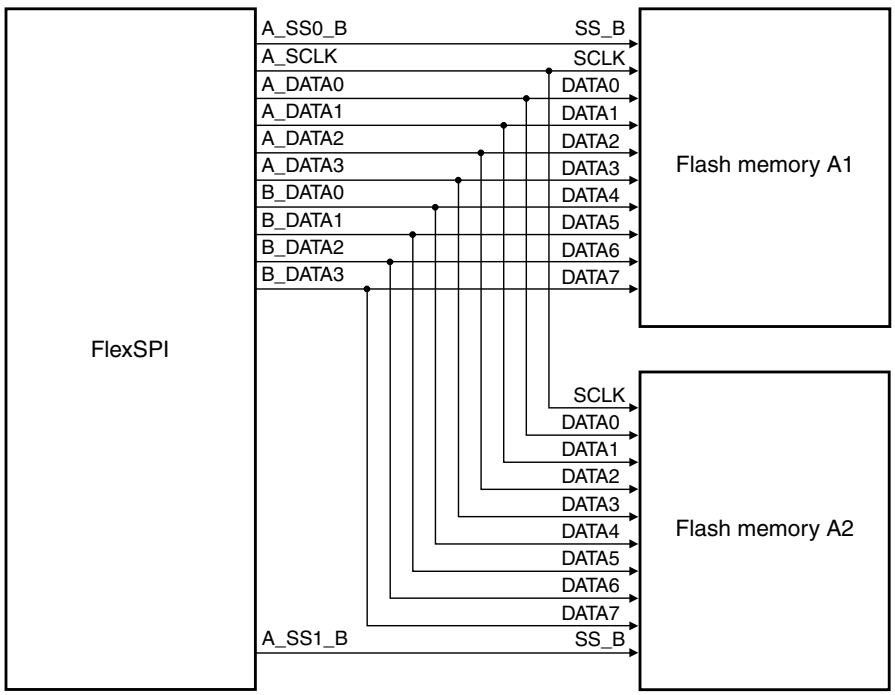


Figure 30-3. FlexSPI connection diagram for combination mode

30.3.2 Flash memory access mode

30.3.2.1 SPI clock mode

FlexSPI only supports SPI clock mode 0: clock polarity (CPOL) = 0 and clock phase (CPHA) = 0. When the SPI bus is idle, SCLK stays at a logic low state.

30.3.2.2 Individual mode and Parallel mode

In individual mode, read and write data are received and transmitted on port A or port B.

In parallel mode, read and write data are received and transmitted on port A and port B in parallel. FlexSPI merges or splits the read and program data automatically (READ/WRITE instruction). Only read and program data is merged or split (READ/WRITE

instruction). For other instructions (such as Command, Address, Mode, and Data size), the same associated information is transmitted to the devices connected to port A and port B. For more details, see Executing instructions on SPI interface.

IPCR1[IPAREN] (for IP command) or AHBCR[APAREN] (for AHB command) statistically determines Individual and Parallel mode.

NOTE

FlexSPI does not support 16-bit read and write for the SPI interface.

30.3.2.3 SDR mode and DDR mode

In SDR mode, flash memory receives data on the rising edge of SCLK and transmits data on the falling edge of SCLK.

In DDR mode, flash memory receives and transmits data on both the rising and falling edges of SCLK.

The instruction (opcode) in the LUT sequence dynamically determines SDR and DDR modes. There is no static configuration register field setting for SDR and DDR modes. See FlexSPI input timing and FlexSPI output timing for details about input and output timing.

30.3.2.4 Single, Dual, Quad, and Octal mode

In Single mode, FlexSPI transmits and receives data on one data pin (DATA0 for transmitting, DATA1 for receiving).

In Dual mode, FlexSPI transmits and receives data on two data pins (DATA0–DATA1 for both transmitting and receiving).

In Quad mode, FlexSPI transmits and receives data on four data pins (DATA0–DATA3 for both transmitting and receiving).

In Octal mode, FlexSPI transmits and receives data on eight data pins (DATA0–DATA7 for both transmitting and receiving).

Instruction (num_pads) in the LUT sequence dynamically determine Single, Dual, Quad, and Octal modes. There is no static configuration register field setting for Single, Dual, Quad, and Octal modes.

30.3.3 Modes of operation

FlexSPI operates with these modes.

Table 30-7. Operating modes

Mode	Description
Module Disable	This mode is a low-power mode for FlexSPI. In Module Disable mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock domain is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, and IP transmit FIFO cannot be accessed. Serial memory access is also not available. Write 1 to MCR0[MDIS] to enter this mode. Write 0 to MCR0[MDIS] to exit this mode.
Doze	This mode is a low-power mode for the chip. When the chip requires FlexSPI to enter Doze mode and MCR0[DOZEEN] = 1, FlexSPI enters Doze mode after all transactions are completed (STS0[ARBIDLE] = 1). In Doze mode, the AHB clock and serial clock domains are gated off internally, but the IPS bus clock is not gated off. Read and write access to the control and status register are available, but the LUT, IP receive FIFO, the IP transmit FIFO cannot be accessed. Serial memory access is also not available. This mode is entered via system request, and exited by deasserting this system request.
Stop	This mode is a low-power mode for the chip. When the chip requires the FlexSPI to enter Stop mode, FlexSPI waits for all transactions to complete (STS0[ARBIDLE] = 1) and return ACK handshake to the system. After the ACK handshake is returned, FlexSPI gates off the AHB clock and serial clock domains internally. The system can gate off the AHB bus clock, IPS bus clock, and serial clock at the system level. This mode is entered via system request. This mode is exited by deasserting this system request, and the ACK handshake message is also deasserted immediately.
Normal	No clock is gated off internally. Normal register access and serial memory access are available.

30.3.4 AHB access memory map

FlexSPI allocates AHB memory space for each memory device starting from the FlexSPI region base address of the system memory map. The FLSHxCR0[FLSHSZ] field determines the amount of memory allocated for each memory device in KB, Fx_SIZE. The memory is then allocated sequentially and contiguously.

NOTE

See the chip-specific section for the maximum flash size supported.

When MCR2[SAMEDEVICEEN] = 1:

- FA1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FA2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB

- FB1_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHA1CR0[FLSHSZ] × 1 KB

When MCR2[SAMEDEVICEEN] = 0:

- FA1_SIZE = FLSHSZ × 1 KB
- FA2_SIZE = FLSHA2CR0[FLSHSZ] × 1 KB
- FB1_SIZE = FLSHB1CR0[FLSHSZ] × 1 KB
- FB2_SIZE = FLSHB2CR0[FLSHSZ] × 1 KB

Flash B1 and Flash B2 size settings are ignored in parallel mode (FLSHB1CR0[FLSHSZ], FLSHB2CR0[FLSHSZ]). To execute in parallel mode, Flash B1 should be the same device as Flash A1, and Flash B2 should be the same device as Flash A2.

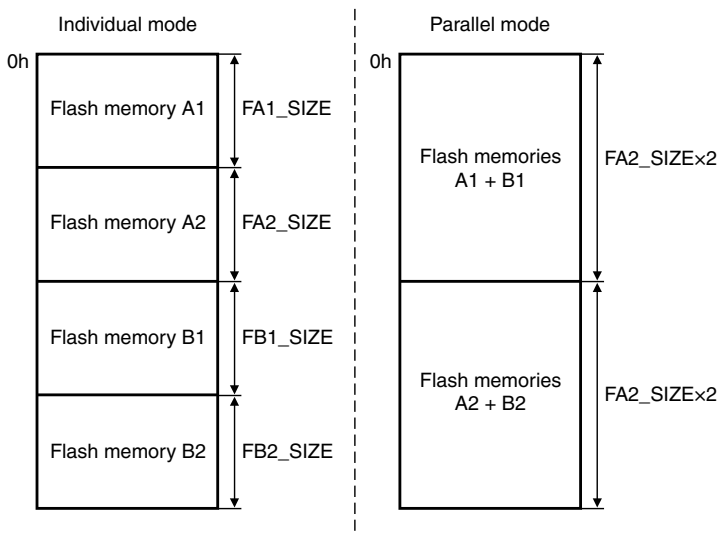


Figure 30-4. AHB access memory map in individual and parallel mode

AHB access memory map in individual mode:

- Flash A1 address range: 0000_0000h to FA1_SIZE
- Flash A2 address range: FA1_SIZE to (FA1_SIZE + FA2_SIZE)
- Flash B1 address range: (FA1_SIZE + FA2_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE)
- Flash B2 address range: (FA1_SIZE + FA2_SIZE + FB1_SIZE) to (FA1_SIZE + FA2_SIZE + FB1_SIZE + FB2_SIZE)

AHB access memory map in parallel mode:

- Flash A1+B1 address range: 0000_0000h to FA1_SIZE × 2
- Flash A2+B2 address range: FA1_SIZE × 2 to (FA1_SIZE × 2 + FA2_SIZE × 2)

NOTE

The address used in Figure 30-4 and equations in this topic is the address presented to the memory. The base address for FlexSPI in the system memory map has already been removed, so it not used or shown here.

30.3.4.1 Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers

The FlexSPI controller has a remap feature that supports the storage of dual images in memory. The HADDRSTART and HADDREND registers can be used to configure an address range to which an offset (that the HADDROFFSET register configures) is applied. This feature allows a single system bus address range to be mapped to two different memory ranges: the range that is remapped or the range that is not remapped. You can enable and disable the remapping to switch between accessing the two images at the same system memory address. This feature is used for booting.

NOTE

- Remapping in this table includes the REMAP register (HADDRSTART[0]) and SWAP register (HADDRSTART[2]).
- The swap function impacts both the AHB and IPS address remapping with the same mechanism. The remap function only impacts the AHB address.

Table 30-8. FlexSPI address translation

	AHB or IPS address	Memory address	Description
Enable remap			
AHB WR	ADDR	ADDR + OFFSET	The remap function is available if HADDRSTART <= ADDR < HADDREND, otherwise no change in address.
AHB RD	ADDR	ADDR + OFFSET	The remap function is available if HADDRSTART <= ADDR <HADDREND, otherwise no change in address.
Disable remap			
AHB WR	ADDR	ADDR	No change in address.
AHB RD	ADDR	ADDR	No change in address.

NOTE

- HADDRSTART[0] is used to enable the remap feature.
- The ADDR above is the address from the IPS or AHB interface. The OFFSET is same as the HADDROFFSET register.

30.3.4.2 Flash address sent to flash memory devices

The AHB address (AHB command) or IPCR0[SFAR] (IP command) determines the flash memory access start address. See Flash memory access via AHB command and Flash memory access via IP command for more details.

For AHB commands, the FlexSPI controller removes the flash memory base address automatically when sending flash addresses to flash devices. The flash address is sent to devices in two parts: row address and column address. For flash devices that do not support the column address, set FLSHxCR1[CAS] to 0. This setting causes all flash address bits to be sent to flash devices as row addresses.

For word-addressable flash devices, the last bit of the address is not needed, because flash memory is read and programmed in terms of two bytes. For parallel mode, Flash A1 and Flash B1 (or Flash A2 and Flash B2) are accessed in parallel. As a result, the flash memory address sent to flash devices should be divided by 2. Table 30-9 indicates the relationship of row address, column address, and flash address (FA).

Table 30-9. Flash address, row address, and column address

Parallel mode	Word-addressable	Row address	Column address	Comment
0	0	FA[31:CAS]	FA[CAS-1:0]	There is no limitation on FA and data size alignment.
0	1	FA[31:CAS+1]	FA[CAS:1]	FA and data size should be two-byte aligned.
1	0	FA[31:CAS+1]	FA[CAS:1]	FA and data size should be two-byte aligned.
1	1	FA[31:CAS+2]	FA[CAS+1:2]	FA and data size should be four-byte aligned.

NOTE

- FA is the flash address without base address.
- If the bit number of the row or column flash address is more than the valid value, the high position bits are

supplemented with zero. See Programmable sequence engine for details about row or column address instructions.

When parallel mode is enabled or word-addressable flash memory is used, there are limitations on the flash memory start address and data size. You can address this requirement by aligning the AHB bus access address (for AHB command) or IP command address IPCR0[SFAR] (for IP command) in software. There are two ways to avoid these limitations in the specified case.

1. For AHB Read Command only:

   When AHBCR[READADDROPT] and AHBCR[PREFETCHEN] are both 1, or OTFAD is enabled, FlexSPI guarantees the start address and data size for the flash memory access are eight-byte aligned by hardware.

   When AHBCR[READADDROPT] is 1, or OTFAD is enabled, FlexSPI fetches redundant data to guarantee the flash memory start address is eight-byte aligned. When prefetch is enabled (AHBCR[PREFETCHEN] is 1), the size of the eight-byte-aligned AHB receive buffer determines the flash memory read data size.

2. For AHB Write Command and Individual Mode only:

   By default, FlexSPI guarantees the flash memory write access start address and data size are two-byte aligned when using DQS as write mask (FLSHCR4[WMENB or WMENA] = 1).

   This feature is not applied in parallel mode and must be used if the external device supports write mask feature.

30.3.5 Lookup table (LUT)

The LUT is an internal memory that preserves a number of preprogrammed sequences. Each sequence consists of up to eight instructions which are executed sequentially. When an IP command or an AHB command triggers a flash memory access, the FlexSPI controller:

1. Fetches the sequence from LUT (sequence index or number).
2. Executes the flash memory access to generate a valid flash transaction on the SPI interface.

Figure 30-5 shows the LUT structures, sequences, and instructions.

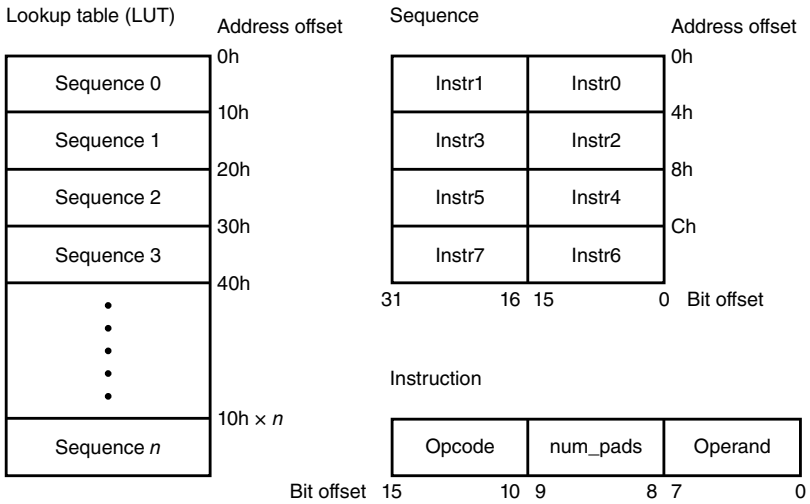


Figure 30-5. LUT structure, sequences, and instructions

NOTE

If the number of instructions needed for a flash transaction is less than eight, the STOP instruction (code 0000_0000h) must be programmed for unneeded instructions.

For IP and AHB write commands, the FlexSPI controller always executes from instruction pointer 0. For AHB read commands, the FlexSPI controller executes from a saved instruction start pointer. The FlexSPI controller saves the instruction start pointers separately for each flash device. All these saved instruction pointers are zero before the JMP_ON_CS instruction is executed. When the JMP_ON_CS instruction is executed, the operand in the JMP_ON_CS instruction is saved as instruction start pointer. See Execute-In-Place (XIP) enhanced mode.

The reset value of the LUT is unknown because it is implemented as internal memory. The LUT must be programmed according to the connected device. To protect its contents during a code runover, the LUT can be locked or unlocked to prevent unwanted changes after the LUT has been configured. The key to lock or unlock the LUT is **5AF05AF0h**.

To lock the LUT:

1. Write the key (5AF05AF0h) to LUT Key (LUTKEY).
2. Write 1 to LUTCR[LOCK] and write 0 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

To unlock the LUT:

1. Write the key (5AF05AF0h) to the LUT Key Register LUT Key (LUTKEY).

2. Write 0 to LUTCR[LOCK] and write 1 to LUTCR[UNLOCK]. When there is another register write access to FlexSPI between these two write accesses, this LUT is not successfully locked.

The lock status of the LUT can be read from LUTCR[LOCK] and LUTCR[UNLOCK].

30.3.6 Programmable sequence engine

The FlexSPI controller implements a programmable sequence engine that executes the sequence from the LUT. The FlexSPI controller executes the instructions sequentially, and generates flash transactions on the SPI interface accordingly. Table 30-10 is a complete list of the supported instructions.

Table 30-10. Instruction set

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
CMD_SDR	00_0001h	00h - one pad (Single mode) 01h - two pads (Dual mode) 02h - four pads (Quad mode) 03h - eight pads (Octal mode)	Transmit command code to flash memory	Command code: Operand[7:0]	Bit number: 8
CMD_DDR	00_0021h				
RADDR_SDR	00_0002h		Transmit row address to flash memory See Flash address sent to flash memory devices	Row_Address[31:0] Row_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	Bit number: operand[7:0] Value of operand determines number of bits sent in Row_Address.
RADDR_DDR	00_0022h				
CADDR_SDR	00_0003h		Transmit column address to flash memory. See Flash address sent to flash memory devices.	Column_Address[31:0] Column_Address comes from AHB bus (AHB command) or IPCR0[SFAR] (IP command).	Bit number: operand[7:0] Value of operand determines number of bits sent in Column_Address.
CADDR_DDR	00_0023h				
MODE1_SDR	00_0004h		Transmit mode bits to flash memory	Mode bits: Operand[0]	Bit number: 1
MODE1_DDR	00_0024h				
MODE2_SDR	00_0005h			Mode bits: Operand[1:0]	Bit number: 2
MODE2_DDR	00_0025h				
MODE4_SDR	00_0006h			Mode bits: Operand[3:0]	Bit number: 4
MODE4_DDR	00_0026h				
MODE8_SDR	00_0007h			Mode bits: Operand[7:0]	Bit number: 8
MODE8_DDR	00_0027h				
WRITE_SDR	00_0008h		Transmit program data to flash device	Program data in IP_TX_FIFO or AHB_TX_BUF	AHB burst size and burst type (AHB command) or IPCR1[IDATSZ] (IP
WRITE_DDR	00_0028h				

Table continues on the next page...

Table 30-10. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
READ_SDR	00_0009h		Receive read data from flash device Read data is put into AHB_RX_BUF or IP_RX_FIFO.	-	command) determines byte number (data size). For details about flash read or program data size, see Flash memory access via AHB command and Flash memory access via IP command.
READ_DDR	00_0029h				
DATSZ_SDR	00_000Bh		Transmit read or program data size (byte number) to flash device	Internal logic Read or program data size for current command sequence	Bit number: operand[7:0] Never set operand to zero or greater than 64 for DATSZ instruction.
DATSZ_DDR	00_002Bh				
DUMMY_SDR	00_000Ch		Leave data lines undriven by FlexSPI controller. Turnaround cycles are provided from host driving to device driving. num_pads determines number of pads in input mode.	-	Dummy cycle number (in serial root clock): Operand[7:0] Dummy cycle (N), described in data sheet of flash device, is in number of SCLK cycles. This number may be configurable. In SDR mode, SCLK cycle is same as serial root clock. Operand value must be set to N. In DDR mode, SCLK cycle is double serial root clock cycle. Operand value must be set to 2N, 2N-1 or 2N+1 depending on definition of dummy in data sheet of flash device. See Flash memory access sequence examples and dummy cycle definition in data sheet of external memory.
DUMMY_DDR	00_002Ch				
DUMMY_RWDS_SDR	00_000Dh		Similar to DUMMY_SDR/ DUMMY_DDR instruction. Difference is in dummy cycle number. DQS pin is called RWDS in HyperBus specification. See Dummy instruction for details. **Set operand to "Latency count" for HyperBus devices.**	-	For read command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 1) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 1) if RWDS (DQS pin) is low; For write command, dummy cycle number (in serial root clock): (operand[7:0] × 4 - 2) if RWDS (DQS pin) is high; (operand[7:0] × 2 - 2) if RWDS (DQS pin) is low;
DUMMY_RWDS_DDR	00_002Dh				

Table continues on the next page...

Table 30-10. Instruction set (continued)

Name	Opcode	num_pads	Action on SPI interface	Transmit data	Bits/bytes/cycle number
JMP_ON_CS	00_001Fh	Num_pads setting ignored.	Stop execution, deassert CS and save operand[7:0] as instruction start pointer for next sequence. Normally this instruction is used to support Execute-In-Place enhanced mode. See Execute-In-Place (XIP) enhanced mode. This instruction is only allowed for AHB read commands. When using this instruction in IP command or AHB write command, interrupt status flag set (INTR[IPCMDERR] or INTR[AHBCMDERR]).	-	No transaction on SPI interface.
STOP	00_0000h		Stop execution and deassert CS. Next command sequence (to same flash device) starts from instruction pointer 0.	-	

The programmable sequence engine allows configuration of the LUT according to the connected external serial device. The flexible LUT structure easily adapts to new command or protocol changes from different vendors.

The DDR sequence is a flash memory access sequence that contains DDR instructions other than DUMMY_DDR, and may contain SDR instructions. The output and input timing on FlexSPI differs for SDR and DDR sequences. When included as part of a DDR sequence, SDR instructions execute differently from SDR instructions in an SDR sequence. See FlexSPI input timing and FlexSPI output timing.

30.3.6.1 Executing instructions on SPI interface

This section describes the execution of instructions on the SPI interface. For all instructions that receive bits from or transmit bits to flash devices:

- The bit order in one byte is higher on DATA3 than DATA0.
- The bit order is higher on port B than port A.

This order is shown in Figure 30-6.

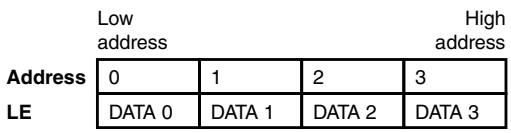


Figure 30-6. Bus bit order

Table 30-11. SPI instructions

Instructions	Description
Command (CMD_SDR, CMD_DDR)	Normally used to transmit a command code to the external device. The command code is the 8-bit operand in the instructions. Command code is sent to both port A and port B in parallel mode. See Flash memory access sequence examples.
Address (RADDR_SDR, RADDR_DDR, CADDR_SDR, CADDR_DDR)	Normally used to send the flash memory access start address (row address or column address) to an external device. FlexSPI determines the bits of the row address or column address according to the AHB access address or IP command address. See Flash address sent to flash memory devices. The operand value in the instruction code represents the number of address bits to send. The bits in the row address or column address are sent to both port A and port B in parallel mode. For example, when NUM_PADSn = 8, a memory reading the flash memory address on four SCK edges requires the sum of bits in CADDR and RADDR instructions to be 32. Otherwise, FlexSPI does not generate four SCK edges of address phase. See Flash memory access sequence examples.
Mode (MODEx_SDR, MODEx_DDR)	Normally used to send mode bits to external devices. Mode bits are the lower bits of an operand. For example, the bit number is 1 for MODE1_SDR and MODE1_DDR, 2 for MODE2_SDR and MODE2_DDR, 4 for MODE4_SDR and MODE4_DDR, and 8 for MODE8_SDR and MODE8_DDR. The pad number should not be greater than the mode bit number. For example, you cannot set NUM_PADSn to 11b (Octal mode) for MODE4_* instructions. Mode bits are sent to both port A and port B in parallel mode. See Flash memory access sequence examples for more details.
Data Size (DATSZ_SDR, DATSZ_DDR)	Used to send the size of program data or read data (byte number) to external devices. This instruction is normally used in FPGA applications where the memory space in external device acts like a FIFO. The external device requires data size information to determine how much data are popped from or pushed into the internal FIFO. The operation value in the data size instructions is the bit number. Data size is sent to both port A and port B in parallel mode. See Flash memory access sequence examples.
Write (WRITE_SDR, WRITE_DDR)	Normally used to send program data to external device. Programming data is fetched from IP_TX_FIFO (IP Command) or AHB_TX_Buffer (AHB command). For details about flash program data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for program data is always from low to high. Odd bytes are sent on port A and even bytes are sent on port B in parallel mode. See Flash memory access sequence examples for more details.

Table continues on the next page...

Table 30-11. SPI instructions (continued)

Instructions	Description
Read (READ_SDR, READ_DDR)	Normally used to receive flash memory data from external devices. Received data is put into IP_RX_FIFO (IP Command) or AHB_RX_Buffer (AHB command). For information about flash memory read data size, see Flash memory access via AHB command and Flash memory access via IP command. The byte order for reading data is always from low to high. Odd bytes are received from port A and even bytes are received from port B in parallel mode. See Flash memory access sequence examples.
Dummy (DUMMY_SDR, DUMMY_DDR, DUMMY_RWDS_SDR, DUMMY_RWDS_DDR)	Used to provide turnaround cycles on the SPI interface. During dummy instruction, the FlexSPI controller and external devices do not drive the SPI interface. See Programmable sequence engine for details about dummy cycle number. DUMMY_RWDS_DDR can be used for a HyperBus device that uses RWDS pin to indicate whether extra latency is needed. DUMMY_RWDS_SDR is reserved. The FlexSPI controller checks the DQS pin input level at the fourth cycle after SCLK output toggling is enabled. The DQS pin is called RWDS in HyperBus specification. See Flash memory access sequence examples. **NOTE:** FlexSPI releases the bus after at least one cycle. To avoid data contention, the NUM_PADSn value for the dummy commands should be configured to match the number of data lines used for the external memory.

30.3.6.2 Flash memory access sequence examples

Diagrams below all assume MCR0[SERCLKDIV] = 0.

Figure 30-7 shows an example SDR single I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

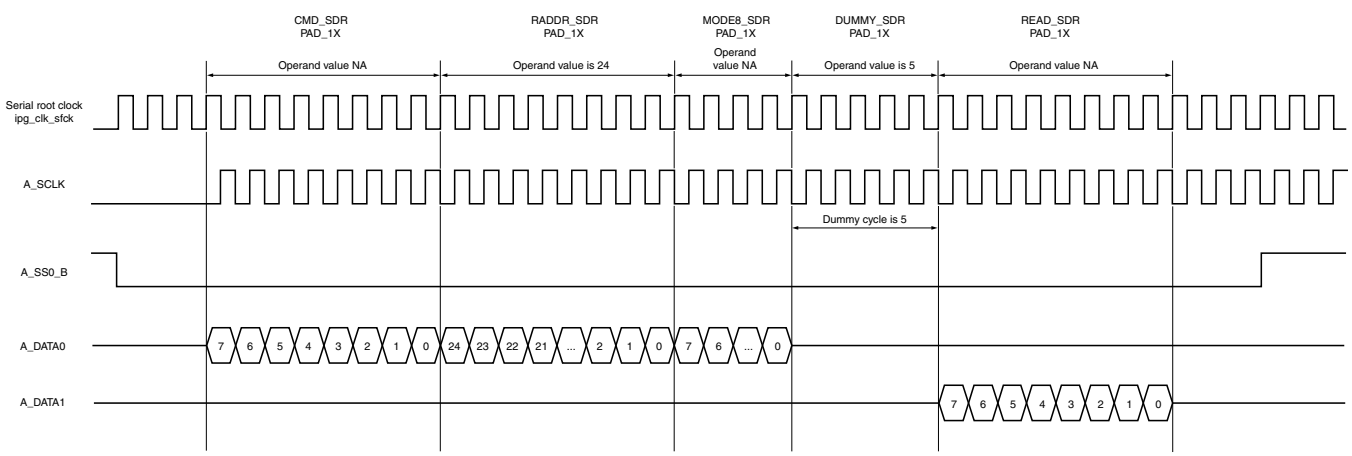


Figure 30-7. Flash memory access sequence example (SDR single I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 30-8 shows an example SDR quad I/O read sequence (Cypress Serial Nor Flash S25FS512S) in individual mode.

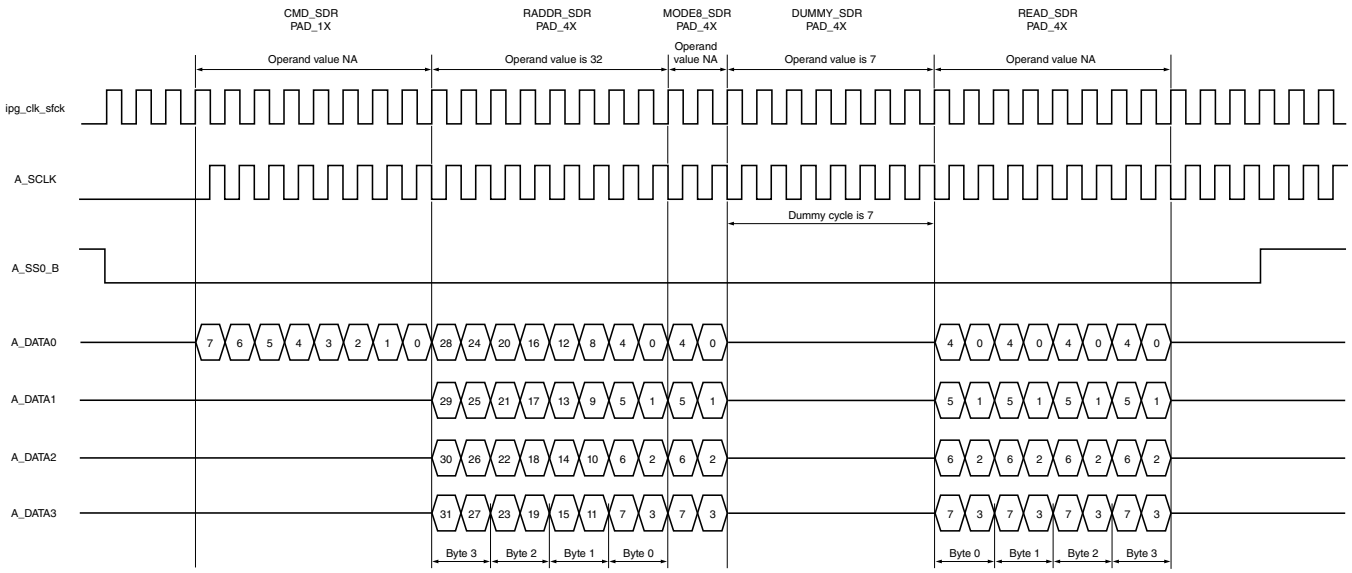


Figure 30-8. Flash memory access sequence example (SDR quad I/O read sequence)

NOTE

- FlexSPI dummy instruction starts and ends at the rising edge of serial root clock.
- Device dummy cycle starts and ends at the falling edge of SCLK (on Cypress S25FS512S data sheet).

Figure 30-9 shows an example DDR quad I/O read sequence (Cypress Serial Nor Flash S25FS512S) in parallel mode.

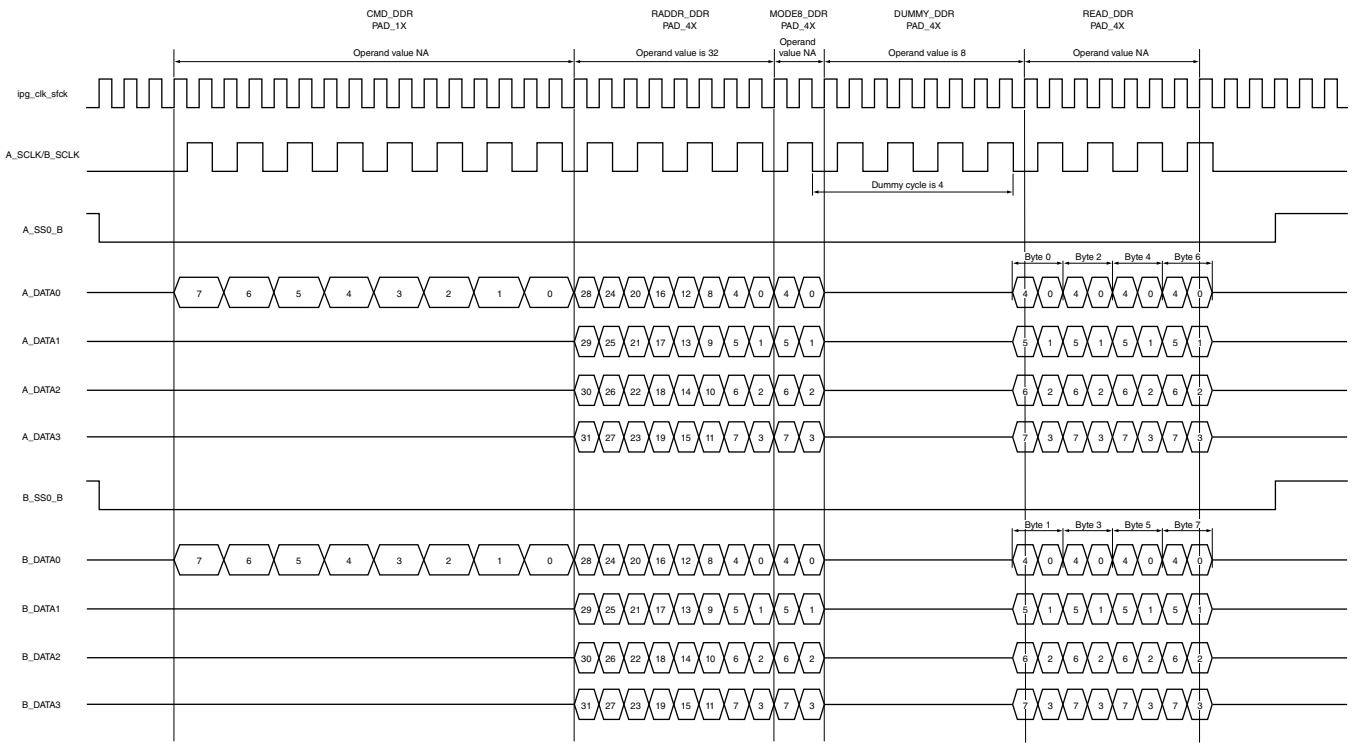


Figure 30-9. Flash memory access sequence example (DDR quad I/O read sequence)

Figure 30-10 shows an example HyperBus device read transaction (single latency count) in individual mode.

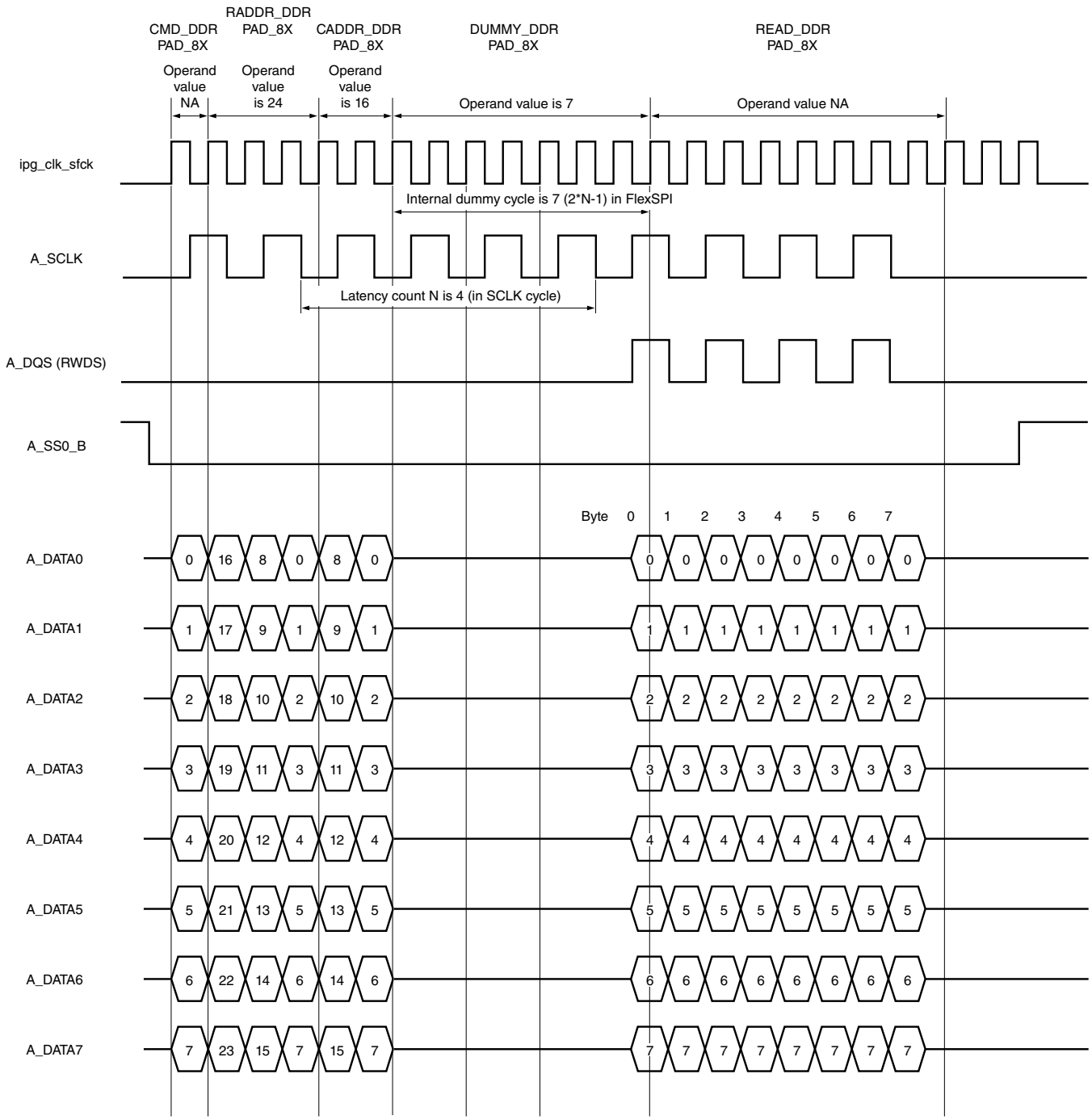


Figure 30-10. HyperBus device read transaction with single latency count

NOTE

FlexSPI continues to drive the clock until it has latched all of the read data it is expecting. On-chip delay for the data to reach FlexSPI can increase the number of SCK clock pulses.

Figure 30-11 shows an example HyperBus device read transaction (additional latency count) in individual mode.

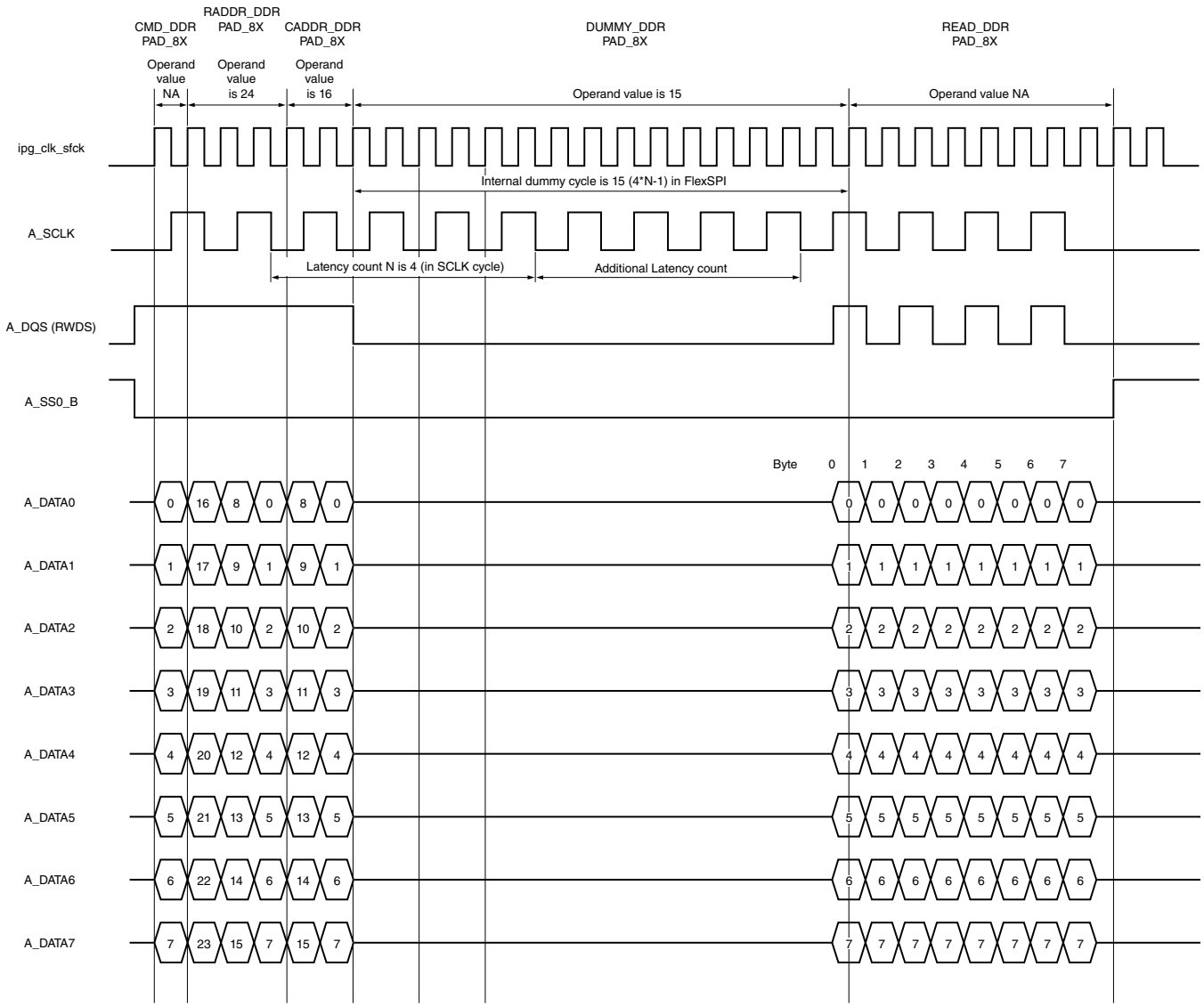


Figure 30-11. HyperBus device read transaction with additional latency count

Figure 30-12 shows an example HyperBus device write transaction (single latency count) in individual mode.

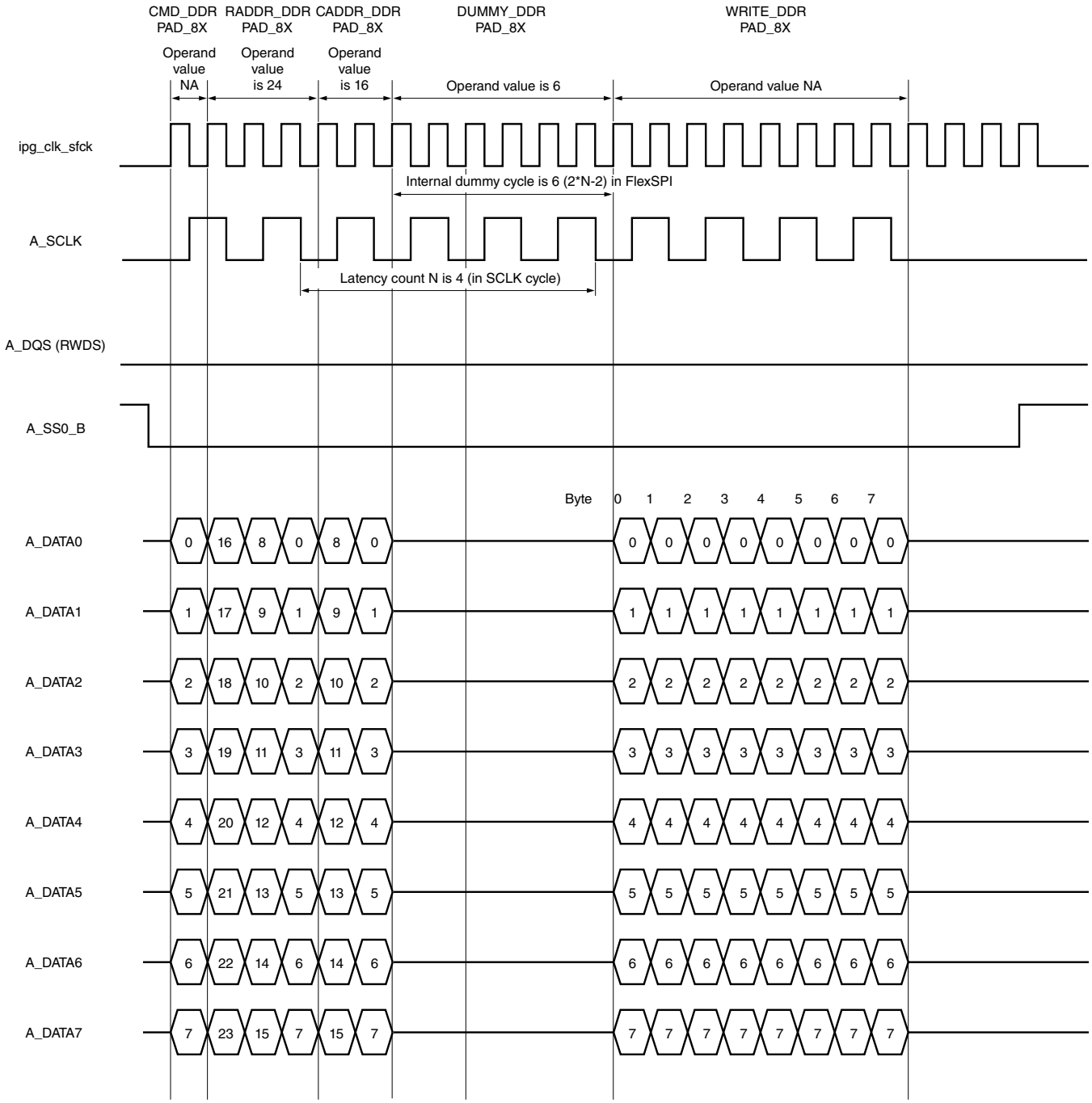


Figure 30-12. HyperBus device write transaction with single latency count

Figure 30-13 shows an example HyperBus device write transaction (additional latency count) in individual mode.

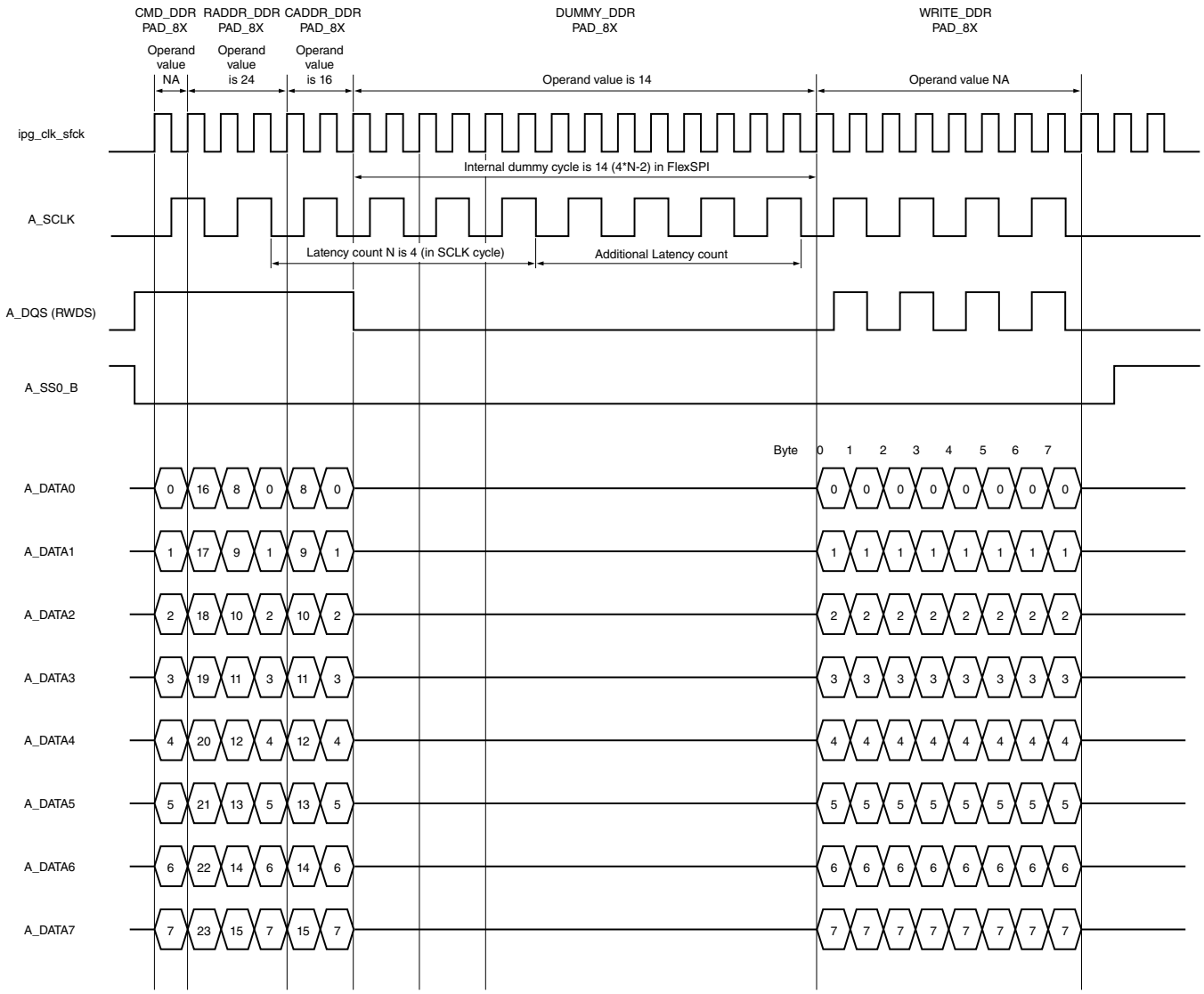


Figure 30-13. HyperBus device write transaction with additional latency count

30.3.7 Clocking

This section describes FlexSPI clocks and special clocking requirements.

Table 30-12. Clock usage

Clock name	Description	Comment
serial clock root (ipg_clk_sfck)	Root clock for Serial domain	FlexSPI core clock, used to generate the serial clock output. The clock is optionally divided using MCR0[SERCLKDIV] to obtain the SCLK.

Table continues on the next page...

Table 30-12. Clock usage (continued)

Clock name	Description	Comment
		Its frequency is same as SPI SCK frequency in SDR mode if MCR0[SERCLKDIV] = 0.
ahb clock (hclk)	AHB bus clock	hclk frequency is higher than 1/4 of serial root clock, so internal async FIFO is never full. This clock is used for internal logic in FlexSPI.
ipg clock (ipg_clk)	IPS bus clock	Register access clock. ipg_clk must be equal to hclk or an integer division of hclk. For example, ipg_clk = hclk/2.
SCLK	FlexSPI serial clock. Output clock on SCLK pin	Half clock frequency of serial clock root in DDR mode, and same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
DQS_OUT	Dummy Read Strobe output	Same frequency as SCLK. Clock output toggles during READ and LEARN instructions. The DQS ensures that the data can be strobed correctly. It is only used for data strobe, instead of command or address strobe. The data includes useful read data and data learning patterns.
DQS_IN	Sample clock for receive data	Same frequency as SCLK. Sample clock comes from looped-back dummy read strobe, looped-back SCLK, or flash-memory-provided read strobe.

30.3.8 Interrupts

Table 30-13 describes all interrupts that FlexSPI generates.

Table 30-13. Flash memory address, row address, and column address

Interrupt	Enable	Condition
IP command done	INTEN[IPCMDDONEEN]	An IP command is finished.
IP command grant error	INTEN[IPCMDGEEN]	An IP command grant timeout occurs (bus not granted after MCR0[IPGRANTWAIT] × 1024 AHB clock cycles). See Overview of error categories, flags, and triggered sources.
AHB command grant error	INTEN[AHBCMDGEEN]	An AHB command grant timeout occurs (bus not granted after MCR0[AHBGRANTWAIT] × 1024 AHB clock cycles). See Overview of error categories, flags, and triggered sources.

Table continues on the next page...

Table 30-13. Flash memory address, row address, and column address (continued)

IP command error	INTEN[IPCMDERREN]	A command check error or a command execution error for an IP command occurs. See Overview of error categories, flags, and triggered sources.
AHB command error	INTEN[AHBCMDERREN]	A command check error or a command execution error for AHB command occurs. See Overview of error categories, flags, and triggered sources.
IP_RX_FIFO watermark available	INTEN[IPRXWAEN]	The fill level of IP_RX_FIFO reaches the watermark level (IPRXFCR[RXWMRK]).
IP_TX_FIFO watermark empty	INTEN[IPTXWEEN]	The empty level of IP_TX_FIFO reaches the watermark level (IPTXFCR[TXWMRK]).
Sequence execution timeout	INTEN[SEQTIMEOUTEN]	A sequence execution time exceeds the timeout wait time (MCR1[SEQWAIT]). For example, the following flash memory read command sequence lasts about 800_0000h cycles (ipg_clk_sfck). If SEQWAIT is set to FFFFh, a sequence timeout interrupt is generated. • IP command triggers timeout. • Flash memory read data size is 100_0000h bytes. • Flash memory accesses in single mode and SDR mode.
AHB bus error	INTEN[AHBBUSERROREN]	AHB bus response timeout occurs, or AHB read access occurs while OTFAD is enabled and FlexSPI is still fetching the key blob. For example, AHB read sequence is not configured properly in LUT (such as without a READ instruction). No data is read from the external device. As a result, FlexSPI cannot reach the read data in AHB_RX_Buffers for the AHB read command.
Write command stops SCLK	INTEN[SCKSTOPBYWREN]	IP_TX_FIFO is empty while executing a write command sequence. FlexSPI stops SCLK output clock toggling and waits for write data filling.
Read command stops SCLK	INTEN[SCKSTOPBYRDEN]	IP_RX_FIFO is full while executing a read command sequence. FlexSPI stops SCLK output clock toggling and waits to read the data from IP_RX_FIFO.
Key done	INTEN[KEYDONEEN]	OTFAD is enabled and key blob processing is enabled, key blob processing is done, and no error is detected.
Key error	INTEN[KEYERROREN]	OTFAD is enabled and key blob processing is enabled, and an error occurs during key blob processing. Once

Table continues on the next page...

Table 30-13. Flash memory address, row address, and column address (continued)

		this interrupt is set, only reset can clear it.
ECC multibit error	INTEN[ECCMULTIERREN]	ECC is enabled, the current AHB read data has a multibit ECC error, and the ECC engine cannot correct this error.
ECC single-bit error	INTEN[ECCSINGLEERREN]	ECC is enabled, and the current AHB read data has a single-bit error

30.3.9 On-The-Fly AES Decryption (OTFAD) support

After power-up, if OTFAD and key fetches are enabled in the system, FlexSPI automatically starts fetching the key blobs from location 0 of flash A1. The key fetches continue until OTFAD receives and processes all key blobs. During the fetch operation, any AHB access causes an error response, and INTR[AHBBUSERROR] is set. No IPS access is allowed during the fetching of key blobs. INTR[KEYDONE] or INTR[KEYERROR] represents the key blob processing result.

After OTFAD processes all key blobs, the OTFAD engine decrypts any reads from FlexSPI and returns them on the AHB bus. When OTFAD is enabled, FlexSPI always enables prefetch; only AHBCR[READSZALIGN] can disable the prefetch. The Read Resume function is also disabled. All read access addresses are eight-byte-aligned automatically, regardless the setting of AHBCR[READADDROPT]. If OTFAD is enabled, the prefetch never crosses the 1 KB boundary.

30.3.10 AHB read ECC support

The purpose of ECC is to protect data transactions between the chip and external memory. When AHBCR[ECCEN] = 1:

- All data that the AHB bus reads from flash memory conducts ECC checking.
- Prefetch and address alignment are enabled.
- The resume function is disabled.

FlexSPI supports two wdata formats: swap wdata and raw wdata.

- When wdata is raw data, AHBCR[ECCSWAPEN] should be 0.
- When wdata is swap data, AHBCR[ECCSWAPEN] should be 1.

If the chip does not have an automatic ECC write data generation circuit, software should follow these steps to generate the write data:

1. To convert wdata to wdata_swap, perform a bit swap.

Table 30-14. wdata to wdata_swap

Bit Offset of Packed data	Byte 0	Byte 1	Byte 2	Byte 3
7	B3[7]	B0[7]	B1[7]	B2[7]
6	B2[6]	B3[6]	B0[6]	B1[6]
5	B1[5]	B2[5]	B3[5]	B0[5]
4	B0[4]	B1[4]	B2[4]	B3[4]
3	B3[3]	B0[3]	B1[3]	B2[3]
2	B2[2]	B3[2]	B0[2]	B1[2]
1	B1[1]	B2[1]	B3[1]	B0[1]
0	B0[0]	B1[0]	B2[0]	B3[0]

2. To convert 32 bits of wdata_swap into wdata_ECC, use the matrix shown below. wdata_swap[3:0] becomes wdata_ECC[3:0], wdata_swap[7:4] becomes wdata_ECC[7:4], and so on.

   0 1 1 1

   1 0 1 1

   1 1 0 1

   1 1 1 0

3. To convert wdata_ECC into wdata_ECC_swap, perform a nibble swap.

Table 30-15. wdata_ECC to wdata_ECC_swap

Bit offset of packed data	Byte 0	Byte 1	Byte 2	Byte 3
7	B0[3]	B1[3]	B2[3]	B3[3]
6	B0[2]	B1[2]	B2[2]	B3[2]
5	B0[1]	B1[1]	B2[1]	B3[1]
4	B0[0]	B1[0]	B2[0]	B3[0]
3	B0[7]	B1[7]	B1[7]	B2[7]
2	B0[6]	B1[6]	B2[6]	B3[6]
1	B0[5]	B1[5]	B2[5]	B3[5]
0	B0[4]	B1[4]	B2[4]	B3[4]

4. Write wdata_ECC_swap and wdata to memory.

For a read operation, FlexSPI uses the above steps in reverse.

Every 32 bits of actual data have one ECC single-bit error indication bit and one ECC multibit error indication bit. Therefore, every 64 bits of data have four bits of ECC information. 68 bits of data are stored in the AHB receive buffer for every 64 bits of data read from an external device.

When the AHB receive buffer read data indicates an ECC error in the current data, FlexSPI asserts a multibit error or single-bit error interrupt (if the corresponding interrupt is enabled).

In ECC single-bit error interrupt service, you can check Miscellaneous Control 5 (MISCCR5) and Miscellaneous Control 6 (MISCCR6). MISCCR5 and MISCCR6 store the information for the first ECC single bit-error that FlexSPI encountered since the last ECC single-bit error interrupt was served or reset. Before exiting service, toggle AHBCR[ECCSINGLEERRCLR] to clear the valid bit of MISCCR6.

In ECC multi error interrupt service, you can check Misc Control 4 (MISCCR4) and Miscellaneous Control 7 (MISCCR7). MISCCR4 and MISCCR7 store the information for the first ECC multibit error that FlexSPI encountered since the last ECC multibit error interrupt was served or reset. Before leaving service, software should toggle AHBCR[ECCMULTIERRCLR] to clear the valid bit of MISCCR7. Software can assert AHBCR[CLRAHBRXBUF] to clear the AHB receive buffer. FlexSPI then rereads the data from the external flash memory.

30.3.11 AHB transaction split

When AHBCR[SPLITEN] = 1, and FLSHxCR0[SPLITWREN] = 1 or FLSHxCR0[SPLITRDEN] = 1 , and the corresponding flash memory is accessed, all read and write transactions to external memory devices are split into subtransactions. This feature is used for RAM devices with a self-refresh limit that has a maximum chip select assertion time interval requirement. If FlexSPI operates at a low frequency, a single AHB burst can violate this timing requirement.

Consider an example where a DDR RAM with SCK = 20 MHz has only one data line. If an AHB burst is 128 bytes long, FlexSPI requires about 26 µs to finish the whole data reading. If the timing interval is 6 µs, this read operation violates the RAM refresh limitation. The RAM cannot refresh the data during the read operation, which may cause data corruption. The situation is similar for AHB write operations. This feature can be enabled to reduce the chip select assertion time.

NOTE

When this feature is enabled and the corresponding flash memory is accessed, FlexSPI splits one single burst into smaller

bursts based on AHBCR[SPLIT_LIMIT]. A long read burst is aligned to the next eight-byte-aligned data size. Prefetch is disabled.

Consider an example where the AHB read burst length = 42 bytes and AHCBR[SPLIT_LIMIT] = 00b. The 42-byte AHB read is aligned to 48 bytes, with an extra six bytes fetched. The entire burst is split into six subbursts of eight bytes each (8-8-8-8-8-8). The chip select signal to the external device asserts and deasserts six times.

Consider another example where the AHB write burst length = 42 bytes and AHCBR[SPLIT_LIMIT] = 00b. In this case, there is no burst size modification. The long write burst is split into five subbursts of eight bytes, and one subburst of two bytes (8-8-8-8-8-2).

NOTE

This feature is transparent to software and it significantly impacts performance, so enable it only when necessary. Base your decision on the maximum supported AHB burst size of the current chip, the clock frequency, and the refresh limitation of external RAM.

30.3.12 Flash memory access via IP command

Follow these steps to trigger a flash memory access via IP command.

1. If this command is a programing command, fill the IP transmit FIFO with programming data (for instance, programming flash data and flash memory status registers.)
2. Write the flash memory access start address to IPCR0[SFAR].
3. Write the read or program data size to IPCR1[IDATSZ]. Write the sequence index to IPCR1[ISEQID]. Write the sequence number to IPCR1[ISEQNUM].
4. Write 1 to IPCMD[TRG] to trigger a flash memory access command.
5. Wait for the INTR[IPCMDDONE] flag to set or for the IP command done interrupt to fire, indicating the command has completed on the FlexSPI interface.

NOTE

- The IP transmit FIFO can be filled before or after writing the IPCR0, IPCR1, and IPCMD registers. If the SFM command starts with the IP transmit FIFO empty, FlexSPI stops SCLK toggling to wait for transmit data ready automatically.
- The IPCMD register must be written after writing the IPCR0 and IPCR1 registers.

- One IP command can issue up to eight command sequences.
- You cannot issue another IP command before the previous IP command is finished. Performing this action results in unknown behavior.

If this command is a read command, all read data from flash memory is put into the IP receive FIFO. Software must read data from the IP receive FIFO via AHB bus or IP bus. When the IP receive FIFO is full and the flash device still contains data to read, FlexSPI stops SCLK toggling until the FIFO has free space. See SCLK stop.

A triggered serial flash command contains:

- A flash memory access start address. IPCR0[SFAR] determines this address.
- A flash chip select. The flash memory access address and flash memory size setting (FLSHxCR0[FLSHSZ]) determine the chip select.
- A flash command sequence index and sequence number. FlexSPI sequentially executes the sequences indexed from IPCR1[ISEQID] to (IPCR1[ISEQID] + IPCR1[ISEQNUM]) in LUT.
- Flash individual/parallel memory access mode:

  Determined by IPCR1[IPAREN].

- The flash memory read or program data size, in bytes.
  - If the value of IPCR1[IDATSZ] is nonzero, the data size is IPCR1[IDATSZ].
  - If the value of IPCR1[IDATSZ] is zero, the operand value in the read or write instruction determines the data size.

NOTE

- Software must ensure that the last sequence index never exceeds the LUT sequence number. That is, IPCR1[ISEQID] + IPCR1[ISEQNUM] must be less than 16.
- For sequence numbers greater than one, the data size is applied to every command sequence.
- If there is no write or read instruction in the command sequence, the data size is ignored.

The IP command request is sent to the arbitrator after software triggers it. It is not executed on the FlexSPI interface until the arbitrator grants it. See Command arbitration.

30.3.12.1 Reading data from IP receive FIFO

FlexSPI puts read data from the external device into the IP receive FIFO for IP commands.

The data stored in the IP receive FIFO can be read using IPS and register or system and AHB bus transactions:

- Via IP bus
- Via AHB bus

If MCR0[ARDFEN] = 1, only the AHB bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using IP bus always returns with data zero, and no bus error occurs.

If MCR0[ARDFEN] = 0, only the IPS bus can read the read data in the IP receive FIFO. Read access to the IP receive FIFO using the AHB bus triggers a bus error.

FlexSPI pushes read data into the IP receive FIFO in sets of 64 bits each time it receives 64 bits of data from an external device. When the number of read data bits is not 64-bit-aligned, FlexSPI pushes additional zero bits into the IP receive FIFO for the last push.

The processor or DMA can read the IP receive FIFO.

30.3.12.1.1 Reading via processor

To read data from the IP receive FIFO via processor, configure the items below.

- Write 0 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPRXWAEN] to enable the IP receive FIFO watermark available interrupt (optional).

The processor must poll INTR[IPRXWA] or wait for the IP receive FIFO watermark available interrupt before reading data from the FIFO. Waiting ensures that a watermark level amount of data is filled in the IP receive FIFO before reading.

After reading a watermark level amount of data from the FIFO, software must set the INTR[IPRXWA] flag to pop that data from the FIFO.

Figure 30-14 shows the reading flow from the IP receive FIFO via processor.

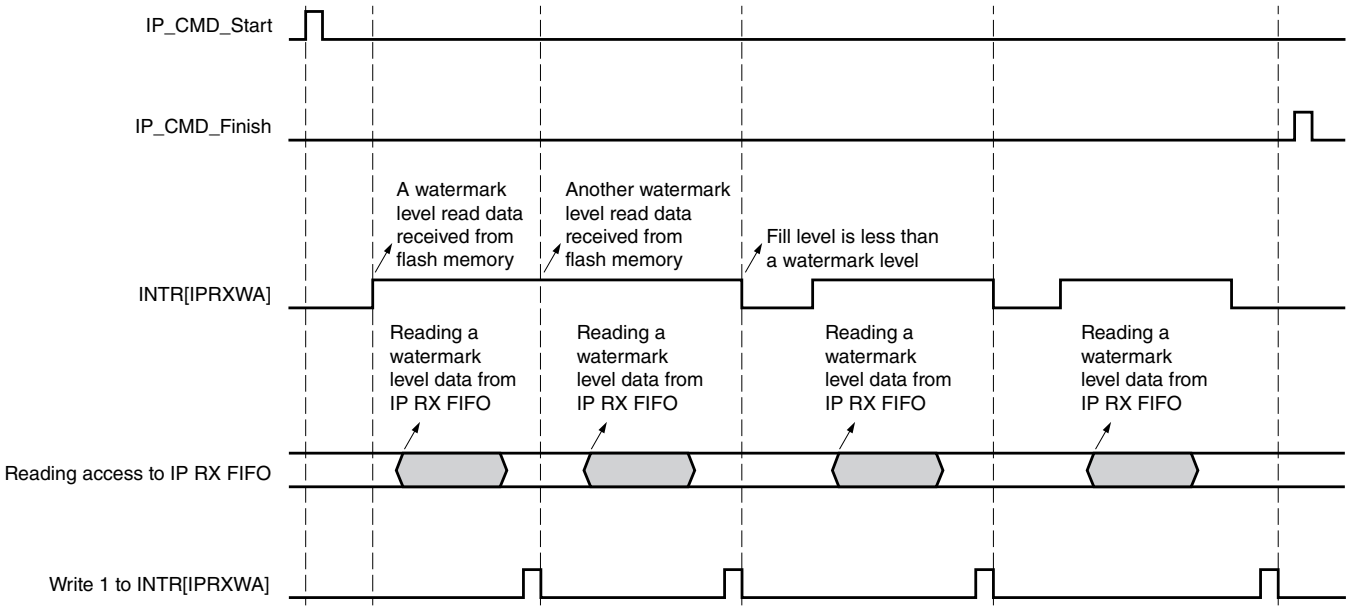


Figure 30-14. Reading IP receive FIFO via processor

NOTE

- The processor must read a watermark level of data from the IP receive FIFO before setting the INTR[IPRXWA] flag.
- The total flash read or program data size is not required to be a multiple of the watermark level. When the reading data size from the FIFO is less than a watermark level for the last time, software polls IPRXFSTS[FILL], not INTR[IPRXWA]. After all data from the FIFO are copied and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), software sets IPRXFCR[CLRIPRXF] to clear the FIFO. If the FIFO is not cleared, the reading data will be incorrect for the next reading command to flash memory.
- Setting the INTR[IPRXWA] flag pops the IP receive FIFO data. Each read access to the FIFO does not pop the data.

30.3.12.1.2 Reading via DMA

To read data from the IP receive FIFO via DMA, configure the items below.

- Write 1 to IPRXFCR[RXDMAEN].
- Write the watermark level to IPRXFCR[RXWMRK]. The watermark level is (IPRXFCR[RXWMRK] + 1) × 8 bytes.
- Configure the DMA transfer minor loop size to match the watermark level.

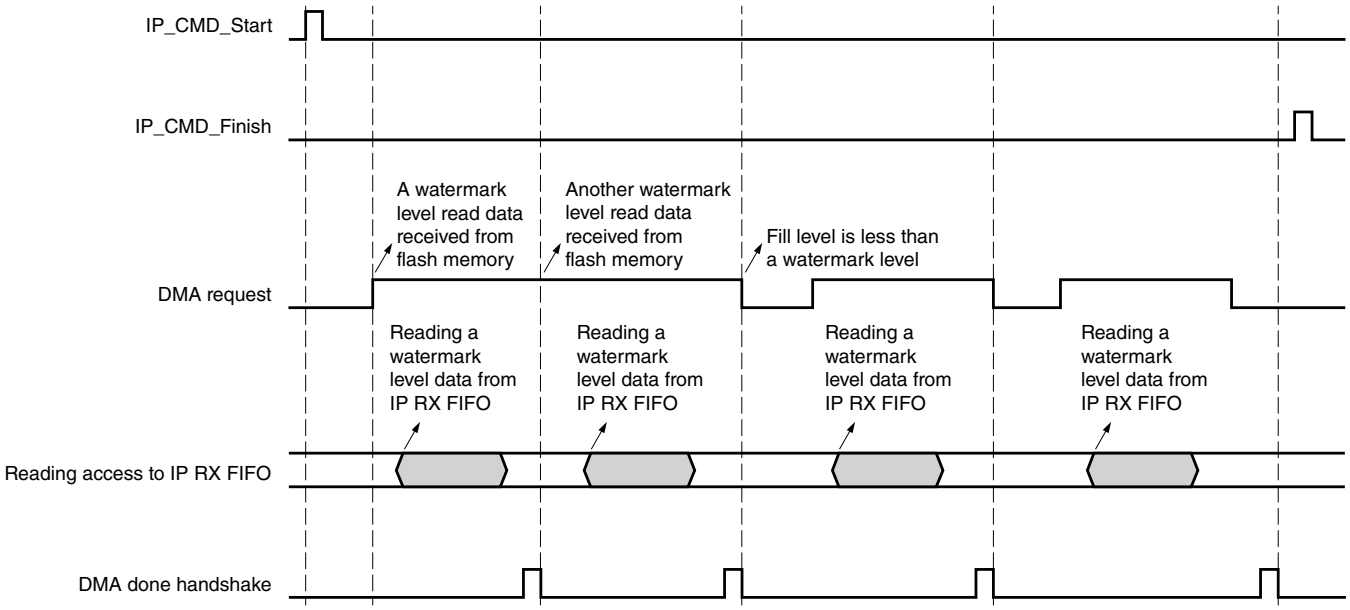


Figure 30-15. Reading from IP receive FIFO via DMA

NOTE

- When the fill level of the FIFO meets or exceeds the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA reads a watermark level amount of data from the IP receive FIFO each time. (Configure the minor loop size to match the watermark level).
- DMA must return a done handshake (pulse valid) to FlexSPI each time it finishes reading a watermark level amount of data.
- The DMA done handshake (not each read access) pops the IP receive FIFO data.
- The total read and write data size (major loop size) must be a multiple of the watermark level. The DMA does not know when the data is ready for the last reading. When the data size is not a multiple of the watermark level, it is too complex for the DMA driver to poll IPRXFSTS[FILL].

30.3.12.2 Writing data to IP transmit FIFO

The programming data is put into the IP transmit FIFO, then FlexSPI transmits it to the flash memory. IPS or AHB access can write it:

- Via IP bus
- Via AHB bus

To write the programming data into the IP transmit FIFO via AHB bus, write 1 to MCR0[ATDFEN]. The IP bus write access to the FIFO is ignored, but no bus error occurs.

To write the programming data into the IP transmit FIFO via IP bus, write 0 to MCR0[ATDFEN]. The AHB bus write access to the FIFO returns a bus error.

When FlexSPI fetches data to transmit, 64 bits of data from the IP transmit FIFO are popped. If the programming data size is not a multiple of 64 bits, the number of last popped valid bits is less than 64 bits. No error occurs; these invalid bits are not transmitted to the flash memory.

The processor or DMA can fill the IP transmit FIFO with programming data.

30.3.12.2.1 Writing via processor

To write data into the IP transmit FIFO via processor, configure the items below.

- Write 0 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.
- Write 1 to INTEN[IPTXWEEN] to enable the IP transmit FIFO watermark available interrupt (optional).

The processor must poll INTR[IPTXWE] or wait for the IP transmit FIFO watermark empty interrupt before writing data to the FIFO. Waiting ensures that a watermark level amount of data is available in the IP transmit FIFO before writing.

After writing a watermark level amount of data to the FIFO, software must set the INTR[IPTXWE] flag to push that data into the FIFO. (The write pointer is incremented.)

NOTE

Setting the INTR[IPTXWE] flag pushes the IP transmit FIFO data. Each write access to the FIFO does not push the data.

Figure 30-16 shows the writing flow to the IP transmit FIFO via processor.

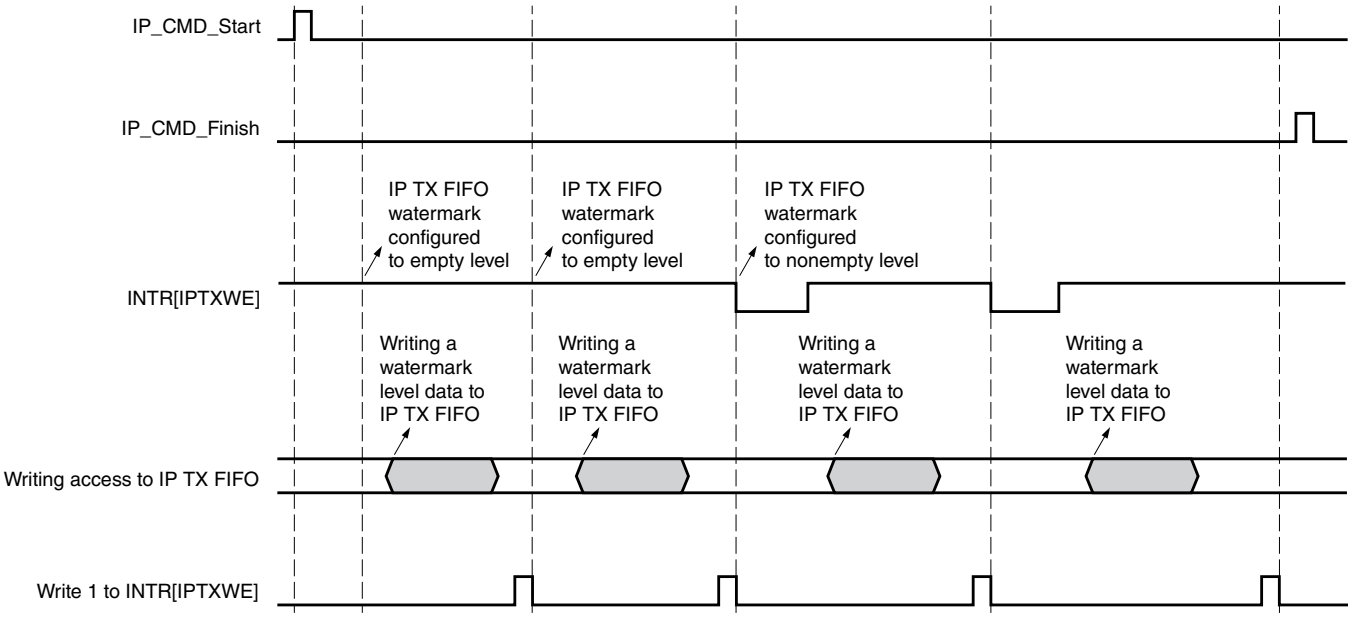


Figure 30-16. Writing to IP transmit FIFO via processor

NOTE

- The processor must write a watermark level amount of data into the IP transmit FIFO each time.
- The total flash write data size is not required to be a multiple of the watermark level. In this case, the writing data size to the FIFO is less than a watermark level for the last time. After all data to the FIFO are written and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), the processor sets IPTXFCR[CLRIPTXF] to clear the FIFO. If the FIFO is not cleared, the programming data will be incorrect for the next programming command to flash memory.

30.3.12.2.2 Writing via DMA

To write data into the IP transmit FIFO via DMA, configure the items below.

- Write 1 to IPTXFCR[TXDMAEN].
- Write the watermark level to IPTXFCR[TXWMRK]. The watermark level is (IPTXFCR[TXWMRK] + 1) × 8 bytes.
- Configure the DMA transfer minor loop size to match the watermark level.

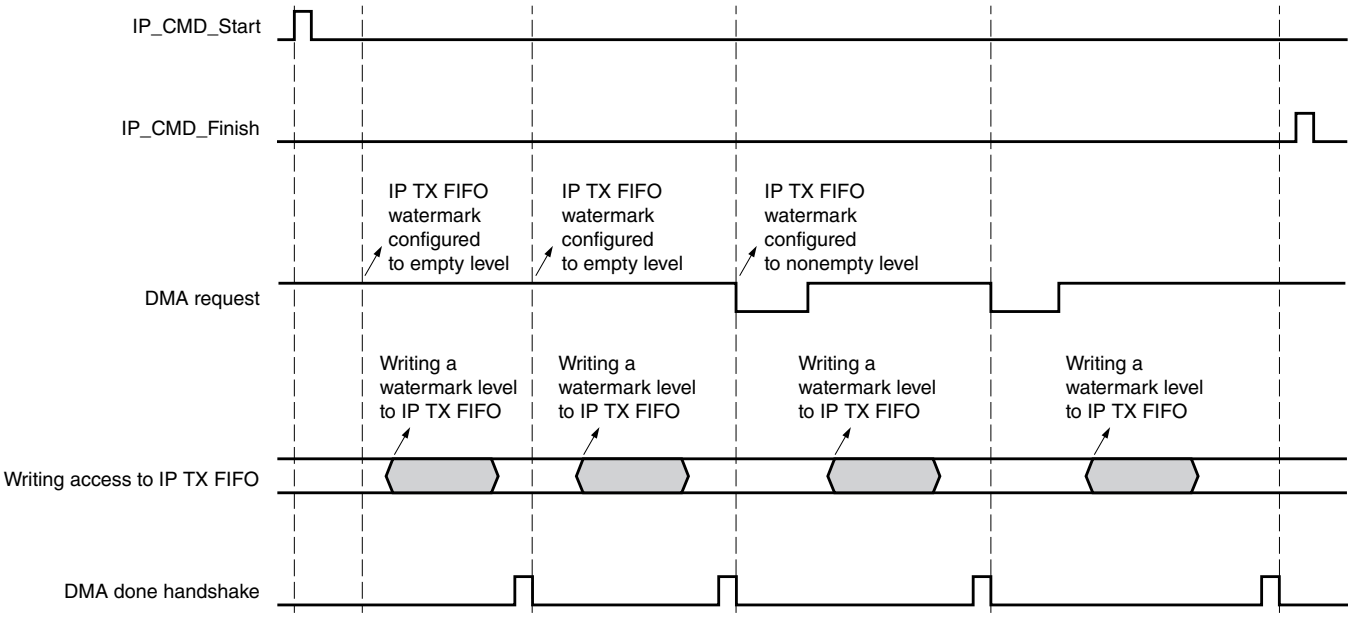


Figure 30-17. Writing into IP transfer FIFO via DMA

NOTE

- When there is empty space in the IP transmit FIFO greater than the watermark level, a DMA request is generated. This request is level-valid, not pulse-valid.
- DMA writes a watermark level amount of data into the FIFO each time. (Configure the minor loop size to match the watermark level.
- DMA returns a done handshake (pulse valid) to FlexSPI each time it finishes writing a watermark level amount of data.
- The DMA done handshake (not each write access) pushes the IP transmit FIFO data.
- The total program data size (major loop size) is not required to be a multiple of the watermark level. After writing all data into the IP transmit FIFO and all command sequences to flash memory are finished (INTR[IPCMDDONE] = 1), write 1 to IPTXFCR[CLRIPTXF] to clear the FIFO. Failure to clear the FIFO results in incorrect programming data for the next programming command to flash memory.

30.3.13 Flash memory access via AHB command

Flash memory can be accessed via AHB bus directly on the AHB address space. This address space is mapped to serial flash memory in FlexSPI. AHB bus accesses to this address space trigger flash memory access command sequences as needed.

For AHB read access to serial flash memory, FlexSPI fetches data from flash memory into the AHB receive buffer, then returns the data on the AHB bus. For AHB write access to serial flash memory, FlexSPI buffers AHB bus write data into the AHB transmit buffer, then transmits the data to serial flash memory.

No software configuration or polling is needed for AHB commands, except for FlexSPI initialization. As with a normal AHB target, the AHB controller accesses the external flash device transparently.

AHB commands are normally used to access serial flash memory space. IP commands must be used to access the control and status registers or other spaces, such as a one-time programmable (OTP) space in an external flash device.

The AHB commands for read and write access are discussed in detail below.

30.3.13.1 AHB write access to flash memory

For AHB write access to flash memory, FlexSPI buffers the write data from the AHB bus into the internal AHB transmit buffer. FlexSPI then transmits the data to flash memory. FlexSPI only buffers write data for one AHB burst. Figure 30-18 shows the hardware operation in response to an AHB write access to flash memory.

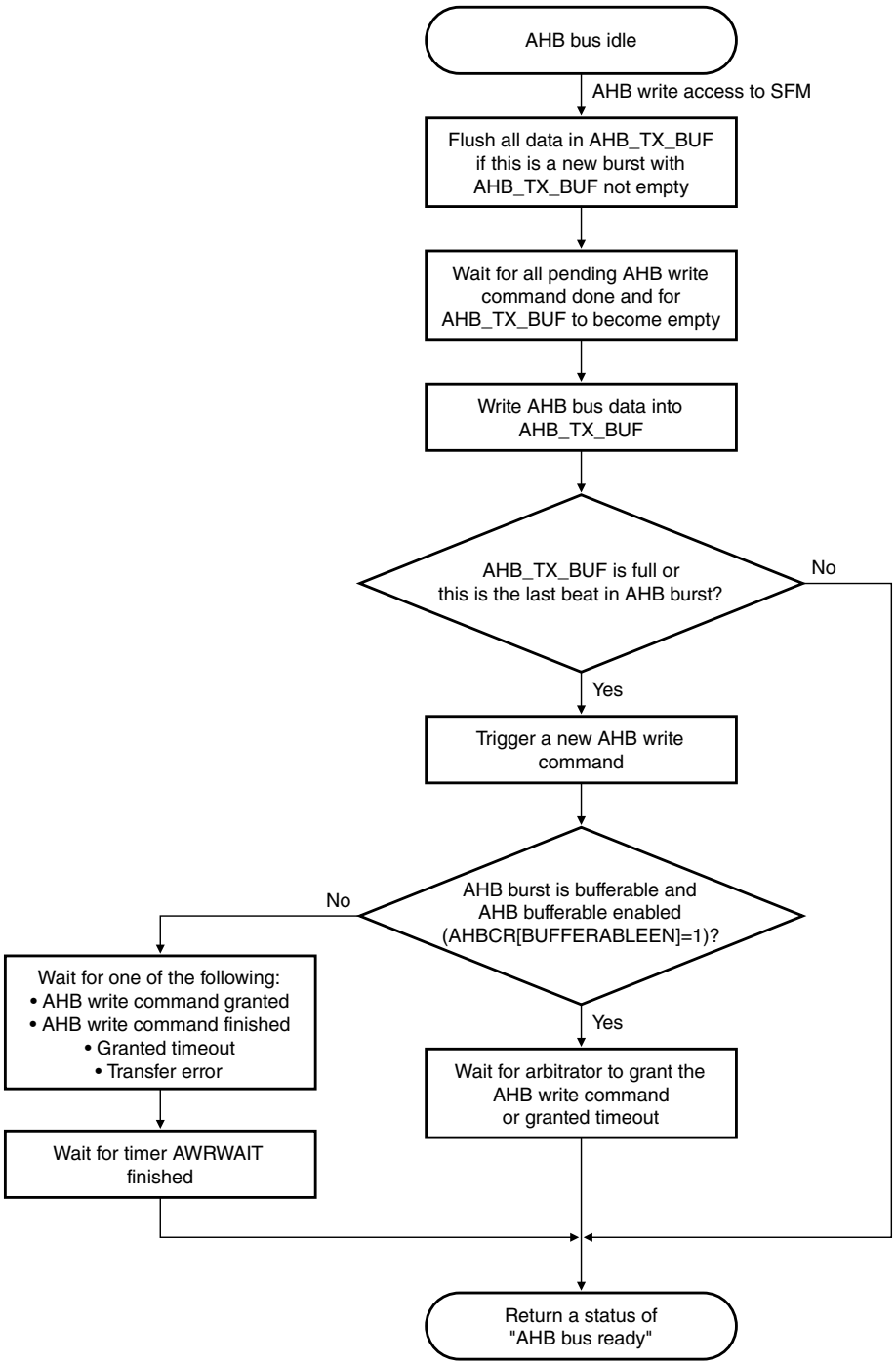


Figure 30-18. Hardware operation in response to AHB write access to flash memory

NOTE

1. If previous AHB burst is INCR write burst, AHB transmit buffer may not be empty when AHB bus is idle or new burst arrives. A new AHB write command is triggered to flush all data in AHB transmit buffer to external flash device. The new burst is held until this AHB write command finishes.

2. If the current access is bufferable, when the AHB write command is triggered, AHB ready is returned after the write command is granted. If the current access is non-bufferable, AHB ready is returned after the write command finishes.

FlexSPI triggers a new AHB write command in these cases:

- This beat is the last one in the current AHB burst, for any burst type except INCR.
- AHB transmit buffer is full after the current beat data is buffered.
- AHB bus becomes idle or a new burst arrives with the AHB transmit buffer not empty.

Table 30-16 describes the details of a triggered AHB write command.

Table 30-16. Triggered AHB write command

Item	Description
Flash memory access start address	Determined by AHB burst address. FlexSPI records the start address for the data in AHB transmit buffer, which is used as flash memory access start address.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip selection.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB write command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual/ Parallel)	Determined by AHBCR[APAREN].
Flash memory data size	Determined by byte number of buffered data in AHB transmit buffer.

The following examples indicate internal logic for AHB write access to flash memory, where AHB_TX_BUF is 64 bytes :

- AHB INCR 64-bit bufferable burst with address sequence 8h, 10h, 18h, 20h, ..., 50h (10 beats total).

  Two AHB write commands are triggered: the first command with flash memory start address 8h and data size 40h; the second command with flash memory start address 48h and data size 10h. See Figure 30-19.

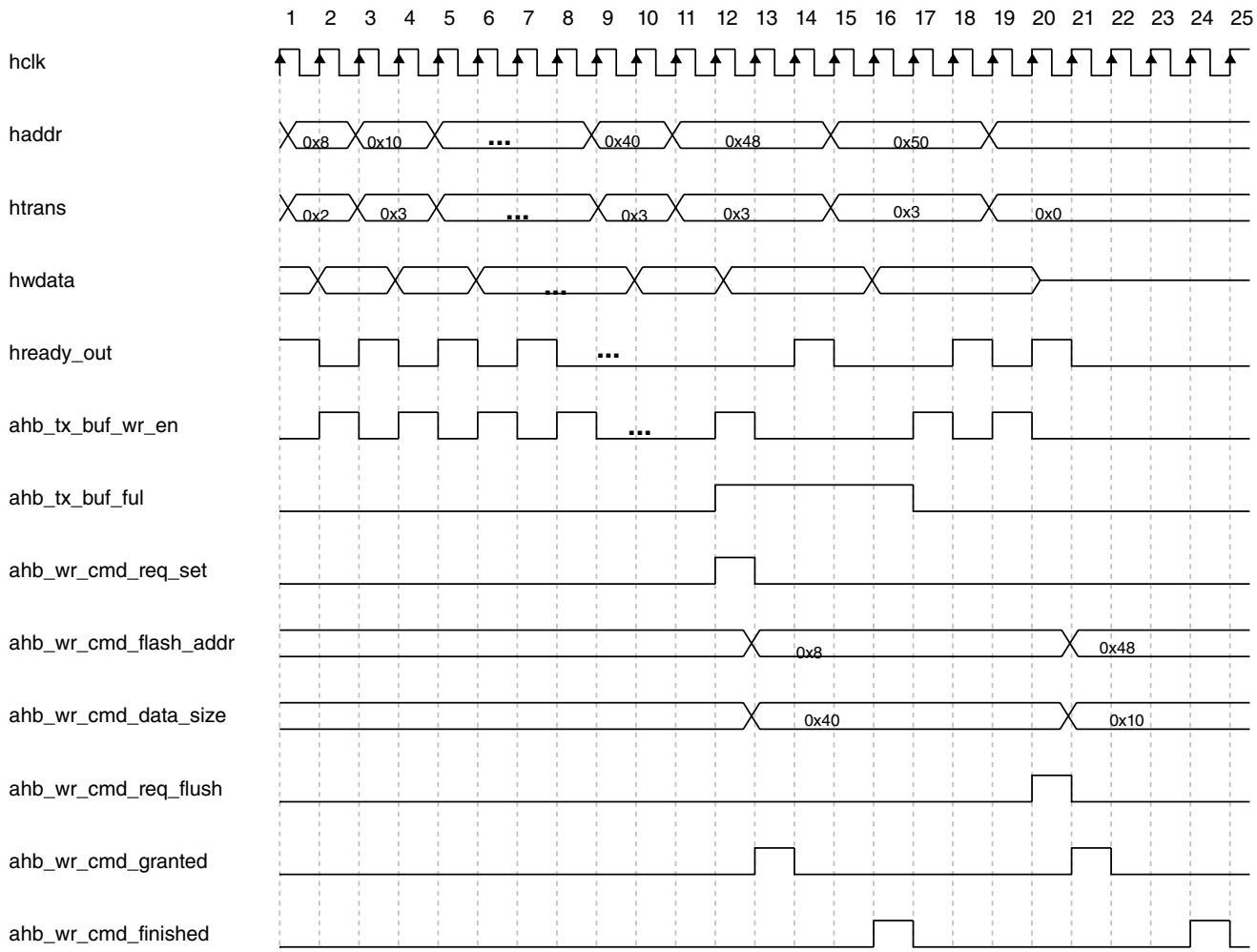


Figure 30-19. AHB write access (INCR, 64-bit, bufferable)

- AHB WRAP8 64-bit bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h. See Figure 30-20.

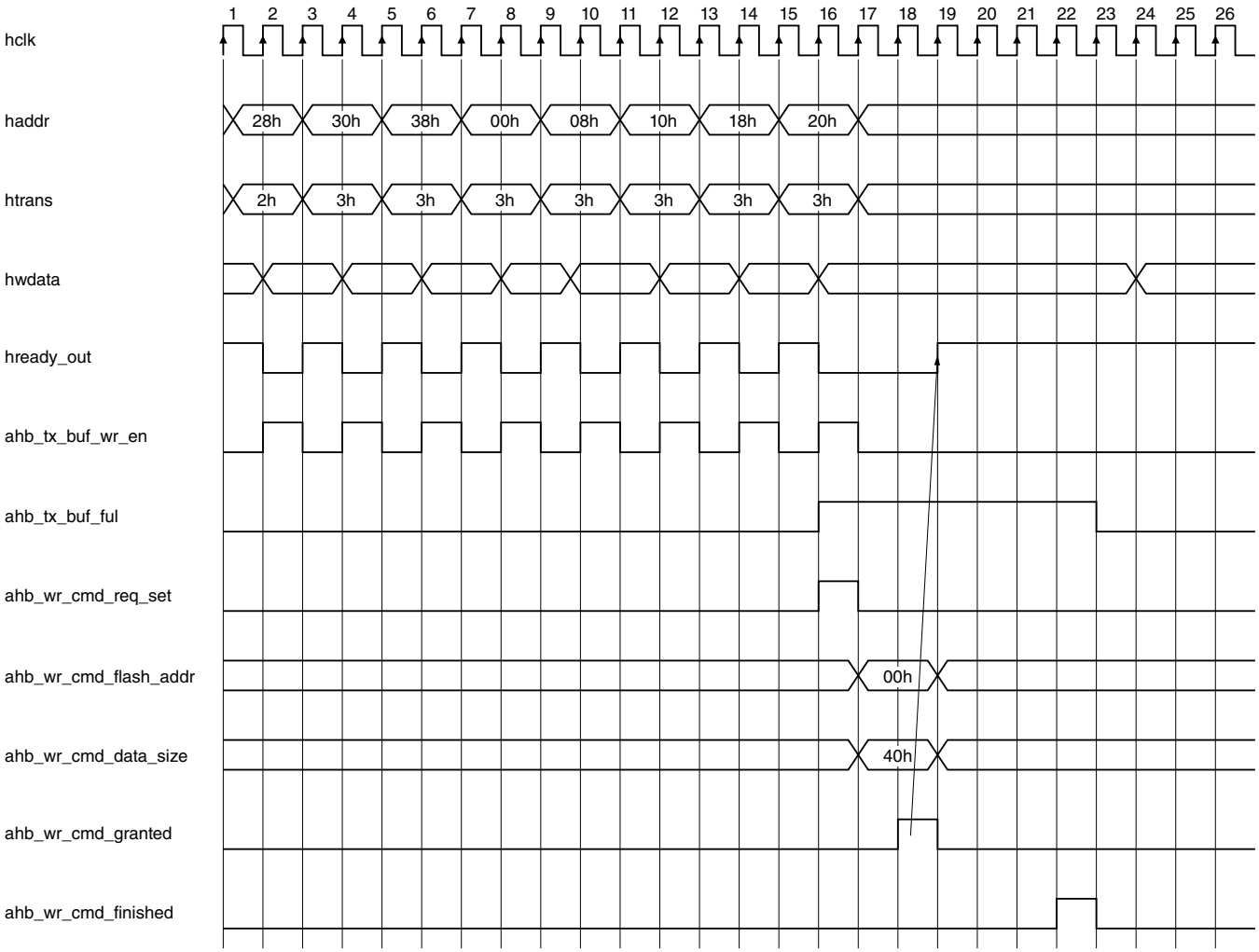


Figure 30-20. AHB write access (WRAP8, 64-bit, bufferable)

- AHB WRAP8 64-bit non-bufferable burst with address sequence 28h, 30h, 38h, 0h, 8h, 10h, 18h, 20h.

  One AHB write command is triggered with flash memory start address 0h and data size 40h.

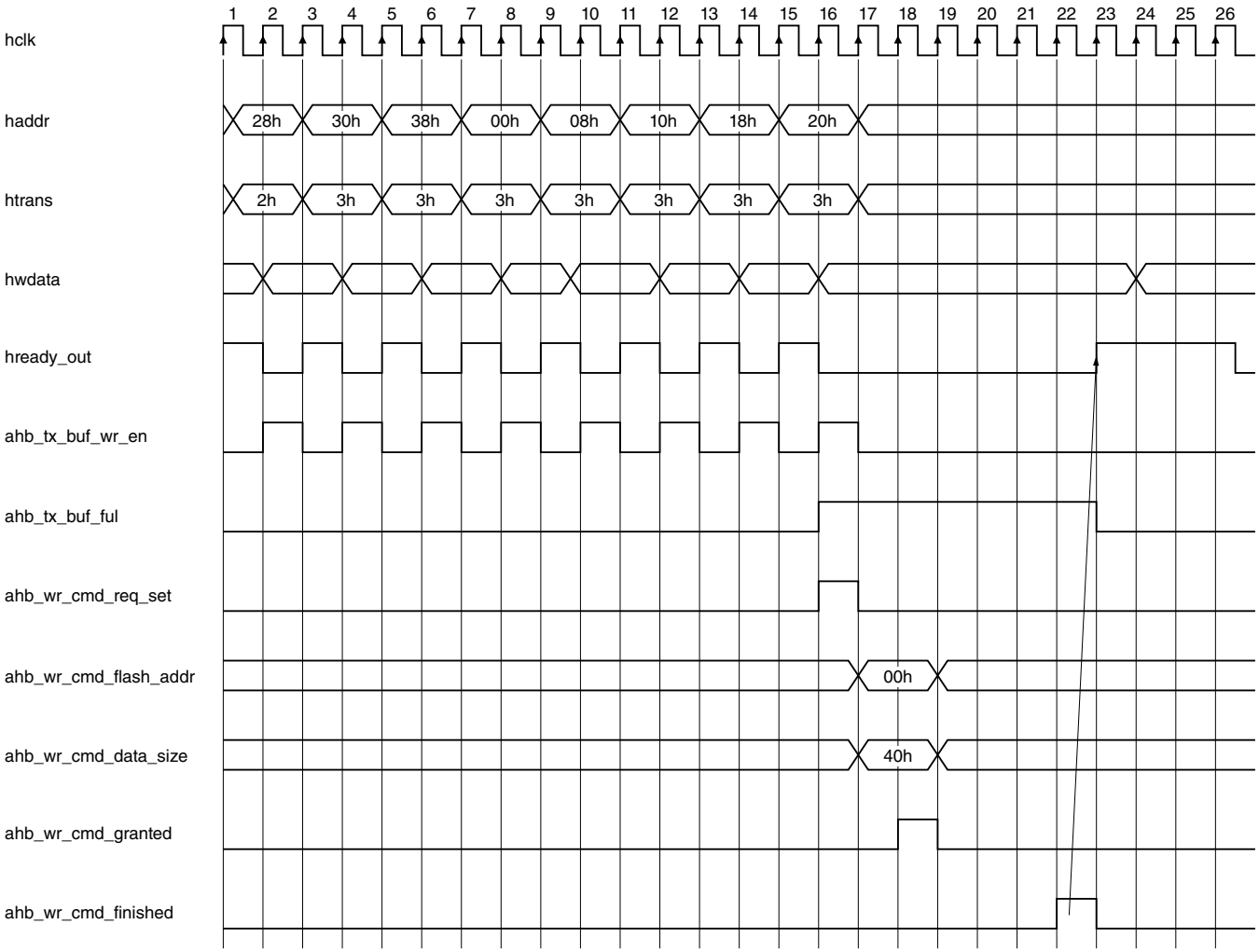


Figure 30-21. AHB write access (WRAP8, 64-bit, non-bufferable)

NOTE

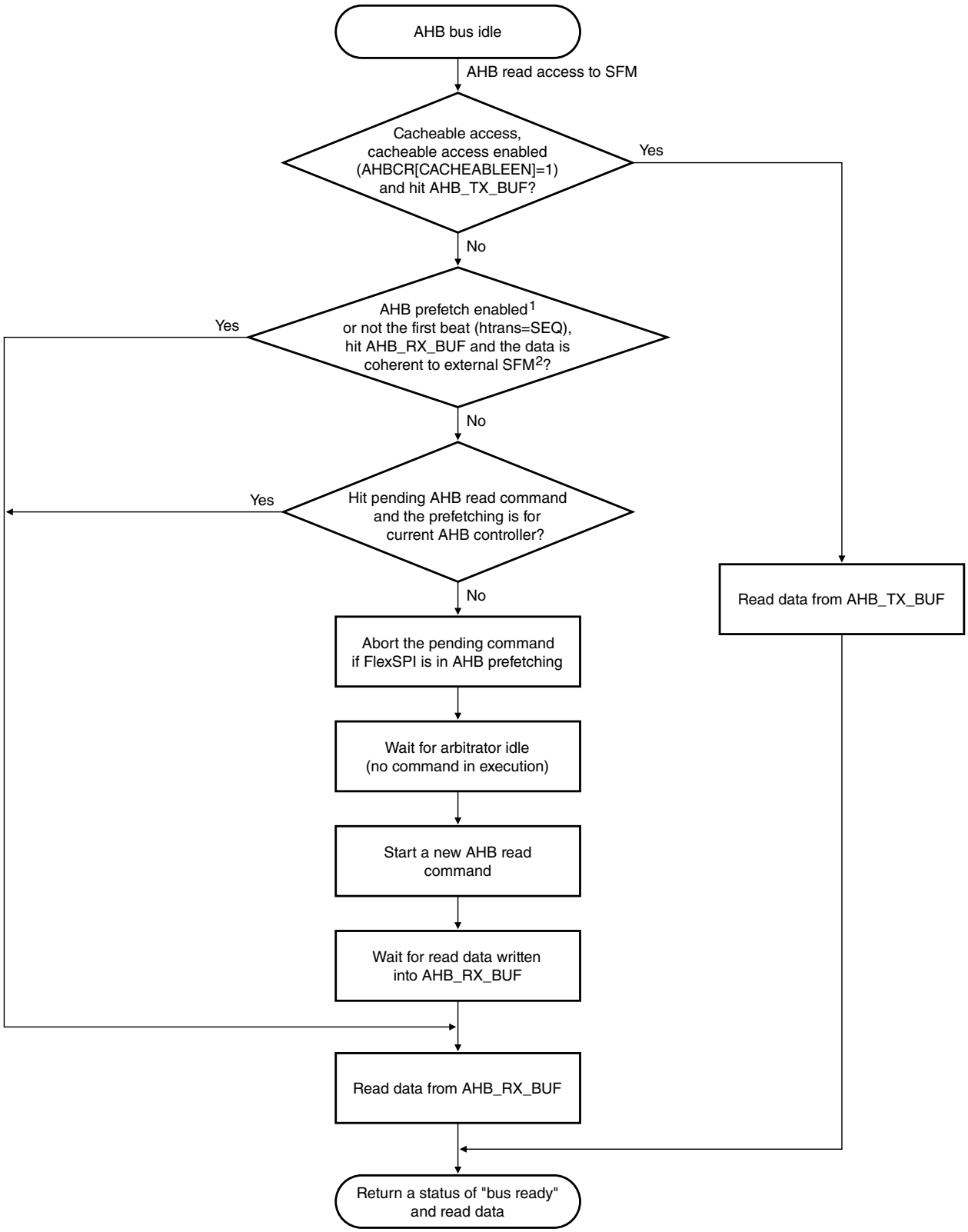
If the burst data size (in bytes) is greater than the AHB transmit buffer size, wrapper burst is not supported. For example, if AHB_TX_BUF is 64 bytes, AHB WRAP16 (16 × 64 bits = 128 bytes) write burst access is not supported.

30.3.13.2 AHB read access to flash memory

For AHB read access to flash memory, FlexSPI checks the burst access type and register setting to determine whether to access these locations:

- The AHB transmit buffer
- The AHB receive buffer
- A pending AHB read command

If all of these options are missed, FlexSPI triggers a new AHB read command to fetch data from flash memory. Figure 30-22 shows the hardware operation in response to AHB read access to flash memory.



Notes:
1) AHB prefetch is enabled when both AHBCR[PREFETCHEN]=1 and AHBRXBUFxCR0[PREFETCHEN]=1 (x is the AHB RX buffer ID for the current AHB controller) or OTFAD is enabled.
2) The AHB RX buffer holds the data read from flash memory. This data may become incoherent if AHB writes to the same flash-memory address.

Figure 30-22. Hardware operation in response to AHB read access to flash memory

Table 30-17 describes the details of a triggered AHB read command.

Table 30-17. Triggered AHB read command

Item	Description
Flash memory access start address and data size	Determined by AHB address, burst type, and burst size. FlexSPI fetches read data from the start address for the current burst or beat. See Table 30-18 for details.
Flash memory chip select	FlexSPI uses the settings of the flash memory access start address and flash memory size to determine the chip select.
Flash memory command sequence index	Determined by FLSHxCR2[AWRSEQID].
Flash memory command sequence number	Determined by FLSHxCR2[AWRSEQNUM]. If AWRSEQNUM is not zero, multiple flash memory access command sequences are triggered every time for an AHB read command. The sequences indexed from AWRSEQID to (AWRSEQID + AWRSEQNUM) in LUT are executed sequentially.
Flash memory access mode (Individual/ Parallel)	Determined by AHBCR[APAREN].

NOTE

- FlexSPI automatically determines which ARDSEQNUM and ARDSEQID fields that the flash device uses as sequence ID for chip selection. See MCR2[SAMEDEVICEEN].
- FlexSPI determines which AHB receive buffer to use for the current AHB read access by controller ID. For details about the AHB receive buffer ID and AHB controller ID mapping, see AHB receive buffer management.
- You cannot allocate the size of the AHB receive buffer to be less than the AHB Burst size. This allocation results in unknown behavior.

Table 30-18. AHB read command flash memory start address and data size

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [27:0]	Data size
0	Never cross flash boundary, because AHB burst never crosses 1 KB boundary.	SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(hburst_end_address-hbeat_start_address)
		INCR	hbeat_start_address	byte size of current beat For INCR burst with prefetch disabled, each beat is handled same as SINGLE burst.
		WRAP4, WRAP8, WRAP16	hburst_start_address	(hburst_end_address-hburst_start_address)

Table continues on the next page...

Table 30-18. AHB read command flash memory start address and data size (continued)

Prefetch enable	Cross flash boundary	Burst type	Flash memory start address [27:0]	Data size
1	No	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	ahb_rx_buf_sz
	No	WRAP4, WRAP8, WRAP16	hburst_start_address	ahb_rx_buf_sz
	Yes	INCR, SINGLE, INCR4, INCR8, INCR16	hbeat_start_address	(flash_top_address-hbeat_start_address)
	Yes	WRAP4, WRAP8, WRAP16	hburst_start_address	(flash_top_address-hburst_start_address)

NOTE

- hbeat_start_address is HADDR input from the AHB controller for current beat.
- hburst_start_address (for example 00h) is the lowest address for current burst. hburst_end_address (for example 10h) is the highest address in current burst plus 1. For example, WRAP4 burst with HADDR 8h, Ch, 0h, 4h.
- ahb_rx_buf_sz is the buffer size in bytes of AHB receive buffer that the current AHB controller uses.
- flash_top_address is the top address of currently accessed flash memory.

30.3.13.3 AHB receive buffer management

There are 8 buffers (Buffer 0–7) in the AHB receive buffer. These buffers are transparent to AHB controllers. FlexSPI fetches flash memory data and returns it on the AHB bus automatically. No status register polling is needed for AHB read access to Serial Flash Memory (SFM).

See the chip-specific section for AHB buffer size.Use AHBRXBUF0CR0[BUFSZ]–AHBRXBUFCR7[BUFSZ] to configure the buffer size of each AHB receive buffer. The buffer size for Buffer 0 to Buffer 7 could be set to 0. If the buffer size is set to 0, FlexSPI ignores the related setting of AHBRXBUFxCR0[MSTRID]. Buffer 7 is used for all AHB controllers not assigned to Buffer 0–6. FlexSPI ignores the Buffer 7 size setting field (AHBRXBUF7CR0[BUFSZ]). Its buffer size is: (AHB receive buffer total size) - (the sum of the sizes of Buffer 0–6).

When an AHB read access to serial flash memory occurs, FlexSPI determines which buffer to use via this method:

1. If the controller ID equals AHBRXBUF0CR0[MSTRID] and AHBRXBUF0CR0[BUFSZ] is not zero, Buffer 0 is used.
2. If the controller ID equals AHBRXBUF1CR0[MSTRID] and AHBRXBUF1CR0[BUFSZ] is not zero, Buffer 1 is used.
3. If the controller ID equals AHBRXBUF2CR0[MSTRID] and AHBRXBUF2CR0[BUFSZ] is not zero, Buffer 2 is used.
4. If the controller ID equals AHBRXBUF3CR0[MSTRID] and AHBRXBUF3CR0[BUFSZ] is not zero, Buffer 3 is used.
5. If the controller ID equals AHBRXBUF4CR0[MSTRID] and AHBRXBUF4CR0[BUFSZ] is not zero, Buffer 4 is used.
6. If the controller ID equals AHBRXBUF5CR0[MSTRID] and AHBRXBUF5CR0[BUFSZ] is not zero, Buffer 5 is used.
7. If the controller ID equals AHBRXBUF6CR0[MSTRID] and AHBRXBUF6CR0[BUFSZ] is not zero, Buffer 6 is used.
8. If all the above cases are not met, Buffer 7 is used.

NOTE

- Software must ensure that the size of each buffer is not less than the maximum burst size of an AHB read access from the AHB controller using this buffer. Otherwise, the behavior is undefined. The minimum buffer size request depends on the chip implementation. For example, if the capability of the chip to send out a maximum burst is 64 bytes, the minimum AHB receive buffer is 64 bytes.
- Assigning multiple buffers to a single AHB controller is not supported, unless the prefetch buffer enhancement feature is being used.
- When AHB read prefetch is enabled (AHBCR[PREFETCHEN] = 1), the size of the buffer determines the prefetch data size. If it does not cross the flash boundary, FlexSPI fetches data from the external flash device with that buffer size.
- The priority setting of AHB controller (AHBRXBUFxCR0[PRIORITY] = 1) is used only for suspending control of AHB prefetching. See Command abort and suspend.

30.3.13.4 AHB controller ID remapping

The AHB controller uses AHB user bits instead of HMASTER to support multiple AHB receive buffers on one AHB controller. For example, a single AHB controller with two sub threads can be distinguished by using different AHB user bits to access FLEXSPI. The controller ID remapping feature is designed to support such user cases.

The related AHB user bits mask and expected controller value are defined in HMSTRnCR0-7. The function enable is AHBCR[HMSTRIDREMAP]

When there is an AHB read access to serial flash memory, FlexSPI remaps the controller ID as shown below.

1. If AHBCR[HMSTRIDREMAP] = 0, the remapping function is disabled and bus signal HMASTER[3:0] is used directly.
2. If AHBCR[HMSTRIDREMAP] = 1, AHB user bits [15:0] are ANDed with HMSTRCR0~7's bits [15:0], which is a masking field. This combining results in eight new user bits (for example, masked_user[15:0]_0-7_).
3. If masked_user[15:0]_0_ = HMSTRCR0[31:16], controller ID is mapped to 0h internally, which is compared to AHBRXBUFxCR0[MSTRID].
4. If masked_user[15:0]_1_ = HMSTRCR1[31:16], controller ID is mapped to 1h internally, which is compared to AHBRXBUFxCR0[MSTRID].
5. If masked_user[15:0]_2_ = HMSTRCR2[31:16], controller ID is mapped to 2h internally, which is compared to AHBRXBUFxCR0[MSTRID].
6. If masked_user[15:0]_3_ = HMSTRCR3[31:16], controller ID is mapped to 3h internally, which is compared to AHBRXBUFxCR0[MSTRID].
7. If masked_user[15:0]_4_ = HMSTRCR4[31:16], controller ID is mapped to 4h internally, which is compared to AHBRXBUFxCR0[MSTRID].
8. If masked_user[15:0]_5_ = HMSTRCR5[31:16], controller ID is mapped to 5h internally, which is compared to AHBRXBUFxCR0[MSTRID].
9. If masked_user[15:0]_6_ = HMSTRCR6[31:16], controller ID is mapped to 6h internally, which is compared to AHBRXBUFxCR0[MSTRID].
10. If none of the above listed cases are met, Buffer 7 is used.

30.3.14 Command arbitration

There are four flash memory access command sources:

1. AHB command, that AHB write access to serial flash memory space triggers
2. AHB command, that AHB read access to serial flash memory space triggers
3. IP command, that writing IPCMD[TRG] triggers
4. Suspended command, an AHB read prefetch sequence which is suspended

An AHB bus access never triggers a write command and a read command at the same time.

The AHB prefetch sequence is an AHB command sequence triggered via AHB read access when AHB prefetch is enabled. After all read data is fetched for the current AHB read burst, FlexSPI prefetches more data to reduce read latency for the next AHB read access. An AHB command for a read operation is never aborted while fetching read data for the current AHB read burst. However, a new IP command or AHB command request while prefetching data (not for the current read burst) can abort an AHB read command.

When the arbitrator is idle (STS0[ARBIDLE] = 1), the granted priority of these command sources is:

1. AHB command (read or write)
2. IP command
3. Suspended command

NOTE

When the arbitrator is idle and there is no AHB or IP command request, a suspended command is not granted immediately. The arbitrator waits MCR2[RESUMEWAIT] AHB clock cycle idle states before resuming the suspended command to avoid AHB prefetch sequence being resumed and suspended frequently.

If the arbitrator is busy executing AHB or IP commands (not suspended commands), no new command requests are granted. If the grant times out, an AHB or IP command granted error occurs.

If a new AHB or IP command request comes while arbitrator is executing AHB read prefetch sequence, AHB read prefetch sequence is aborted (but not immediately). Arbitrator grants the AHB or IP command request after AHB read prefetch sequence is aborted on the FlexSPI interface and saved all internal data pointers.

30.3.14.1 Command abort and suspend

Table 30-19 describes the command abort and suspend mechanism.

Table 30-19. Command abort and suspend

Action	Description
Aborting a command	As mentioned above, if a new AHB or IP command request arrives, an AHB read prefetch sequence can be aborted.
Suspending a command	When an AHB read prefetch sequence is aborted on the FlexSPI interface, FlexSPI saves this suspended sequence in the cases listed below. FlexSPI resumes this sequence after the arbitrator has been idle for enough time.

Table continues on the next page...

Table 30-19. Command abort and suspend (continued)

Action	Description
	• There is no valid suspended command (AHBSPNDSTS[ACTIVE] = 0). This condition can occur when no command is suspended or when the suspended command is resumed. • The priority of aborted AHB read prefetch sequence is higher than the active suspend sequence. **NOTE:** FlexSPI ignores and never resumes the original suspended sequence.
Activating a suspended command	The suspended command (internal status) becomes active when an AHB prefetch command is aborted and suspended. It becomes inactive in the following cases: • The arbitrator resumes the suspended command. • The AHB controller triggers a new AHB read command request using the same AHB receive buffer (buffer ID).

Figure 30-23 shows an example of the flow of command abort, suspend, or resume.

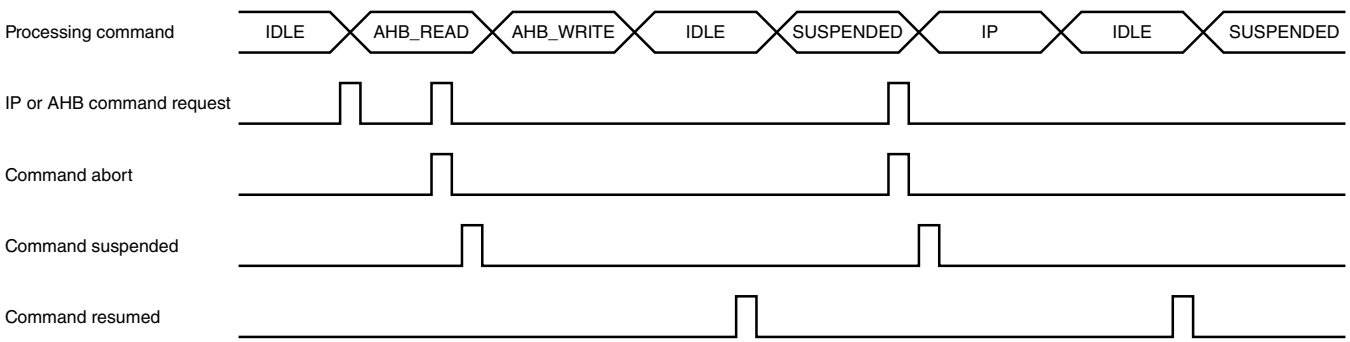


Figure 30-23. Command instruction execution on FlexSPI interface

NOTE

- The AHB prefetch sequence is aborted, suspended, or resumed twice in Figure 30-23. A new AHB write command request aborts the sequence the first time, and a new IP command request aborts the sequence the second time.
- The suspended sequence is resumed after a wait time of idle state. If there are frequent IP or AHB command requests and the wait time is not set reasonably, the suspend and resume activities may be triggered frequently.

30.3.15 SCLK stop

FlexSPI stops SCLK output toggling:

- When programming data is not ready for a programming command sequence.
- When an internal FIFO has no space to receive data for reading command sequence.

Certain devices may not support the stopping of SCLK during a command sequence (chip select is valid). SCLK stopping can be avoided via the methods described below.

- For flash memory reading that an IP command triggers:
  - Never trigger a read command with data size larger than IP receive FIFO size.
  - Internal asynchronous FIFO for flash memory reading should never be full.

    FlexSPI pops 64 bits of data per AHB clock cycle from this FIFO, and receives data from FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual or Parallel mode) determines the receiving speed. For example, in Octal mode and Parallel mode, FlexSPI receives 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never full.

- For flash memory programming that an IP command triggers:
  - Never trigger a program command with data size larger than the IP transmit FIFO size. Write all programming data into the IP transmit FIFO before triggering the IP command.
  - The internal asynchronous FIFO for flash memory programming must never be empty.

    FlexSPI fetches 64 bits of programming data per AHB clock cycle into this FIFO, and transmits data to the FlexSPI interface on the serial root clock. The flash memory access mode (Single, Dual, Quad, or Octal mode and Individual or Parallel mode) determines the transmitting speed. For example, in Octal mode and Parallel mode, FlexSPI transmits 16 bits per serial root clock cycle. If the AHB clock frequency is faster than 1/4 of the serial root clock, this asynchronous FIFO is never empty.

- For flash memory reading or programming that an AHB command triggers:
  - The internal asynchronous FIFO for flash memory reading and programming must never be full or empty. The frequency ratio limitation is same as it is for flash memory reading or flash memory programming that an IP command triggers.

NOTE

- FlexSPI never triggers an AHB read command with data size larger than the internal AHB receive buffer size.

- FlexSPI never triggers an AHB program command with data size larger than the internal AHB transmit buffer size.
- All programming data is buffered into the AHB transmit buffer before triggering an AHB program command in FlexSPI.

30.3.16 FlexSPI output timing

30.3.16.1 Output timing between data and SCLK

This section describes the output timing relationship of data (on A_DATA and B_DATA) and SCLK. There are three cases.

- **SDR instruction in SDR sequence**
  An SDR sequence contains only SDR instructions. In this case, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 30-24 shows the relationship between the serial root clock, data, and SCLK:

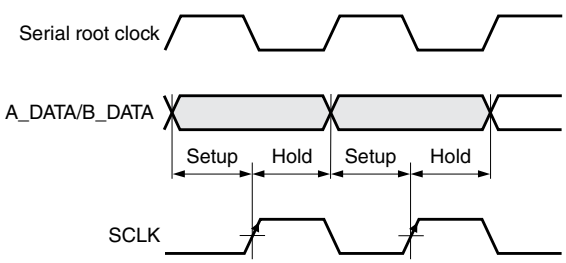


Figure 30-24. SDR instruction in SDR sequence

- **SDR instructions in a DDR sequence**

  A DDR sequence is a flash memory access command sequence that contains DDR instructions other than DUMMY_DDR. It may contain SDR instructions. If there are SDR instructions in a DDR sequence, all data bits last for two serial root clock cycles on the FlexSPI interface. Figure 30-25 shows the relationships between the serial root clock, data, and SCLK.

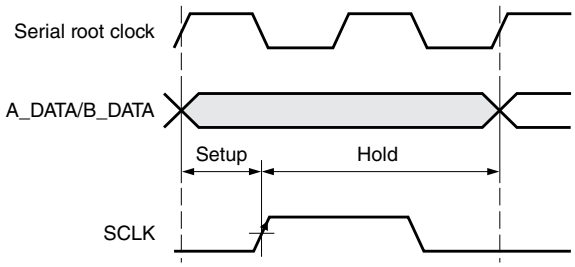


Figure 30-25. SDR instruction in DDR sequence

- **DDR instructions in a DDR sequence**

  For DDR instructions in a DDR sequence, all data bits last one serial root clock cycle on the FlexSPI interface. Figure 30-26 shows the relationships between the serial root clock, data, and SCLK.

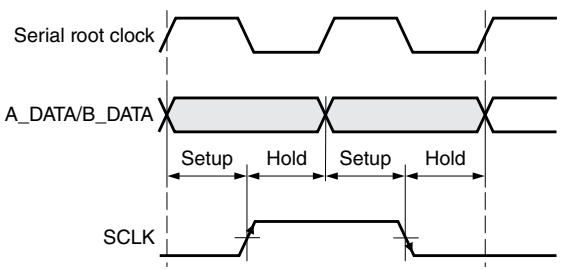


Figure 30-26. DDR instruction in DDR sequence

30.3.16.2 Output timing between chip select and SCLK

This section describes the output timing relationship between chip select (on A_SS0_B, A_SS1_B, B_SS0_B, and B_SS1_B) and SCLK. The timing relationship for the SDR sequence differs a little from the timing relationship for the DDR sequence.

- **Chip select timing in SDR sequence**

  For an SDR sequence, the delay from chip select assertion to the SCLK rising edge is (FLSHxCR1[TCSS] + 0.5) cycles of the serial root clock. The delay from SCLK falling edge to chip select deassertion is FLSHxCR1[TCSH] cycles of serial root clock. Figure 30-27 shows the timing relationship between the chip select and SCLK.

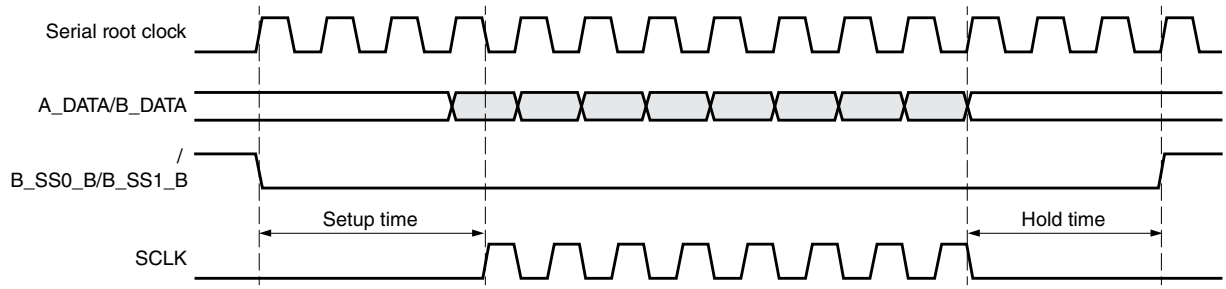


Figure 30-27. Chip select output timing for SDR sequence

- **Chip select timing in DDR sequence**

  For a DDR sequence, the delay from chip select assertion to the SCLK rising edge is **(FLSHxCR1[TCSS] + 0.5)** cycles of the serial root clock. The delay from the SCLK falling edge to chip select deassertion is **(FLSHxCR1[TCSH] + 0.5)** cycles of serial root clock.

  NOTE

  When AHB receive prefetch is enabled, and prefetch can be aborted, configure TCSH to 1 or greater. Configuring TCSH in this way guarantees a positive chip select hold time after the SCLK falling edge.

  Figure 30-28 shows the timing relationship between chip select and SCLK.

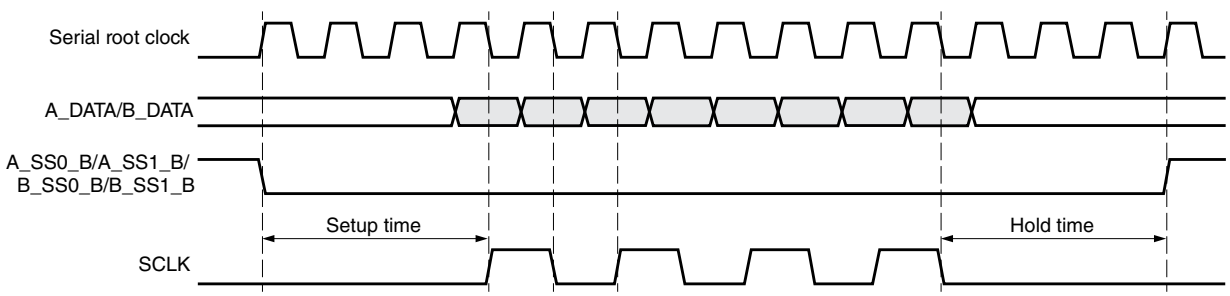


Figure 30-28. Chip select output timing for DDR sequence

For devices such as an FPGA device, a minimum chip select negation time (time between two chip select assertions) may be required. If the value of FLSHxCR1[CSINTERVAL] is nonzero, FlexSPI ensures a delay between chip select valid assertions. The delay time is CSINTERVAL × 256 cycles of serial root clock for an SDR or DDR sequence. If the external device has no limitation, configure the value of this field to zero. Figure 30-29 shows the timing of the chip select interval.

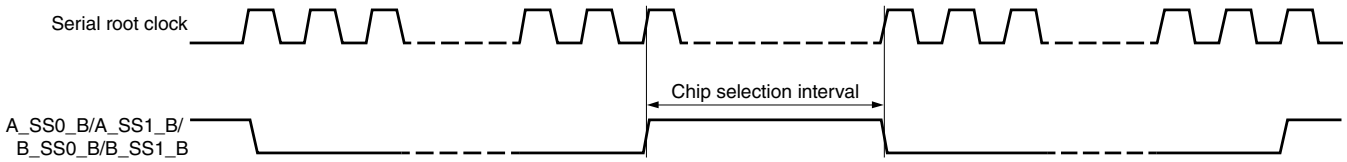


Figure 30-29. Chip select valid interval

30.3.17 FlexSPI input timing

This section describes the input timing of FlexSPI.

30.3.17.1 Receiving block

This section describes the receiving block in FlexSPI.

FlexSPI decodes instruction code to determine SDR mode or DDR mode. It also determines whether both the rising and falling edges are used for flash data sampling. FlexSPI samples the data line for learn instructions and read instructions.

MCR0[RXCLKSRC] selects the clock source from these four sources for the sampling clock:

- Internal dummy read strobe and looped back internally
- Internal dummy read strobe and looped back from DQS pad
- SCK output and looped back from SCK pad
- Flash-memory-provided read strobe

For details about DLL configuration, see DLL configuration for sampling.

Figure 30-30 shows the sampling block in FlexSPI.

FlexSPI uses both rising and falling edges to sample data from flash_A and flash_B. Two muxes with four clock source inputs generate the signals ipp_ind_io_fa, ipp_ind_io_fb. See Receive clock source features. ipp_ind_io_fa_int and ipp_ind_io_fb_int are reversed from ipp_ind_io_fa and ipp_ind_io_fb. Both sampling flip-flops are used when FlexSPI is in DDR mode. Only the falling edge sampling flip-flop is used in SDR mode.

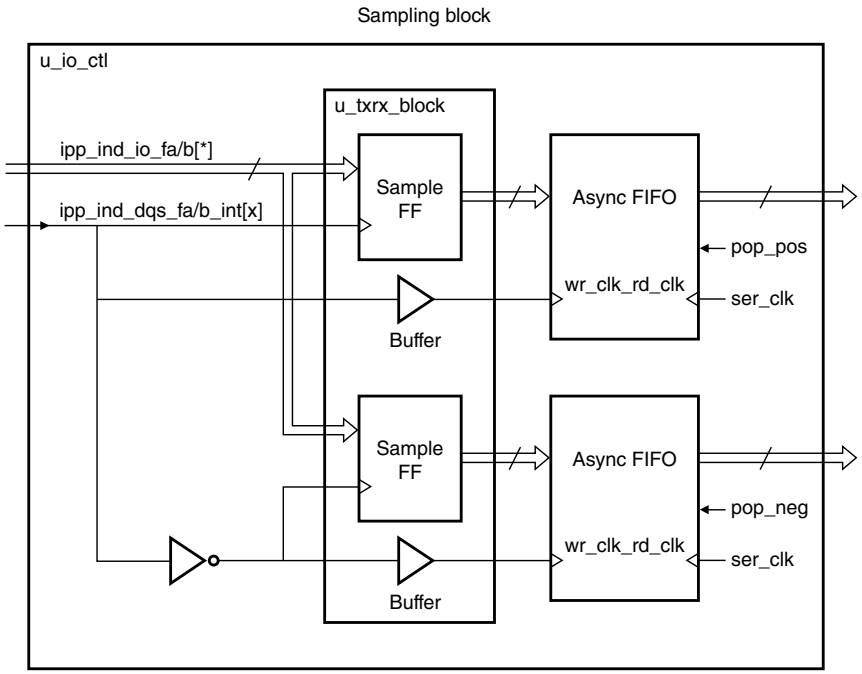


Figure 30-30. Sampling block in FlexSPI

There are 1 clock phases for sampling data. An internal delay chain delays the clock phases.

FlexSPI does not support read operations with a zero-turnaround cycle for direction switching on I/O pins. This read operation often occurs in QuadSPI configuration register reads. In Figure 30-31 and Figure 30-32, SIO[3:0] is an output from FlexSPI in the CMD phase, and is an input to FlexSPI in the DATA phase. No time remains for FlexSPI to switch the direction. This kind of read is not supported. FlexSPI and flash memory drive the I/O pins at the same time, which may cause a problem. To avoid this behavior on I/O pins, using SPI mode or reading with dummy cycles is required.

NOTE

The cycle numbers are not accurate in the examples below. See the device spec for accurate cycle numbers.

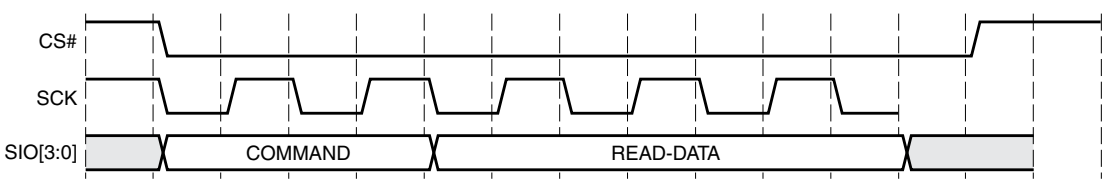


Figure 30-31. Unsupported read behavior

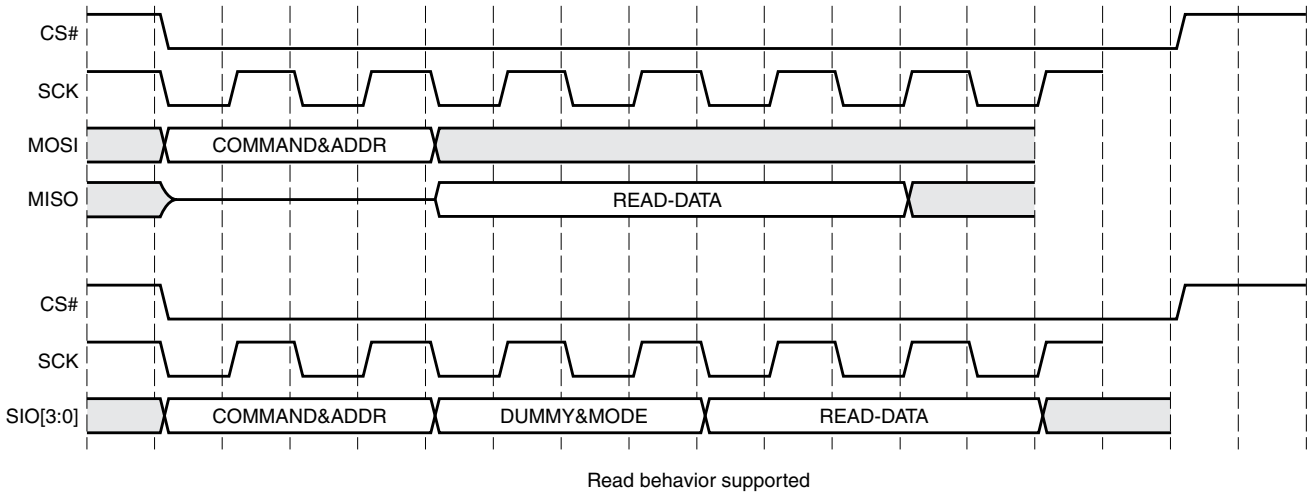


Figure 30-32. Supported read behavior

30.3.17.2 Receive clock source features

FlexSPI uses a read strobe to sample incoming data. There are several options for generating the read clock strobe that MCR0[RXCLKSRC] selects. Using the receive clock source method has implications for the board connections and can affect the maximum frequency that the interface supports. See the device data sheet for details on the maximum frequency supported for each RXCLKSRC option.

Table 30-20. RXCLKSRC options

MCR0[RXCLKSRC]	Description	Board connection	Max Frequency
0	Internal dummy read strobe and internal loopback	DQS pin is not used. The DQS pin can be configured for an alternate signal function.	Lowest
1	Internal dummy read strobe and loopback from DQS pad	FlexSPI uses the DQS pin, and it must be configured for its FlexSPI function. The internally generated read strobe is sent to the DQS pin and is sampled at the pin to match more closely the data pin timings. The DQS pin is typically left floating. External capacitance can be added to adjust timing.	Medium
3	Flash-memory-provided read strobe	DQS pin is connected to the DQS or RWDS signal that the memory provides.	Highest

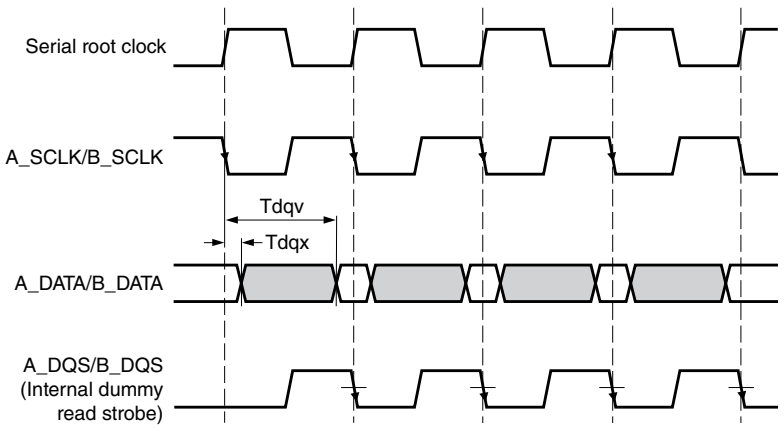
30.3.17.3 Input timing for sampling with dummy read strobe

This section describes the input timing when sampling with the internal dummy read strobe (MCR0[RXCLKSRC] = 0 or 1). The timing for sampling with an internal dummy read strobe loopback is very similar to the timing for loopback from pad. Sampling with a dummy read strobe loopback from the DQS pad can achieve a higher read frequency. It compensates for the delay of the SCLK output path and data pin input path.

The input timing is different for SDR mode and DDR mode.

- **Input timing for sampling with dummy read strobe in SDR mode**

  For SDR read or learn instructions, FlexSPI samples input data pins with the falling edge of the dummy read strobe. Figure 30-33 shows the input timing for sampling with dummy read strobe in SDR mode.



Tdqv is the setup time from SCLK to the next valid data.
Tdqx is the hold time from SCLK to the current valid data.

Figure 30-33. Input timing for sampling with dummy read strobe in SDR mode

- **Input timing for sampling with dummy read strobe in DDR mode**

  For DDR read or learn instructions, FlexSPI samples input data pins on rising and falling edges of the dummy read strobe. Figure 30-34 shows the input timing for sampling with dummy read strobe in DDR mode.

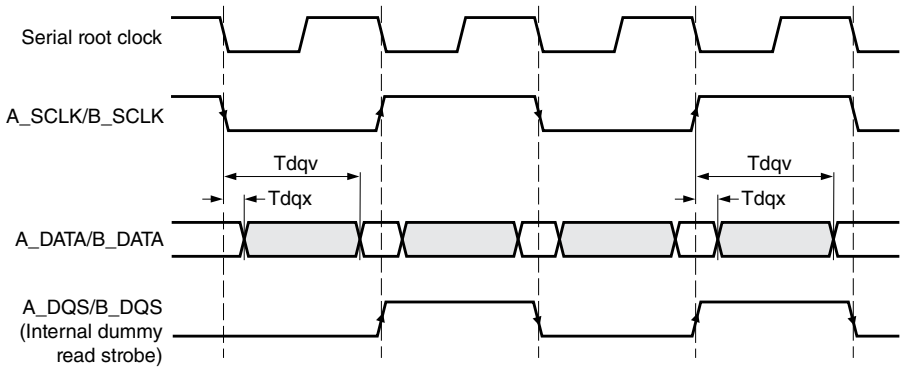


Figure 30-34. Input timing for sampling with dummy read strobe in DDR mode

30.3.17.4 Input timing for sampling with flash-memory-provided read strobe

This section describes the input timing when sampling with flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3). The input timing is different for SDR mode and DDR mode.

NOTE

There are no known devices that provide a read strobe and support SDR mode operation.

- **Flash-memory-provided read strobe with SCLK**

  Certain flash devices provide both read data and read strobes with SCLK. Then the read strobe edge is aligned with the read data change. The FlexSPI controller delays the read strobe for one half cycle of the serial root clock (with DLL), then samples read data with the delayed strobe. See DLL configuration for sampling. Figure 30-35 and Figure 30-36 show the input timing for sampling with flash memory read strobe in SDR mode and DDR mode.

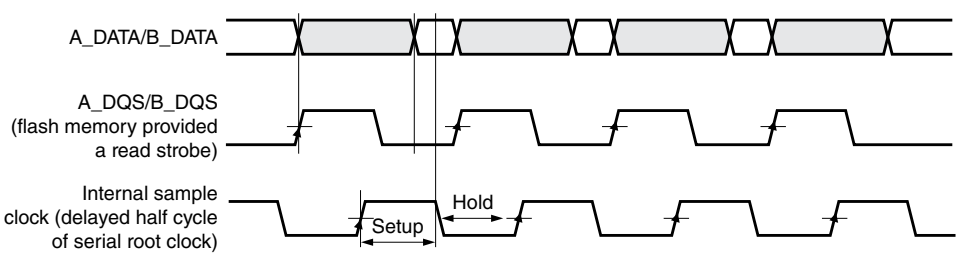


Figure 30-35. Input timing 2 for flash-memory-provided read strobe in SDR mode

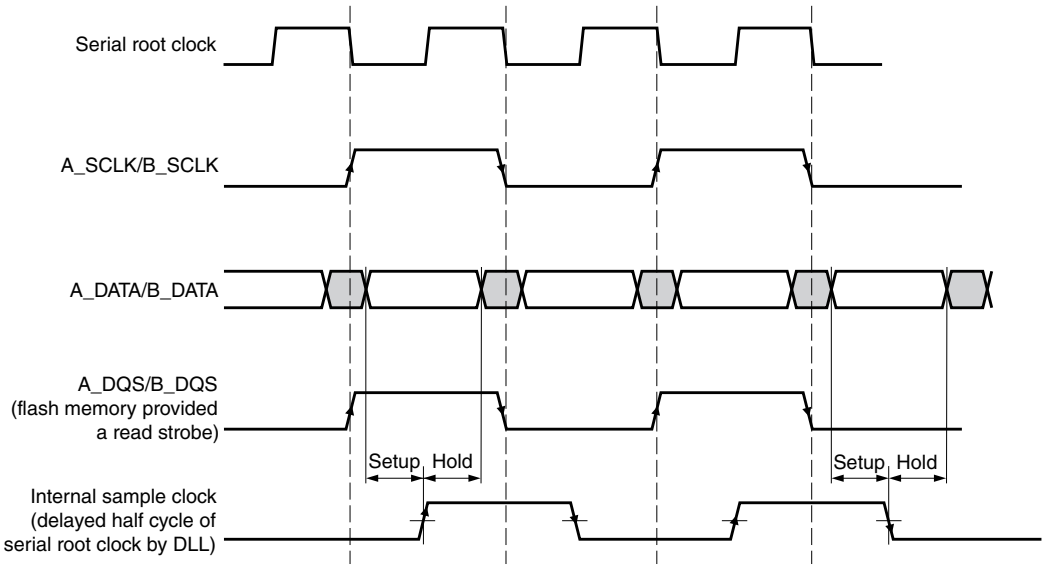


Figure 30-36. Input timing 2 for flash-memory-provided read strobe in DDR mode

30.3.17.5 DLL configuration for sampling

For the four sampling clock sources described previously, the input timing differs depending on the DLLxCR register configuration, according to the sampling clock source mode. The DLL is a delay-line chain that can be set to a fixed number of delay cells. It can also be auto-adjusted to lock on a certain phase delay to the reference clock.

- DLLxCR must be set to 0000_0100h (one fixed delay cell in DLL delay chain) in the following cases:
  - Sampling data with dummy read strobe looped back internally (MCR0[RXCLKSRC] = 0h)
  - Sampling data with dummy read strobe looped back from DQS pad (MCR0[RXCLKSRC] = 1h)
- When data is sampled with a flash-memory-provided read strobe (MCR0[RXCLKSRC] = 3h) and flash memory provides the read strobe with SCLK, DLL must be configured as follows. These settings ensure a lock of the reference clock (serial root clock).
  - DLLxCR[SLVDLYTARGET] is up to flash parameters used.
  - DLLxCR[REFPHASEGAP] = 2h
  - DLLxCR[DLLEN] = 1h
  - DLLxCR[OVRDEN] = 0h
  - Other fields in DLLxCR must be kept at their reset values (all zeroes).

NOTE

If the serial root clock is slower than 100 MHz, DLL cannot lock on a half cycle of serial root clock. The delay cell number is limited in the delay chain. DLL must be configured as follows instead:

- DLLxCR[OVRDEN] = 1
- DLLxCR[OVRDVAL] = N

  Each delay cell in DLL is about 75 ps to 225 ps. The delay of DLL delay chain is (N × Delay_cell_delay). N is set based on the maximum DDR frequency (N ≤ equation). This value is calculated based on the MAX_DDR_FREQ parameter. N = 18. This value is a recommended value. If a failure occurs, the value might require adjustment in a real application.

- Other fields in DLLxCR must be kept at their reset values (all zeroes).

30.3.18 Execute-In-Place (XIP) enhanced mode

FlexSPI always supports XIP, regardless of whether external devices provide XIP enhanced mode. To support XIP, put program codes on an external device, then read or execute these codes directly via AHB read access to serial flash memory space. There is no configuration or status polling needed during AHB read access to the memory of the external device. The AHB receive buffer is fully transparent to software.

Certain devices provide XIP enhanced mode to improve code execution. In this mode, the command code is only sent in the first read instruction. It saves many cycles for command instructions and improves code execution. Devices enter or exit XIP enhanced mode via a special sequence that external devices specify. See the external device data sheet for more details.

Normally, a device enters XIP enhanced mode via this sequence:

1. Enable XIP enhanced mode in external flash device via IP command.
2. Send the first read sequence to external flash device with correct mode bits. Command code is included in this read sequence.
3. Send the subsequent read sequences to external flash device with correct mode bits. Command code is not included in these read sequences. Mode bits must be sent according to the flash specified; otherwise, the flash exits XIP enhanced mode.

The instruction JMP_ON_CS in FlexSPI must be used to support XIP enhanced mode in external devices. This instruction is only allowed in the AHB read command; otherwise, the instruction generates an IPCMDERR or AHBCDMERR interrupt (when the interrupt is enabled). When external devices do not support XIP enhanced mode, JMP_ON_CS must never be used. To support XIP enhanced mode:

- The first instruction in the read sequence must be the command instruction.
- The last valid instruction must be JMP_ON_CS (with operand 1h).

For the first AHB read command triggered, FlexSPI executes the instructions from the instruction pointer 0 in the sequence (which is the command instruction). After this sequence is executed, FlexSPI saves the operand in the JMP_ON_CS instruction as the start pointer for the next command to current device internally. For the next AHB read command triggered, FlexSPI executes from the instruction pointer 1h, bypassing the command instruction.

After XIP enhanced mode is started, it is possible to access flash via other commands like program commands. You must, however, use a special sequence to exit enhanced XIP mode before you can run a different command. Figure 30-37 indicates XIP operation with Flash XIP enhanced mode.

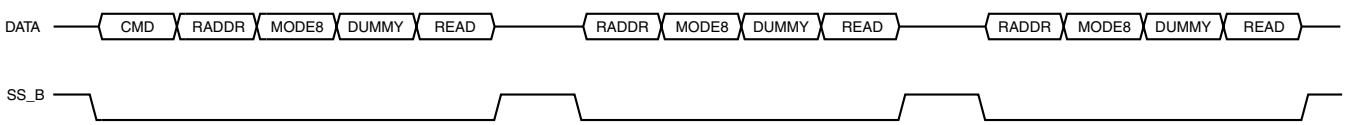


Figure 30-37. XIP Enhanced Mode Operation

30.4 External signals

This section provides the external signal information for the FlexSPI module.

Table 30-21. External Signal List

Signal name	Function	Direction	Description
A_SS0_B	Peripheral Chip Select Flash A1	O	Chip select for the serial flash device A1
A_SS1_B	Peripheral Chip Select Flash A2	O	Chip select for the serial flash device A2
B_SS0_B	Peripheral Chip Select Flash B1	O	Chip select for the serial flash device B1
B_SS1_B	Peripheral Chip Select Flash B2	O	Chip select for the serial flash device B2
A_SCLK	Serial Clock Flash A	O	Serial clock output to the serial flash device A.

Table continues on the next page...

Table 30-21. External Signal List (continued)

Signal name	Function	Direction	Description
			This clock runs at half the frequency of serial clock root in DDR mode, and at the same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence.
B_SCLK	Serial Clock Flash B	O	Serial clock output to the serial flash device B. This clock runs at half the frequency of serial clock root in DDR mode, and at the same frequency as serial clock root in SDR mode. Clock output toggles during the entire flash memory access sequence. **NOTE:** B_SCLK may be used as the differential clock output of A_SCLK by writing 1 to MCR2[SCKBDIFFOPT].
A_DATAn	Serial I/O Flash A	I/O	Data I/O lines to and from the serial flash device A
B_DATAn	Serial I/O Flash B	I/O	Data I/O lines to and from the serial flash device B
A_DQS	Data Strobe Signal Flash A	I/O	Data strobe signal for port A. This signal has three functions: • Driven with read strobe by external device. Some flash devices provide the read strobe signal together with read data. In this case, if the external device drives this pad only when reading flash memory data, this pad may require a **pull-down** resistor. • Driven with latency information by external device. Some devices use this pin to indicate the dummy cycles needed (before program or read data transfer) such as HyperRAM or HyperFlash. • Loopback dummy read strobe. The FlexSPI controller provides internal dummy read strobe for flash memory read data. Higher read frequency can be achieved by looping back this dummy read strobe from the pad. This pin can be floated or have some capacitive load added at the board level to compensate for load on DATA or SCLK pins.
B_DQS	Data Strobe Signal Flash B	I/O	Similar to A_DQS

30.5 Initialization

The FlexSPI controller initialization sequence is:

1. Enable controller clocks (AHB clock, IP bus clock, or serial root clock) at system level.
2. To enter module stop mode, write 1 to MCR0[MDIS].
3. Configure module control registers: MCR0, MCR1, and MCR2. Do not change MCR0[MDIS].
4. When AHB commands are used, configure AHB Bus Control (AHBCR) and AHB Receive Buffer Control (AHBRXBUFxCR0) registers.
5. Configure flash control registers (FLSHxCR0, FLSHxCR1, or FLSHxCR2) according to external device type.
6. Configure DLL control register (DLLxCR) according to sample clock source selection.
7. To exit module stop mode, write 0 to MCR0[MDIS].
8. Configure LUT as needed for AHB command or IP command.
9. Optionally, to reset controller, write 1 to MCR0[SWRESET].

External devices must be configured via IP command normally after initialization. For example, the WRITE STATUS command performs device configuration for most serial NOR flash memory.

30.6 Application information

This section describes applications that FlexSPI supports.

30.6.1 Application on serial NOR flash device

This section provides the example LUT instruction sequences for serial NOR flash device (Cypress Flash S25FS128S).

30.6.1.1 Write enable command

Table 30-22 shows the WRITE ENABLE command sequence.

Table 30-22. WRITE ENABLE command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h	06h	Command name: WREN
1-7	STOP (00h)	0h	00h	

30.6.1.2 Write registers command

Table 30-23 shows the Write Registers command sequence.

Table 30-23. Write Registers command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	01h	Command name: WRR
1	WRITE_SDR	0h (Single)	1h or 2h	Data size is 1 byte or 2 bytes. Byte 0 is write data for Status Register 1; Byte 1 is write data for Configuration Register 1. IPCR1[IDATSZ] can override this value.
2-7	STOP (00h)	0h	00h	

30.6.1.3 Page Program command

Table 30-24 shows Page Program command sequence.

Table 30-24. PAGE PROGRAM command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	02h or 12h	Command name: PP or 4PP PP is 3-byte address mode. 4PP is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

Table 30-25 shows Page Program command sequence (QPI mode).

Table 30-25. PAGE PROGRAM (QPI mode) command (Cypress serial NOR flash)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	02h or 12h	Command name: PP or 4PP

Table continues on the next page...

Table 30-25. PAGE PROGRAM (QPI mode) command (Cypress serial NOR flash) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				PP is 3-byte address mode. 4PP is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	WRITE_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default programming data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

30.6.1.4 Read Status 1 command

Table 30-26 shows READ STATUS 1 command sequence.

Table 30-26. READ STATUS 1 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	05h	Command name: RDSR1
1	READ_SDR	0h (Single)	1h	1 byte for Status register 1
2-7	STOP (00h)	0h	00h	

30.6.1.5 Read command

Table 30-27 shows READ command sequence.

Table 30-27. READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	03h or 13h	Command name: READ or 4READ READ is 3-byte address mode. 4READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading

Table continues on the next page...

Table 30-27. READ command (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				data size. This value is ignored for AHB command.
3-7	STOP (00h)	0h	00h	

30.6.1.6 Fast Read command

Table 30-28 shows Fast Read command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h.

Table 30-28. FAST_READ command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Bh or 0Ch	Command name: FAST_READ or 4FAST_READ FAST_READ is 3-byte address mode. 4FAST_READ is 4-byte address mode.
1	ADDR_SDR	0h (Single)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	DUMMY_SDR	0h (Single)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
3	READ_SDR	0h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
4-7	STOP (00h)	0h	00h	

30.6.1.7 Dual IO Fast Read command

Table 30-29 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Continuous Read mode.

Table 30-29. Dual IO FAST_READ command (Continuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode.

Table continues on the next page...

Table 30-29. Dual IO FAST_READ command (Continuous Read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Axh	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 30-30 shows Dual IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[3:0] = 8h, Noncontinuous Read mode.

Table 30-30. Dual IO FAST_READ command (Noncontinuous Read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	BBh or BCh	Command name: DIOR or 4DIOR DIOR is 3-byte address mode. 4DIOR is 4-byte address mode.
1	ADDR_SDR	1h (Dual)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	1h (Dual)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	1h (Dual)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	1h (Dual)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

30.6.1.8 Quad IO Fast Read command

Table 30-31 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Noncontinuous read mode.

Table 30-31. Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or remain in noncontinuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

Table 30-32 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 30-32. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.

Table continues on the next page...

Table 30-32. Quad IO FAST_READ command (Non-QPI mode, Continuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

Table 30-33 shows Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 30-33. Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EBh or ECh	Command name: QIOR or 4QIOR QIOR is 3-byte address mode. 4QIOR is 4-byte address mode.
1	ADDR_SDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_SDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
5	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
6-7	STOP (00h)	0h	00h	

30.6.1.9 DDR Quad IO Fast Read command

Table 30-34 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Non-continuous read mode.

Table 30-34. DDR Quad IO FAST_READ command (Non-QPI mode, Noncontinuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	Any value other than Axh	Exit continuous read mode or keep in noncontinuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6-7	STOP (00h)	0h	00h	

Table 30-35 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration register bits CR2V[7] = 0, CR2V[6] = 0, CR2V[3:0] = 8h, Non-QPI mode, Continuous read mode.

Table 30-35. DDR Quad IO FAST_READ command (Non-QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.

Table continues on the next page...

Table 30-35. DDR Quad IO FAST_READ command (Non-QPI mode, Continuous read mode) (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

Table 30-36 shows DDR Quad IO FAST_READ command sequence. This table shows Cypress SPI Configuration Register bits CR2V[7] = 0, CR2V[6] = 1, CR2V[3:0] = 8h, QPI mode, Continuous read mode.

Table 30-36. DDR Quad IO FAST_READ command (QPI mode, Continuous read mode)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	2h (Quad)	EDh or EEh	Command name: QIOR_DDR or 4QIOR_DDR QIOR_DDR is 3-byte address mode. 4QIOR_DDR is 4-byte address mode.
1	ADDR_DDR	2h (Quad)	18h or 20h	Address bit number (operand value) should be 24 in 3-byte address mode and 32 in 4-byte address mode.
2	MODE8_DDR	2h (Quad)	A0h	Enter continuous read mode or remain in continuous read mode.
3	DUMMY_DDR	2h (Quad)	08h	Dummy cycle is 8 (in serial root clock) as CR2V[3:0] = 8h.
4	LEARN_DDR	2h (Quad)	1h	DLP is 8 bits.
5	READ_DDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
6	JMP_ON_CS	0h (or Don't care)	01h	CMD instruction is bypassed after the first read access.
7	STOP (00h)	0h	00h	

30.6.2 Application on HyperBus device

This section provides the example LUT instruction sequences for HyperBus device (Cypress RPC flash, HyperRam, or HyperFlash).

30.6.2.1 HyperFlash

This section provides example sequences for HyperFlash devices (Cypress S26KS series).

Table 30-37 shows the read status command sequence.

Table 30-37. Read status command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = 70h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W# = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0070h
	7	CMD_DDR	3h (Octal)	70h	
1 (Read - Addr = xxx, Data = Status register data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	0h (Octal)	4h	4-Byte read
	5-7	STOP (00h)	0h	00h	

Table 30-38 shows the read (memory) command sequence.

Table 30-38. Read (memory) command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Read - Addr = xxx, Data = memory data)	0	CMD_DDR	3h (Octal)	A0h	CA bit 47: (R/W#) = 1h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits

Table continues on the next page...

Table 30-38. Read (memory) command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
	4	READ_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size. This value is ignored for AHB command.
	5-7	STOP (00h)	0h	00h	

Table 30-39 shows the word program command sequence.

Table 30-39. Word program command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1 (Write - Addr = 2AAh, Data = 55h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h

Table continues on the next page...

Table 30-39. Word program command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
(Write - Addr = 555h, Data = A0h)					CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	55h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00A0h
	7	CMD_DDR	3h (Octal)	A0h	
3 (Word Program)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
	2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
	3	WRITE_DDR	3h (Octal)	02h	2-Byte written data
	4-7	STOP (0h)	0h	00h	

Table 30-40 shows Write-to-Buffer and Program-Buffer-to-Flash command sequence.

Table 30-40. Write-to-Buffer and Program-Buffer-to-Flash command

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0 (Write - Addr = 555h, Data = AAh)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 0000AAh (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	AAh	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 05h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	05h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 00AAh
	7	CMD_DDR	3h (Octal)	AAh	
1 (Write - Addr = 2AAh, Data = 55h)	0	CMD_DDR	0h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h

Table continues on the next page...

Table 30-40. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
					CA bit 44-40: All reserved = 0h
	1	CMD_DDR	3h (Octal)	00h	Row Address: 000055h (24 bit)
	2	CMD_DDR	3h (Octal)	00h	
	3	CMD_DDR	3h (Octal)	55h	
	4	CMD_DDR	3h (Octal)	00h	Column Address: 02h (13 zero bits + 3 valid bits)
	5	CMD_DDR	3h (Octal)	02h	
	6	CMD_DDR	3h (Octal)	00h	Write Data: 0055h
	7	CMD_DDR	3h (Octal)	55h	
2 (Write - Addr = SA, Data = 25h)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 0025h
	4	CMD_DDR	3h (Octal)	25h	
2 (Write - Addr = SA, Data = WC)	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	WC	Write Data: WC
	4	CMD_DDR	3h (Octal)		WC is word count.
3 - N (Write - Addr = WBL, Data = PD) N is the word count + 2	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: WBL (24 bit) WBL is write buffer location. Write WBL to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	WRITE_DDR	3h (Octal)	02h	2-Byte write data

Table continues on the next page...

Table 30-40. Write-to-Buffer and Program-Buffer-to-Flash command (continued)

Seq num	Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
	4-7	STOP (0h)	0h	00h	
N + 1 (Write - Addr = SA, Data = 29) Program Buffer to Flash	0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
	1	RADDR_DDR	3h (Octal)	18h	Row Address: SA (24 bit) SA is sector address. Write SA to IPCR0[SFAR].
	2	CADDR_DDR	3h (Octal)	10h	Column Address: 13 zero bits + 3 valid bits
	3	CMD_DDR	3h (Octal)	00h	Write Data: 29h
	4	CMD_DDR	3h (Octal)	29h	
	5-7	STOP (0h)	0h	00h	

30.6.2.2 HyperRAM

This section provides example sequences for HyperRAM (Cypress S27KL series).

The Read (memory) command sequence is same as HyperFlash. Table 30-41 shows the Write (memory) command sequence.

Table 30-41. Write (memory) command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_DDR	3h (Octal)	20h	CA bit 47: (R/W#) = 0h CA bit 46: (Target) = 0h CA bit 45: (Burst Type) = 1h CA bit 44-40: All reserved = 0h
1	RADDR_DDR	3h (Octal)	18h	Row Address: 24 bits
2	CADDR_DDR	3h (Octal)	10h	Column Address: (13 zero bits + 3 valid bits)
3	DUMMY_RWDS_DDR	3h (Octal)	0Bh	When latency count = 11
4	WRITE_DDR	3h (Octal)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing data size. This value is ignored for AHB command.
5-7	STOP (00h)	0h	00h	

30.6.3 Application on Serial NAND flash device

This section provides example LUT instruction sequences for serial NAND flash device (Micron Flash MT29 series). The operation of serial NAND flash is similar to serial NOR flash.

The read operation sequence is:

- Page read. Transfer the data from the NAND flash array to the cache register.
- Get feature to read the status.
- Random data read

Table 30-42 shows the page read command sequence.

Table 30-42. Page Read command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	13h	Command code: 13h
1	RADDR_SDR	1h (Single)	18h	Row address: 24 bit
2-7	STOP (0h)	0h	00h	

Table 30-43 shows Get Feature command sequence.

Table 30-43. Get Feature command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	0h (Single)	0Fh	Command code: 0Fh
1	CMD_SDR	1h (Single)	C0h	Status register address (C0h)
2	READ_SDR	0x1 (Single)	02h	2 bytes read data
3-7	STOP (0h)	0h	00h	

Table 30-44 shows Random Data Read command sequence.

Table 30-44. Read from Cache x4 command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	6Bh	Command code: 6Bh
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection is zero, software should write 0h to mode bits.

Table continues on the next page...

Table 30-44. Read from Cache x4 command (continued)

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
				Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	DUMMY_SDR	2h (Quad)	08h	Dummy cycle number: 8 (serial root clock)
4	READ_SDR	2h (Quad)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default reading data size.
5-7	STOP (0x0)	0x0	0x00	

The program operation sequence is:

- Write enable
- Program Load. Transfer the write data to the cache register.
- Program Execute
- Get Feature to read the status.

Table 30-45 shows Program Load command sequence.

Table 30-45. Program Load command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	02h	Command code: 02h
1	MODE4_SDR	1h (Single)	0h or 1h	If plane selection is one, software should decode the flash address and write 1h to mode bits. If plane selection is zero, software should write 0h to mode bits. Plane selection bit is 18th bit of flash address. If NAND flash size is less than 4 Gbit, plane selection is always zero.
2	CADDR_SDR	1h (Single)	0Ch	Column address: 12 bit
3	WRITE_SDR	1h (Single)	Any nonzero value	If IPCR1[IDATSZ] is zero, this operand value can be used as default writing data size.
4-7	STOP (0h)	0h	00h	

Table 30-46 shows Program Execute command sequence.

Table 30-46. Program Execute command

Instruction number	Instruction opcode[5:0]	Instruction num_pads[1:0]	Instruction operand[7:0]	Comment
0	CMD_SDR	1h (Single)	10h	Command code: 10h
1	RADDR_SDR	1h (Single)	18h	Row Address: 24 bit
2-7	STOP (0h)	0h	00h	

30.6.4 Application on FPGA device

An FPGA device must be accessed via AHB command. All AHB accesses to FPGA are transparent to the software driver (no software intervention). An FPGA device may have some special requirements.

Table 30-47. Special requirements for FPGA devices

Condition	How to manage the condition
Device type may be different on A1, A2, B1, or B2.	Write 0 to MCR2[SAMEDEVICEEN] and configure the FLSHxCR0 and FLSHxCR1 registers separately for up to four external devices.
Device requires different wait cycle for programming.	The AHB write wait cycle number can be set separately for these four external devices (via FLSHxCR2[AWRWAIT]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle). FlexSPI holds the AHB bus ready for this wait time; AHB bus performance may decrease when this wait time is very long.
Device requires different wait cycle for reading.	The AHB read sequence index and sequence number can be set separately for these four external devices (via FLSHxCR2[ARDSEQID] and FLSHxCR2[ARDSEQNUM]). Software can configure the sequences in LUT with different dummy instructions (operand determines dummy cycle).
Device may be sensitive to read instruction clock cycle number.	If its internal memory is implemented similar to a FIFO, the device is sensitive to the read instruction clock cycle number. Software can send the data size information to the external device via DATSZ instruction. The FPGA device decodes the data size information and determines how many data bytes must be popped.
Device may require interval time between chip select valid.	This interval can be managed via FLSHxCR1[CSINTERVAL].
Device may use SCLK as reference clock for its internal PLL.	SCLK must be free-running and the clock frequency must be stable. To achieve this configuration, write 1 to MCR0[SCKFREERUNEN] and only use SDR sequences.

30.6.5 Overview of error categories, flags, and triggered sources

Table 30-48 provides an overview of the categories, flags, and triggered sources of errors in FlexSPI.

Table 30-48. Error category, triggered sources, and flags

Error Category	Triggered Sources	Description	Error Flags
Command grant error	AHB write command	Command grant timeout	INTR[AHBCMDGE] is set. AHB bus error response
	AHB read command		INTR[AHBCMDGE] is set. AHB bus error response
	IP command		INTR[IPCMDGE] is set.
Command check error	AHB write command	• AHB write command with JMP_ON_CS instruction used in the sequence • Unknown instruction opcode in the sequence. • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence.	INTR[AHBCMDERR] is set. Command is not executed when an error is detected in command check
	AHB read command	• Unknown instruction opcode in the sequence. • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence.	INTR[AHBCMDERR] is set. Command is not executed when an error is detected in command check
	IP command	• IP command with JMP_ON_CS instruction used in the sequence. • Unknown instruction opcode in the sequence • Instruction DUMMY_SDR or DUMMY_RWDS_SDR used in DDR sequence. • Instruction DUMMY_DDR or DUMMY_RWDS_DDR used in SDR sequence. • Flash boundary across	INTR[IPCMDERR] is set. Command is not executed when an error is detected in command check.
Command execution error	AHB write command	Command timeout during execution	INTR[AHBCMDERR] is set. INTR[SEQTIMEOUT] is set. An AHB bus error response occurs, except in following cases: • Flush triggers AHB write command (INCR

Table continues on the next page...

Table 30-48. Error category, triggered sources, and flags (continued)

Error Category	Triggered Sources	Description	Error Flags
			burst ended with AHB_TX_BUF not empty). • AHB bufferable write access and bufferable enabled (AHBCR[BUFFERABLEEN] = 1).
	AHB read command		INTR[AHBCMDERR] is set. INTR[SEQTIMEOUT] is set. AHB bus error response
	IP command		INTR[IPCMDERR] is set. INTR[SEQTIMEOUT] is set.
AHB bus timeout	AHB write command	AHB bus timeout (no bus ready return)	AHB bus error response
	AHB read command		

30.7 Memory map and register definition

This section includes the FlexSPI module memory map and detailed descriptions of all registers.

30.7.1 Register access

All registers can be accessed with 8-bit, 16-bit, and 32-bit width operations. Never change the values of reserved fields in control registers. Changing the values of reserved fields may impact the normal functioning of the controller.

NOTE

When FlexSPI can access sensitive memory contents, the FlexSPI controller should protect those contents using resource isolation such as XRDC or TrustZone. Ensure that only trusted software is allowed to access the FlexSPI controller register interface.

30.7.2 FlexSPI register descriptions

30.7.2.1 FlexSPI memory map

FLEXSPI1 base address: 400C_C000h

FLEXSPI2 base address: 400D_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Control 0 (MCR0)	32	Table 30-48	FFFF_80C2h
4h	Module Control 1 (MCR1)	32	RW	FFFF_FFFFh
8h	Module Control 2 (MCR2)	32	Table 30-48	2000_81F7h
Ch	AHB Bus Control (AHBCR)	32	Table 30-48	0000_0018h
10h	Interrupt Enable (INTEN)	32	Table 30-48	0000_0000h
14h	Interrupt (INTR)	32	Table 30-48	0000_0000h
18h	LUT Key (LUTKEY)	32	RW	5AF0_5AF0h
1Ch	LUT Control (LUTCR)	32	Table 30-48	0000_0002h
20h	AHB Receive Buffer 0 Control 0 (AHBRXBUF0CR0)	32	Table 30-48	8000_0040h
24h	AHB Receive Buffer 1 Control 0 (AHBRXBUF1CR0)	32	Table 30-48	8001_0040h
28h	AHB Receive Buffer 2 Control 0 (AHBRXBUF2CR0)	32	Table 30-48	8002_0040h
2Ch	AHB Receive Buffer 3 Control 0 (AHBRXBUF3CR0)	32	Table 30-48	8003_0040h
30h	AHB Receive Buffer 4 Control 0 (AHBRXBUF4CR0)	32	Table 30-48	8004_0040h
34h	AHB Receive Buffer 5 Control 0 (AHBRXBUF5CR0)	32	Table 30-48	8005_0040h
38h	AHB Receive Buffer 6 Control 0 (AHBRXBUF6CR0)	32	Table 30-48	8006_0040h
3Ch	AHB Receive Buffer 7 Control 0 (AHBRXBUF7CR0)	32	Table 30-48	8007_0040h
60h	Flash Control 0 (FLSHA1CR0)	32	Table 30-48	0001_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
64h	Flash Control 0 (FLSHA2CR0)	32	Table 30-48	0001_0000h
68h	Flash Control 0 (FLSHB1CR0)	32	Table 30-48	0001_0000h
6Ch	Flash Control 0 (FLSHB2CR0)	32	Table 30-48	0001_0000h
70h	Flash Control 1 (FLSHA1CR1)	32	RW	0000_0063h
74h	Flash Control 1 (FLSHA2CR1)	32	RW	0000_0063h
78h	Flash Control 1 (FLSHB1CR1)	32	RW	0000_0063h
7Ch	Flash Control 1 (FLSHB2CR1)	32	RW	0000_0063h
80h	Flash Control 2 (FLSHA1CR2)	32	Table 30-48	0000_0000h
84h	Flash Control 2 (FLSHA2CR2)	32	Table 30-48	0000_0000h
88h	Flash Control 2 (FLSHB1CR2)	32	Table 30-48	0000_0000h
8Ch	Flash Control 2 (FLSHB2CR2)	32	Table 30-48	0000_0000h
94h	Flash Control 4 (FLSHCR4)	32	Table 30-48	0000_00C3h
A0h	IP Control 0 (IPCR0)	32	RW	0000_0000h
A4h	IP Control 1 (IPCR1)	32	Table 30-48	0000_0000h
B0h	IP Command (IPCMD)	32	Table 30-48	0000_0000h
B8h	IP Receive FIFO Control (IPRXFCR)	32	Table 30-48	0000_0000h
BCh	IP Transmit FIFO Control (IPTXFCR)	32	Table 30-48	0000_0000h
C0h	DLL Control 0 (DLLACR)	32	Table 30-48	0000_0100h
C4h	DLL Control 0 (DLLBCR)	32	Table 30-48	0000_0100h
D0h	Misc Control 4 (MISCCR4)	32	RO	0000_0000h
D4h	Miscellaneous Control 5 (MISCCR5)	32	RO	0000_0000h
D8h	Miscellaneous Control 6 (MISCCR6)	32	RO	0000_0000h
DCh	Miscellaneous Control 7 (MISCCR7)	32	RO	0000_0000h
E0h	Status 0 (STS0)	32	RO	0000_0002h
E4h	Status 1 (STS1)	32	RO	0000_0000h
E8h	Status 2 (STS2)	32	RO	0100_0100h
ECh	AHB Suspend Status (AHBSPNDSTS)	32	RO	0000_0000h
F0h	IP Receive FIFO Status (IPRXFSTS)	32	RO	0000_0000h
F4h	IP Transmit FIFO Status (IPTXFSTS)	32	RO	0000_0000h
100h - 17Ch	IP Receive FIFO Data a (RFDR0 - RFDR31)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
180h - 1FCh	IP TX FIFO Data a (TFDR0 - TFDR31)	32	WO	0000_0000h
200h - 2FCh	Lookup Table a (LUT0 - LUT63)	32	RW	Table 30-48
400h	AHB Controller ID 0 Control (HMSTR0CR)	32	RW	0040_FFCFh
404h	AHB Controller ID 1 Control (HMSTR1CR)	32	RW	0000_FFCFh
408h	AHB Controller ID 2 Control (HMSTR2CR)	32	RW	0004_F00Fh
40Ch	AHB Controller ID 3 Control (HMSTR3CR)	32	RW	0022_807Fh
410h	AHB Controller ID 4 Control (HMSTR4CR)	32	RW	0032_F87Fh
414h	AHB Controller ID 5 Control (HMSTR5CR)	32	RW	0011_F87Fh
418h	AHB Controller ID 6 Control (HMSTR6CR)	32	RW	0000_0000h
41Ch	AHB Controller ID 7 Control (HMSTR7CR)	32	RW	0000_0000h
420h	HADDR REMAP Start Address (HADDRSTART)	32	Table 30-48	0000_0000h
424h	HADDR REMAP END ADDR (HADDREND)	32	Table 30-48	0000_0000h
428h	HADDR Remap Offset (HADDROFFSET)	32	Table 30-48	0000_0000h

30.7.2.2 Module Control 0 (MCR0)

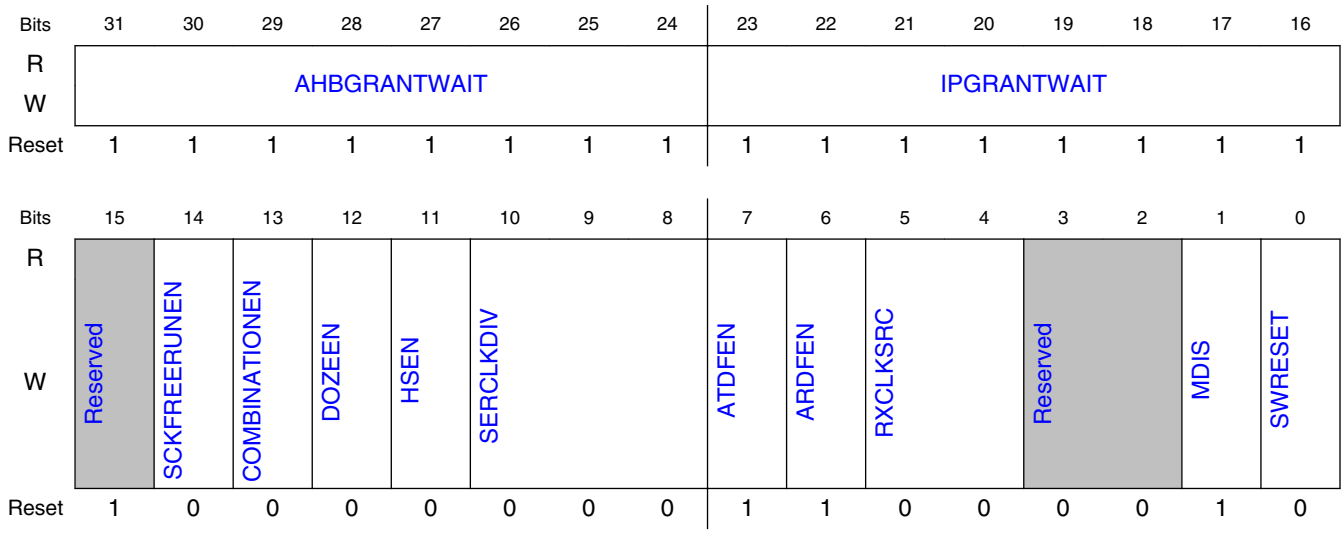
30.7.2.2.1 Offset

Register	Offset
MCR0	0h

30.7.2.2.2 Function

Controls basic functions of FlexSPI module

30.7.2.2.3 Diagram



30.7.2.2.4 Fields

Field	Function
31-24 AHBGRANTWAIT	Timeouts Wait Cycle for AHB command Grant Sets duration of timeout wait cycle for AHB commands. If the arbitrator does not grant the AHB-triggered command, it times out after (AHBGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is IP-triggered and the read or write data size is too large, this grant timeout may occur. When an AHB command grant timeout occurs, an INTR[AHBCMDGE] interrupt is generated. When INTEN[AHBCMDGEEN] = 1, the arbitrator ignores AHB commands. **NOTE:** This field is for debugging only. Do not change the value from its default.
23-16 IPGRANTWAIT	Timeout Wait Cycle for IP Command Grant Sets duration of timeout wait cycle for IP commands. If the arbitrator does not grant the IP-triggered command, it times out after (IPGRANTTIMEOUT × 1024) AHB clock cycles. When the pending command sequence is AHB-triggered and the read or write data size is too large, this grant timeout may occur. When IP command grant timeout occurs, an INTR[IPCMDGE] interrupt is generated. When INTEN[IPCMDGEEN] = 1, the arbitrator ignores IP commands. **NOTE:** This field is for debugging only. Do not change the value from its default.
15 —	Reserved
14 SCKFREERUNEN	SCLK Free-running Enable Enables free-running SCLK output. For FPGA applications, the external device may use SCLK as a reference clock to its internal PLL. 0b - Disable 1b - Enable
13 COMBINATIONEN	Combination Mode Enable Enables combination mode, which supports flash Octal mode access. When Port A and Port B data are four bits wide, this mode combines Port A and B Data pins (A_DATA[3:0] and B_DATA[3:0]). When Port A and Port B data are eight bits wide, combination mode is not supported; write 0 to this field in this case.

Table continues on the next page...

Field	Function
	0b - Disable 1b - Enable
12 DOZEEN	Doze Mode Enable Enables Doze mode. When enabled, AHB clock and serial clock are gated off when there is a Doze-mode request from the system. 0b - Disable 1b - Enable
11 HSEN	Half Speed Serial Flash Memory Access Enable Enables the clock divider to provide a half-speed clock to external serial flash devices (A_SCLK/B_SCLK) for all commands in SDR and DDR mode. Write 1 to MCR0[MDIS] before changing the value of this field. Failure to do so may cause issues in the internal logic or state machine. 0b - Disable 1b - Enable
10-8 SERCLKDIV	Serial Root Clock Divider Sets divider value for serial root clock. The serial root clock can be divided inside FlexSPI . See Clocking. **NOTE:** Do not modify this field while FlexSPI is active (MCR0[MDIS] = 0). 000b - Divided by 1 001b - Divided by 2 010b - Divided by 3 011b - Divided by 4 100b - Divided by 5 101b - Divided by 6 110b - Divided by 7 111b - Divided by 8
7 ATDFEN	AHB Write Access to IP Transmit FIFO Enable Enables AHB write access to IP transmit FIFO. See Writing data to IP transmit FIFO. 0b - AHB write access disabled. IP bus writes to IP transmit FIFO. AHB bus write access to IP transmit FIFO memory space produces bus error. 1b - AHB write access enabled. AHB bus writes to IP transmit FIFO. IP Bus write access to IP transmit FIFO memory space is ignored and causes no bus error.
6 ARDFEN	AHB Read Access to IP Receive FIFO Enable Enables AHB read access to IP receive FIFO. See Reading data from IP receive FIFO. 0b - AHB read access disabled. IP bus reads IP receive FIFO. AHB Bus read access to IP receive FIFO memory space produces bus error. 1b - AHB read access enabled. AHB bus reads IP receive FIFO. IP Bus read access to IP receive FIFO memory space returns data zero and causes no bus error.
5-4 RXCLKSRC	Sample Clock Source for Flash Reading Selects sampling clock source for flash memory reading. See Receive clock source features. 00b - Dummy Read strobe that FlexSPI generates, looped back internally 01b - Dummy Read strobe that FlexSPI generates, looped back from DQS pad 10b - Reserved 11b - Flash-memory-provided read strobe and input from DQS pad
3-2 —	Reserved
1 MDIS	Module Disable Disables FlexSPI module. When the module is disabled, AHB and serial clock are gated off internally to save power. Only register access is allowed, except for the LUT, IP receive FIFO, and IP transmit FIFO. 0b - No impact

Table continues on the next page...

Field	Function
	1b - Module disable
0 SWRESET	Software Reset Resets the internal FlexSPI state machine and aborts any transactions in progress. Hardware automatically writes 0 to this field after software reset is done. Configuration registers are not reset. 0b - No impact 1b - Software reset

30.7.2.3 Module Control 1 (MCR1)

30.7.2.3.1 Offset

Register	Offset
MCR1	4h

30.7.2.3.2 Function

Controls basic functions of FlexSPI module

30.7.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SEQWAIT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AHBBUSWAIT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

30.7.2.3.4 Fields

Field	Function
31-16 SEQWAIT	Command Sequence Wait Sets wait time for command sequence. Command sequence execution times out and aborts after (SEQWAIT × 1024) serial root clock cycles. When this timeout occurs, if the interrupt is enabled

Table continues on the next page...

Field	Function
	(INTEN[SEQTIMEOUTEN] = 1), an INTR[SEQTIMEOUT] interrupt is generated. Also, the arbitrator ignores AHB commands. **NOTE:** You cannot write 0 to this field.
15-0 AHBBUSWAIT	AHB Bus Wait Sets wait time for AHB bus. When data is not received from or transmitted to serial flash memory space after (AHBBUSWAIT × 1024) AHB clock cycles, the read or write access times out. When this timeout occurs, the AHB bus receives an error response, and an interrupt is generated (INTEN[AHBBUSERROREN]). When INTR[AHBBUSERROREN] = 1, the arbitrator ignores AHB commands. **NOTE:** You cannot write 0 to this field.

30.7.2.4 Module Control 2 (MCR2)

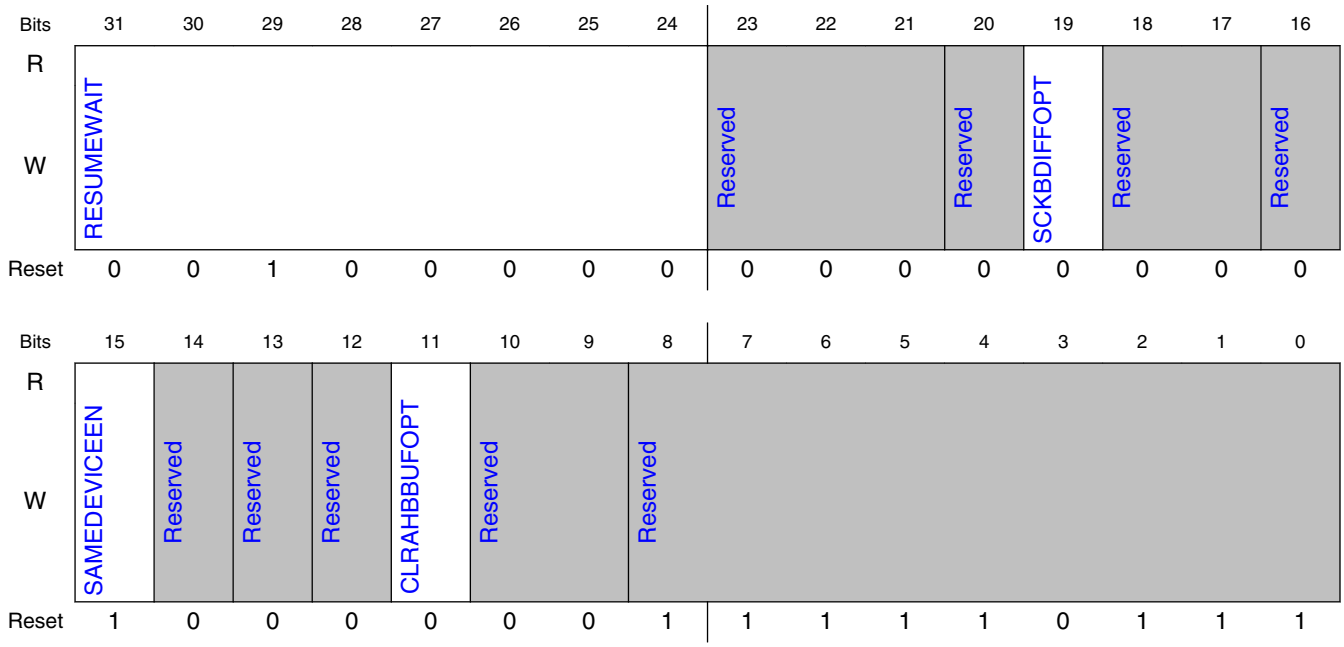
30.7.2.4.1 Offset

Register	Offset
MCR2	8h

30.7.2.4.2 Function

Controls basic functions of FlexSPI module.

30.7.2.4.3 Diagram



30.7.2.4.4 Fields

Field	Function
31-24 RESUMEWAIT	Resume Wait Duration Determines duration (in AHB clock cycles) to remain in idle state before suspended command sequence is resumed. See Command abort and suspend.
23-21 —	Reserved
20 —	Reserved
19 SCKBDIFFOPT	SCLK Port B Differential Output Controls whether to use B_SCLK pad as the Port B SCLK output. Alternatively, B_SCLK pad can be used as A_SCLK differential clock output (inverted clock to A_SCLK). Before changing the value of this field, write 1 to MCR0[MDIS]. After changing the value of this field, write 1 to MCR0[SWRESET]. 0b - Use B_SCLK pad as port B SCLK clock output. Port B flash memory access is available. 1b - Use B_SCLK pad as port A SCLK inverted clock output (Differential clock to A_SCLK). Port B flash memory access is not available.
18-17 —	Reserved
16 —	Reserved
15	Same Device Enable

Table continues on the next page...

Field	Function
SAMEDEVICEEN	Sets all external devices (A1, A2, B1, and B2) to be the same devices (in type and in size). 0b - In Individual mode, FLSHA1CRx and FLSHA2CRx, FLSHB1CRx and FLSHB2CRx settings are applied to Flash A1, A2, B1, B2 separately. In Parallel mode, FLSHA1CRx register setting is applied to Flash A1 and B1, FLSHA2CRx register setting is applied to Flash A2 and B2. FLSHB1CRx and FLSHB2CRx register settings are ignored. 1b - FLSHA1CR0, FLSHA1CR1, and FLSHA1CR2 register settings are applied to Flash A1, A2, B1, B2. FLSHA2CRx, FLSHB1CRx, and FLSHB2CRx settings are ignored.
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 CLRAHBBUFOPT	Clear AHB Buffer Determines whether AHB receive and transmit buffers are cleared automatically when FlexSPI returns Stop mode ACK. If an AHB receive buffer or transmit buffer will be powered off in Stop mode, software should write 1 to this field. Otherwise, an AHB read access after exiting Stop mode may hit either buffer, but their data entries are invalid. 0b - Not cleared automatically 1b - Cleared automatically
10-9 —	Reserved
8-0 —	Reserved

30.7.2.5 AHB Bus Control (AHBCR)

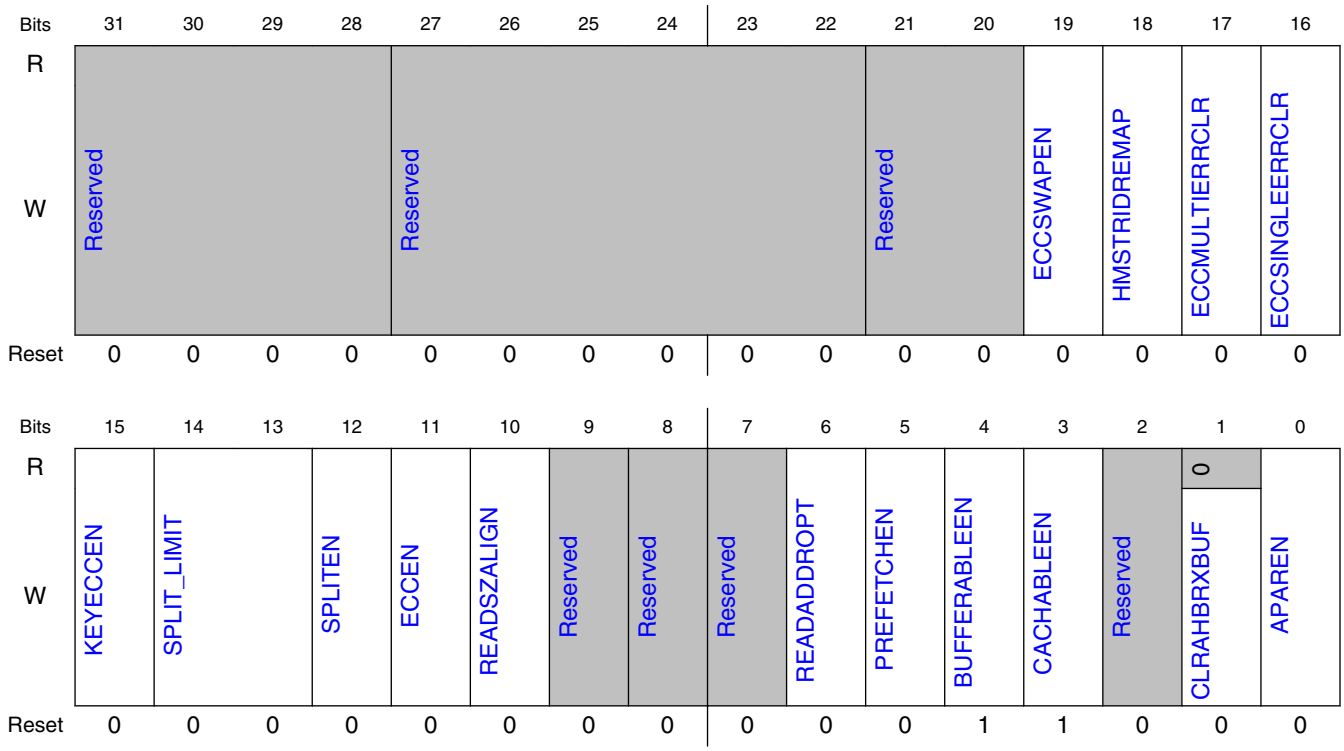
30.7.2.5.1 Offset

Register	Offset
AHBCR	Ch

30.7.2.5.2 Function

Controls AHB bus interface functions.

30.7.2.5.3 Diagram



30.7.2.5.4 Fields

Field	Function
31-28 —	Reserved
27-22 —	Reserved
21-20 —	Reserved
19 ECCSWAPEN	ECC Read Data Swap Enable Enables ECC data swapping when read data is sent to an ECC check. If write data swapping is enabled, read data swapping must also be enabled. Otherwise, ECC function fails. For swap methodology, see AHB read ECC support. 0b - Disable 1b - Enable
18 HMSTRIDREMAP	AHB Controller ID Remapping Enable Enables AHB controller ID remapping. See AHB controller ID remapping. 0b - Disable 1b - Enable
17	AHB ECC Multibit Error Clear

Table continues on the next page...

Field	Function
ECCMULTIERRCLR	Clears the valid bit in ECC multibit error information registers. Do not toggle this field during interrupt service; otherwise, ECC multibit error information is not updated. 0b - No function 1b - Clear
16 ECCSINGLEERRCLR	AHB ECC Single-Bit Error Clear Clears the valid bit in ECC single-bit error information registers. Do not toggle this field during interrupt service; otherwise, ECC single-bit error information is not updated. 0b - No function 1b - Clear
15 KEYECCEN	OTFAD Key Blob ECC Enable Enables ECC checking on OTFAD key blob. 0b - Disable 1b - Enable
14-13 SPLIT_LIMIT	AHB Split Limit Size Configures the AHB Split Size when AHBCR[SPLITEN] = 1. See AHB transaction split. 00b - 8 bytes 01b - 16 bytes 10b - 32 bytes 11b - 64 bytes
12 SPLITEN	AHB Transaction Split Enable Enables the AHB transaction split feature. When enabled, all AHB read and write bursts are split into smaller transactions on the flash SPI bus side. This function is used for RAM devices with a self-refresh limit. See AHB transaction split. 0b - Disable 1b - Enable
11 ECCEN	AHB Read ECC Enable Enables ECC checks for AHB read operations. When enabled, AHB read operations perform ECC checking. All ECC-related logic and registers are activated. Prefetching is enabled by default. Address fetches are eight-bytes aligned. The prefetching resume function is disabled. 0b - Disable 1b - Enable
10 READSZALIGN	AHB Read Size Alignment Configures how AHB read size is determined. See Table 30-18. **NOTE:** When 1, data prefetching is disabled regardless of other field settings such as PREFETCH_EN and OTFAD_EN. Read size is rounded up to the nearest eight-byte multiple. For example, if current AHB read size is 12 bytes, it is aligned to 16 bytes. When the fetch size must be eight-byte aligned but no speculative fetching is wanted, write 1 to this field. 0b - Register settings such as PREFETCH_EN and OTFAD_EN determine AHB read size. 1b - AHB read size to up size to 8 bytes aligned, no prefetching
9 —	Reserved
8 —	Reserved
7 —	Reserved
6	AHB Read Address Option

Table continues on the next page...

Field	Function
READADDROPT	Removes AHB burst start address alignment limitation. When the FlexSPI controller is used for FPGA application, this field may be required for FlexSPI to fetch the exact byte number as an AHB burst. In this case, FPGA device should be non-word-addressable and this field must be 0. **NOTE:** When OTFAD is enabled, FlexSPI functions as it does when READADDROPT = 1 regardless the value of this field. 0b - AHB read burst start address alignment is limited when flash memory is accessed in parallel mode or flash is word-addressable. 1b - AHB read burst start address alignment is not limited. FlexSPI fetches more data than the AHB burst requires for address alignment.
5 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch. When enabled, FlexSPI fetches more flash read data than the current AHB burst requires. This action reduces the read latency for next AHB read access. **NOTE:** When OTFAD is enabled, the AHB prefetch function is enabled regardless of the value of this field. **NOTE:** AHB read prefetch is enabled only when both this field and AHBRXBUFnCR0[PREFETCHEN] are 1. 0b - Disable 1b - Enable
4 BUFFERABLEEN	Bufferable Write Access Enable Enables AHB bus bufferable write access. This field affects the last beat of an AHB write access. See AHB write access to flash memory. 0b - Disabled. For all AHB write accesses (bufferable or nonbufferable), FlexSPI returns AHB Bus Ready after transmitting all data and finishing command. 1b - Enabled. For AHB bufferable write access, FlexSPI returns AHB Bus Ready when the arbitrator grants the AHB command. FlexSPI does not wait for the AHB command to finish.
3 CACHABLEEN	Cacheable Read Access Enable Enables AHB bus cacheable read access. 0b - Disabled. When an AHB bus cacheable read access occurs, FlexSPI does not check whether it hit the AHB transmit buffer. 1b - Enabled. When an AHB bus cacheable read access occurs, FlexSPI first checks whether the access hit the AHB transmit buffer.
2 —	Reserved
1 CLRAHBRXBUF	Clear AHB Receive Buffer Clears status and pointers of AHB receive buffer. When the clear operation completes, this field clears automatically, so it always reads as 0. 0b - No impact. 1b - Enable clear operation.
0 APAREN	AHB Parallel Mode Enable Enables Parallel mode for AHB-triggered read and write commands. 0b - Flash is accessed in Individual mode. 1b - Flash is accessed in Parallel mode.

30.7.2.6 Interrupt Enable (INTEN)

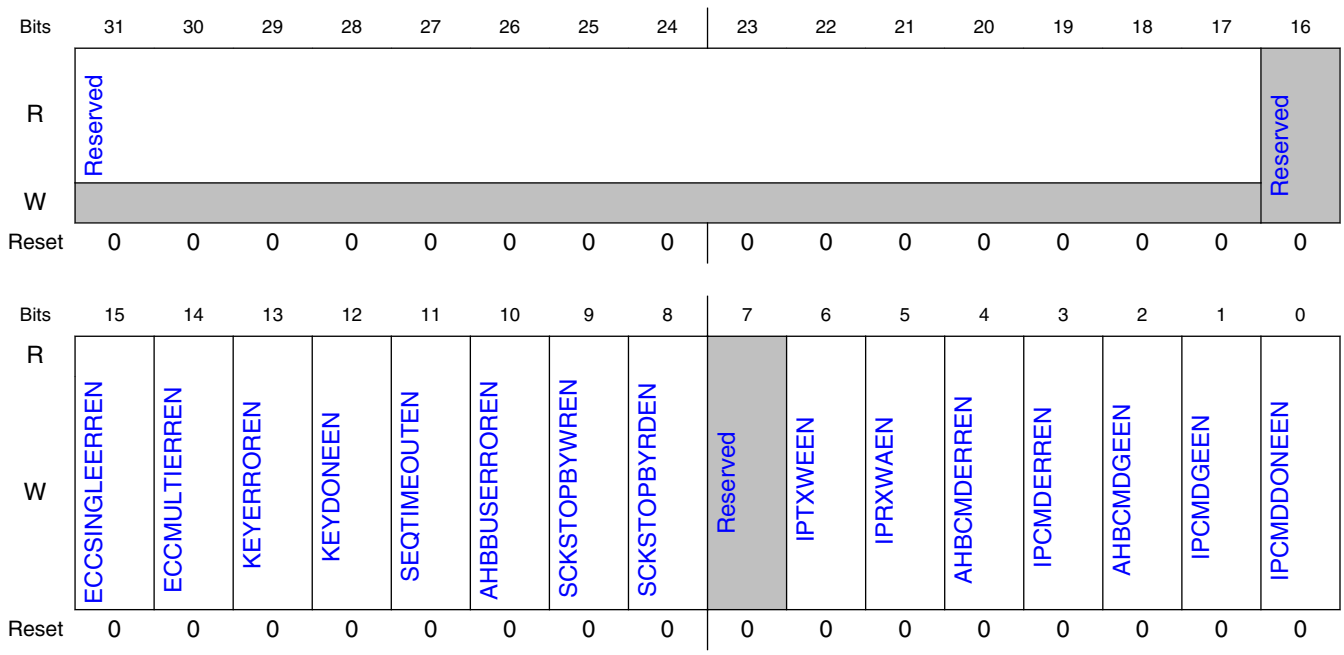
30.7.2.6.1 Offset

Register	Offset
INTEN	10h

30.7.2.6.2 Function

Includes AHB error response, IPS error response, and ECC error interrupt enables. See Interrupts.

30.7.2.6.3 Diagram



30.7.2.6.4 Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15 ECCSINGLEERREN	ECC Single-Bit Error Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
14	ECC Multibit Error Interrupt Enable

Table continues on the next page...

Field	Function
ECCMULTIERREN	0b - Disable interrupt or no impact 1b - Enable interrupt
13 KEYERROREN	OTFAD Key Blob Processing Error Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
12 KEYDONEEN	OTFAD Key Blob Processing Done Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
11 SEQTIMEOUTEN	Sequence execution Timeout Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
10 AHBBUSERROREN	AHB Bus Error Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
9 SCKSTOPBYWREN	SCLK Stopped By Write Interrupt Enable Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous transmit FIFO is empty. 0b - Disable interrupt or no impact 1b - Enable interrupt
8 SCKSTOPBYRDEN	SCLK Stopped By Read Interrupt Enable Enables interrupt that indicates SCLK is stopped during command sequence because asynchronous receive FIFO is full. 0b - Disable interrupt or no impact 1b - Enable interrupt
7 —	Reserved
6 IPTXWEEN	IP Transmit FIFO Watermark Empty Interrupt Enable Enables interrupt that indicates IP transmit FIFO contains more empty space than watermark level. 0b - Disable interrupt or no impact 1b - Enable interrupt
5 IPRXWAEN	IP Receive FIFO Watermark Available Interrupt Enable Enables interrupt that indicates IP receive FIFO contains more valid data than the watermark level. 0b - Disable interrupt or no impact 1b - Enable interrupt
4 AHBCMDERREN	AHB-Triggered Command Sequences Error Detected Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
3 IPCMDERREN	IP-Triggered Command Sequences Error Detected Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
2 AHBCMDGEEN	AHB-Triggered Command Sequences Grant Timeout Interrupt Enable. 0b - Disable interrupt or no impact 1b - Enable interrupt
1 IPCMDGEEN	IP-Triggered Command Sequences Grant Timeout Interrupt Enable 0b - Disable interrupt or no impact 1b - Enable interrupt
0	IP-Triggered Command Sequences Execution Finished Interrupt Enable 0b - Disable interrupt or no impact

Field	Function
IPCMDDONEEN	1b - Enable interrupt

30.7.2.7 Interrupt (INTR)

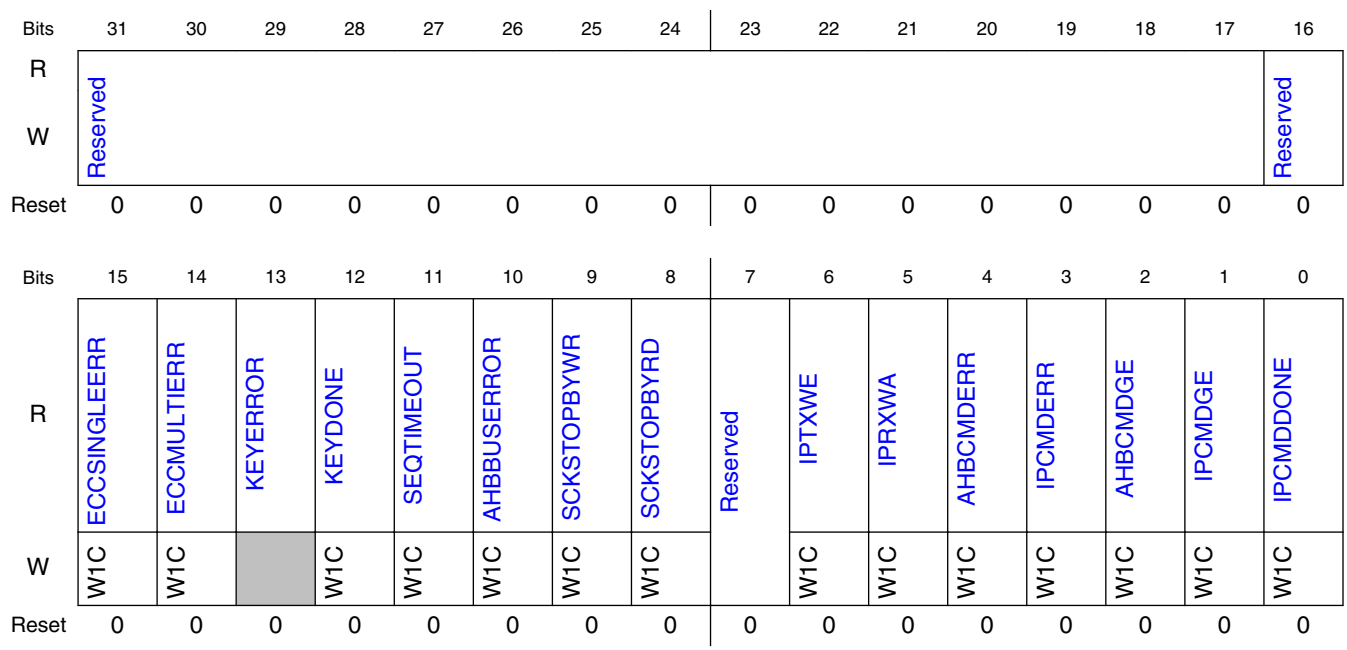
30.7.2.7.1 Offset

Register	Offset
INTR	14h

30.7.2.7.2 Function

Includes AHB error response, IPS error response, and ECC error interrupts. See Interrupts.

30.7.2.7.3 Diagram



30.7.2.7.4 Fields

Field	Function
31-17	Reserved

Table continues on the next page...

Field	Function
—	
16 —	Reserved
15 ECCSINGLEERR	ECC Single-Bit Error **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
14 ECCMULTIERR	ECC Multibit Error **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
13 KEYERROR	OTFAD Key Blob Processing Error **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
12 KEYDONE	OTFAD key blob processing done interrupt.
11 SEQTIMEOUT	Sequence Execution Timeout **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
10 AHBBUSERROR	AHB Bus Error Generated upon an AHB bus timeout or an AHB bus illegal access to flash during OTFAD key blob processing.

Table continues on the next page...

Field	Function
	NOTE: This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
9 SCKSTOPBYWR	SCLK Stopped Due To Empty Transmit FIFO Generated when SCLK is stopped during command sequence because the asynchronous transmit FIFO is empty. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
8 SCKSTOPBYRD	SCLK Stopped Due To Full Receive FIFO Generated when SCLK is stopped during command sequence because the asynchronous receive FIFO is full. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
7 —	Reserved
6 IPTXWE	IP Transmit FIFO Watermark Empty Generated when the IP transmit FIFO contains more empty space than the watermark level. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
5 IPRXWA	IP Receive FIFO Watermark Available Generated when the IP receive FIFO contains more valid data than the watermark level.

Table continues on the next page...

Field	Function
	NOTE: This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
4 AHBCMDERR	AHB-Triggered Command Sequences Error Generated upon an AHB-triggered command sequence error. When an error is detected for an AHB command, the command is ignored and not executed. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
3 IPCMDERR	IP-Triggered Command Sequences Error Generated upon an IP-triggered command sequence error. When an error is detected for an IP command, the command is ignored and not executed. **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
2 AHBCMDGE	AHB-Triggered Command Sequences Grant Timeout **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag
1 IPCMDGE	IP-Triggered Command Sequences Grant Timeout **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred

Table continues on the next page...

Field	Function
	When writing 0b - No effect 1b - Clear the flag
0 IPCMDDONE	IP-Triggered Command Sequences Execution Finished Generated upon completion of an IP-triggered command sequence. Also generated when IPCMDGE or IPCMDERR interrupt is generated **NOTE:** This field behaves differently for read and write operations. When reading 0b - Interrupt condition has not occurred 1b - Interrupt condition has occurred When writing 0b - No effect 1b - Clear the flag

30.7.2.8 LUT Key (LUTKEY)

30.7.2.8.1 Offset

Register	Offset
LUTKEY	18h

30.7.2.8.2 Function

Contains the key to lock and unlock LUT. See Lookup table (LUT).

30.7.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

30.7.2.8.4 Fields

Field	Function
31-0 KEY	LUT Key Contains key to lock or unlock LUT. The key is 5AF05AF0h. Read value is always 5AF05AF0h.

30.7.2.9 LUT Control (LUTCR)

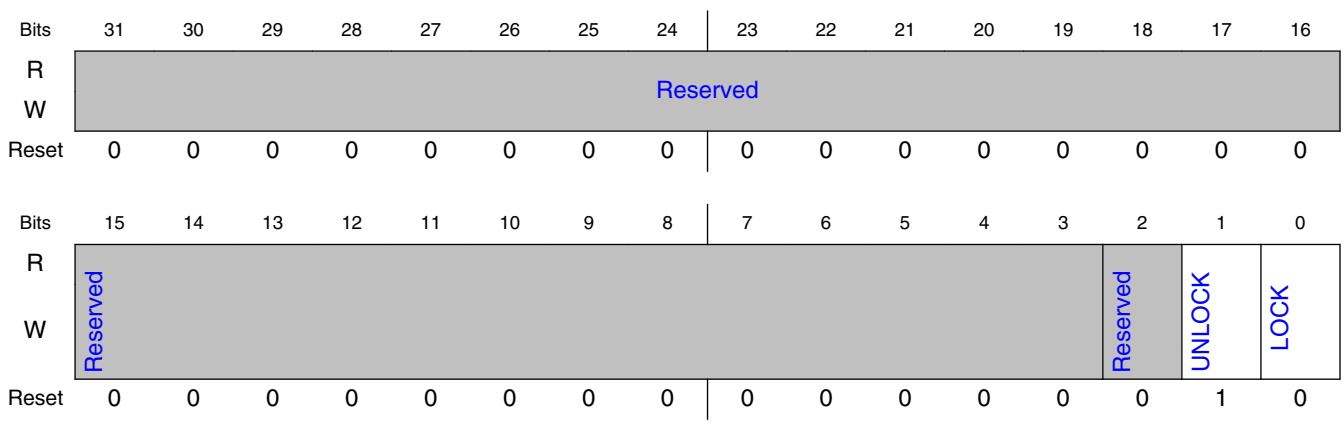
30.7.2.9.1 Offset

Register	Offset
LUTCR	1Ch

30.7.2.9.2 Function

Used with LUTKEY register to lock or unlock LUT. For the lock or unlock operation to be successful, this register must be written immediately after writing 5AF05AF0h to the LUTKEY register. See Lookup table (LUT) for details on locking and unlocking LUT. You cannot write 00 or 11 to the LOCK and UNLOCK fields.

30.7.2.9.3 Diagram



30.7.2.9.4 Fields

Field	Function
31-3	Reserved

Table continues on the next page...

Field	Function
—	
2 —	Reserved
1 UNLOCK	Unlock LUT 0b - LUT is locked (LUTCR[LOCK] must be 1) 1b - LUT is unlocked and can be written
0 LOCK	Lock LUT 0b - LUT is unlocked (LUTCR[UNLOCK] must be 1) 1b - LUT is locked and cannot be written

30.7.2.10 AHB Receive Buffer n Control 0 (AHBRXBUF0CR0 - AHBRXBUF7CR0)

30.7.2.10.1 Offset

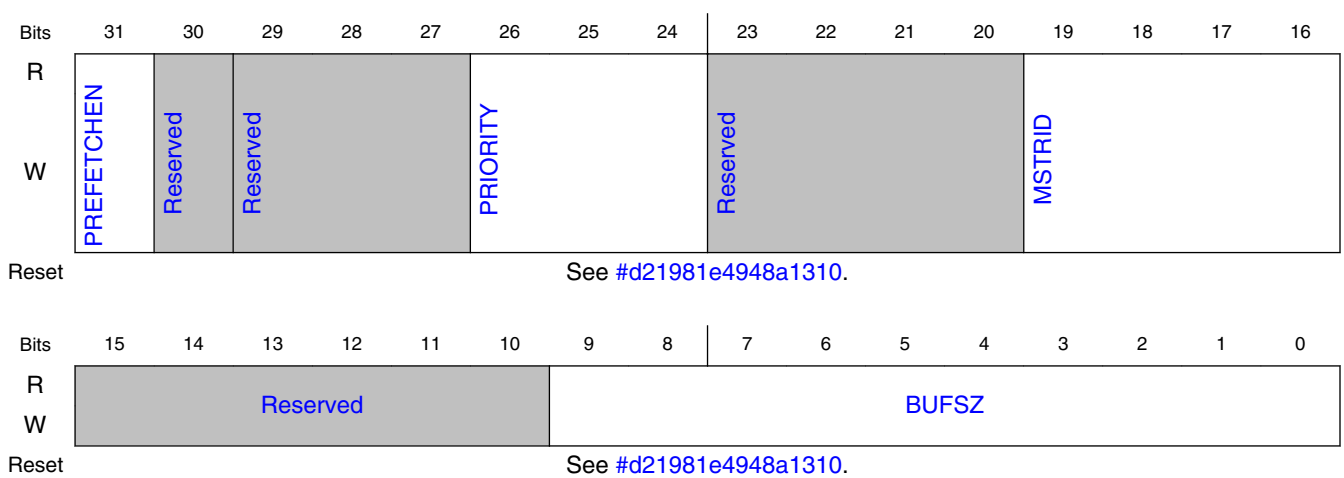
For n = 0 to 7:

Register	Offset
AHBRXBUFnCR0	20h + (n × 4h)

30.7.2.10.2 Function

Stores the read data from the SPI interface.

30.7.2.10.3 Diagram



30.7.2.10.4 Register reset values

Register	Reset value
AHBRXBUF0CR0	FLEXSPI1,FLEXSPI2: 8000_0040h
AHBRXBUF1CR0	FLEXSPI1,FLEXSPI2: 8001_0040h
AHBRXBUF2CR0	FLEXSPI1,FLEXSPI2: 8002_0040h
AHBRXBUF3CR0	FLEXSPI1,FLEXSPI2: 8003_0040h
AHBRXBUF4CR0	FLEXSPI1,FLEXSPI2: 8004_0040h
AHBRXBUF5CR0	FLEXSPI1,FLEXSPI2: 8005_0040h
AHBRXBUF6CR0	FLEXSPI1,FLEXSPI2: 8006_0040h
AHBRXBUF7CR0	FLEXSPI1,FLEXSPI2: 8007_0040h

30.7.2.10.5 Fields

Field	Function
31 PREFETCHEN	AHB Read Prefetch Enable Enables AHB read prefetch for the controller corresponding to the current AHB receive buffer. The prefetch feature is disabled when AHBCR[PREFETCHEN] is 0. You can use this field to enable or disable prefetch separately for each controller. 0b - Disabled 1b - Enabled when AHBCR[PREFETCHEN] is enabled.
30 —	Reserved
29-27 —	Reserved
26-24 PRIORITY	AHB Controller Read Priority Configure the priority for the AHB Controller Read to which this AHB receive buffer is assigned. 7 is the highest priority, 0 the lowest. See Command abort and suspend.
23-20 —	Reserved
19-16 MSTRID	AHB Controller ID Configures the ID of the AHB controller to which this AHB receive buffer is assigned. See AHB receive buffer management.
15-10 —	Reserved
9-0 BUFSZ	AHB Receive Buffer Size Configures the size of the AHB receive buffer in multiples of 64 bits. See AHB receive buffer management.

30.7.2.11 Flash Control 0 (FLSHA1CR0 - FLSHB2CR0)

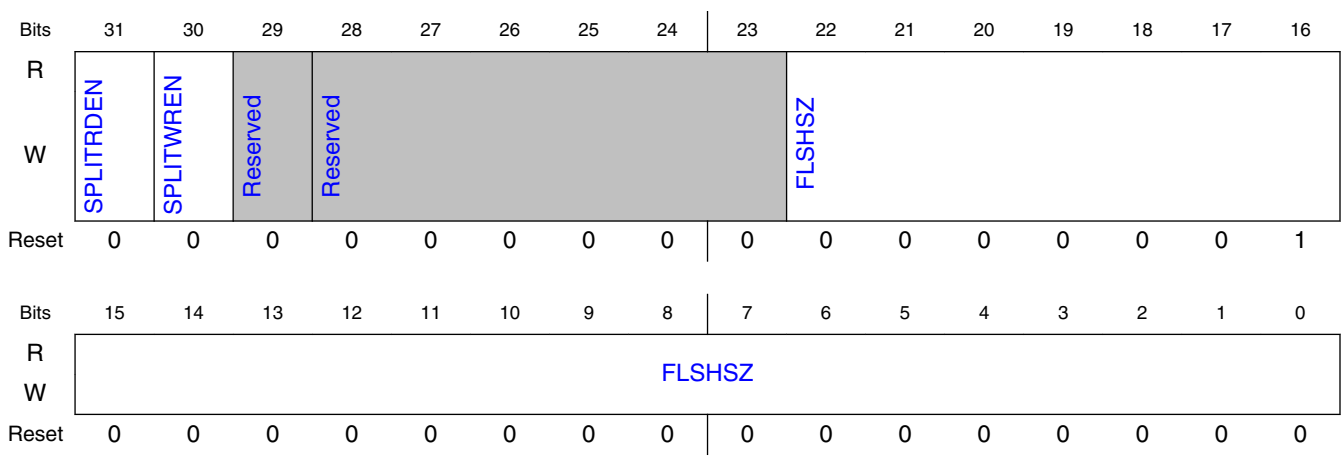
30.7.2.11.1 Offset

Register	Offset
FLSHA1CR0	60h
FLSHA2CR0	64h
FLSHB1CR0	68h
FLSHB2CR0	6Ch

30.7.2.11.2 Function

Contains flash memory size setting. FlexSPI determines which device is accessed (Chip Select) via this register.

30.7.2.11.3 Diagram



30.7.2.11.4 Fields

Field	Function
31 SPLITRDEN	AHB Read Access Split Function Enable Combined with AHBCR[SPLITEN], used to control the split function enable and disable, based on which flash is accessed. See AHB transaction split. 0b - Disable 1b - Enable

Table continues on the next page...

Field	Function
30 SPLITWREN	AHB Write Access Split Function Enable Combined with AHBCR[SPLITEN], used to control the split function enable and disable, based on which flash memory is accessed. See AHB transaction split 0b - Disable 1b - Enable
29 —	Reserved
28-23 —	Reserved
22-0 FLSHSZ	Flash Size in KB Configures the maximum flash memory size. See the chip-specific section for the maximum flash size supported. When the value of this field is greater than the maximum size, the device flash size is interpreted as the maximum. The maximum flash size is the largest flash size supported for a single device. It is also the maximum total flash size supported for all devices (up to four). If the total flash size is larger than the maximum size, only the maximum flash size address space is accessible.

30.7.2.12 Flash Control 1 (FLSHA1CR1 - FLSHB2CR1)

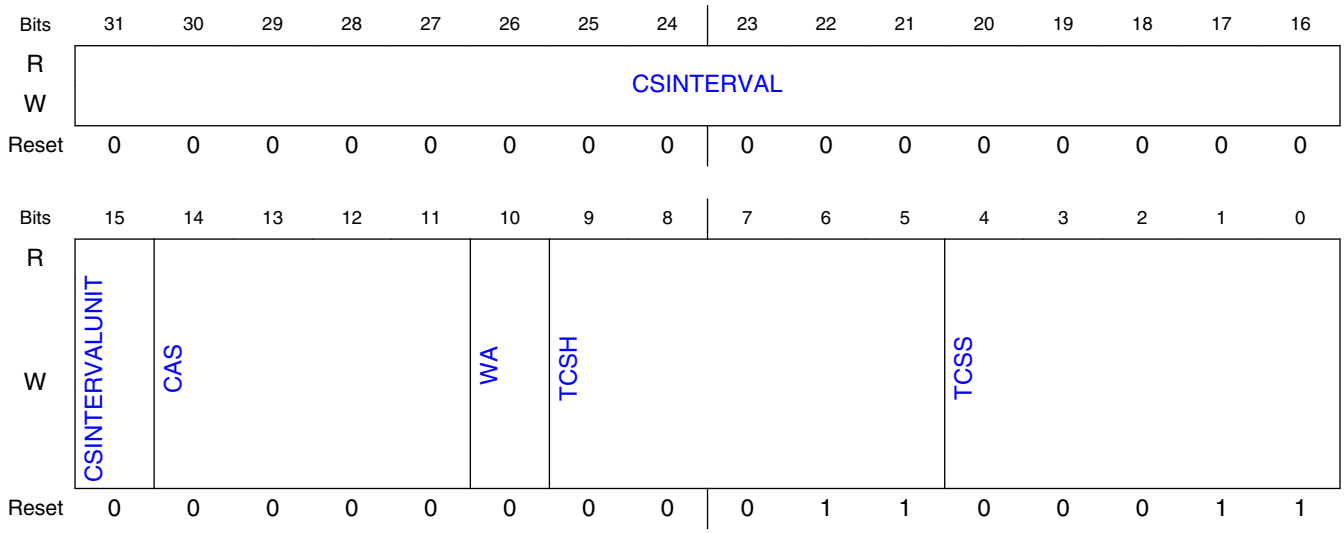
30.7.2.12.1 Offset

Register	Offset
FLSHA1CR1	70h
FLSHA2CR1	74h
FLSHB1CR1	78h
FLSHB2CR1	7Ch

30.7.2.12.2 Function

Contains settings for flash device-specific timings and flash internal address space.

30.7.2.12.3 Diagram



30.7.2.12.4 Fields

Field	Function
31-16 CSINTERVAL	Chip Select Interval Configures the minimum interval between flash device chip select deassertion and chip select assertion. If the external flash device has a limitation on the interval between command sequences, configure this field accordingly. If there is no limitation, write 0h to this field. When CSINTERVALUNIT = 0, the chip select invalid interval is: CSINTERVAL × 1 serial clock cycle; When CSINTERVALUNIT = 1, the chip select invalid interval is: CSINTERVAL × 256 serial clock cycles. **NOTE:** The minimum chip select interval is 2 cycles, even when the value of CSINTERVAL is less than 2.
15 CSINTERVALUNIT	Chip Select Interval Unit Configures the interval unit for chip select. 0b - 1 serial clock cycle 1b - 256 serial clock cycles
14-11 CAS	Column Address Size When external flash memory has a separate address field for rows and columns, this field configures the flash column address bit width. FlexSPI automatically splits a flash-mapped address into row address and column address according to the values of this field and the WA field. When the external flash memory does not support column address, write 0 to this field. FlexSPI transmits all flash address bits as Row address. For flash address mapping, see Flash address sent to flash memory devices.
10 WA	Word-Addressable Configures whether external flash memory is word-addressable or byte-addressable. If flash memory is word-addressable, it should be accessed in multiples of 16 bits. Currently, FlexSPI does not transmit flash address bit 0 to external flash memory. For flash address mapping, see Flash address sent to flash memory devices. 0b - Byte-addressable 1b - Word-addressable

Table continues on the next page...

Field	Function
9-5 TCSH	Serial Flash CS Hold Time Used to meet flash TCSH timing requirement. Serial flash CS Hold time that FlexSPI promises is: (TCSH + 1/2) serial clock cycles for DDR mode, and TCSH serial clock cycles for SDR mode. See Output timing between chip select and SCLK.
4-0 TCSS	Serial Flash CS Setup Time Used to meet flash TCSS timing requirement. Serial flash CS Setup time that FlexSPI promises is: (TCSS + 1/2) serial root clock cycles for SDR and DDR mode. See Output timing between chip select and SCLK.

30.7.2.13 Flash Control 2 (FLSHA1CR2 - FLSHB2CR2)

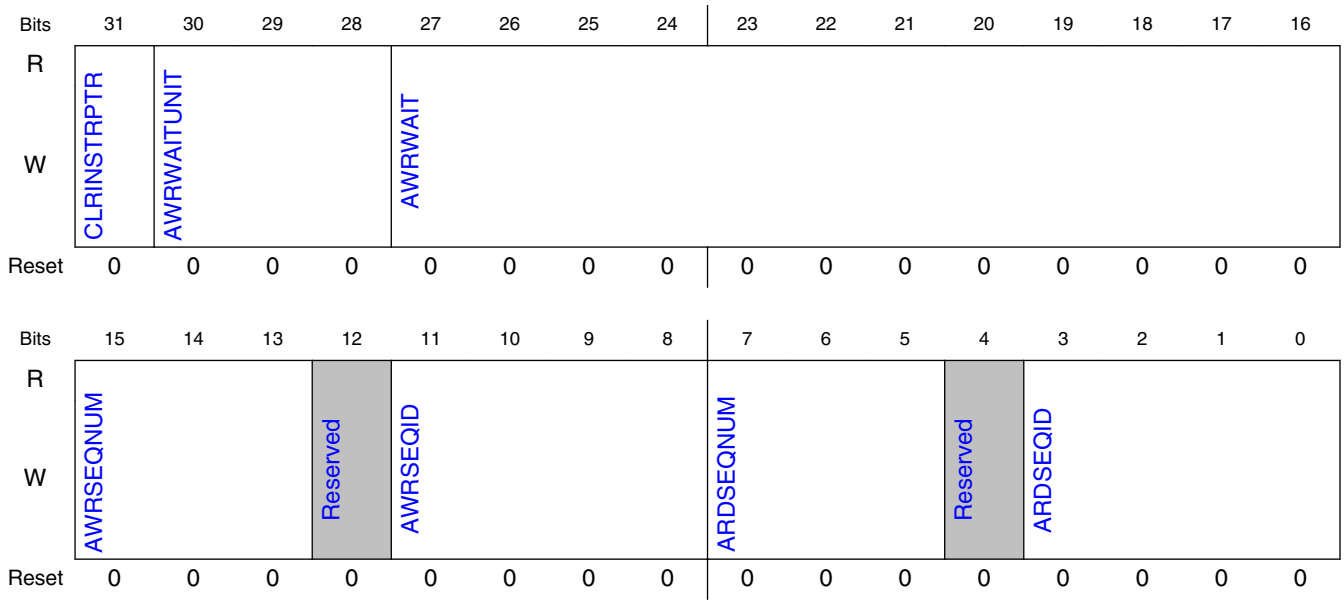
30.7.2.13.1 Offset

Register	Offset
FLSHA1CR2	80h
FLSHA2CR2	84h
FLSHB1CR2	88h
FLSHB2CR2	8Ch

30.7.2.13.2 Function

Contains fields to configure AHB bus access. If the four external devices are different types, AHB read and write commands may use different command sequences; AHB bus ready wait time may also differ.

30.7.2.13.3 Diagram



30.7.2.13.4 Fields

Field	Function
31 CLRINSTRPTR	
30-28 AWRWAITUNIT	AWRWAIT Unit Configures the unit of AHB write wait time, as the value of AWRWAIT determines, in terms of AHB clock cycles. 000b - 2 001b - 8 010b - 32 011b - 128 100b - 512 101b - 2048 110b - 8192 111b - 32768
27-16 AWRWAIT	AHB Write Wait Configures AHB write wait time, with the AWRWAITUNIT field. Certain devices (such as FPGA) require time to write data into internal memory after the command sequences finished on the FlexSPI interface. If another read command sequence arrives before the current programming finishes internally, the read data may be wrong. This field is used to hold the AHB bus ready for AHB write access, waiting until the programming is finished in the external device. This hold ensures that no AHB read command is triggered before the programming finishes in the external device. The wait cycle between an AHB-triggered command sequence finishing on FlexSPI and the AHB return bus being ready is: AWRWAIT × AWRWAITUNIT.
15-13 AWRSEQNUM	Sequence Number for AHB Write-Triggered Command

Table continues on the next page...

Field	Function
	Configures the sequence number of an AHB read-triggered command. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash programming access is done via several command sequences. An AHB write command triggers (AWRSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE:** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: AWRSEQID + AWRSEQNUM < 16 • Software must ensure that the AWRSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
12 —	Reserved
11-8 AWRSEQID	Sequence Index for AHB Write-Triggered Command
7-5 ARDSEQNUM	Sequence Number for AHB Read-Triggered Command Configures the sequence number of an AHB read-triggered command in the lookup table. For certain flash devices (for example, HyperFlash, HyperRam, and Serial NAND flash), a flash read access is done via several command sequences. An AHB read command triggers (ARDSEQNUM + 1) command sequences to external flash memory each time. FlexSPI executes the sequences in LUT incrementally. **NOTE:** • Software should ensure that the last sequence index never exceeds LUT sequence numbers: ARDSEQID + ARDSEQNUM ≤ 16. • Software must ensure that the ARDSEQNUM and LUT fields are configured correctly according to the external device specification. FlexSPI does not check the sequence; it executes the sequences one by one.
4 —	Reserved
3-0 ARDSEQID	Sequence Index for AHB Read-Triggered Command in LUT

30.7.2.14 Flash Control 4 (FLSHCR4)

30.7.2.14.1 Offset

Register	Offset
FLSHCR4	94h

30.7.2.14.2 Function

Provides configuration for all external devices.

30.7.2.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved			Reserved			Reserved	Reserved	WMENB	WMENA	WMOPT2	WMOPT1
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

30.7.2.14.4 Fields

Field	Function
31-12 —	Reserved
11-9 —	Reserved
8-6 —	Reserved
5 —	Reserved
4 —	Reserved
3 WMENB	Write Mask Enable for Port B Enables write mask for flash device on port B. 0b - Disabled. When writing to external device, DQS(RWDS) pin is not driven. 1b - Enabled. When writing to external device, FlexSPI drives DQS(RWDS) pin as write mask output.
2 WMENA	Write Mask Enable for Port A Enables write mask for flash device on port A. 0b - Disabled. When writing to external device, DQS(RWDS) pin is not driven. 1b - Enabled. When writing to external device, FlexSPI drives DQS(RWDS) pin as write mask output.
1 WMOPT2	Write Mask Option 2 When using AP memory, used to remove AHB or IP write burst minimum length limitation. When using this field, FLSHCR4[WMOPT1] should be 1. 0b - When writing to an external device, DQS pin is used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst length is not limited. 1b - When writing to an external device, DQS pin is not used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst length is limited. The minimum write burst length should be 4.

Table continues on the next page...

Field	Function
0 WMOPT1	Write Mask Option 1 Used to remove AHB and IP write burst start address alignment limitation. 0b - When writing to an external device, DQS pin is used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is not limited. 1b - When writing to an external device, DQS pin is not used as write mask. When flash memory is accessed in individual mode, AHB or IP write burst start address alignment is limited.

30.7.2.15 IP Control 0 (IPCR0)

30.7.2.15.1 Offset

Register	Offset
IPCR0	A0h

30.7.2.15.2 Function

Provides all configuration required for IP commands. Provides the start address for the flash device, instead of the chip address, to be accessed for IP command. FlexSPI determines the chip select automatically according to this start address.

NOTE

- You cannot issue an IP command that crosses flash device boundaries. If you do so, it generates an IPCMDERR interrupt.
- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

30.7.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.15.4 Fields

Field	Function
31-0 SFAR	Serial Flash Address Configures the serial flash address for IP commands. The address should be the address of the flash device without the base address.

30.7.2.16 IP Control 1 (IPCR1)

30.7.2.16.1 Offset

Register	Offset
IPCR1	A4h

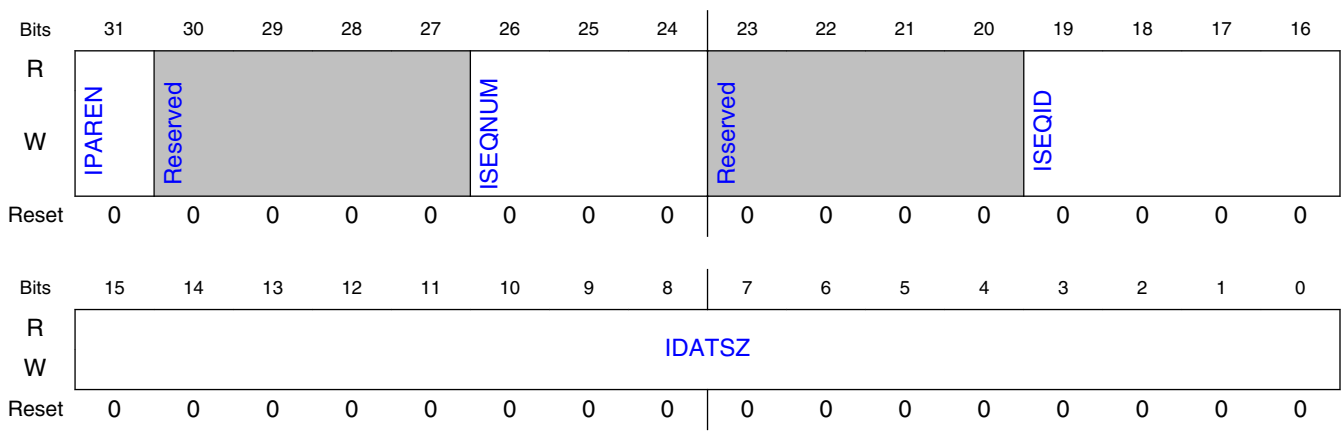
30.7.2.16.2 Function

Provides all configuration required for IP commands. Provides the flash read and program data size, sequence index in LUT, sequence number and individual/parallel mode settings for IP commands.

NOTE

- Configure this register before an IP command is triggered.
- Do not change the values in this register while an IP command is in progress.

30.7.2.16.3 Diagram



30.7.2.16.4 Fields

Field	Function
31 IPAREN	Parallel Mode Enable for IP Commands 0b - Disabled. Flash memory is accessed in Individual mode. 1b - Enabled. Flash memory is accessed in Parallel mode.
30-27 —	Reserved
26-24 ISEQNUM	Sequence Number for IP command: ISEQNUM+1.
23-20 —	Reserved
19-16 ISEQID	Sequence Index in LUT for IP command.
15-0 IDATSZ	Flash Read/Program Data Size (in bytes) for IP command.

30.7.2.17 IP Command (IPCMD)

30.7.2.17.1 Offset

Register	Offset
IPCMD	B0h

30.7.2.17.2 Function

Used to trigger an IP command to access external flash device. When the arbitrator grants the IP command, the command is executed on the FlexSPI interface.

30.7.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															0
W																TRG
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.17.4 Fields

Field	Function
31-1 —	Reserved
0 TRG	Command Trigger Triggers an IP command. This field clears automatically after it has been written and always reads as 0. **NOTE:** You cannot trigger another IP command before the previous IP command finishes on the FlexSPI interface. Software must poll INTR[IPCMDDONE] or wait for this interrupt. 0b - No action 1b - Start the IP command that the IPCR0 and IPCR1 registers define.

30.7.2.18 IP Receive FIFO Control (IPRXFCR)

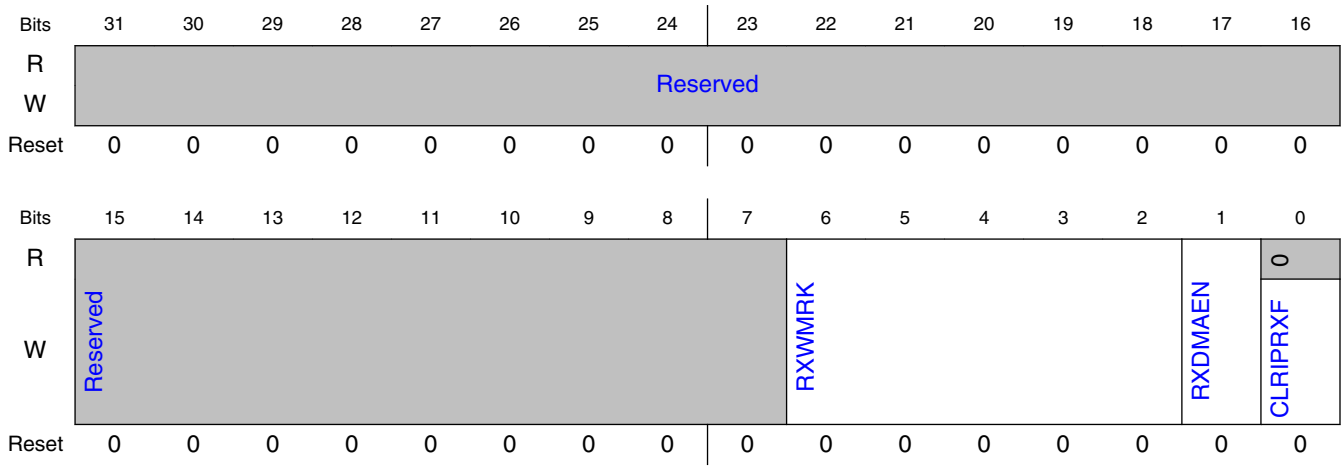
30.7.2.18.1 Offset

Register	Offset
IPRXFCR	B8h

30.7.2.18.2 Function

Provides the configuration fields for IP receive FIFO management.

30.7.2.18.3 Diagram



30.7.2.18.4 Fields

Field	Function
31-7 —	Reserved
6-2 RXWMRK	IP Receive FIFO Watermark Level Configures the watermark level for the IP receive FIFO. The watermark level is (RXWMRK + 1) × 64 bits. The INTR[IPRXWA] interrupt is set when FlexSPI fills the IP receive FIFO ≥ the watermark level. A DMA request occurs when the fill level is ≥ the watermark level and IPRXFCR[RXDMAEN] = 1. When the fill level is ≥ the watermark level and INTEN[IPRXWAEN] = 1, an INTR[IPRXWA] interrupt is generated. **NOTE:** After writing 1 to INTR[IPRXWA] to clear it, the read address should be rolled back to the start address (memory mapped). If the IP bus reads the IP receive FIFO, the read address should roll back to RFDR0. If the AHB bus reads the IP receive FIFO, the read address should roll back to ARDF_BASE.
1 RXDMAEN	IP Receive FIFO Reading by DMA Enable 0b - Disabled. The processor reads the FIFO. 1b - Enabled. DMA reads the FIFO.
0 CLRIPRXF	Clear IP Receive FIFO Clears all valid data entries in IP receive FIFO. Resets the read and write pointers in IP receive FIFO. 0b - No function 1b - A clock cycle pulse clears all valid data entries in IP receive FIFO.

30.7.2.19 IP Transmit FIFO Control (IPTXFCR)

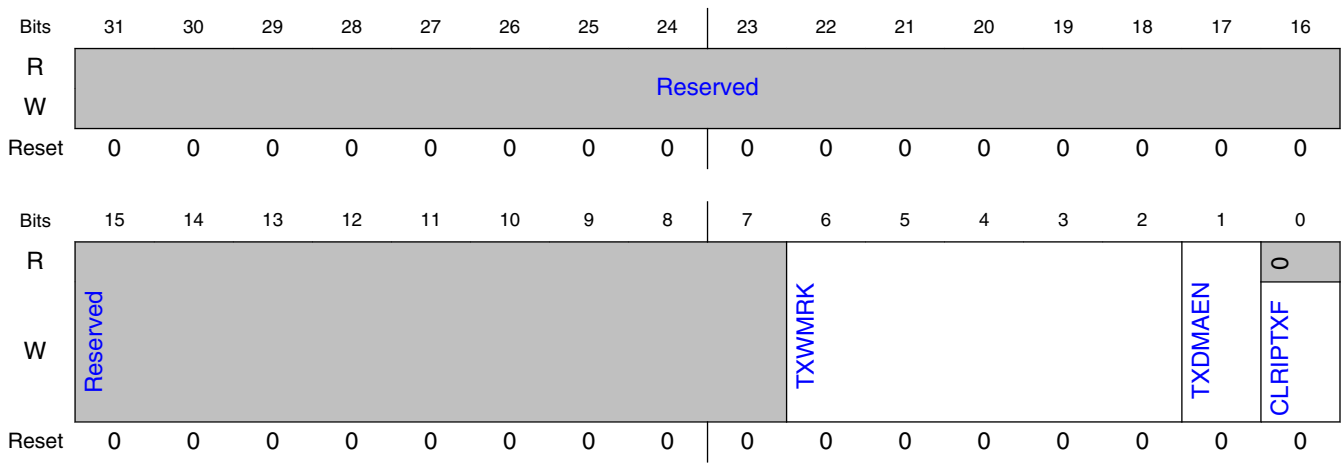
30.7.2.19.1 Offset

Register	Offset
IPTXFCR	BCh

30.7.2.19.2 Function

Provides the configuration fields for IP transmit FIFO management.

30.7.2.19.3 Diagram



30.7.2.19.4 Fields

Field	Function
31-7 —	Reserved
6-2 TXWMRK	Transmit Watermark Level Sets the transmit watermark level. The watermark level is (TXWMRK + 1) × 64 bits. INTR[IPTXWE] is set when FlexSPI empties the IP transmit FIFO ≥ the watermark level. When the empty level ≥ the watermark level and IPTXFCR[TXDMAEN] = 1, a DMA request occurs. When the empty level ≥ the watermark level and INTEN[IPTXWEEN] = 1, an INTR[IPTXWE] interrupt is generated. **NOTE:** • The watermark level should not be larger than the write window. • The watermark level should not be larger than the IP transmit FIFO size. • The write address to the IP receive FIFO should roll back to the start address of the write window. After pushing the IP transmit fifo, you should write 1 to INTR[IPTXWE] to clear it, which will perform the rollback.
1 TXDMAEN	Transmit FIFO DMA Enable Selects whether DMA or processor fills IP transmit FIFO. 0b - Processor 1b - DMA

Table continues on the next page...

Field	Function
0 CLRIPTXF	Clear IP Transmit FIFO Clears all valid data entries in IP transmit FIFO. The read and write pointers in IP transmit FIFO are reset. 0b - No function 1b - A clock cycle pulse clears all valid data entries in the IP transmit FIFO.

30.7.2.20 DLL Control 0 (DLLACR - DLLBCR)

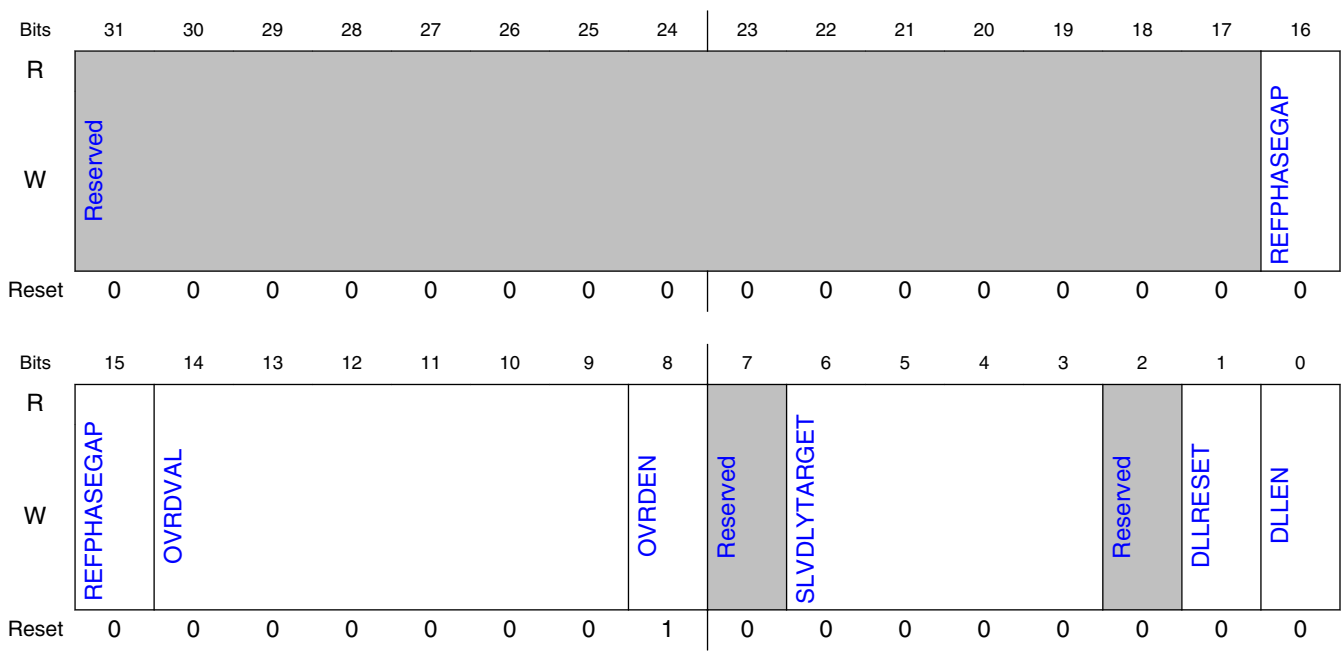
30.7.2.20.1 Offset

Register	Offset
DLLACR	C0h
DLLBCR	C4h

30.7.2.20.2 Function

Configures Flash A and B sample clock DLL.

30.7.2.20.3 Diagram



30.7.2.20.4 Fields

Field	Function
31-17 —	Reserved
16-15 REFPHASEGAP	Reference Clock Delay Line Phase Adjust Gap. REFPHASEGAP setting of 2h is recommended if DLLEN is set.
14-9 OVRDVAL	Target Clock Delay Line Override Value Configures the override value for the target clock line delay cell number. When OVRDEN = 1, the delay cell number in DLL is OVRDVAL + 1. See DLL configuration for sampling.
8 OVRDEN	Target Clock Delay Line Override Value Enable Enables the override value for the target clock line delay cell number. 0b - Disable 1b - Enable
7 —	Reserved
6-3 SLVDLYTARGET	Target Delay Line Configures the target delay line for the target. The delay target for target delay line is: ((SLVDLYTARGET +1) × 1/32 × clock cycle of reference clock (serial root clock). If serial root clock is ≥ 100 MHz, DLLEN is set to 1, OVRDEN is set to 0, then SLVDLYTARGET setting is up to flash parameters used.
2 —	Reserved
1 DLLRESET	DLL reset Forces a DLL reset. The forced reset causes the DLL to lose lock and recalibrate to detect a ref_clock half-period phase shift. The reset action is edge-triggered, so software must write 0 to this field after writing 1 to it (no delay limitation). 0b - No function 1b - Force DLL reset.
0 DLLEN	DLL Calibration Enable Enables DLL calibration. When DLL calibration is disabled, the delay cell number in the target delay line is always 1. When DLLxCR[OVRDEN] = 1, the target delay line is overridden and this field is ignored. 0b - Disable 1b - Enable

30.7.2.21 Misc Control 4 (MISCCR4)

30.7.2.21.1 Offset

Register	Offset
MISCCR4	D0h

30.7.2.21.2 Function

Contains information about ECC multibit error interrupt.

30.7.2.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AHBADDRESS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AHBADDRESS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.21.4 Fields

Field	Function
31-0 AHBADDRESS	AHB Address Contains the AHB address that triggered the current ECC multibit error interrupt.

30.7.2.22 Miscellaneous Control 5 (MISCCR5)

30.7.2.22.1 Offset

Register	Offset
MISCCR5	D4h

30.7.2.22.2 Function

Contains information about ECC single-bit error interrupt.

30.7.2.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ECCSINGLEERRORCORR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECCSINGLEERRORCORR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.22.4 Fields

Field	Function
31-0 ECCSINGLEERRORCORR	ECC Single-Bit Error Correction Indicator Stores single-bit ECC error correction information for first single ECC error after exiting single-bit ECC error interrupt.

30.7.2.23 Miscellaneous Control 6 (MISCCR6)

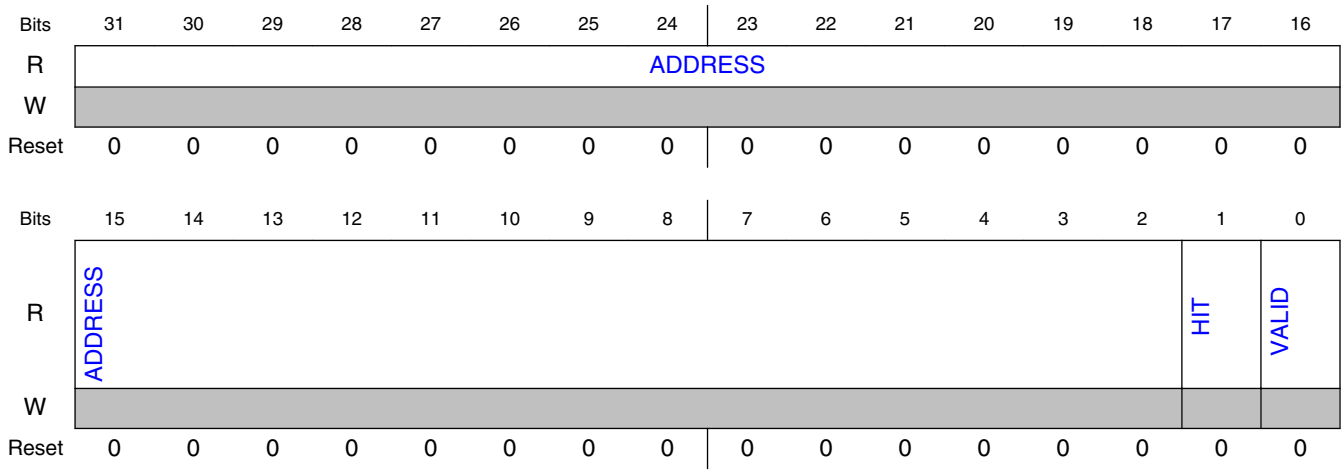
30.7.2.23.1 Offset

Register	Offset
MISCCR6	D8h

30.7.2.23.2 Function

Stores information about the first ECC single-bit error after exiting an ECC single-bit error interrupt.

30.7.2.23.3 Diagram



30.7.2.23.4 Fields

Field	Function
31-2 ADDRESS	ECC Single-Bit Error Address Contains the AHB bus address of the first ECC single-bit error encountered since the last ECC single-bit error interrupt.
1 HIT	ECC Single-Bit Error Information Hit Indicates whether the current read address on the AHB bus with the single-bit error has hit the information stored in MISCCR6[ADDRESS]. Only the address that caused the INTR[ECCSINGLEERR] interrupt is stored and compared; all subsequent single-bit errors are ignored. 0b - Address not hit 1b - Address hit
0 VALID	ECC Single-Bit Error Information Valid Indicates whether the information in MISCCR5 and MISCCR6 registers is valid. AHBCR[ECCSINGLEERRCLR] or reset can clear this field. When 0, do not use any other fields in MISCCR5 and MISCCR6 registers. 0b - Invalid 1b - Valid

30.7.2.24 Miscellaneous Control 7 (MISCCR7)

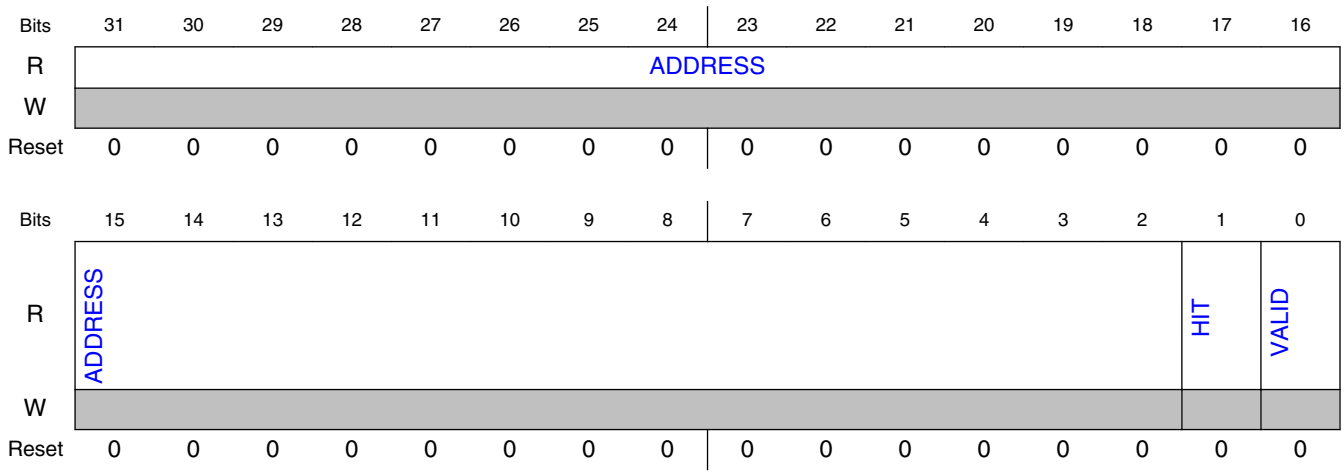
30.7.2.24.1 Offset

Register	Offset
MISCCR7	DCh

30.7.2.24.2 Function

Stores information about the first ECC multibit error after exiting ECC multibit error interrupt.

30.7.2.24.3 Diagram



30.7.2.24.4 Fields

Field	Function
31-2 ADDRESS	ECC multi error address Contains the AHB bus address of the first ECC multibit error encountered since the last ECC multibit error interrupt.
1 HIT	ECC Multibit Error Information Hit Indicates whether the current read address on the AHB bus with the multibit error has hit the information stored in MISCCR7[ADDRESS]. Only the address that caused the INTR[ECCMULTIERR] interrupt is stored and compared; all subsequent multibit errors are ignored. 0b - Address not hit 1b - Address hit
0 VALID	ECC Multibit Error Information Valid Indicates that the information in MISCCR7 register is valid. AHBCR[ECCMULTIERRCLR] or reset can clear this field. When 0, do not use any other fields in MISCCR7 register.

30.7.2.25 Status 0 (STS0)

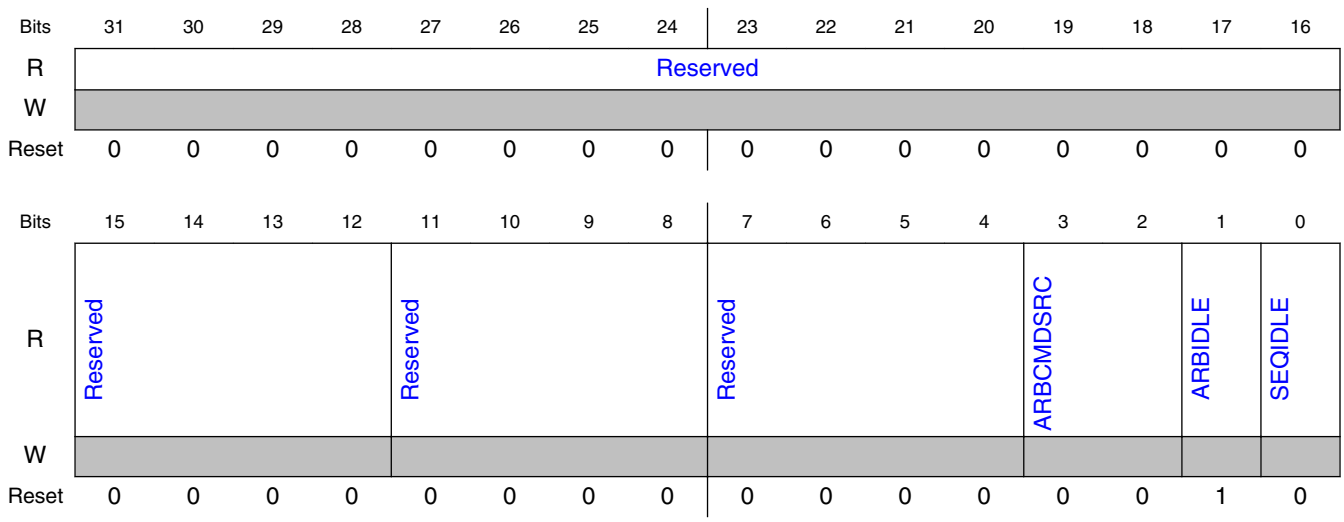
30.7.2.25.1 Offset

Register	Offset
STS0	E0h

30.7.2.25.2 Function

Indicates the status of the internal state machine.

30.7.2.25.3 Diagram



30.7.2.25.4 Fields

Field	Function
31-12 —	Reserved
11-8 —	Reserved
7-4 —	Reserved
3-2 ARBCMDSRC	ARB Command Source Indicates the trigger source of current command sequence that the arbitrator has granted. When ARB_CTL is not busy (STS0[ARBIDLE] = 1), the value of this field does not matter. 00b - Trigger source is AHB read command. 01b - Trigger source is AHB write command. 10b - Trigger source is IP command (by writing 1 to IPCMD[TRG]). 11b - Trigger source is a suspended command that has resumed.

Table continues on the next page...

Field	Function
1 ARBIDLE	ARB_CTL State Machine Idle Indicates whether the state machine in ARB_CTL is idle, or a command sequence that the arbitrator granted has not finished on the FlexSPI interface. When idle, no transaction is occurring on the FlexSPI interface (STS0[SEQIDLE] = 1). When waiting for the FlexSPI controller to become idle, poll this field instead of STS0[SEQIDLE]. 0b - Not idle 1b - Idle
0 SEQIDLE	SEQ_CTL State Machine Idle Indicates whether the state machine in SEQ_CTL is idle, or a command sequence is executing on the FlexSPI interface. 0b - Not idle 1b - Idle

30.7.2.26 Status 1 (STS1)

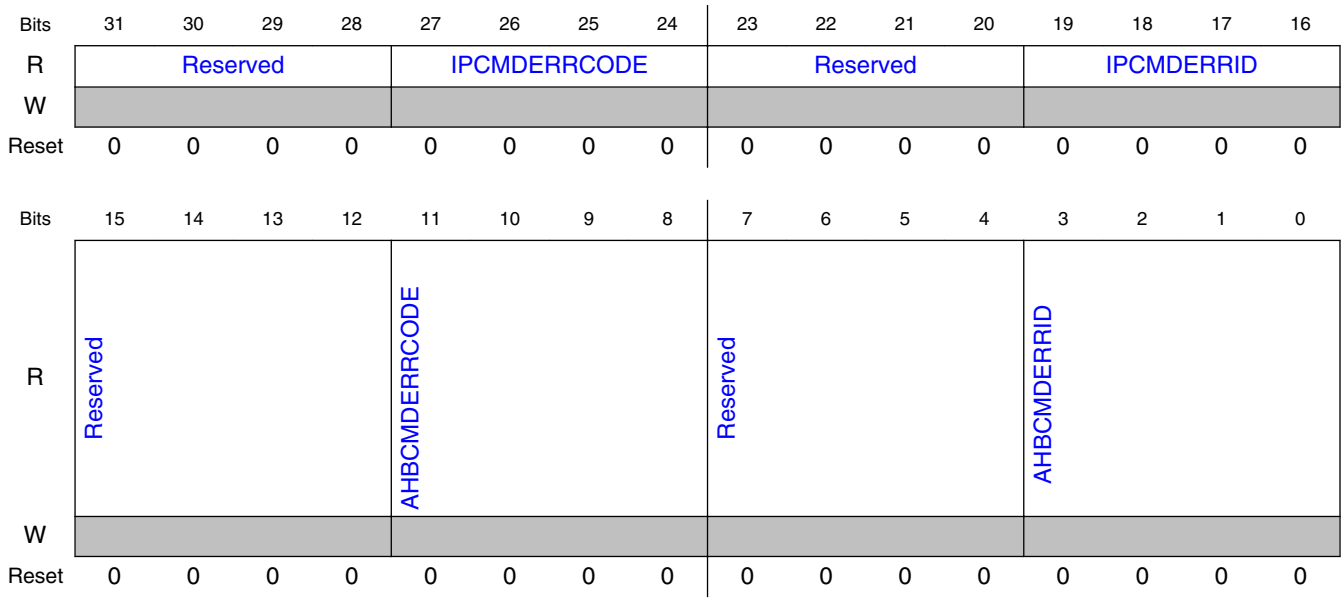
30.7.2.26.1 Offset

Register	Offset
STS1	E4h

30.7.2.26.2 Function

Indicates the error code of the AHB or IPS interface error response.

30.7.2.26.3 Diagram



30.7.2.26.4 Fields

Field	Function
31-28 —	Reserved
27-24 IPCMDERRCODE	IP Command Error Code When an IP command Error is detected, indicates the error code. This field returns to zero when INTR[IPCMDERR] is cleared. 0000b - No error 0010b - IP command with JMP_ON_CS instruction used in the sequence 0011b - Unknown instruction opcode in the sequence 0100b - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101b - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 0110b - Flash memory access start address exceeds entire flash address range (A1, A2, B1, and B2) 1110b - Sequence execution timeout 1111b - Flash boundary crossed
23-20 —	Reserved
19-16 IPCMDERRID	IP Command Error ID When an IP command error is detected, indicates the sequence index. This field returns to zero when INTR[IPCMDERR] is cleared.
15-12 —	Reserved
11-8	AHB Command Error Code

Table continues on the next page...

Field	Function
AHBCMDERRCODE	Contains the error code when an AHB command error is detected. This field returns to zero when INTR[AHBCMDERR] is cleared. 0000b - No error 0010b - AHB Write command with JMP_ON_CS instruction used in the sequence 0011b - Unknown instruction opcode in the sequence 0100b - DUMMY_SDR or DUMMY_RWDS_SDR instruction used in DDR sequence 0101b - DUMMY_DDR or DUMMY_RWDS_DDR instruction used in SDR sequence 1110b - Sequence execution timeout
7-4 —	Reserved
3-0 AHBCMDERRID	AHB Command Error ID When an AHB command error is detected, indicates the sequence index. This field returns to zero when INTR[AHBCMDERR] is cleared.

30.7.2.27 Status 2 (STS2)

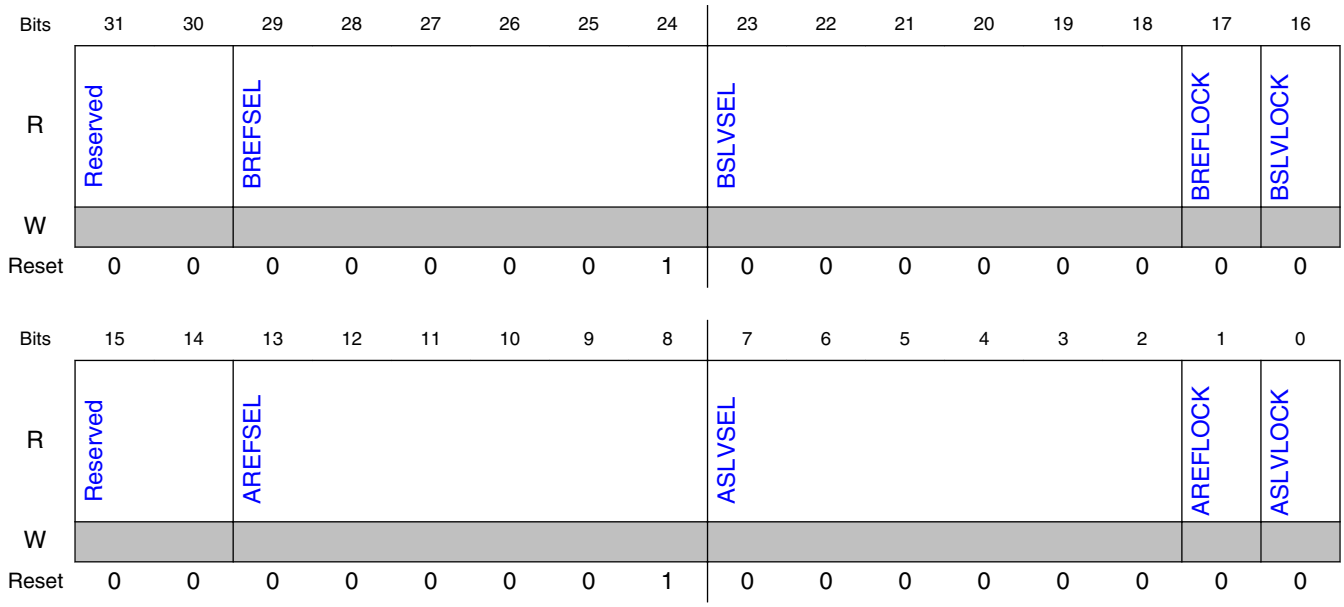
30.7.2.27.1 Offset

Register	Offset
STS2	E8h

30.7.2.27.2 Function

Indicates the status of Flash A and B sample clock DLLs.

30.7.2.27.3 Diagram



30.7.2.27.4 Fields

Field	Function
31-30 —	Reserved
29-24 BREFSEL	Flash B Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
23-18 BSLVSEL	Flash B Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash B. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
17 BREFLOCK	Flash B Sample Clock Reference Delay Line Locked 0b - Not locked 1b - Locked
16 BSLVLOCK	Flash B Sample Target Reference Delay Line Locked 0b - Not locked 1b - Locked
15-14 —	Reserved
13-8 AREFSEL	Flash A Sample Clock Reference Delay Line Delay Cell Number Indicates the sample clock reference delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.

Table continues on the next page...

Field	Function
7-2 ASLVSEL	Flash A Sample Clock Target Delay Line Delay Cell Number Indicates the sample clock target delay line delay cell number for Flash A. There are 0-63 phases. • 000001b - Indicates lock phase 0. • 100000b - Indicates lock phase 63.
1 AREFLOCK	Flash A Sample Clock Reference Delay Line Locked 0b - Not locked 1b - Locked
0 ASLVLOCK	Flash A Sample Target Delay Line Locked 0b - Not locked 1b - Locked

30.7.2.28 AHB Suspend Status (AHBSPNDSTS)

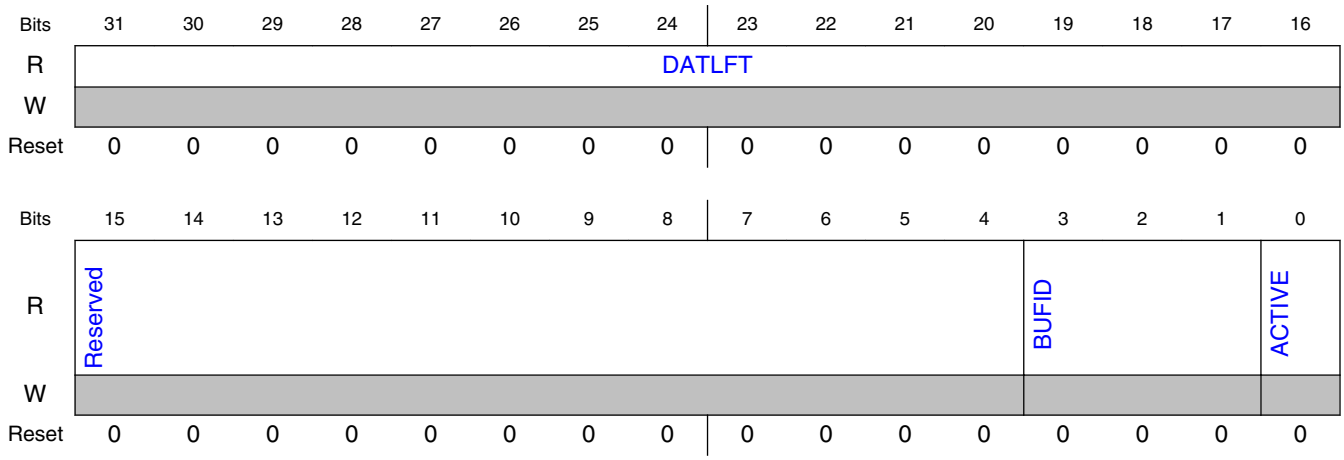
30.7.2.28.1 Offset

Register	Offset
AHBSPNDSTS	ECh

30.7.2.28.2 Function

Indicates the status of suspended AHB read prefetch command sequence. When an IP or AHB command is triggered while the arbitrator processes an AHB read sequence (prefetching additional data not for current AHB burst), the prefetch sequence is suspended. When there are no longer transactions on FlexSPI, the prefetch sequence may be resumed. FlexSPI saves only one AHB prefetch sequence. When a new prefetch sequence is suspended with an active sequence already suspended, the previous suspended sequence is removed and never resumed. See Command abort and suspend.

30.7.2.28.3 Diagram



30.7.2.28.4 Fields

Field	Function
31-16 DATLFT	Data Left Contains the data size remaining (in bytes) for a suspended command sequence.
15-4 —	Reserved
3-1 BUFID	AHB Receive Buffer ID for Suspended Command Sequence
0 ACTIVE	Active AHB Read Prefetch Suspended Indicates whether an AHB read prefetch command sequence has been suspended. 0b - No suspended AHB read prefetch command. 1b - An AHB read prefetch command sequence has been suspended.

30.7.2.29 IP Receive FIFO Status (IPRXFSTS)

30.7.2.29.1 Offset

Register	Offset
IPRXFSTS	F0h

30.7.2.29.2 Function

Indicates the status of IP receive FIFO.

30.7.2.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.29.4 Fields

Field	Function
31-16 RDCNTR	Read Data Counter Contains counter for total data read. The data read is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Receive FIFO Indicates how full the IP receive FIFO is. The fill level is the value of this field × 64 bits.

30.7.2.30 IP Transmit FIFO Status (IPTXFSTS)

30.7.2.30.1 Offset

Register	Offset
IPTXFSTS	F4h

30.7.2.30.2 Function

Indicates the status of IP transmit FIFO.

30.7.2.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRCNTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FILL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.30.4 Fields

Field	Function
31-16 WRCNTR	Write Data Counter Contains counter for total data written. The data written is the value of this field × 64 bits.
15-8 —	Reserved
7-0 FILL	Fill Level of IP Transmit FIFO Indicates how full the IP transmit FIFO is. The fill level is the value of this field × 64 bits.

30.7.2.31 IP Receive FIFO Data a (RFDR0 - RFDR31)

30.7.2.31.1 Offset

For a = 0 to 31:

Register	Offset
RFDRa	100h + (a × 4h)

30.7.2.31.2 Function

Provides read access to IP receive FIFO via IPS bus. The read value is unknown for read access to invalid entries in the IP receive FIFO.

30.7.2.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.31.4 Fields

Field	Function
31-0 RXDATA	Receive Data Contains receive data. See Reading data from IP receive FIFO.

30.7.2.32 IP TX FIFO Data a (TFDR0 - TFDR31)

30.7.2.32.1 Offset

For a = 0 to 31:

Register	Offset
TFDRa	180h + (a × 4h)

30.7.2.32.2 Function

Provides write access to IP transmit FIFO via IPS bus.

30.7.2.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.32.4 Fields

Field	Function
31-0 TXDATA	Transmit Data Contains transmit data. See Writing data to IP transmit FIFO.

30.7.2.33 Lookup Table a (LUT0 - LUT63)

30.7.2.33.1 Offset

For a = 0 to 63:

Register	Offset
LUTa	200h + (a × 4h)

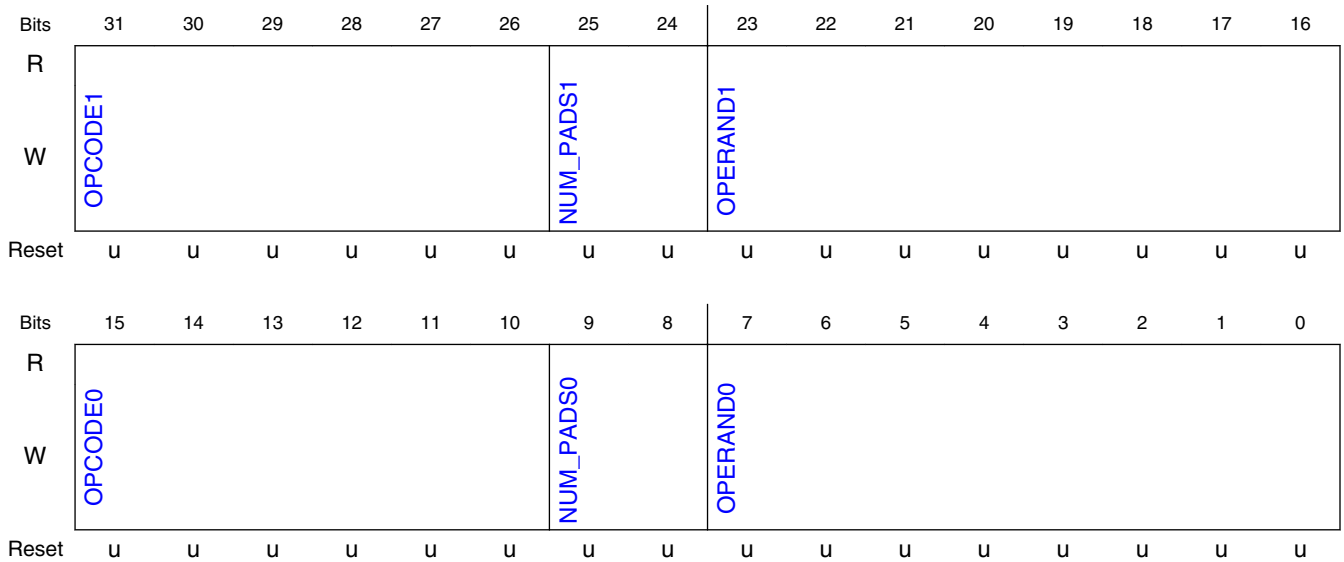
30.7.2.33.2 Function

Contains a lookup table for command sequences. Software should set the sequence index before triggering an IP command or AHB command. FlexSPI fetches the command sequence from LUT when an IP or AHB command is triggered. There are 16 command sequences in LUT. See Lookup table (LUT) and Programmable sequence engine.

NOTE

LUT is implemented as memory, so the reset value is unknown.

30.7.2.33.3 Diagram



30.7.2.33.4 Fields

Field	Function
31-26 OPCODE1	OPCODE1
25-24 NUM_PADS1	NUM_PADS1
23-16 OPERAND1	OPERAND1
15-10 OPCODE0	OPCODE
9-8 NUM_PADS0	NUM_PADS0
7-0 OPERAND0	OPERAND0

30.7.2.34 AHB Controller ID 0 Control (HMSTR0CR)

30.7.2.34.1 Offset

Register	Offset
HMSTR0CR	400h

30.7.2.34.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

30.7.2.34.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.35 AHB Controller ID 1 Control (HMSTR1CR)

30.7.2.35.1 Offset

Register	Offset
HMSTR1CR	404h

30.7.2.35.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

30.7.2.35.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.36 AHB Controller ID 2 Control (HMSTR2CR)

30.7.2.36.1 Offset

Register	Offset
HMSTR2CR	408h

30.7.2.36.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1

30.7.2.36.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.37 AHB Controller ID 3 Control (HMSTR3CR)

30.7.2.37.1 Offset

Register	Offset
HMSTR3CR	40Ch

30.7.2.37.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

30.7.2.37.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.38 AHB Controller ID 4 Control (HMSTR4CR)

30.7.2.38.1 Offset

Register	Offset
HMSTR4CR	410h

30.7.2.38.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1

30.7.2.38.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.39 AHB Controller ID 5 Control (HMSTR5CR)

30.7.2.39.1 Offset

Register	Offset
HMSTR5CR	414h

30.7.2.39.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1

30.7.2.39.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.40 AHB Controller ID n Control (HMSTR6CR - HMSTR7CR)

30.7.2.40.1 Offset

Register	Offset
HMSTR6CR	418h
HMSTR7CR	41Ch

30.7.2.40.2 Function

Contains the expected controller ID and the ID mask.

30.7.2.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSTRID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.40.4 Fields

Field	Function
31-16 MSTRID	Controller ID Contains the controller ID. This field is compared with the value of MASTER_ID ANDed with the MASK field. When there is a bit-by-bit match, the values are considered a match.
15-0 MASK	Mask bits for AHB Controller ID. Contains the mask for the AHB controller ID. This field is ANDed with the incoming controller ID on AHB user bits. 0000_0000_0000_0000b - Mask 0000_0000_0000_0001b - Unmask

30.7.2.41 HADDR REMAP Start Address (HADDRSTART)

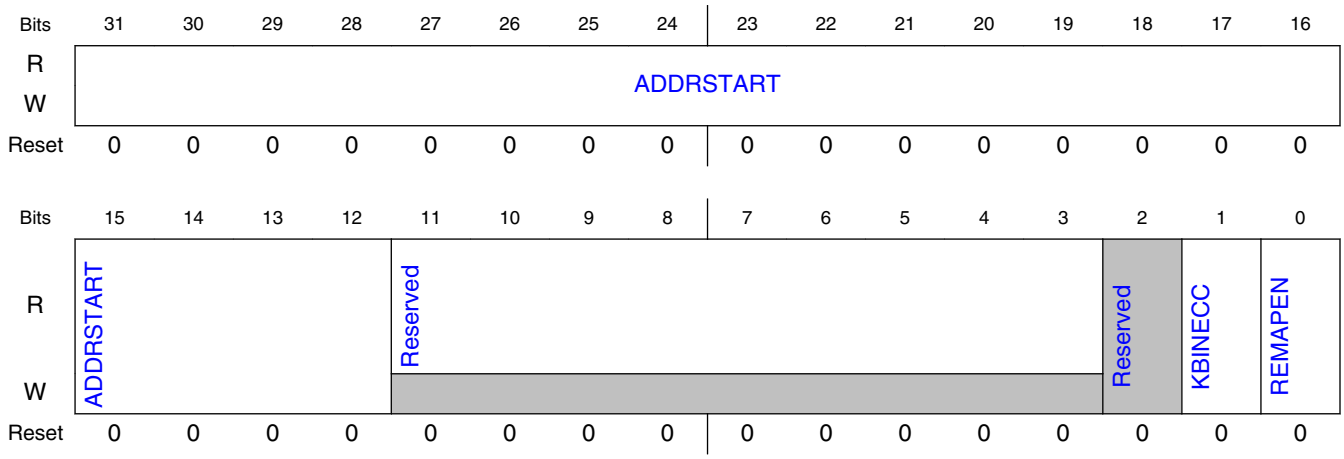
30.7.2.41.1 Offset

Register	Offset
HADDRSTART	420h

30.7.2.41.2 Function

Used to configure options for the dual-image remap feature. See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

30.7.2.41.3 Diagram



30.7.2.41.4 Fields

Field	Function
31-12 ADDRSTART	HADDR Start Address Contains the start address of the HADDR remap range, aligned to 4 KB. **NOTE:** When ADDRSTART setting is same as ASFM_BASE, and OTFAD keyblob function is enabled, keyblob is also remapped. Remapping location is based on the value of KBINECC.
11-3 —	Reserved
2 —	Reserved
1 KBINECC	OTFAD Keyblob in ECC When the keyblob is in the ECC region and must be remapped, indicates where FlexSPI fetches the OTFAD keyblob. 0b - FlexSPI fetches keyblob at base address + offset 1b - FlexSPI fetches keyblob at base address + offset × 2
0 REMAPEN	AHB Bus Address Remap Enable 0b - HADDR REMAP Disabled 1b - HADDR REMAP Enabled

30.7.2.42 HADDR REMAP END ADDR (HADDREND)

30.7.2.42.1 Offset

Register	Offset
HADDREND	424h

30.7.2.42.2 Function

Used to configure options for the dual-image remap feature. See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

30.7.2.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ENDSTART															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENDSTART				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

30.7.2.42.4 Fields

Field	Function
31-12 ENDSTART	End Address of HADDR Remap Range Contains the end address of the HADDR remap range, aligned to 4 KB.
11-0 —	Reserved

30.7.2.43 HADDR Remap Offset (HADDROFFSET)

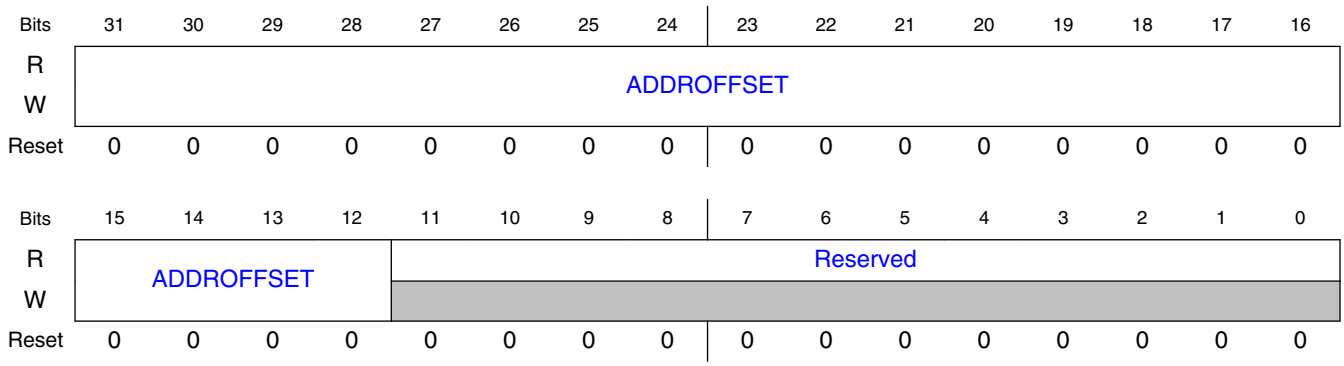
30.7.2.43.1 Offset

Register	Offset
HADDROFFSET	428h

30.7.2.43.2 Function

Used to configure options for the dual-image remap feature. See Dual image use case using HADDRSTART, HADDREND, and HADDROFFSET registers.

30.7.2.43.3 Diagram



30.7.2.43.4 Fields

Field	Function
31-12 ADDROFFSET	HADDR Offset Contains the offset for HADDR. The remapped address is ADDR[31:12] = ADDR_original[31:12] + ADDROFFSET.
11-0 —	Reserved

30.8 AHB memory map definition

This section describes FlexSPI module AHB memory map in detail.

30.8.1 AHB memory map for serial flash memory and RAM access

AHB read and write access for serial flash and RAM memory are mapped to a specific address range. See the system memory map for specific address ranges supported.

AHB bus features supported for serial flash and RAM memory reading:

- Cacheable and non-cacheable access
- Prefetch enable and disable
- Burst size: 8, 16, 32, 64 bits
- All burst types: types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

AHB bus features for Serial RAM writing:

- Bufferable and Non-Bufferable access
- Burst size: 8, 16, 32, 64 bits
- All burst types: SINGLE, INCR, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16

See Flash memory access via AHB command for details about AHB access to serial flash memory and RAM.

30.8.2 AHB Memory Map for IP RX FIFO read access

See chip-specific section for the address range mapped for AHB read access.

AHB Bus feature supported for IP RX FIFO reading:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Reading data from IP receive FIFO for more details about IP RX FIFO reading.

30.8.3 AHB Memory Map for IP TX FIFO write access

See chip-specific section for the address range mapped for AHB write access.

AHB Bus feature supported for IP TX FIFO writing:

- Burst size: 8/16/32/64 bits
- All burst type: SINGLE/INCR/WRAP4/INCR4/WRAP8/INCR8/WRAP16/INCR16

Refer Writing data to IP transmit FIFO for more details about IP TX FIFO filling.

Chapter 31
External ECC Controller (XECC)

31.1 Chip-specific XECC information

Table 31-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

31.2 Overview

The External ECC Controller (XECC) can be used as a gasket to provide ECC function for external memories. External memory address space can be divided into ECC regions and non-ECC regions, where there can be up to four ECC regions. XECC supports Single Error Correction and Double Error Detection (SEC-DED) ECC functions to provide reliability for external memory ECC region accesses. The XECC generates a 4-bit ECC code by Hsiao Hamming algorithm for every 4 bits in 32 bits data. 32 bits data is extended to 64 bits (32 bits data followed by 32 bits ECC code) before be written into external memory.

31.2.1 Block diagram

The following diagram shows 3 kinds of possible data flows for AXI access:

- When ECC is disabled, NIC301 and RECC/WECC modules are bypassed. AXI accesses directly pass through XECC.
- When ECC is enabled and AXI access address is out of all the ECC regions, AXI accesses pass through NIC301 between m_0 port and s_0 port and RECC/WECC is bypassed.
- When ECC is enabled and AXI access address hits one of the defined ECC regions, AXI accesses enter the WECC/RECC channel for ECC encode and decode.

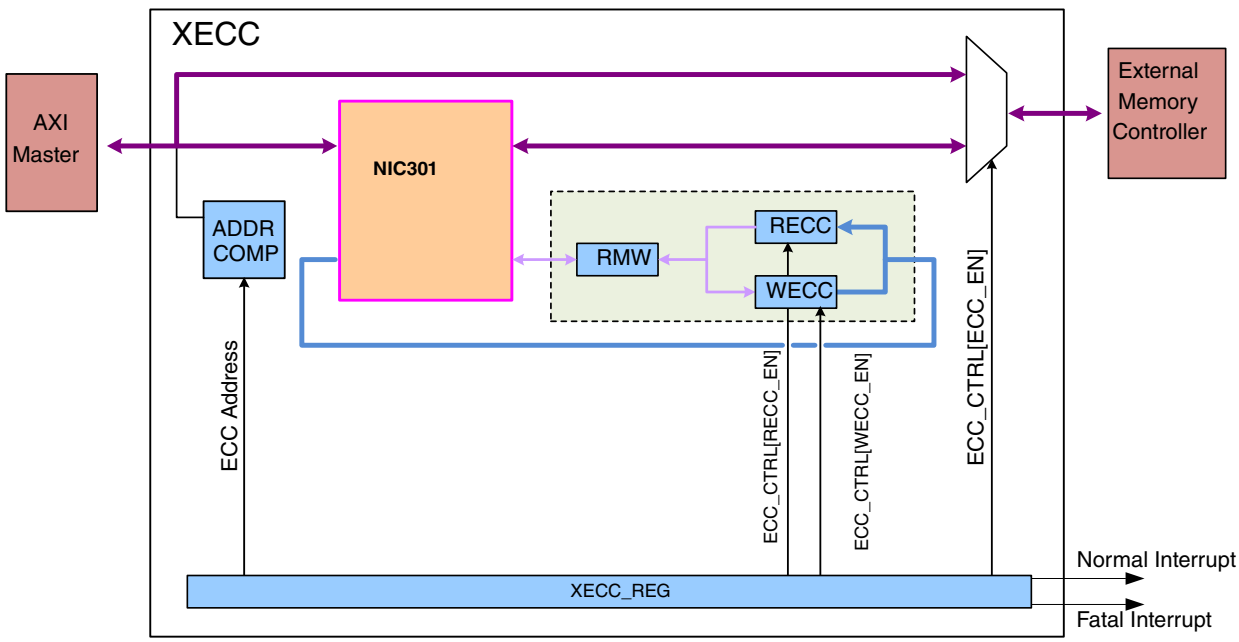


Figure 31-1. XECC block diagram

31.2.2 Features

The XECC includes the following features:

- Supports four configurable ECC regions for external memory space
- Supports SEC-DED ECC on every 4 bits in 32 bits data
- Supports error injection on data and ECC code
- ECC encode and decode functions can be disabled separately
- Separated interrupts for single bit error and multiple bits error

31.3 Functional description

The XECC can provide ECC function for the external memory ECC region. When ECC is off, XECC is totally transparent. When ECC is on, the number of total clock periods needed to pass the XECC module varies. Different AXI bursts have different conditions and pass times. For one AXI burst, the XECC waits for all transfers of this burst, then estimates if this burst is aligned or not. If it is not aligned, then the XECC reads memory data back, and combines them together to do the ECC calculation. The ECC calculation time is transparent.

31.3.1 ECC region

The XECC can provide four ECC region allocation for external memory space. The ECC regions address ranges are assigned by ECC_BASE_ADDRn and ECC_END_ADDRn registers (n=0,1,2,3). When ECC is enabled, the XECC checks if the AXI access address hits one of 4 ECC regions first. If the AXI access address is outside of ECC region, the AXI access directly passes through NIC301 and no ECC operation is executed. Otherwise, the AXI access enters the WECC and RECC channel for ECC encode and decode.

31.3.2 ECC encode

When ECC_CTRL[WECC_EN] and ECC_CTRL[ECC_EN] are set and AXI write address hits ECC regions, write data is encoded. The XECC generates 4 bits ECC code by Hsiao Hamming algorithm for every 4 bits in 32 bits data. After encode, 32 bits data is extended to 64 bits (32 bits data following by 32 bits ECC code) before a write into external memory. Therefore, there is address mapping from AXI master port to AXI slave port when write ECC is enabled. Address space should be doubled for ECC region in external memory.

If the write data is less than 32 bits, the XECC first executes a read-modify-write operation to extend the write data to 32 bits. The XECC then generates 32 bits ECC code. The resulting data written to external memory is 64 bits (32 bits data and 32 bits ECC code).

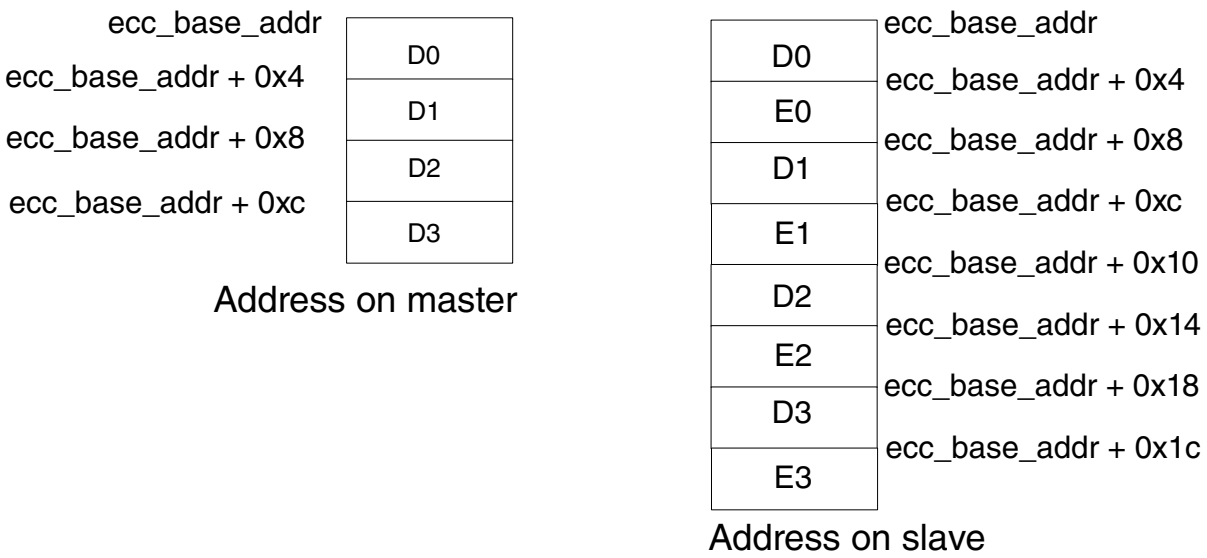


Figure 31-2. Address mapping after ECC encode

31.3.3 Error injection

XECC can inject error in write data and ECC code by configuring ERR_DATA_INJ and ERR_ECC_INJ registers. The 32 bits write data and 32 bits ECC code can be inverted if corresponding bits in ERR_DATA_INJ and ERR_ECC_INJ registers are set.

ERR_ECC_INJ registers are used for verification to test the ECC function. ERR_ECC_INJ registers keep injecting errors in write data, so bus error and interrupt continue to be received. For real cases (cases where ERR_ECC_INJ registers are not used for testing), bus error and interrupt only occur on real error data.

31.3.4 ECC decode

When ECC_CTRL[RECC_EN] and ECC_CTRL[ECC_EN] are set and AXI read address hits ECC regions, 32 bit read access must be extended to 64 bit read access to read data and ECC code back together. Therefore, there is also address mapping from AXI master port to AXI slave port when read ECC is enabled. After read address mapping, 64 bits read data from external memory enters into the RECC module and is decoded. The higher 32 bits in 64 bits read data are used as ECC code to decode low 32 bits in 64 bits read data. It can correct one bit error or detect 2 or more bits error for every 4 bits in low 32 bits read data. At the same time, error information is stored into XECC error registers for debug. Since 32 bits data have eight 4-bits fields, it is possible to report single bit error

and multiple bits error for different 4-bits field in one 32 bits data at same time. SINGLE_ERR_BIT_FIELD and MULTI_ERR_BIT_FIELD registers indicate which 4-bits fields are corrected or have multiple bits error.

NOTE

The XECC returns an error on the internal bus if an uncorrectable multi-bit error is detected.

31.3.5 ECC encode and decode bypass

To bypass all the ECC functions, SW should clear ECC_CTRL[ECC_EN]. To bypass either the ECC encode or decode, SW can clear ECC_CTRL[WECC_EN] or ECC_CTRL[RECC_EN]. If either the ECC encode or decode is bypassed, the same function should be implemented in external memory controller.

31.3.6 Data swap

The XECC can provide one bit correction for every 4 bits in 32 bits data. Though the sum of correctable bits can be up to 8 in 32 bits data, it cannot correct if 8 bits error are all in a single byte of 32 bits data. After the ECC_CTRL[SWAP_EN] bit is set, every bit in one byte of 32 bits data can be dispersed to different 4-bit fields before ECC encoding. That way, even if 8 bits in a single byte are all wrong, the XECC can still correct them.

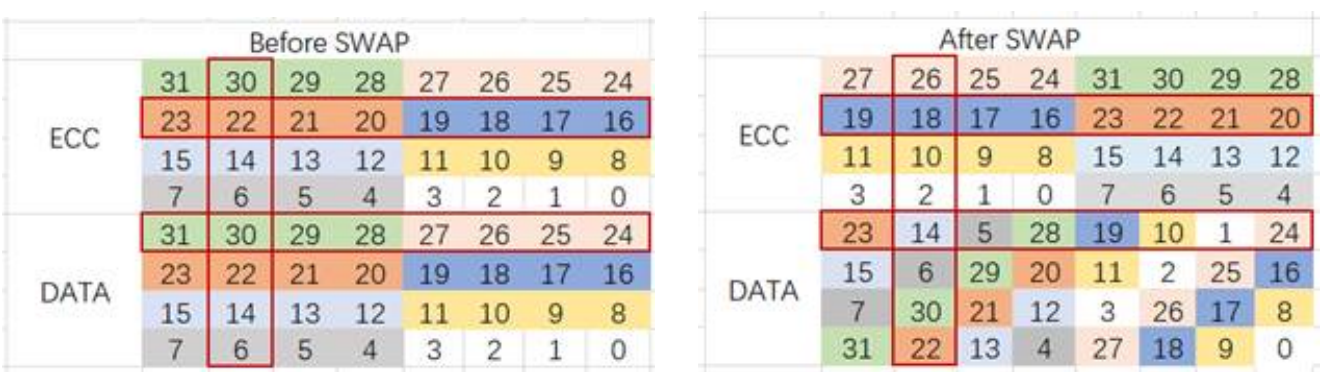


Figure 31-3. XECC Data Swap

In the figure above, the image on the left is ECC and data before swap, and the right image is ECC and data after swap. Every 4 bits of data has 4 bits ECC data. For example, write data [31:28] has its corresponding ECC data [31:28], indicated as green in the before-swap image. Corresponding post-swap data and ECC are marked as the same color in the after-swap image. The XECC can detect and correct the error if only one bit

error occurs in 8 bits (4 bits data + 4 bits ECC), no matter if the error occurs in data or ECC. The XECC can detect the error but cannot correct it if two bits of error or more occur in 8 bits.

In most situations, error bits in memory are more likely to be all in one row or one column like the marked red boxes indicated above.

If error bits happen in one column before swap, there are 2 bits error in 8 bits. But after swap, these errors could be dispersed to different 4-bit fields.

If error bits happen in one row and in the data area before swap, there are 4 bits error in 8 bits. But after swap, they could also be dispersed to different fields.

If error bits happen at one row and at ECC area, the situation is the same. Data is correct and only ECC is wrong in this case. Data is not changed. A fatal interrupt (fatal_int) is asserted for this error.

31.3.7 Clocks

There are two clocks in the XECC module:

Clock	Description
clk	Main XECC clock
ipg_clk	Used to read/write registers

See clocking chapter for this device for more information.

31.3.8 Interrupts

XECC can generate 2 kinds of interrupts:

- Normal interrupts - A normal interrupt can be generated when a single bit error is corrected for read.
- Fatal interrupts - A fatal interrupt can be generated when multiple bits error is detected for read. If fatal error interrupts are not needed, they can be disabled by changing ERR_SIG_EN[MULT_ERR_SIG_EN]=0. When this is disabled, only bus error is sent.

31.4 External Signals

The XECC does not have external signals.

31.5 Initialization

The following registers need to be configured to provide ECC function for external memories:

1. Configure ECC Region registers (ECC_BASE_ADDR*x*, ECC_END_ADDR*x*)
2. Configure ECC Control register (ECC_CTRL[ECC_EN]) to enable ECC

After steps 1 and 2 above are complete, ECC is enabled and AXI bursts can be sent to XECC

NOTE

The code to enable XECC should be placed in internal RAM to avoid bus hung.

31.6 Memory Map and register definition

This section includes the XECC module memory map and detailed descriptions of all registers.

31.6.1 XECC register descriptions

31.6.1.1 XECC memory map

XECC_FLEXSPI1 base address: 4001_C000h

XECC_FLEXSPI2 base address: 4002_0000h

XECC_SEMC base address: 4002_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	ECC Control Register (ECC_CTRL)	32	Table 31-1	0000_0000h
4h	Error Interrupt Status Register (ERR_STATUS)	32	Table 31-1	0000_0000h
8h	Error Interrupt Status Enable Register (ERR_STAT_EN)	32	Table 31-1	0000_0000h
Ch	Error Interrupt Enable Register (ERR_SIG_EN)	32	Table 31-1	0000_0000h
10h	Error Injection On Write Data (ERR_DATA_INJ)	32	RW	0000_0000h
14h	Error Injection On ECC Code of Write Data (ERR_ECC_INJ)	32	RW	0000_0000h
18h	Single Error Address (SINGLE_ERR_ADDR)	32	RO	0000_0000h
1Ch	Single Error Read Data (SINGLE_ERR_DATA)	32	RO	0000_0000h
20h	Single Error ECC Code (SINGLE_ERR_ECC)	32	RO	0000_0000h
24h	Single Error Bit Position (SINGLE_ERR_POS)	32	RO	0000_0000h
28h	Single Error Bit Field (SINGLE_ERR_BIT_FIELD)	32	RO	0000_0000h
2Ch	Multiple Error Address (MULTI_ERR_ADDR)	32	RO	0000_0000h
30h	Multiple Error Read Data (MULTI_ERR_DATA)	32	RO	0000_0000h
34h	Multiple Error ECC code (MULTI_ERR_ECC)	32	RO	0000_0000h
38h	Multiple Error Bit Field (MULTI_ERR_BIT_FIELD)	32	RO	0000_0000h
3Ch	ECC Region 0 Base Address (ECC_BASE_ADDR0)	32	RW	0000_0000h
40h	ECC Region 0 End Address (ECC_END_ADDR0)	32	RW	0000_0000h
44h	ECC Region 1 Base Address (ECC_BASE_ADDR1)	32	RW	0000_0000h
48h	ECC Region 1 End Address (ECC_END_ADDR1)	32	RW	0000_0000h
4Ch	ECC Region 2 Base Address (ECC_BASE_ADDR2)	32	RW	0000_0000h
50h	ECC Region 2 End Address (ECC_END_ADDR2)	32	RW	0000_0000h
54h	ECC Region 3 Base Address (ECC_BASE_ADDR3)	32	RW	0000_0000h
58h	ECC Region 3 End Address (ECC_END_ADDR3)	32	RW	0000_0000h

31.6.1.2 ECC Control Register (ECC_CTRL)

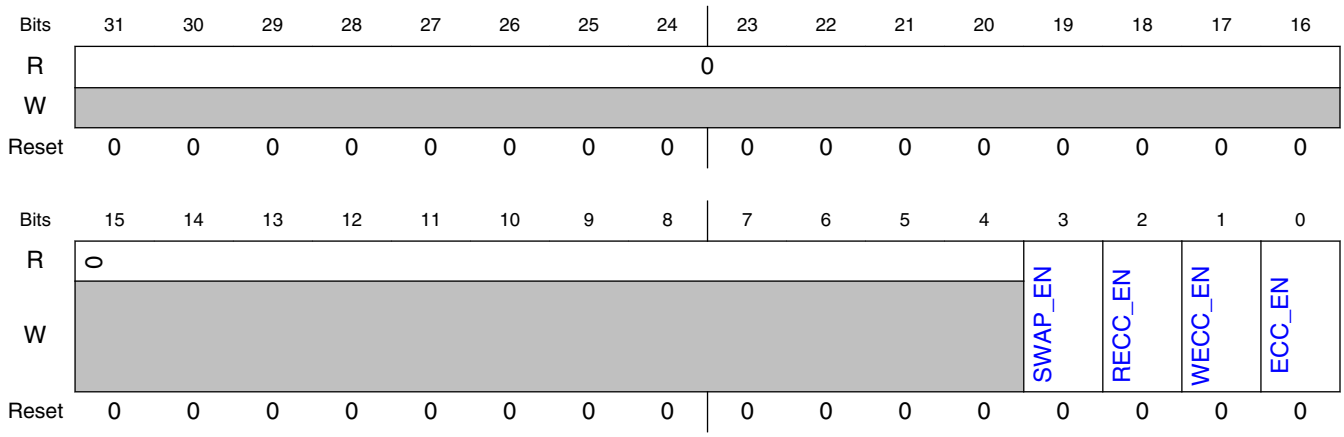
31.6.1.2.1 Offset

Register	Offset
ECC_CTRL	0h

31.6.1.2.2 Function

ECC function enable

31.6.1.2.3 Diagram



31.6.1.2.4 Fields

Field	Function
31-4 —	Reserved
3 SWAP_EN	Swap Data Enable Swap data for ECC encode and decode. This register bit needs to be set with ECC_EN. 0b - Disable 1b - Enable
2 RECC_EN	Read ECC Function Enable Enable ECC decode function for read. This register bit needs to be set with ECC_EN. 0b - Disable 1b - Enable
1 WECC_EN	Write ECC Encode Function Enable Enable ECC encode function for write. This register bit needs to be set with ECC_EN. 0b - Disable 1b - Enable
0 ECC_EN	ECC Function Enable Enable ECC function. 0b - Disable 1b - Enable

31.6.1.3 Error Interrupt Status Register (ERR_STATUS)

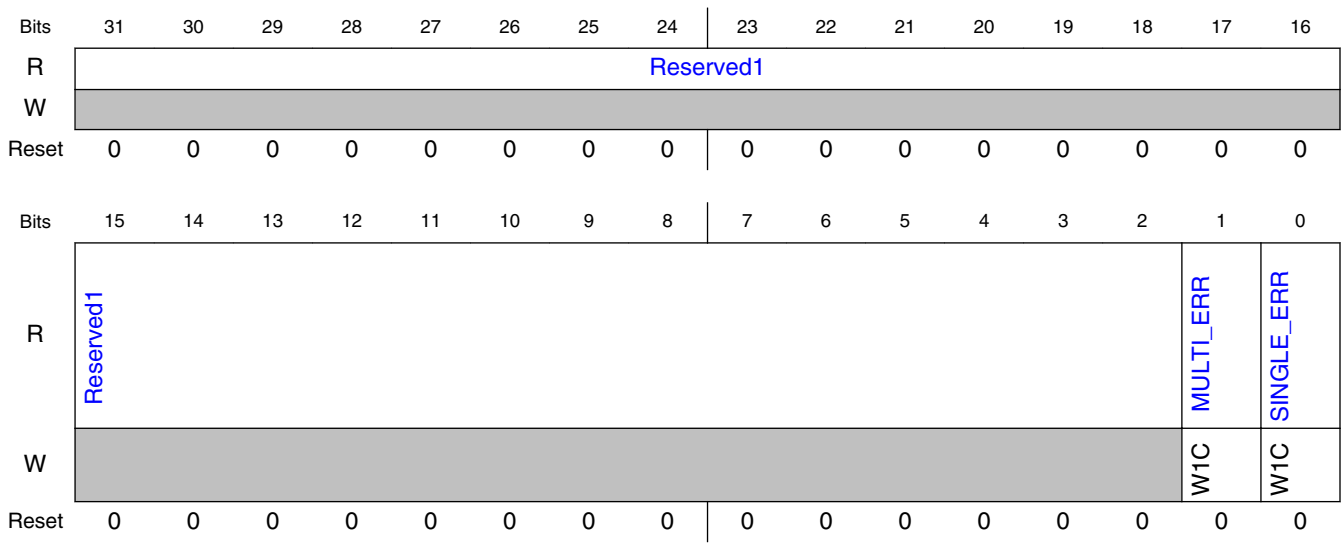
31.6.1.3.1 Offset

Register	Offset
ERR_STATUS	4h

31.6.1.3.2 Function

The Error Interrupt Status Register reports if a single bit error or multiple bits error occurs.

31.6.1.3.3 Diagram



31.6.1.3.4 Fields

Field	Function
31-2 Reserved1	Reserved
1 MULTI_ERR	Multiple Bits Error When a multiple bits error is detected on read data, this bit is asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0b - Multiple bits error does not happen. 1b - Multiple bits error happens.
0 SINGLE_ERR	Single Bit Error When detect single bit error on read data, this bit is asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0b - Single bit error does not happen. 1b - Single bit error happens.

31.6.1.4 Error Interrupt Status Enable Register (ERR_STAT_EN)

31.6.1.4.1 Offset

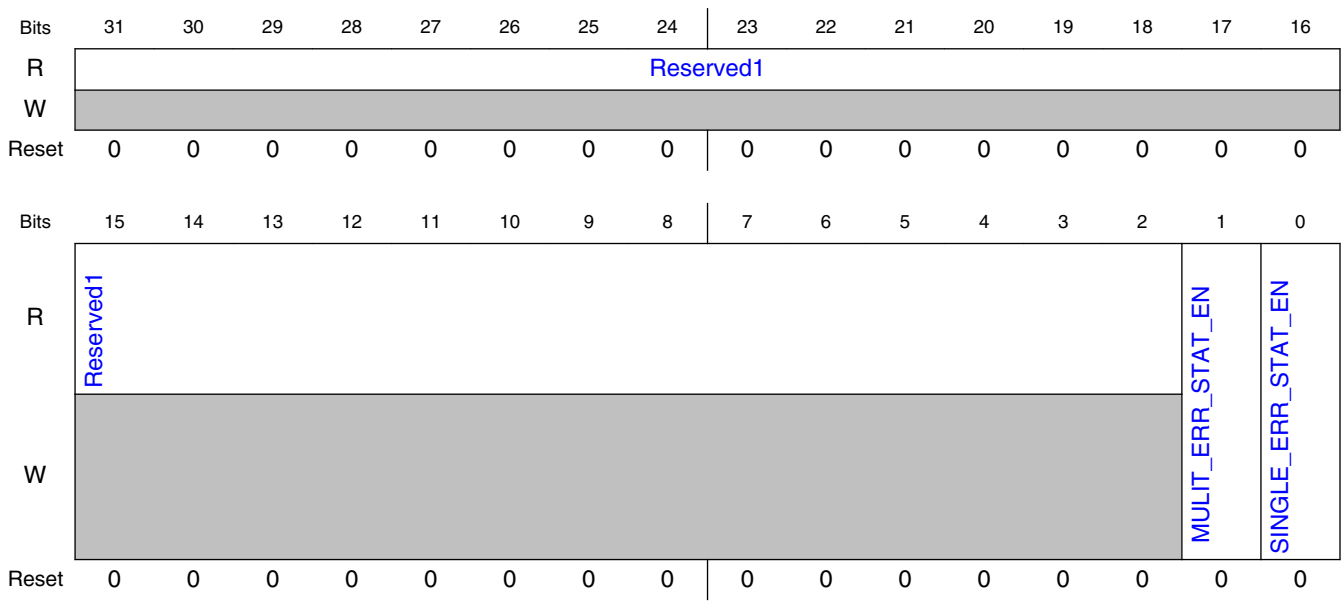
Register	Offset
ERR_STAT_EN	8h

31.6.1.4.2 Function

Setting the bits in this register to 1 enables the corresponding Error Interrupt Status to be set by the specified event.

Both ERR_SIG_EN and ERR_STAT_EN bitfields need to be set in order to generate interrupts.

31.6.1.4.3 Diagram



31.6.1.4.4 Fields

Field	Function
31-2 Reserved1	Reserved

Table continues on the next page...

Field	Function
1 MULIT_ERR_STAT_EN	Multiple Bits Error Status Enable 0b - Masked 1b - Enabled
0 SINGLE_ERR_STAT_EN	Single Bit Error Status Enable 0b - Masked 1b - Enabled

31.6.1.5 Error Interrupt Enable Register (ERR_SIG_EN)

31.6.1.5.1 Offset

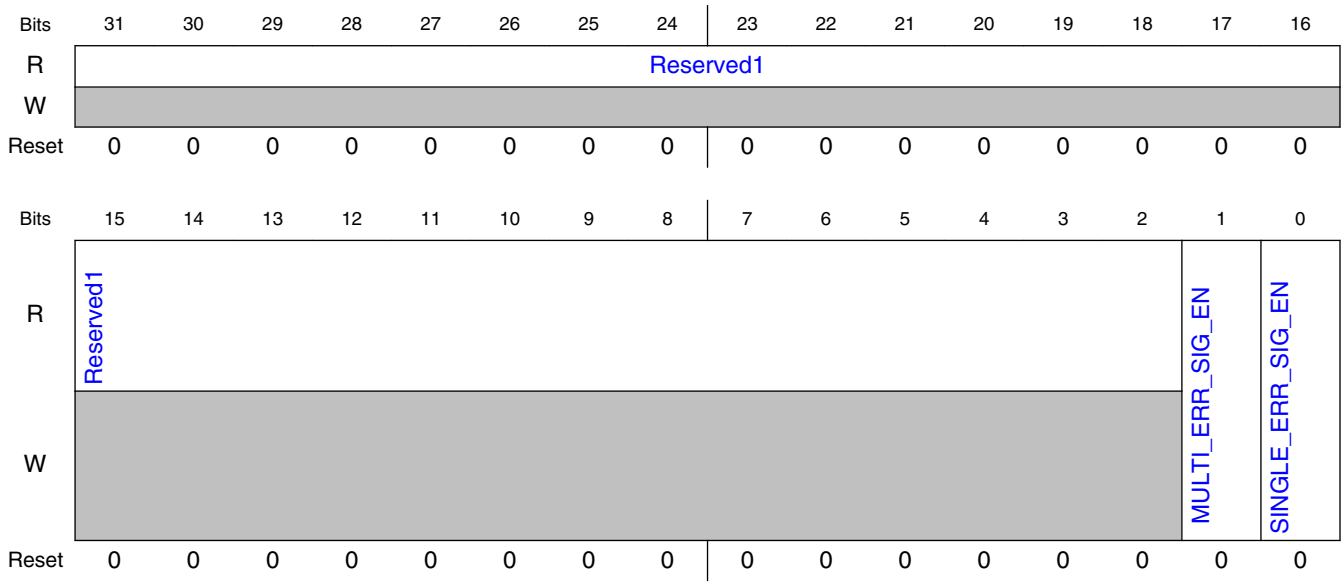
Register	Offset
ERR_SIG_EN	Ch

31.6.1.5.2 Function

This register is used to select which interrupt status is indicated to the Host System as the interrupt. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit generates an interrupt when the corresponding interrupt signal enable bit is set.

Both ERR_SIG_EN and ERR_STAT_EN bitfields need to be set in order to generate interrupts.

31.6.1.5.3 Diagram



31.6.1.5.4 Fields

Field	Function
31-2 Reserved1	Reserved
1 MULTI_ERR_SIG_EN	Multiple Bits Error Interrupt Enable 0b - Masked 1b - Enabled
0 SINGLE_ERR_SIG_EN	Single Bit Error Interrupt Enable 0b - Masked 1b - Enabled

31.6.1.6 Error Injection On Write Data (ERR_DATA_INJ)

31.6.1.6.1 Offset

Register	Offset
ERR_DATA_INJ	10h

31.6.1.6.2 Function

Write data is inverted if corresponding bits in this register are set.

31.6.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.6.4 Fields

Field	Function
31-0 ERR_DATA_INJ	Error Injection On Write Data Write data is inverted if corresponding bits in this register are set.

31.6.1.7 Error Injection On ECC Code of Write Data (ERR_ECC_INJ)

31.6.1.7.1 Offset

Register	Offset
ERR_ECC_INJ	14h

31.6.1.7.2 Function

ECC code of write data is inverted if corresponding bits in this register are set.

31.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_ECC_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_ECC_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.7.4 Fields

Field	Function
31-0 ERR_ECC_INJ	Error Injection On ECC Code of Write Data ECC code of write data is inverted if corresponding bits in this register are set.

31.6.1.8 Single Error Address (SINGLE_ERR_ADDR)

31.6.1.8.1 Offset

Register	Offset
SINGLE_ERR_ADDR	18h

31.6.1.8.2 Function

When a single bit error is corrected in an AXI read access, this stores the AXI read address. This register stores the address for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[SINGLE_ERR] flag is cleared and then another error occurs.

31.6.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.8.4 Fields

Field	Function
31-0 SINGLE_ERR_ADDR	Single Error Address When a single bit error is corrected in an AXI read access, this store the AXI read address.

31.6.1.9 Single Error Read Data (SINGLE_ERR_DATA)

31.6.1.9.1 Offset

Register	Offset
SINGLE_ERR_DATA	1Ch

31.6.1.9.2 Function

When a single bit error is corrected in an AXI read access, this stores the uncorrected read data. This register stores the read data for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[SINGLE_ERR] flag is cleared and then another error occurs.

31.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.9.4 Fields

Field	Function
31-0 SINGLE_ERR_DATA	Single Error Read Data When a single bit error is corrected in an AXI read access, this stores uncorrected read data.

31.6.1.10 Single Error ECC Code (SINGLE_ERR_ECC)

31.6.1.10.1 Offset

Register	Offset
SINGLE_ERR_ECC	20h

31.6.1.10.2 Function

When a single bit error is corrected in an AXI read access, this store the read ECC code. This register stores the ECC code for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[SINGLE_ERR] flag is cleared and then another error occurs.

31.6.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_ECC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_ECC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.10.4 Fields

Field	Function
31-0 SINGLE_ERR_ECC	Single Error ECC code When a single bit error is corrected in an AXI read access, this stores the read ECC code.

31.6.1.11 Single Error Bit Position (SINGLE_ERR_POS)

31.6.1.11.1 Offset

Register	Offset
SINGLE_ERR_POS	24h

31.6.1.11.2 Function

When a single bit error is corrected in an AXI read access, this indicates which bit is corrected in an AXI read data. This register stores the error bit position for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[SINGLE_ERR] flag is cleared and then another error occurs.

31.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.11.4 Fields

Field	Function
31-0 SINGLE_ERR_POS	Single Error bit Position When a single bit error is corrected in an AXI read access, this indicates which bit is corrected in an AXI read data.

31.6.1.12 Single Error Bit Field (SINGLE_ERR_BIT_FIELD)

31.6.1.12.1 Offset

Register	Offset
SINGLE_ERR_BIT_FIELD	28h

31.6.1.12.2 Function

When a single bit error is corrected in an AXI read access, SINGLE_ERR_BIT_FIELD indicates which 4-bit field is corrected in the read data. This register stores the error bit field for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[SINGLE_ERR] flag is cleared and then another error occurs.

31.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved1								SINGLE_ERR_BIT_FIELD							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.12.4 Fields

Field	Function
31-8 Reserved1	Reserved
7-0 SINGLE_ERR_BIT_FIELD	Single Error Bit Field When a single bit error is corrected in an AXI read access, this bit field indicates which 4-bit field is corrected in the read data. If ECC_CTRL[SWAP_EN] is ON, This bit field indicates which 4-bit field has a single error in the read data which has been swapped. If ECC_CTRL[SWAP_EN] is OFF, this bit field indicates which 4-bit field has a single error in the read data with original order.

31.6.1.13 Multiple Error Address (MULTI_ERR_ADDR)

31.6.1.13.1 Offset

Register	Offset
MULTI_ERR_ADDR	2Ch

31.6.1.13.2 Function

When a multiple bits error is detected in an AXI read access, this stores the AXI read address. This register stores the address for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[MULTI_ERR] flag is cleared and then another error occurs.

31.6.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.13.4 Fields

Field	Function
31-0 MULTI_ERR_ADDR	Multiple Error Address When a multiple bits error is detected in an AXI read access, this stores the AXI read address.

31.6.1.14 Multiple Error Read Data (MULTI_ERR_DATA)

31.6.1.14.1 Offset

Register	Offset
MULTI_ERR_DATA	30h

31.6.1.14.2 Function

When a multiple bits error is detected in an AXI read access, this stores the error read data. This register stores the read data for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[MULTI_ERR] flag is cleared and then another error occurs.

31.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.14.4 Fields

Field	Function
31-0 MULTI_ERR_DATA	Multiple Error Read Data When a multiple bits error is detected in an AXI read access, this stores the error read data.

31.6.1.15 Multiple Error ECC code (MULTI_ERR_ECC)

31.6.1.15.1 Offset

Register	Offset
MULTI_ERR_ECC	34h

31.6.1.15.2 Function

When a multiple bits error is detected in an AXI read access, this stores the ECC code of the read data. This register stores the ECC code for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[MULTI_ERR] flag is cleared and then another error occurs.

31.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MULTI_ERR_ECC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MULTI_ERR_ECC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.15.4 Fields

Field	Function
31-0 MULTI_ERR_ECC	Multiple Error ECC code When a multiple bits error is detected in an AXI read access, this stores the ECC code of read data.

31.6.1.16 Multiple Error Bit Field (MULTI_ERR_BIT_FIELD)

31.6.1.16.1 Offset

Register	Offset
MULTI_ERR_BIT_FIELD	38h

31.6.1.16.2 Function

When a multiple bits error is detected in an AXI read access, MULTI_ERR_BIT_FIELD indicates which 4-bit field has a multiple bits error in the read data. This register stores the error bit field for the first error that happens. The register does not update again if additional errors occur until the ERR_STATUS[MULTI_ERR] flag is cleared and then another error occurs.

31.6.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved1								MULTI_ERR_BIT_FIELD							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.16.4 Fields

Field	Function
31-8 Reserved1	Reserved
7-0 MULTI_ERR_BIT_FIELD	Multiple Error Bit Field When a multiple bits error is detected in an AXI read access, this bit field indicates which 4-bit field has a multiple bits error in the read data. If ECC_CTRL[SWAP_EN] is ON, this bit field indicates which 4-bit field has a multiple bits error in the read data which has been swapped. If ECC_CTRL[SWAP_EN] is OFF, this bit field indicates which 4-bit field has a multiple bits error in the read data with original order.

31.6.1.17 ECC Region 0 Base Address (ECC_BASE_ADDR0)

31.6.1.17.1 Offset

Register	Offset
ECC_BASE_ADDR0	3Ch

31.6.1.17.2 Function

Base address for ECC region 0.

31.6.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_BASE_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.17.4 Fields

Field	Function
31-0 ECC_BASE_ADDR0	ECC Region 0 Base Address Base address for ECC region 0. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_BASE_ADDR0 register sets the number of 4K increments, the real addr is: 4K*ECC_BASE_ADDR0 For example, if ECC_BASE_ADDR0 = 0 and ECC_END_ADDR0 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.18 ECC Region 0 End Address (ECC_END_ADDR0)

31.6.1.18.1 Offset

Register	Offset
ECC_END_ADDR0	40h

31.6.1.18.2 Function

End address for ECC region 0.

31.6.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_END_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_END_ADDR0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.18.4 Fields

Field	Function
31-0 ECC_END_ADDR0	ECC Region 0 End Address End address for ECC region 0. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_END_ADDR0 register sets the number of 4K increments, the real addr is: 4K*ECC_END_ADDR0 For example, if ECC_BASE_ADDR0 = 0 and ECC_END_ADDR0 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.19 ECC Region 1 Base Address (ECC_BASE_ADDR1)

31.6.1.19.1 Offset

Register	Offset
ECC_BASE_ADDR1	44h

31.6.1.19.2 Function

Base address for ECC region 1.

31.6.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_BASE_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.19.4 Fields

Field	Function
31-0 ECC_BASE_ADDR1	ECC Region 1 Base Address Base address for ECC region 1. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_BASE_ADDR1 register sets the number of 4K increments, the real addr is: 4K*ECC_BASE_ADDR1 For example, if ECC_BASE_ADDR1 = 0 and ECC_END_ADDR1 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.20 ECC Region 1 End Address (ECC_END_ADDR1)

31.6.1.20.1 Offset

Register	Offset
ECC_END_ADDR1	48h

31.6.1.20.2 Function

End address for ECC region 1.

31.6.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_END_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_END_ADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.20.4 Fields

Field	Function
31-0 ECC_END_ADDR1	ECC Region 1 End Address End address for ECC region 1. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_END_ADDR1 register sets the number of 4K increments, the real addr is: 4K*ECC_END_ADDR1 For example, if ECC_BASE_ADDR1 = 0 and ECC_END_ADDR1 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.21 ECC Region 2 Base Address (ECC_BASE_ADDR2)

31.6.1.21.1 Offset

Register	Offset
ECC_BASE_ADDR2	4Ch

31.6.1.21.2 Function

Base address for ECC region 2.

31.6.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_BASE_ADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_BASE_ADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.21.4 Fields

Field	Function
31-0 ECC_BASE_ADDR2	ECC Region 2 Base Address Base address for ECC region 2. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_BASE_ADDR2 register sets the number of 4K increments, the real addr is: 4K*ECC_BASE_ADDR2 For example, if ECC_BASE_ADDR2 = 0 and ECC_END_ADDR2 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.22 ECC Region 2 End Address (ECC_END_ADDR2)

31.6.1.22.1 Offset

Register	Offset
ECC_END_ADDR2	50h

31.6.1.22.2 Function

End address for ECC region 2.

31.6.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_END_ADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_END_ADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.22.4 Fields

Field	Function
31-0 ECC_END_ADDR2	ECC Region 2 End Address End address for ECC region 2. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_END_ADDR2 register sets the number of 4K increments, the real addr is: 4K*ECC_END_ADDR2 For example, if ECC_BASE_ADDR2 = 0 and ECC_END_ADDR2 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.23 ECC Region 3 Base Address (ECC_BASE_ADDR3)

31.6.1.23.1 Offset

Register	Offset
ECC_BASE_ADDR3	54h

31.6.1.23.2 Function

Base address for ECC region 3.

31.6.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_BASE_ADDR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_BASE_ADDR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.23.4 Fields

Field	Function
31-0 ECC_BASE_ADDR3	ECC Region 3 Base Address Base address for ECC region 3. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_BASE_ADDR3 register sets the number of 4K increments, the real addr is: 4K*ECC_BASE_ADDR3 For example, if ECC_BASE_ADDR3 = 0 and ECC_END_ADDR3 = 1, then the resulting ECC region is from 0 to 4K.

31.6.1.24 ECC Region 3 End Address (ECC_END_ADDR3)

31.6.1.24.1 Offset

Register	Offset
ECC_END_ADDR3	58h

31.6.1.24.2 Function

End address for ECC region 3.

31.6.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_END_ADDR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_END_ADDR3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

31.6.1.24.4 Fields

Field	Function
31-0 ECC_END_ADDR3	ECC Region 3 End Address End address for ECC region 3. The minimum ECC region range is 4k, so the lower 20 bits of this register must be 0. The ECC_END_ADDR3 register sets the number of 4K increments, the real addr is: 4K*ECC_END_ADDR3 For example, if ECC_BASE_ADDR3 = 0 and ECC_END_ADDR3 = 1, then the resulting ECC region is from 0 to 4K.

Chapter 32
Ultra Secured Digital Host Controller (uSDHC)

32.1 Chip-specific uSDHC information

Table 32-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

The 32KHz clock used for card detection is sourced from OSC_32K.

32.2 Overview

The Ultra Secured Digital Host Controller (uSDHC) provides the interface between the host system and the SD/SDIO/MMC cards, as shown in Figure 32-2. The module acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards. It handles the SD/SDIO/MMC protocols at the transmission level.

32.2.1 Block Diagram

The following figure illustrates the block diagram for uSDHC.

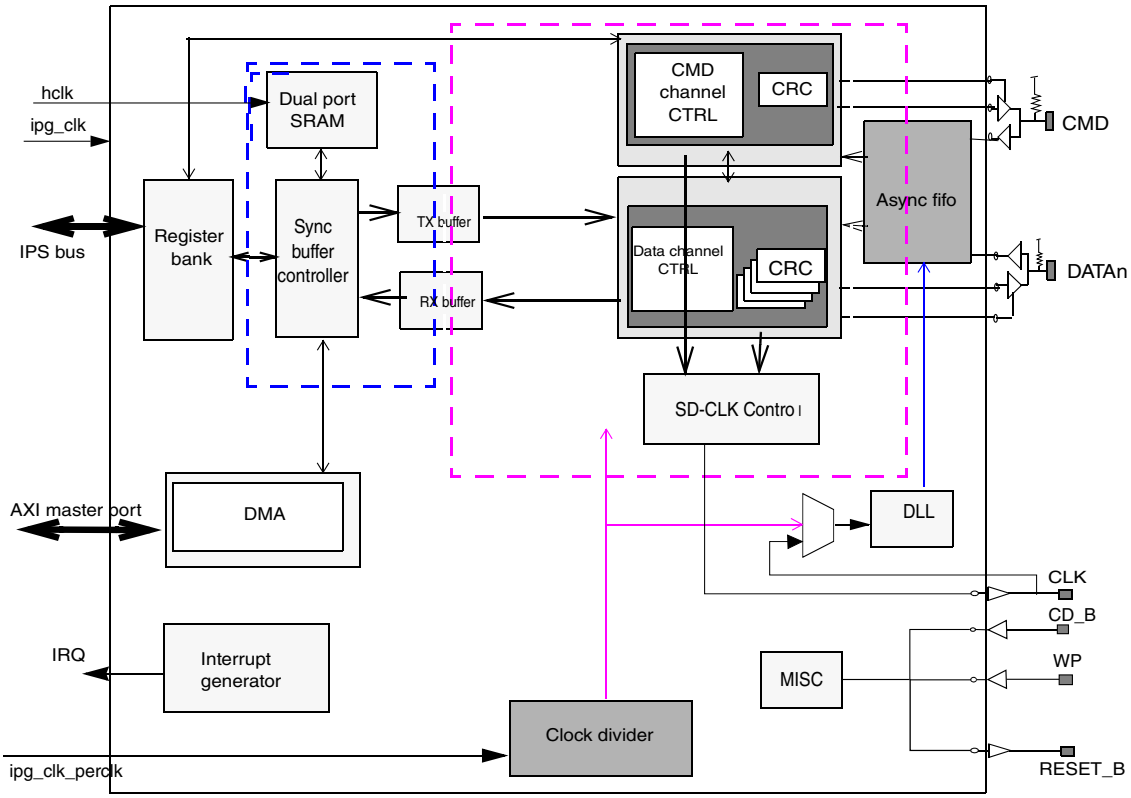


Figure 32-1. uSDHC block diagram

The figure below shows the System connection of uSDHC:

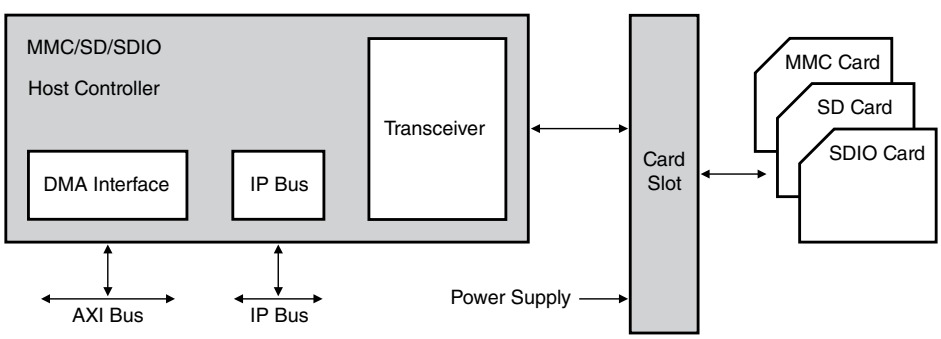


Figure 32-2. System connection of uSDHC

The following are brief descriptions of the cards supported by uSDHC:

- The Multi Media Card (MMC) is a universal low-cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous MMC cards were based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.

- The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are backward compatible with the old MMC (with some additions).
- Under the SD protocol, system connection can be categorized into memory card, I/O card, and combo card, which have both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card, which is also known as SDIO card, provides high-speed data I/O with low-power consumption for mobile electronic devices. For the sake of simplicity, the next figure does not show cards with reduced size or mini cards.

32.2.2 Features

The list given below shows the features of the uSDHC module:

- Conforms to the SD Host Controller Standard Specification version 2.0/3.0
- Compatible with the MMC System Specification version 4.2/4.3/4.4/4.41/4.5/5.0
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 2.0/3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards
- Card bus clock frequency up to 208 MHz
- Supports 1-bit/4-bit SD and SDIO modes, and 1-bit/4-bit/8-bit MMC modes
  - Up to 832 Mbps of data transfer for SDIO cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDIO card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 832 Mbps of data transfer for SDXC cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDXC card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 1600 Mbps of data transfer for MMC cards using eight parallel data lines in the Single Data Rate (SDR) mode
  - Up to 3200 Mbps of data transfer for MMC cards using eight parallel data lines in the Dual Data Rate (DDR) mode
- Supports single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations

- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes; also supports interrupt period
- Embodies a fully configurable 128x32-bit FIFO for read/write data
- Supports internal DMA capabilities
- Support voltage selection by configuring vendor-specific register bit
- Supports Advanced DMA to perform linked memory access

NOTE

This block can support all the above-listed speed mode and maximum data throughput. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

32.3 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA AXI interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

32.3.1 Modes and operations

32.3.1.1 Data transfer modes

The uSDHC module can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification mode (up to 400 kHz)
- MMC full-speed mode (up to 26 MHz)
- MMC high-speed mode (up to 52 MHz)

- MMC HS200 mode(up to 200 MHz)
- MMC HS400 mode (200 MHz both edges)
- MMC DDR mode (52 MHz both edges)
- SD/SDIO full-speed mode (up to 25 MHz)
- SD/SDIO high-speed mode (up to 50 MHz)
- SD/SDIO UHS-I mode(up to 208 MHz in SDR mode, up to 50 MHz in the DDR mode)

NOTE

This block can support all the above listed speed mode and maximum clock frequency. However, these may be specific to the device. See the corresponding chip-specific information or the device data sheet for accurate details.

32.3.2 Data buffer

The uSDHC module uses one configurable data buffer to transfer data between the system bus (IP bus or advanced extensible interface (AXI) bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the master clock).

The buffer is used as a temporary storage for transferring data between the host system and the card. The watermark levels for read and write are both configurable and can range between 1 to 128 words. When using internal DMA mode, The watermark level for read should be configured to 16 and watermark level for write should be configured to integral multiple of 16 to get max AXI bus throughput. The next figure provides the uSDHC buffer scheme.

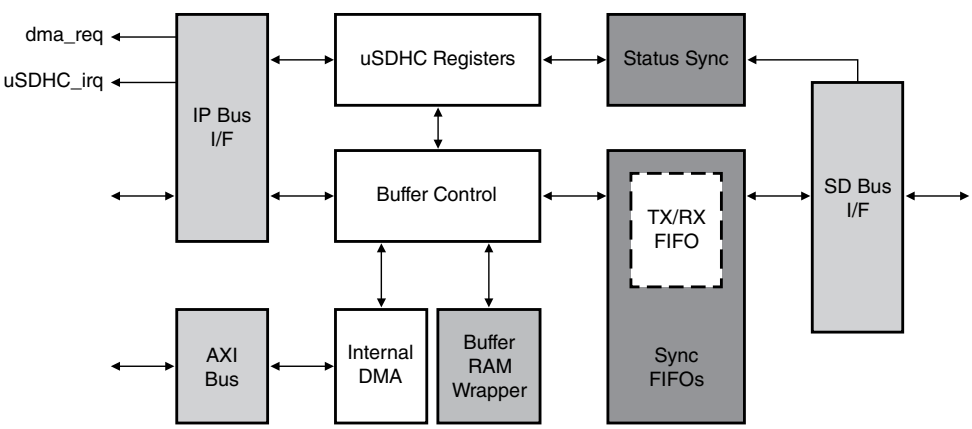


Figure 32-3. uSDHC buffer scheme

Here are 2 transfer modes to access the data buffer:

- CPU polling mode:
  - For a host-read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the host driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar. For more information on the process of writing operation, see Write operation sequence.
- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AXI bus.

For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AXI bus.

The Write operation functions in a similar manner—sequential and contiguous access is necessary to ensure that the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, the byte order is swapped after the data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, the byte order is swapped before the data is stored in the buffer.

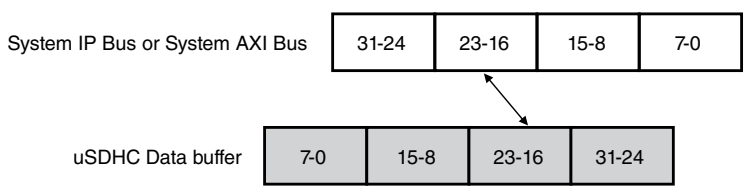


Figure 32-4. Data swap between system bus and uSDHC data buffer in the byte little endian mode

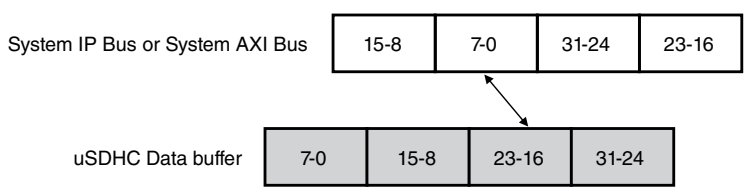


Figure 32-5. Data swap between system bus and uSDHC data buffer in half word big endian mode

32.3.2.1 Write operation sequence

There are 2 ways to write data into the buffer when the user transfers data to the card:

- Processor core polling through the BWR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, the DMAEN bit in the Transfer Type register is not set when the command is sent, uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WR_WML register and is ready for receiving new data. At the same time, uSDHC sets the BWR bit. The buffer write ready interrupt is generated if it is enabled by software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the write operation from that address. It is recommended that a software reset for Data be applied before the transfer is restarted.

The uSDHC module does not start data transmission until the buffer has been filled with the number of words set in the WR_WML register. If the buffer is empty and the host system does not write data in time, uSDHC stops the CLK to avoid the data buffer underrun situation.

32.3.2.2 Read operation sequence

There are 2 ways to read data from the buffer when the data is received from the card:

- Processor core polling through the BRR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, which is available and ready for system fetching data. At the same time, uSDHC sets the BRR bit. The buffer read ready interrupt is generated if it is enabled by the software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the content of the DMA System Address register to get the

starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the read operation from that address. It is recommended that a software reset (in register RSTD) for data be applied before the transfer is restarted.

For any write transfer mode, uSDHC does not start data transmission until the number of words set in the RD_WML register are in buffer. If the buffer is full and the host system does not read data in time, uSDHC stops the CLK to avoid the data buffer overrun situation.

32.3.2.3 Data buffer and block size

In the uSDHC module, the data buffer can hold up to 256 words (32-bit) and the watermark levels for write and read can be configured accordingly. The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way.

The host driver may configure the value according to the system and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes).

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned), stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the software side must write twice for each block. For each block, the ending byte is abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

32.3.2.4 Dividing large data transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in multiples of block size. If the total data length cannot be divided evenly into a multiple of the block size, then there are two options to transfer the data. These two options depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block

size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See the figure below for an example showing the division of large data transfers, assuming a kind of WLAN SDIO card that only supports a block size of up to 64 bytes. Although uSDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size of less than 64 bytes, so the data must be divided (see the example below).

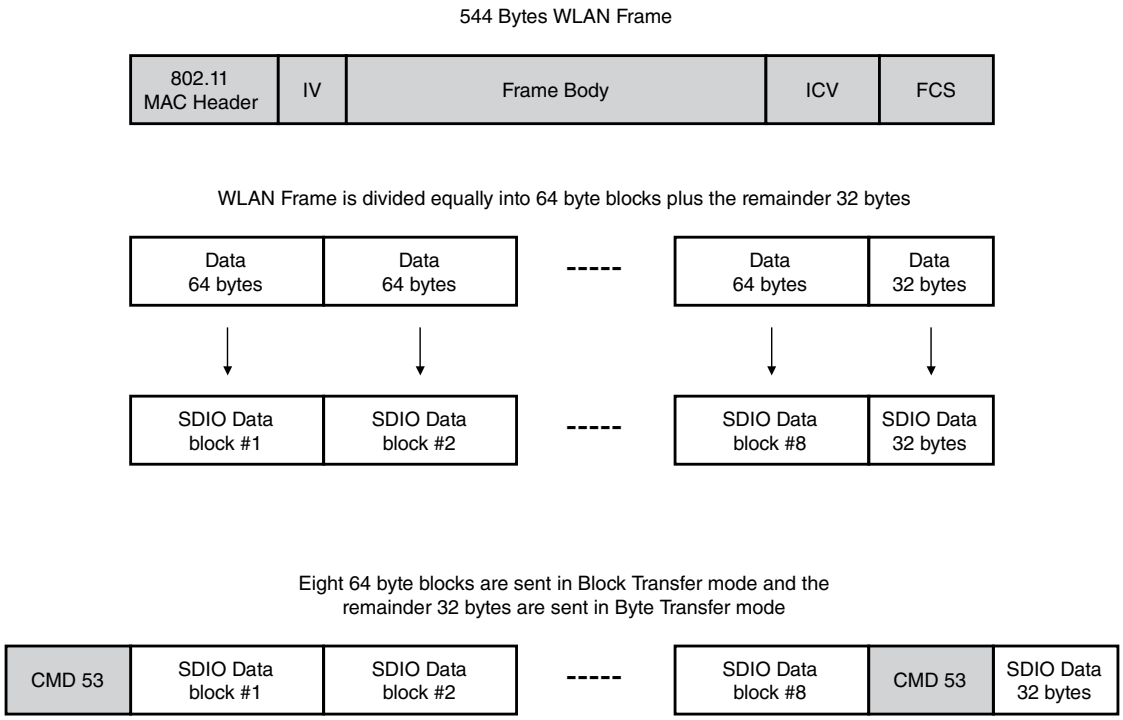


Figure 32-6. Example for dividing large data transfers

32.3.3 DMA AXI interface

The internal DMA implements a DMA engine and the AXI master. When the internal DMA is enabled, but the BWR and BRR bits are set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register.

See the figure below for an illustration of the DMA AXI interface block.

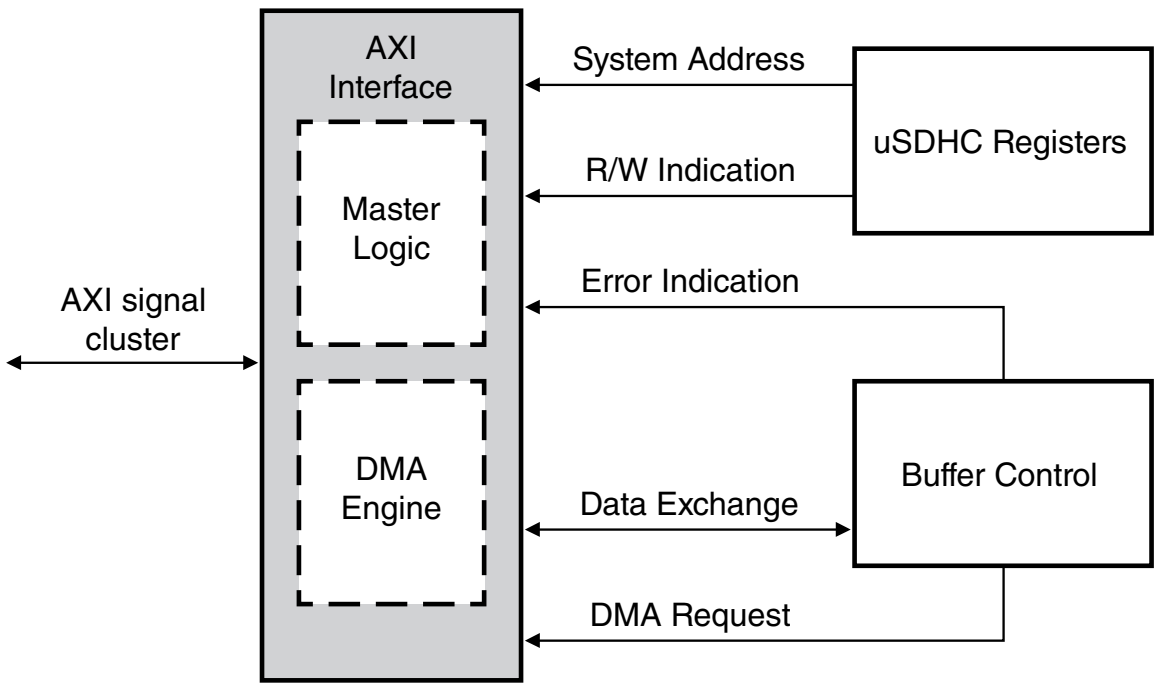


Figure 32-7. DMA AXI interface block

32.3.3.1 Internal DMA request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the data buffer block sends a DMA request to AXI interface.

The delay in response from the internal DMA engine depends on the system AXI bus loading and the priority assigned to uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request if the data buffer space (for write) or bytes in data buffer is smaller than the watermark level. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of the current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always automatically set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the watermark level set to six words (24 bytes). After the first burst transfer, if there are greater than or equal to two words in the buffer, which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to transfer for the current block is ceiling((31 - 6 * 4) / 4) = 2. The uSDHC module reads two words out of the buffer, with seven valid bytes and one stuffed byte.

32.3.3.2 AXI master interface

It is possible that the internal AXI DMA engine could fail during the data transfer. Upon detection of an AXI bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state. At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register is generated to host CPU to report a bus error condition.

After the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and restart the transfer from this address to recover the corrupted block.

32.3.3.3 ADMA engine

In the SD host controller standard, a new DMA transfer algorithm called the Advanced DMA (ADMA) is defined. For simple DMA, after the page boundary is reached, a DMA interrupt is generated and the new system address is programmed by the host driver.

The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention is not be needed during long DMA-based data transfers.

There are two types of ADMA in host controller: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory, and ADMA2 eliminates the restriction so that the data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors defined in SD host controller Standard, and if the "End" flag is detected in the descriptor, ADMA stops after this descriptor is processed.

32.3.3.3.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Rsv descriptor (new in ADMA2)
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA starts read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before the tran type descriptor. Every tran type descriptor triggers a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, the tran type descriptor contains both data length and transfer data address, so the tran type descriptor itself can start a data transfer.

See the figure below for the format of the descriptor table for ADMA1.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 32-8. Format of the ADMA1 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA1.

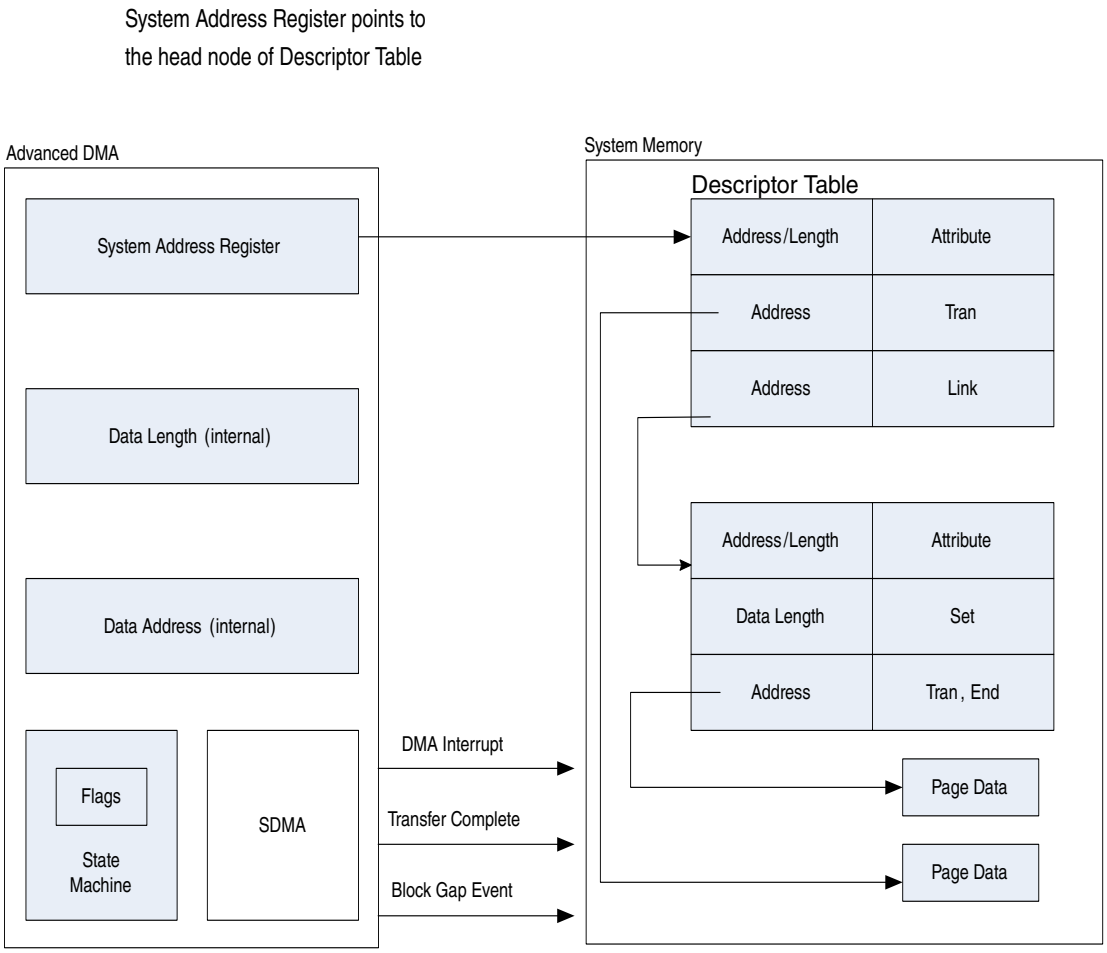


Figure 32-9. Concept and access method of the ADMA1 descriptor table

The figure below explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it ignores the higher 32-bit. The Address field should be set to word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 32-10. Format of the ADMA2 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA2.

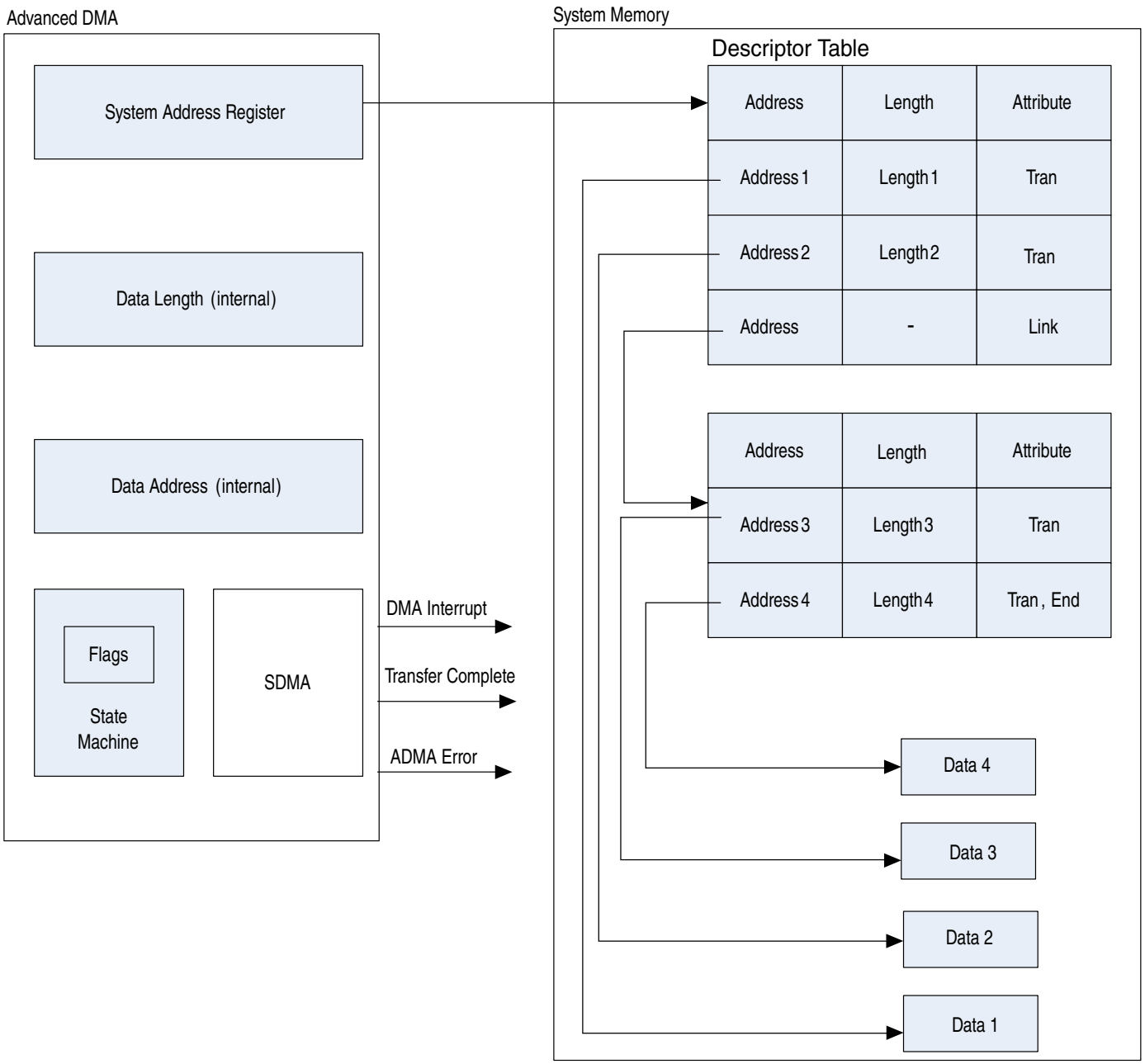


Figure 32-11. Concept and access method of the ADMA2 descriptor table

32.3.3.3.2 ADMA interrupt

If the interrupt flag descriptor is set, ADMA generates an interrupt according to various types of descriptors.

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

32.3.3.3.3 ADMA error

The ADMA stops whenever an error is encountered. These errors include:

- Fetching descriptor error
- AXI response error
- Data length mismatch error

An ADMA descriptor error is generated when it fails to detect a "valid" flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the "Interrupt" flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in register BLK_ATT must be equal to the whole data length set in descriptor, otherwise data length mismatch error is generated.

When BLKCNTEN bit is not set, then the whole data length set in descriptor is a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors occur.

32.3.4 Register bank with IP bus interface

Register accesses through the IP bus interface are on the register bank.

See the figure below for the block diagram.

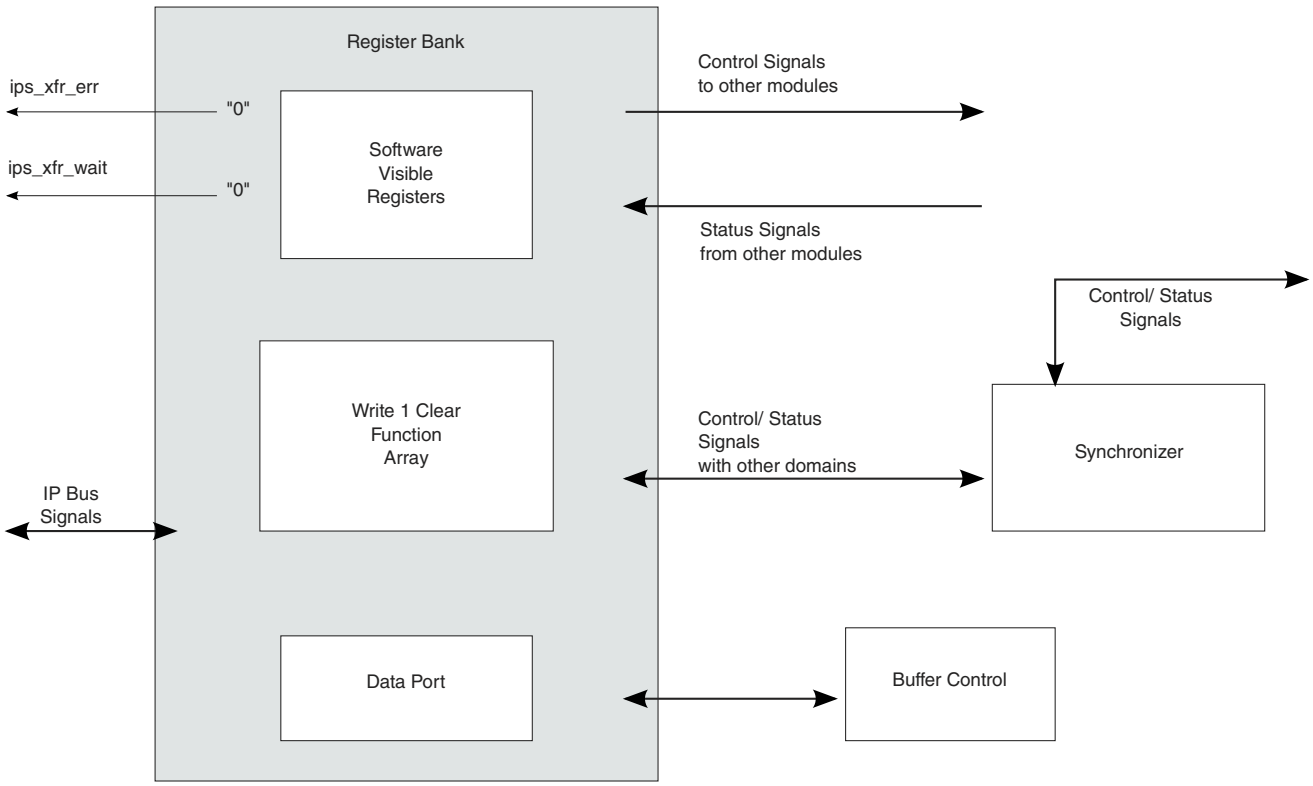


Figure 32-12. Register bank diagram

Only a 32-bit access is allowed, and no partial read/write is supported; therefore, all accesses are word aligned.

Access to an unimplemented address within the register memory map space does not generate a transfer error.

32.3.4.1 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DAT lines
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four submodules:

- SD control misc
- Command control
- Data control
- Clock control

32.3.4.2 SD control misc

In the SD control misc unit:

- The card detection (including CD_B and DATA3 for card detection) and card interrupt are implemented.
- The module monitors the signal level on all the eight data lines and the command lines. It directly routes the level values into the register bank.
- The module also detects the Write Protect (WP) line. If WP is active, writes to the register bank are ignored.

32.3.4.3 SD clock control

If the internal data buffer is near full (for read) or near empty (for write), the SD clock must be gated off to avoid buffer over/under-run. This module asserts the gate of the output SD clock to shut the clock off. After the buffer has space (for read) or has data (for write), the clock gate of this module opens, and the SD clock is active again.

32.3.4.4 Command control

The Command Control module deals with the transactions on the CMD line.

See the figure below for an illustration of the structure for the Command CRC shift register.

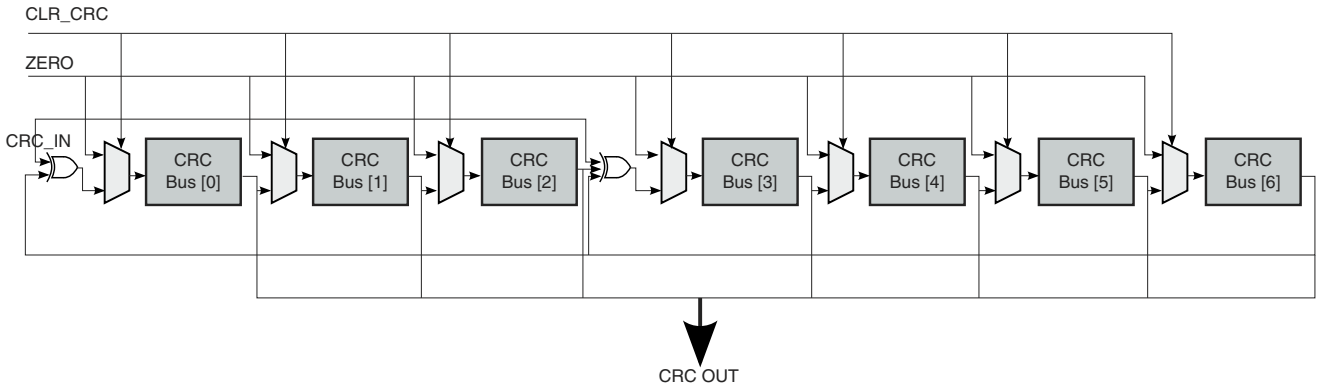


Figure 32-13. Command CRC shift register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

32.3.4.5 Data control

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line and generates the read wait state by the request from the transceiver.

The CRC polynomials for the data are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1

M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0

CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

32.3.5 Card insertion and removal detection

The uSDHC module uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DATA3 is pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the

CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, uSDHC sends an interrupt (if enabled) to inform the Host system that a card is inserted.

32.3.6 Power management and wakeup events

When there is no operation between uSDHC and the card through the SD bus, the user can completely disable the peripheral clock and base clock in the chip-level clock control module to save power. When the user needs to use uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to uSDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC module can generate these interrupts even when there are no clocks enabled. The three interrupts that can be used as wakeup events are these:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO card (not available in multiple block data transfers)

These three wakeup events (or wakeup interrupts) can also be used to wakeup the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to uSDHC are disabled or when the entire system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (PROT_CTRL) for more information on the uSDHC Protocol Control register.

32.3.6.1 Setting wakeup events

For uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters the sleep mode.

Before the software disables the host clock, it should ensure that all the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

32.3.7 MMC fast boot

The Embedded MultiMediaCard (eMMC4.3) specification adds a fast boot feature that requires hardware support. There are two types of fast boot modes in the eMMC4.3 specification: boot operation and alternative boot operation. Each type also has with-acknowledge and without-acknowledge modes.

In the boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

NOTE

For the eMMC4.3 card setting, please see the eMMC4.3 specification.

32.3.7.1 Boot operation

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after the CMD line goes low, the slave starts to send the first boot data to the master on the DATA line(s). The master must keep the CMD line low to read all of the boot data.

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes low. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode with the CMD line high.

The boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

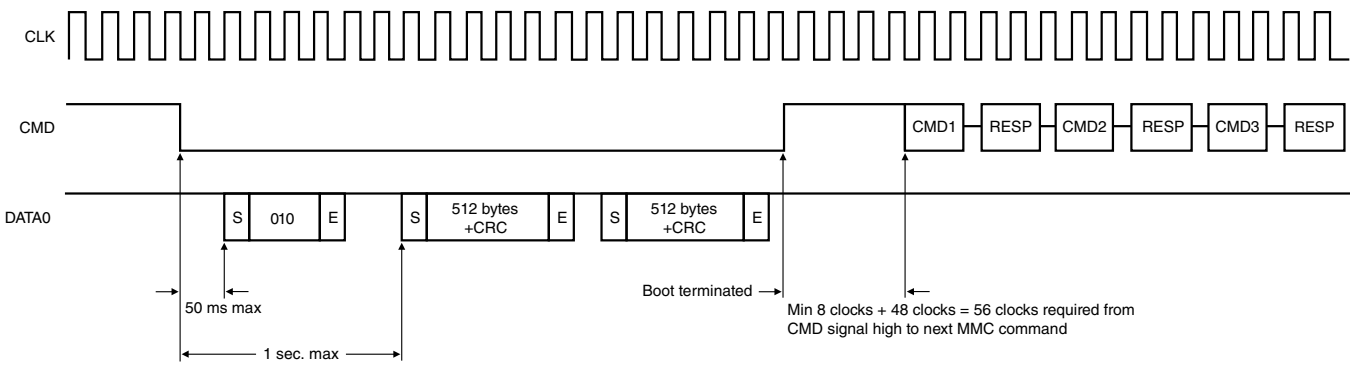


Figure 32-14. MultiMediaCard state diagram (normal boot mode)

32.3.7.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

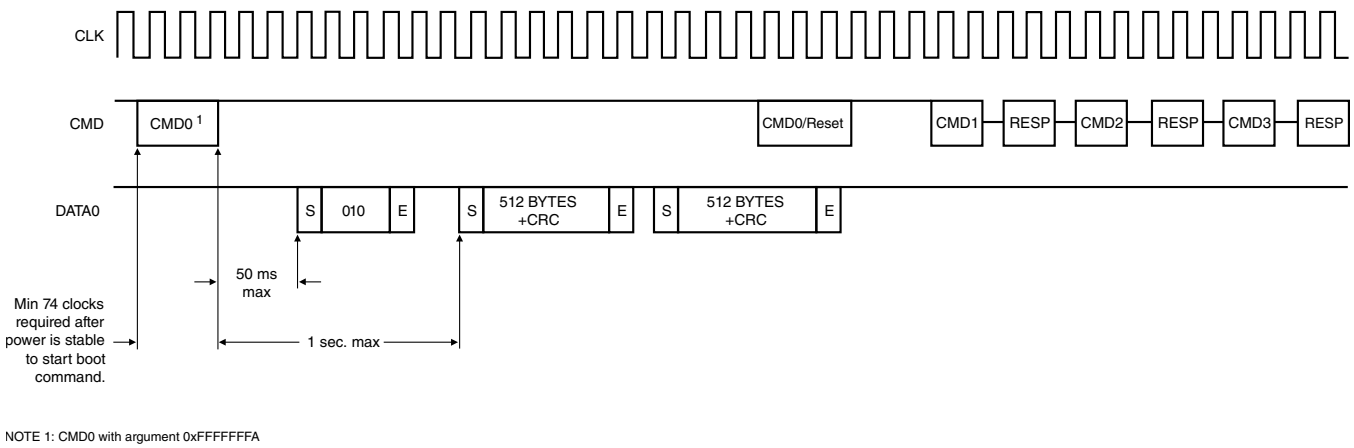
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DATA line(s).

If boot acknowledge is enabled, the slave must send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode by issuing CMD0 (Reset).

Boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE 1: CMD0 with argument 0xFFFFFFFA

Figure 32-15. MultiMediaCard state diagram (alternative boot mode)

32.3.8 Commands for MMC/SD/SDIO

A table containing the list of commands for the MMC/SD/SDIO cards is provided here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DATA
- Addressed (point-to-point) data transfer commands (adtc)

Response: A response is a token that is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Table 32-2. Commands for MMC/SD/SDIO cards

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.

Table continues on the next page...

Table 32-2. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send them operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.

Table continues on the next page...

Table 32-2. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] reserved bits(all 0)	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	128 clocks of tuning pattern (64 byte in 4 bit mode or 128 byte in 8 bit mode) is sent for HS200 optimal sampling point detection.
CMD22-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).

Table continues on the next page...

Table 32-2. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				

Table continues on the next page...

Table 32-2. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 is used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62-63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).

Table continues on the next page...

Table 32-2. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high-speed SD cards. Command SWITCH_FUNC is for high-speed SD cards.
3. Command SWITCH is for high speed MMC cards. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 2.
4. ACMDs is preceded with the APP_CMD command. Commands listed are used for SD only, other SD commands not listed are not supported on this module.

The access bits for the EXT_CSD access modes are listed in the following table.

Table 32-3. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd Set field of the argument.
01	Set bits	The bits in the pointed byte are set, according to the bits set to 1 in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the bits set to 1 in the Value field.
11	Write byte	The Value field is written into the pointed byte.

32.3.9 Clocks

The table found here describes the clock sources for uSDHC. For more information on clocking, see the Clocking chapter.

Table 32-4. Clocks

Clock name	Description
hclk	Bus clock
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock for register access
ipg_clk_perclk	Base clock

32.3.10 Clock and reset manager

This module controls all four kinds reset signals within uSDHC:

- Hardware reset
- Software reset for all logic
- Software reset for the data logic
- Software reset for the command logic

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

32.3.11 Clock generator

The clock generator generates the card CLK by peripheral source clock in two stages. The clock divisor can be configured through register SYS_CTRL [SDCLKFS] for prescaler configuration while the [DVS] for divisor configuration. Details can be found in the register function description. See the following figure for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

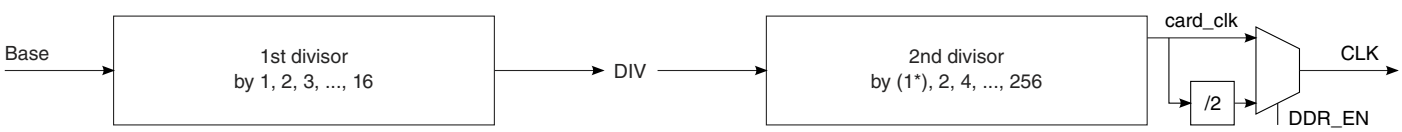


Figure 32-16. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV) that can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler and outputs the actual internal working clock (card_clk). This clock is the driving clock for all the sub modules of the SD protocol unit, and helps in syncing FIFOs (see Figure 32-3) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage can be DIV, DIV/2, DIV/4,..., or DIV/256. Therefore, the highest frequency of the card_clk is base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the duty cycle of the base clock is 50%, the duty cycle of card_clk is also 50%, even when the compound divisor is an odd value.

NOTE

CLK is different for the SDR and DDR modes.

- In the SDR mode, CLK is equal to the internal working clock (card_clk).

- In the DDR mode, CLK is equal to card_clk/2.

32.3.12 SDIO card interrupt

Information on interrupts in 1-bit mode, interrupts in 4-bit mode, and card interrupt handling are detailed in the sections below.

32.3.12.1 Interrupts in 1-bit mode

In this case, the DATA1 pin provides the interrupt function. An interrupt is asserted by pulling the DATA1 low from the SDIO card, until the interrupt service is finished to clear the interrupt.

32.3.12.2 Interrupt in 4-bit mode

As the interrupt and data line 1 share pin 8 in a 4-bit mode, an interrupt is only sent by the card and recognized by the host during a specific time. This is known as the interrupt period. The uSDHC module only provides samples the level on pin 8 during the interrupt period. At all other times, the host treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in a 4-bit mode, there is only a limited period of time that the interrupt period can be active because of the limited period of data line availability between the multiple blocks of data. This requires stricter definition of the interrupt period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line holds low for one clock cycle, then pulls high for the next clock cycle . On completion of the interrupt period, the card releases the DATA1 line into the high Z state. The uSDHC module provides sample of the DATA1 during the interrupt period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10 for further information about the SDIO card interrupt.

32.3.12.3 Card interrupt handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, uSDHC clears the interrupt request to the host system. The host driver should clear this bit before servicing the SDIO interrupt, and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Card Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from the SDIO card does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt source from the external card followed by writing 1 to this bit. In a 1-bit mode, uSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In a 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status is set and the host driver needs to service this interrupt, the SDIO bit in the Interrupt Control Register of SDIO card is cleared. This is required to clear the SDIO interrupt status latched in uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1 and uSDHC starts sampling the interrupt signal again.

See the figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

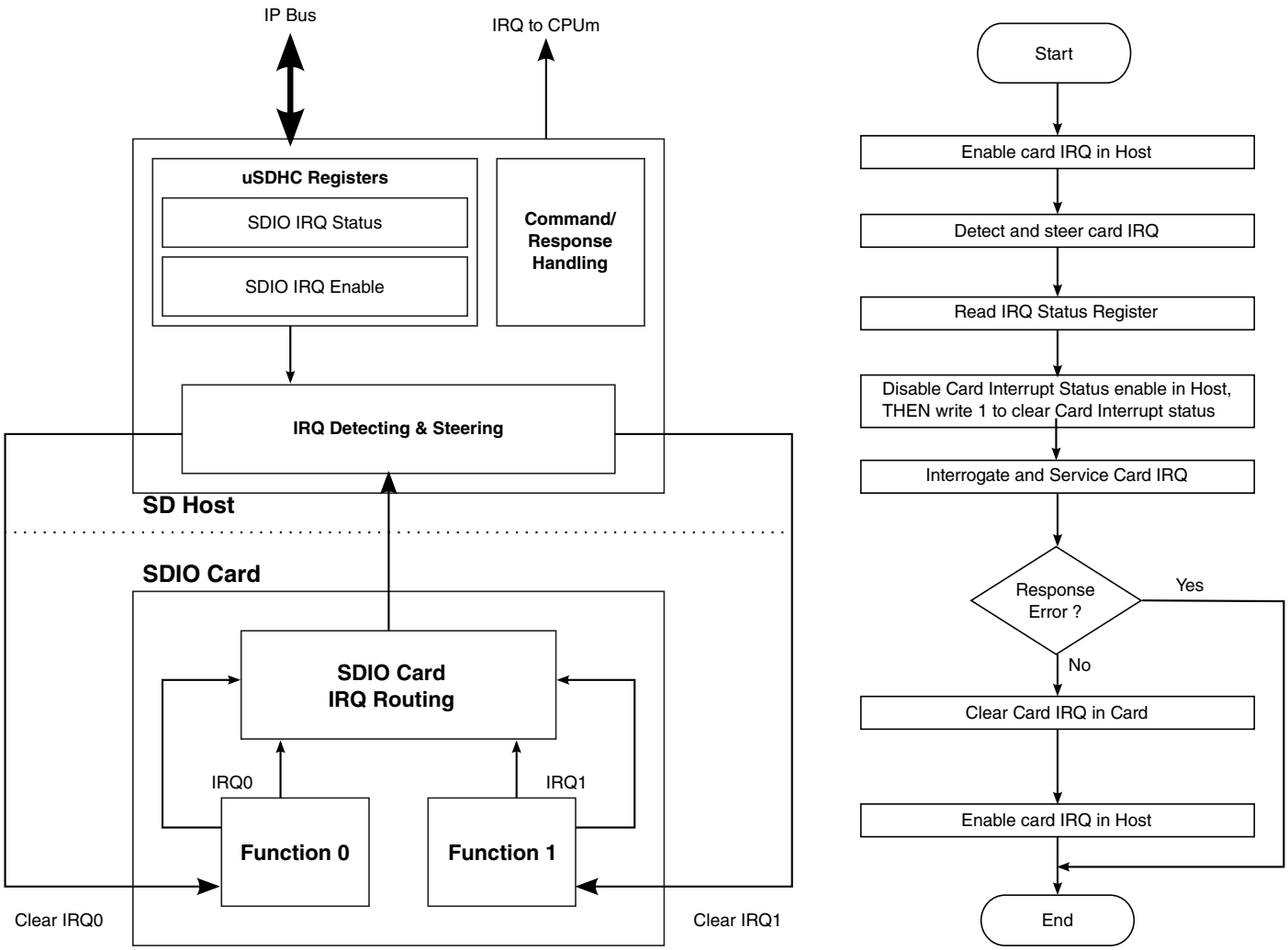


Figure 32-17. Card interrupt scheme, card interrupt detection, and handling procedure

32.3.13 Speed mode selection

To access card with card clock above 52 MHz, you must switch the timing mode to one of supported higher speed timing modes: “High Speed” for frequencies between 26 MHz and 52 MHz. HS200 mode or HS400 mode for frequencies above 52 MHz up to 200 MHz. Note that while the actual timing change is done, the behavior of any command given (such as CMD13) cannot be guaranteed due to the asynchronous operation. Therefore, you must not use CMD13 to check the busy completion of the timing change indication. In case, CMD13 is used, the host must ignore CRC errors, if appear. And, the max frequency for HS200 mode and HS400 mode could be slower than 200 MHz based on the chip implementation, you must confirm with SoC before setting the target frequency for these two speed modes. Following sections describe the mode selections for HS200 mode and HS400 mode in detail.

32.3.13.1 HS200 mode selection

To switch to HS200, the Driver must perform the following steps:

1. Select the chip (through sending CMD7) and make sure it is unlocked (through CMD42).
2. Read the DEVICE_TYPE [196] field of the Extended CSD register to validate if the chip supports HS200 at the IO voltage appropriate for both host and chip (through CMD8).
3. Read the DRIVER_STRENGTH [197] field of the Extended CSD register to find the supported chip Driver Strengths (through CMD8). The default setting is 50 Ohm. The Host Designer might simulate its specific system, using a device driver models. Host can select the optimal Driver Type that might drive the host system load at the required operating frequency with the minimal noise generated.

NOTE

This step can be skipped if changes of Driver strength is not needed.

4. Check the setting of PROT_CTRL[DTW], make sure the data transfer width is set as expected.

   Make sure the I/O pad voltage is 1.8V and corresponding drive strength is set to proper value.

   Set HS200 bit and Driver Strength value in the HS_TIMING [185] field of the Extended CSD register by issuing CMD6. If the host attempts to write an invalid value, the HS_TIMING byte is not changed, the HS200 interface timing is not enabled, the Driver Strength is not changed, and the SWITCH_ERROR field is set. After the chip responds with R1, it might assert Busy signal. Once the busy signal gets de-asserted, the host may send a SEND_STATUS Command (CMD13) using the HS200 timing and after it receives a Trans State indication and No Error it means that the chip is set to HS200 timing and the Driver Strength is set to the selected settings.

5. At this point, the host can set the frequency to ≤ 200 MHz.
6. The host might invoke the HS200 standard tuning procedure, by sending CMD21 to the chip.

NOTE

The host must switch to the required bus width before starting the tuning operation to allow the tuning sequence to be done using the proper bus operating conditions.

32.3.13.2 Standard Tuning Procedure

By default, lower frequency operation, a fixed sampling clock is used to receive signals on CMD and DAT[3:0]. Before using the HS200, HS400, SDR104, or SDR50 modes, the Host Driver executes the tuning procedure at the mode switch sequence.

1. Issue uSDHC SW reset, set SYS_CTRL[RSTT] to 1.
2. Set VEND_SPEC[FRC_SDCLK_ON] to 1.
3. Set TUNING_CTRL[DIS_CMD_CHK_FOR_STD_TUNING] to 1.
4. Start the tuning procedure by setting TUNING_CTRL[STD_TUNING_EN] and MIX_CTRL[EXE_TUNE] to 1.
5. Issue CMD19(SD)/ CMD21(eMMC) with the proper Command Transfer Type (CMD_XFR_TYP) and Mixer Control (MIX_CTRL) settings.
6. Wait for uSDHC BRR (Buffer Read Ready) interrupt signal is 1.
7. Check MIX_CTRL[EXE_TUNE]. If MIX_CTRL[EXE_TUNE] = 1, repeat 5~6. If MIX_CTRL[EXE_TUNE] = 0, standard tuning has completed, or the tuning has not completed within 40 attempts. The Host Driver might abort this loop if the number of loops exceeds 40 or 150ms timeout occurs. In this case, a fixed sampling clock should be used, (AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0).
8. Sampling Clock Select, AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] , is valid after MIX_CTRL[EXE_TUNE] has changed from 1 to 0. AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 1, indicates tuning procedure passed. AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] = 0, indicates tuning procedure failed. The tuning result is applied to the delay chain, CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) [30:16], upon successful tuning procedure completion.
9. Clear VEND_SPEC[FRC_SDCLK_ON].
10. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

While the tuning sequence is being performed, the Host Controller does not generate interrupts (including Command Complete), except Buffer Read Ready. CMD19 response errors are not indicated.

Writing AUTOCMD12_ERR_STATUS[SMP_CLK_SEL] to 0 forces the Host Controller to use a fixed sampling clock and resets the tuning circuit of the Host Controller.

NOTE

- There could be slight difference on delay cell delay in each project, and this difference can lead to different loop

number needed. TUNING_CTRL[TUNING_STEP] can be used to control how may taps are to be added for each step.
- Manual tuning might be required in cases where standard tuning is not able to close passing window with CMD19/21 transfer fail.

32.3.13.3 Manual tuning procedure

Manual tuning controls more detail in the tuning procedure, and some corner cases. Manual tuning provides control to tuning start point, tuning steps, and delay chains. To start manual tuning:

1. Set TUNING_CTRL[STD_TUNING_EN] to 0.
2. Set SYS_CTRL[RSTA] to reset all.
3. Set MIX_CTRL[SMP_CLK_SEL] and MIX_CTRL[EXE_TUNE] to start tuning.
4. Set VEND_SPEC[FRC_SDCLK_ON] to 1.
5. Config delay chain, CLK_TUNE_CTRL_STATUS[DLY_CELL_SET_PRE] with the valueA. The valueA must start from 0.
6. Send CMD19/21 to card.
7. Poll for CC,CIE,CEBE,CCE,CTOE of Interrupt Status (INT_STATUS) assertion, then go to the step 5.
8. Poll for TC, DEBE, DCE and DTOE of Interrupt Status (INT_STATUS) assertion.
9. Check TP of Interrupt Status (INT_STATUS); TP = 1 & no dat/cmd error indicates current CMD19/21 transfer is passed; otherwise, indicates failed.
10. If TP = 0 or command/data error is observed, increase the valueA with the required tuning step(1~127) and repeat step 5~9.
11. If TP = 1 & no dat/cmd error was met after a loop with TP=0 (or cmd/data error), save the current valueA as value_start. Increase valueA with the required tuning step, and repeat step 5~9 until TP = 0 (or cmd/data error) again. Save the current valueA as value_end.
12. Set MIX_CTRL[EXE_TUNE] to 0.
13. Set VEND_SPEC[FRC_SDCLK_ON] to 0.
14. Set SYS_CTRL[RSTA] to reset all.
15. Set MIX_CTRL[SMP_CLK_SEL] to 1.
16. Set bit[14:0] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) with the output of expression {(((value_start+value_end)/2)< < 8)& 0xffffff00 -0x300 } | 0x33.
17. Poll bit[30:16] of CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS) until the value written in the step 16 is read.
18. Set MIX_CTRL[AUTO_TUNE_EN] to 1.

The manual tuning procedure defines a passing window, inside which the tuned clock can capture the correct data, and delay chain is to be configured as centered within the passing window.

It is not a fixed flow as the standard tuning. Any step can be changed based on the real case. In cases where passing window is not closed with a tuning fail, the center of the passing region must be applied for the delay chain setting. In cases where two passing windows separated by tuning fail(s) is observed, the center of the first passing window must be applied for the delay chain setting.

NOTE

Both the standard tuning and manual tuning are only for tuning of SD_CLK, cannot be used to tune STROBE clock of HS400 and HS400 enhanced mode.

32.4 External signals

The following table describes the uSDHC external signals:

Table 32-5. uSDHC external signals

Signal	Description	Direction
CLK	Clock for MMC/SD/SDIO card	O
CMD	CMD line connect to card	I/O
DATA7	DAT7 line in the 8-bit mode — Not used in other modes	I/O
DATA6	DAT6 line in the 8-bit mode — Not used in other modes	I/O
DATA5	DAT5 line in the 8-bit mode — Not used in other modes	I/O
DATA4	DAT4 line in the 8-bit mode — Not used in other modes	I/O
DATA3	DAT3 line in the 4/8-bit mode or configured as card detection pin. The bit may be configured as card detection pin in the 1-bit mode.	I/O
DATA2	DAT2 line or Read Wait in the 4-bit mode Read Wait in 1-bit mode	I/O
DATA1	DAT1 line in the 4/8-bit mode Also, used to detect interrupt in 1/4-bit mode	I/O
DATA0	DAT0 line in all the modes Also, used to detect busy state	I/O

Table continues on the next page...

Table 32-5. uSDHC external signals (continued)

Signal	Description	Direction
CD_B	Card detection pin If not used (for the embedded memory), tie low to indicate that there is a card attached.	I
WP	Card write protect detect If not used (for the embedded memory), tie low to indicate that it is not write protected.	I
RESET_B	Card hardware reset signal, active low	O
VSELECT	IO power voltage selection signal	O

32.4.1 Signals overview

uSDHC has 15 associated I/O signals.

- The CLK is an internally generated clock used to drive the MMC, SD, and SDIO cards.
- The CMD I/O is used to send commands and receive responses to and from the card. Eight data lines (DAT7~DAT0) are used to perform data transfers between the uSDHC module and the card.
- The CD is card detection signal directly routed from the socket. This signal is active low (0). A low on CD# means that a card is inserted. It is optional for system implementation.
- The WP is write protection signals directly routed from the socket. This signal is active low (0). A high on WP means that the write protect switch is active. It is optional for system implementation.
- RST is an output signal used to reset the MMC card.
- VSELECT is an output signal used to change the voltage of the external power supplier. It is optional for system implementation.

If uSDHC needs to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.

32.5 Application of uSDHC

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, and ALL_SEND_CID. In the Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter the standby mode. Addressed type commands are used from this point. In this mode, the CMD/DATA I/O pads turns to the push-pull mode to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

A uSDHC FIFO control problem exists with a software reset for a single block read. See MIX_CTRL[DTDSEL] for more information.

32.5.1 Command send & response receive basic operation

Assuming that the data type WORD is an unsigned 32-bit integer, the flow indicated below presents a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) << 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
```

```
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure that the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and move to the Standby state no response to the Host when CMD2 is sent. The host driver deals with "fake" errors like this with caution.

32.5.2 Card identification mode

When a card is inserted to the socket or the card is reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for MMC cards.

32.5.2.1 Card detect

See the figure below for a flow diagram showing the detection of MMC, SD, and SDIO cards using uSDHC.

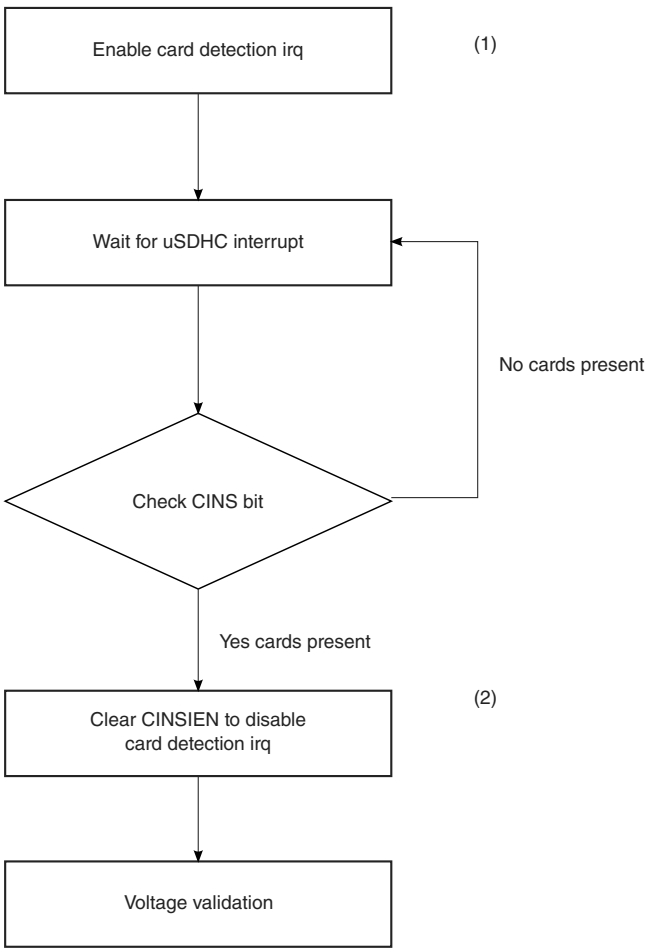


Figure 32-18. Flow diagram for card detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt.
- When an interrupt from uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion.
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards.

32.5.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) that is driven by Power On Reset (POR).

- Software reset (Host only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (Card only): The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SDIO card, CMD52 is used to write an I/O reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both uSDHC and the card.

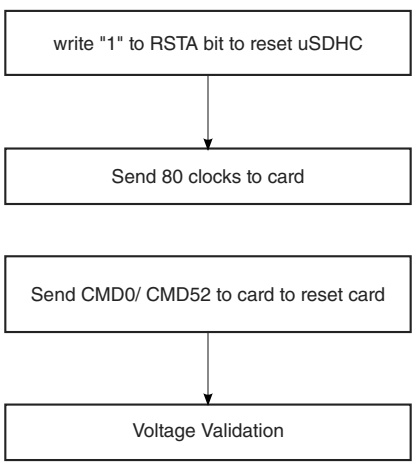


Figure 32-19. Flow chart for resetting uSDHC and SD I/O card

```
software_reset() { set_bit(SYSCTRL, RSTA); // software reset the Host set DTOCV and
      SDCLKFS bit fields to get the CLK of frequency around 400kHz configure IO pad to set
the power
      voltage of external card to around 3.0V poll bits CIHB and CDIHB bits of PRSSTAT to
wait both
      bits are cleared set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
      send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0 or
      send_command(CMD_IO_RW_DIRECT, <other parameters>); }
```

32.5.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for VDD are defined in the Operation Conditions Register (OCR) and may not cover the whole range.

Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer VDD conditions. This means that if the host and card have non-common VDD ranges, the card is neither able to complete the identification cycle nor able to send CSD data.

Therefore, special commands Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory), and IO_Send_Op_Cont (CMD5 for SD I/O) are used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards that do not match the VDD range(s) desired by the host. This is accomplished when the host sends the desired VDD voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive state. This query should be used if the host is able to select a common voltage range or if a notification is sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement) { label the card as UNKNOWN;
      send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
      operation voltage, command argument is zero if (RESP_TIMEOUT !=
      wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted if (0 < number of IO
      functions) { label the card as SDIO; IORDY = 0; while (!(IORDY in IO OCR response))
{ // set
      voltage range for each IO function send_command(IO_SEND_OP_COND, <voltage range>,
      <other parameter>); wait_for_response(IO_SEND_OP_COND); } // end of while ... } // end
      of if (0 < ... if (memory part is present inside SDIO card) Label the card as
SDCombo; //
      this is an SD-Combo card } // end of if (RESP_TIMEOUT ... if (the card is labelled as
SDIO
      card) return; // card type is identified and voltage range is set, so exit the
function;
      send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application
      specific CMD prefix if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55
is
      accepted send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set
      voltage range for memory part or SD card wait_for_response(SD_APP_OP_COND); // voltage
range
      is set if (card type is UNKNOWN) label the card as SD; return; // } // end of if (no
error ...
      else if (errors other than time-out occur) { // command/response pair is corrupted
deal with
      it by program specific manner; } // of else if (response time-out else { // CMD55 is
refuse,
      it must be MMC card if (card is already labelled as SDCombo) { // change label re-
label the
      card as SDIO; ignore the error or report it; return; // card is identified as SDIO
card } //
      of if (card is ... send_command(SEND_OP_COND, <voltage range>, <...>); if
      (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted, either
label the
      card as UNKNOWN; return; } // of if (RESP_TIMEOUT ... } // of else }
```

32.5.2.4 Card registry

This section briefly describes the registry flow. For details, please refer to the card specifications. Card registry for the MMC and SD/SDIO/SD combo cards are different. For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the card spec). Currently, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host requests the card to send their valid operation conditions.

The response to ACMD41 is the operation condition register of the card. The same command is sent to all the new cards in the system. Incompatible cards are put into the Inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA is used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in the system.

For MMC operation, the host starts the card identification process in the open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line to allow parallel card operation during card identification. After the bus is activated, the host requests the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired AND " operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in the Ready state) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. As the CID is unique for each card, only one card can be successfully sent its full CID to the host. This card then goes into the Identification state. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign a relative card address (RCA) to the card. After the RCA is received, the state of the card changes to standby, and the card does not react in further identification cycles. Also, its output driver switches

from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

The following steps show how to perform an operation using MMC cards:

```
card_registry() { do { // decide RCA for each card until response time-out
            if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
            send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to publish its RCA
            retrieve RCA from response; } // end if (card is labelled as SDCombo ... else if
(card
            is labelled as SD) { // for SD card send_command(ALL_SEND_CID, <...>); if
            (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
send_command(SET_RELATIVE_ADDR,
            <...>); retrieve RCA from response; } // else if (card is labelled as SD ... else
            if (card is labelled as MMC) { send_command(ALL_SEND_CID, <...>); rca = 0x1; //
            arbitrarily set RCA, 1 here for example send_command(SET_RELATIVE_ADDR, 0x1 <<
16,
            <...>); // send RCA at upper 16 bits } // end of else if (card is labelled as MMC
            ... } while (response is not time-out); }
```

32.5.3 Card access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

32.5.3.1 Block write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

32.5.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card indicates that the failure on the DATA line. The transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write-protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card responds with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby state and release the DATA line without interrupting the write operation. When re-selecting the card, it reactivates the busy indication by pulling data to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card that incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other methods ((CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   - For SD/MMC cards, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

32.5.3.1.2 DDR write

uSDHC supports the dual data rate mode.

The software flow to write to a card in the DDR mode is described as below:

1. Check the card status and wait until the card is ready for data.

NOTE

For eMMC card, block length only can be set to 512byte.

2. Set the uSDHC number block register (NOB), where nob is 5, for instance.
3. Set the eMMC card to high-speed mode and use SWITCH(CMD6).
4. Set the eMMC card bus with 4-bit/8-bit DDR mode and use SWITCH(CMD6).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred or another error that occurred during the auto12 command sending and response receiving.

32.5.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Similar to the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status and wait until the card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.

5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred or some another error that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The host driver reads the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible that the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is the end of the transfer. These types of requests are ignored, the driver should treat these as a non-pause transfer, and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this, it is recommended to avoid using the Suspend command for the SDIO card. This is because when such a command is sent, uSDHC interprets the system that switches to another function on the SDIO card and flush the data buffer. uSDHC takes the Resume command as a normal command with data transfer, and it is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the uSDHC module automatically sends a CMD12 to mark the end of the multi-block transfer.

32.5.3.2 Block read

Information about Normal read, DDR read, Read with Pause, and Delay Line (DLL) in Read Path are detailed in the sections below.

32.5.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks are continuously transferred until a stop command is issued.

The software flow to read from a card that incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other methods ((CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status and wait until card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer read ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

32.5.3.2.2 DDR read

The uSDHC module supports dual data rate mode.

The software flow to write to a card in the DDR mode is described below:

1. Check the card status and wait until the card is ready for data.

NOTE

For eMMC card, block length can be set to only 512 bytes.

2. Set the uSDHC number block register (NOB) where nob is 5, for instance.
3. Set the eMMC card to high-speed mode and use SWITCH (CMD6).

4. Set the eMMC card bus with 4-bit /8-bit DDR mode and use SWITCH(CMD6).
5. Disable the buffer write ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

32.5.3.2.3 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the host driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks.

Before setting the SABGREQ bit, ensure that the RWCTL bit in the Protocol Control register is set, otherwise uSDHC does not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended to set the RWCTL bit after the Read Wait capability of the SDIO card is recognized.

Similar to the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm that the card supports the Read Wait mode.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
5. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
6. Set the uSDHC number block register (NOB), where nob is 5, for instance.
7. Disable the buffer read ready interrupt, configure the DMA setting, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.

12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

Similar to the Write operation, it is possible to meet the ending block of the transfer when paused. In this case, uSDHC ignores the Stop At Block Gap Request and treats it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause is transferred to the host system. No matter if the Suspend command is sent or not, the internal data buffer is not flushed.

If the Suspend command is sent and the transfer is later resumed by means of a Resume command, uSDHC takes the command as a normal one accompanied with data transfer. It is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the uSDHC automatically sends CMD12 to mark the end of multi-block transfer.

32.5.3.2.4 Delay Line (DLL) in read path

The DLL is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

The reasons why DLL is needed for uSDHC are these:

- The path of read data traveling from card to host varies.
- In the SD/MMC DDR mode, the minimum input setup and hold time are both at 2.5 ns.

The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in the override mode. The override value set is the number of delay cells. In the override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL automatically adjusts the number of delay cells according to the target value set by the user and PVT changes. Be aware that the target value is in units of 1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns; setting target value of 16 means 5ns =(16/32)*10ns. The software can disable automatic update by the setting dll_gate_update bit.

As you might change the frequency of card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. If DLL is to be used, make sure SDCLKFS is used (at least be set to divided by 2) so that the REF_CLK generated are the same frequency as card_clk. There are two DLLs, PARTS DLL and STROBE DLL. PARTS DLL controls the SD_CLK loop back delay for which DLL (Delay Line) Control (DLL_CTRL) and DLL Status (DLL_STATUS) are used. STROBE DLL controls the STROBE loop back delay for which Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) are used. For PARTS DLL, below flow can be taken as reference. For STROBE DLL, similar register field with prefix STROBE in Strobe DLL control (STROBE_DLL_CTRL) and Strobe DLL status (STROBE_DLL_STATUS) must be used.

Step 1: Set the DLL_CTRL_RESET and DLL_CTRL_ENABLE fields

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: Clear the DLL_CTRL_RESET field

Step 5: Wait until both the DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: Set the DLL_CTRL_SLV_FORCE_UPD

Step 7: Clear the DLL_CTRL_SLV_FORCE_UPD

NOTE

The software should make sure that the DLL_CTRL_SLV_FORCE_UPD lasts for at least one card_clk. So, the software may need to add some delay between step 6 and step 7.

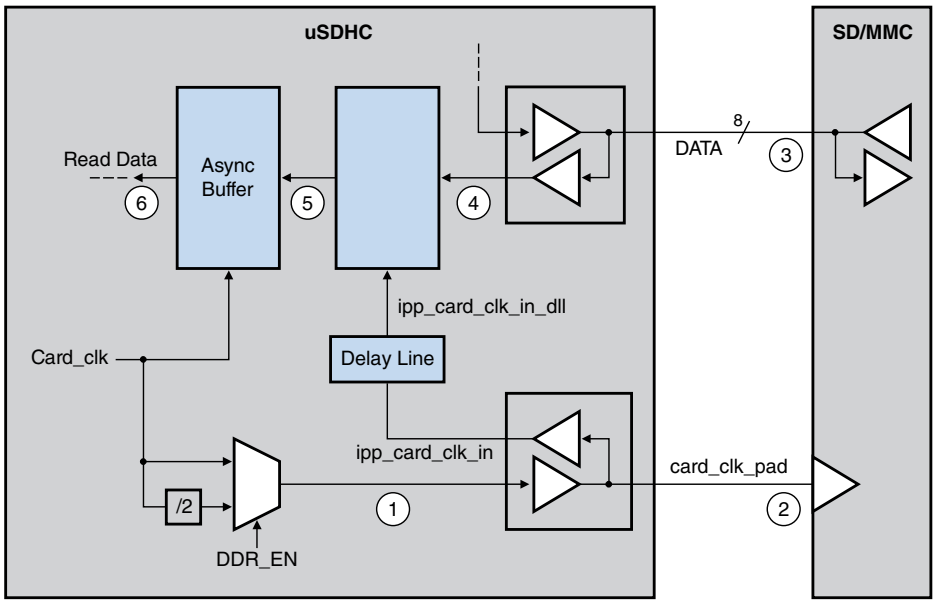


Figure 32-20. DLL in read path

32.5.3.3 Suspend Resume

The uSDHC module supports the Suspend Resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

32.5.3.3.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of the SDIO card. The uSDHC module does not monitor the content of the response, so it does not know if the Suspend command succeeded or not. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the host driver does not mark the Suspend command as "Suspend", (that is, setting the CMDTYP bits to 01). Instead, the driver sends this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform uSDHC that the current transfer is suspended. Here is the sequence for the Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.

4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so uSDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, uSDHC stops driving DATA2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

32.5.3.3.2 Resume

To resume the data transfer, a Resume command is issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation.
2. Send the Resume command: In the Transfer Type register, all the bit fields are set to the value as if this were another ordinary data transfer instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the Resume command has responded, the data transfer is resumed.

32.5.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command.

The steps to prepare the correct descriptor chain are these:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4KB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors matches the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

32.5.3.5 Transfer error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 errors are detailed in the sections below.

32.5.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received is discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one.

For a multi-block transfer, the host driver issues a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, uSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 is sent by uSDHC. In this case, the host driver re-sends or re-obtains the last block with a single block transfer.

32.5.3.5.2 Internal DMA error

During the data transfer with internal Simple DMA, if the DMA engine encounters an error on the AXI bus, the DMA operation is aborted, and the DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the host driver calculates the start address of data block in which the error occurs.

The start address can be calculated by either:

- Reading the DMA System Address register: The error occurs during the previous burst. Considering the block size, and the start address of the next burst transfer, it is straight forward to obtain the start address of the corrupted block.
- Reading the BLKCNT field of the Block Attribute register: Considering the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined.To use this method, MIX_CTRL[BCEN] bit has to be set to enable Block Attribute register update.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

32.5.3.5.3 Transfer ADMA error

There are three kinds of possible ADMA errors: The AXI transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set.

For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

- AXI transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or to the next block if no block is corrupted.
- Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and recreate the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

32.5.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, uSDHC automatically sends a CMD12 to the card to stop the transfer.

When errors with this command occur, it is recommended to the host driver to deal with the situations in the following manner:

- Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The host driver clears the auto CMD12 error status bits and re-send CMD12 until it is accepted by the card.

- Auto CMD12 response CRC error: As the card responds to CMD12, it aborts the transfer. The host driver may ignore the error and clear the error status bit.
- Auto CMD12 conflict error or not sent: The command is not sent; therefore, the host driver sends a CMD12 manually.

32.5.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For the SDIO card, it can be the low-level on the DATA1 line during some special period. The uSDHC module only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by uSDHC, and the host system is informed by uSDHC asserting the uSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt source is controlled by the external card, the interrupt from the SDIO card must be serviced before the CINT bit is cleared. Refer to Card interrupt handling for the card interrupt handling flow.

32.5.4 Switch function

A switch command is issued by the host driver to enable new features added to the SD/MMC spec. The SD/MMC cards can transfer data at bus widths other than 1-bit. Different speed modes are also defined. To enable these features, a switch command is issued by the host driver.

For SDIO cards, the high-speed mode/DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed as high. For SD cards, the high-speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high-speed mode/ HS200/HS400 are queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit, and DDR8-bit width of MMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudocode examples do not show current capability check, which is recommended in the function switch process.

32.5.4.1 Query, enable, and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

32.5.4.2 Query, enable, and disable SD high-speed mode/DDR50/ SDR50/SDR104

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104 x=3; DDR50 x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz for high speed mode, 100MHz for SDR50, 200MHz for SDR104, 50MHz for
DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

32.5.4.3 Query, enable, and disable MMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

32.5.4.4 Set MMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-
bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

32.5.5 ADMA operation

Here are the codes for the ADMA1 and ADMA2 operations.

32.5.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
```

```
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

32.5.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

32.5.6 Fast boot operation

32.5.6.1 Normal fast boot flow

Here are the steps for normal fast boot flow:

1. Software must configure the SYS_CTRL[INITA] bit to make sure that 74 card clocks are finished.
2. Software must configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode. If the data is sent through the DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set Data Transfer Width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN retain the default value, where DPSEL bit is set to 1, DTDSEL is set to 1 and MSBSEL is set to 1.
7. DMAEN should be configured as 0 in the polling mode and if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When step 6 is configured, the boot process begins. The software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt is triggered, and the software must configure MMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This render CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin the normal initialization process.
10. You must reset the host and begin the normal process.

32.5.6.2 Alternative fast boot flow

Here are the steps for alternative fast boot flow:

1. Software needs to configure SYS_CTRL[INITA] to make sure 74 card clocks are finished.

2. Software needs to configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through the DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in the DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software needs to configure the Protocol control register to set the data transfer width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with the 0xFFFFFFFA argument. In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. Note DMAEN should be configured as 0 in the polling mode, and if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in the DDR fast boot mode, DDR_EN needs to be set to 1.
7. When step 6 is configured, the boot process begins. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host sends out the interrupt and the software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process.
8. If there is no time out, the software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed, configure the MMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the slave (card) is ready for the identification step.
9. You must reset the host and begin the normal process.

32.5.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during MMC fast boot.

In fast boot, the host can use Advanced DMA2 (ADMA2) with two destinations.

The detailed flow is described below:

1. The software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. The software needs to configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1). Here VALUE1 is the value of the block count that needs to transfer the first time, so that the host stops at the block gap when the uSDHC controller gets VAULE1 blocks from the device. Also, configure bits[3-0] to select the ack timeout value according to the SD clock frequency.
3. The software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. The software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enable ADMA2 by configuring Protocol Control Register bits [9-8].
6. The software needs to set at least three pairs of ADMA2 descriptor in boot memory (that is, in IRAM, at least six words). The first pair descriptor defines the start address (that is, IRAM) and data length (that is,512byte*VALUE1) of the first part boot code. The software also needs to set the second pair descriptor, the second start address (any value that is writable), and data length is suggested to set 1~2word (record as VALUE2). Note that the second couple desc also transfers useful data even at lease 1 word, because our ADMA2 cannot support 0 data_length data transfer descriptor.
7. The software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot and do not need to be set in normal fast boot.
8. The software needs to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in the DMA mode. And, if BCEN is configured as 1, then configure blk no in Bock Attributes Register to the max value. And, if in the DDR fast boot mode, DDR_EN needs to be set to 1.
9. When step 8 is configured, boot process begins, the first VALUE1 block number data gets transferred. The software needs to poll the TC bit (bit1 in Interrupt Status Register) to determine first transfer is ended. Also, the software needs to polling the BGE bit (bit2 in Interrupt Status Register) to determine if the first transfer stops at the block gap.
10. When TC and BGE bits are set to 1, the software can analyze the first code of VALUE1 block, initializes the new memory device, if required, and sets the third

pair of descriptors to define the start address and length of the remaining part of the boot code (VALUE3, the remain boot code block). Remember to set the last descriptor with END.

11. The software needs to configure the MMC Boot Register (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the ack mode or not. In the DMA mode, configure bit 7 to 1 for enabling the automatically stop at block gap feature. Also, configure bit31-bit16 to set the (BLK_CNT - (VALUE1+1+VALUE3)), that host stops at block gap when the uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequency. Note that the software does not need to configure the BLK_CNT again, because it is counted down automatically by the uSDHC controller.
12. The software needs to clear the TC and BGE bits and the software needs to clear SABGREQ (bit 16 in the Protocol control register) and set CREQ (bit17 in the Protocol control register) to 1 to resume the data transfer. Host transfers the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. The software needs to do poll BGE bit to determine if the fast boot is over.

Note:

- When ADMA boot flow starts, for uSDHC, it is like a normal ADMA read operation. So, set ADMA2 descriptor as the normal ADMA2 transfer.
- Need a few words length memory to keep descriptor.
- For the 1~2-word data in second descriptor setting, it is a useful data, so the software needs to deal the data because of the application case.

32.6 Software restrictions

32.6.1 Initialization active

The driver cannot set INITA bit in System Control register when any of the command line or data lines are active, so the driver must ensure both CDIHB and CIHB bits are cleared.

32.6.2 Software polling procedure

For polling read or write, after the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in the Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block.

For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

32.6.3 Suspend operation

To suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such a 'suspend' command, uSDHC treats the current transfer is aborted and change the BLKCNT register to its original value, instead of retaining the remaining number of blocks.

32.6.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

ADMA1 is 4KB aligned.

32.6.5 (A)DMA address setting

To configure the ADMA1/ADMA2/DMA address register, when TC bit is set, the register always updates itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring the ADMA1/ADMA2/DMA address register.

32.6.6 Data port access

Data port does not support parallel access. For example, during an internal DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or internal DMA. Otherwise the data is corrupted inside the uSDHC buffer.

32.6.7 Change clock frequency

The uSDHC module does not automatically gate off the card clock when the host driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing the clock divisor value (SDCLKFS or DVS in System Control Register) or setting the RSTA bit.

Also, before changing the clock divisor value, the host driver should make sure that the SDSTB bit is high.

32.6.8 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode.

In this case, the card may not respond to this extra abort command and uSDHC gets response timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

32.7 uSDHC memory map/register definition

32.7.1 uSDHC register descriptions

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map of uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

32.7.1.1 uSDHC memory map

USDHC1 base address: 4041_8000h

USDHC2 base address: 4041_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	DMA System Address (DS_ADDR)	32	RW	0000_0000
4	Block Attributes (BLK_ATT)	32	RW	0001_0000
8	Command Argument (CMD_ARG)	32	RW	0000_0000
C	Command Transfer Type (CMD_XFR_TYP)	32	RW	0000_0000
10	Command Response0 (CMD_RSP0)	32	RO	0000_0000
14	Command Response1 (CMD_RSP1)	32	RO	0000_0000
18	Command Response2 (CMD_RSP2)	32	RO	0000_0000
1C	Command Response3 (CMD_RSP3)	32	RO	0000_0000
20	Data Buffer Access Port (DATA_BUFF_ACC_PORT)	32	RW	0000_0000
24	Present State (PRES_STATE)	32	RO	Table 32-8
28	Protocol Control (PROT_CTRL)	32	RW	0880_0020
2C	System Control (SYS_CTRL)	32	RW	0080_800F
30	Interrupt Status (INT_STATUS)	32	W1C	0000_0000
34	Interrupt Status Enable (INT_STATUS_EN)	32	RW	0000_0000
38	Interrupt Signal Enable (INT_SIGNAL_EN)	32	RW	0000_0000
3C	Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)	32	RW	0000_0000
40	Host Controller Capabilities (HOST_CTRL_CAP)	32	RW	07F3_B407
44	Watermark Level (WTMK_LVL)	32	RW	0810_0810
48	Mixer Control (MIX_CTRL)	32	RW	8000_0000
50	Force Event (FORCE_EVENT)	32	WORZ	0000_0000
54	ADMA Error Status (ADMA_ERR_STATUS)	32	RO	0000_0000
58	ADMA System Address (ADMA_SYS_ADDR)	32	RW	0000_0000
60	DLL (Delay Line) Control (DLL_CTRL)	32	RW	0000_0000
64	DLL Status (DLL_STATUS)	32	RO	0000_0200
68	CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)	32	RW	0000_0000
70	Strobe DLL control (STROBE_DLL_CTRL)	32	RW	0000_0000
74	Strobe DLL status (STROBE_DLL_STATUS)	32	RO	0000_0200

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
C0	Vendor Specific Register (VEND_SPEC)	32	RW	3000_7809
C4	MMC Boot (MMC_BOOT)	32	RW	0000_0000
C8	Vendor Specific 2 Register (VEND_SPEC2)	32	RW	0001_9006
CC	Tuning Control (TUNING_CTRL)	32	RW	0021_2800

32.7.1.2 DMA System Address (DS_ADDR)

This register contains the physical system memory address used for DMA transfers.

32.7.1.2.1 Offset

Register	Offset
DS_ADDR	0h

32.7.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.2.3 Fields

Field	Description
31-0 DS_ADDR	System address **DMA system address / argument 2** When ACMD23_ARGU2_EN is set to 0, SDMA uses this register as system address and supports only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When ACMD23_ARGU2_EN is set to 1, SDMA uses ADMA System Address register (05Fh – 058h) instead of this register to support both 32-bit and 64-bit addressing. This register is used only for Argument2 and SDMA may use Auto CMD23.

Field	Description
	1. **SDMA system address** Because the address must be word (4 bytes) aligned, the least two bits are reserved and always set to 0. When uSDHC stops a DMA transfer, this register points out the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The host driver initializes this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver waits until the DLA field in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions. 2. **Argument 2** This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

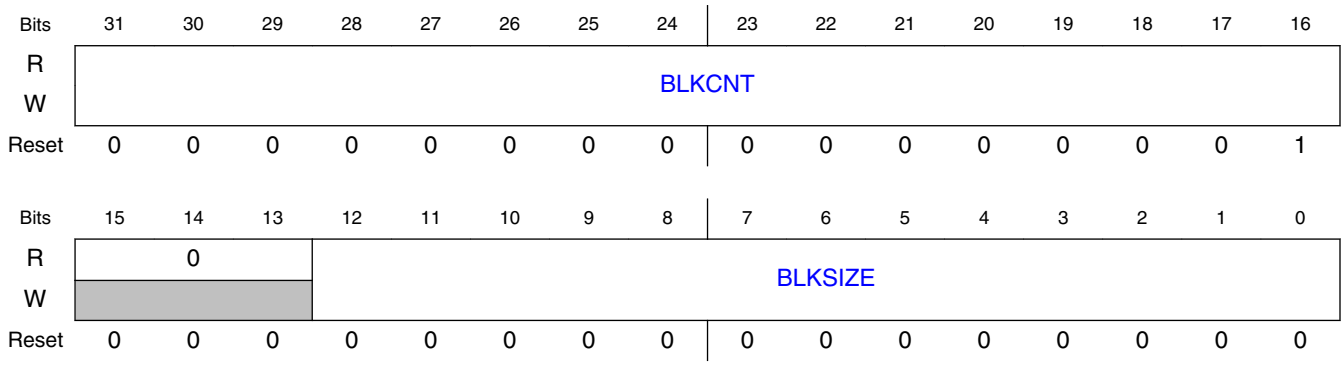
32.7.1.3 Block Attributes (BLK_ATT)

This register is used to configure the number of data blocks and the number of bytes in each block.

32.7.1.3.1 Offset

Register	Offset
BLK_ATT	4h

32.7.1.3.2 Diagram



32.7.1.3.3 Fields

Field	Description
31-16 BLKCNT	Blocks count for current transfer This field is enabled when the Block Count Enable field in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this field always reads as 1. The host driver sets this field to a value between 1 and the maximum block count. The uSDHC module decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to zero results in no data blocks being transferred. This field should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this field may return an invalid value and write operations are ignored. When saving transfer content because of a Suspend command, the number of blocks yet to be transferred can be determined by reading this field. The reading of this field should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when the Suspend command is sent out, uSDHC treats the current transfer as aborted and change the BLKCNT field back to its original value instead of keeping the dynamical indicator of the remaining block count. When restoring transfer content prior to issuing a Resume command, the host driver restores the previously saved block count. **NOTE:** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL field is indicating a single block transfer, the read value of BLKCNT is always 1. 0000000000000000 - Stop count 0000000000000001 - 1 block 0000000000000010 - 2 blocks 1111111111111111 - 65535 blocks
15-13 —	Reserved
12-0 BLKSIZE	Transfer block size This field specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored.

Field	Description
	0000000000000 - No data transfer 0000000000001 - 1 byte 0000000000010 - 2 bytes 0000000000011 - 3 bytes 0000000000100 - 4 bytes 0000111111111 - 511 bytes 0001000000000 - 512 bytes 0100000000000 - 2048 bytes 1000000000000 - 4096 bytes

32.7.1.4 Command Argument (CMD_ARG)

This register contains the SD/MMC command argument.

32.7.1.4.1 Offset

Register	Offset
CMD_ARG	8h

32.7.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.4.3 Fields

Field	Description
31-0	Command argument

Field	Description
CMDARG	The SD/MMC command argument is specified as bits 39-8 of the command format in the SD or MMC specification. This field is write protected when the Command Inhibit (CMD) field in the Present State register is set.

32.7.1.5 Command Transfer Type (CMD_XFR_TYP)

This register is used to control the operation of data transfers. The host driver sets this register before issuing a command followed by a data transfer or before issuing a Resume command. To prevent data loss, uSDHC prevents writing to the bits, which are involved in the data transfer of this register, when data transfer is active. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver checks the Command Inhibit DAT field (PRES_STATE[CDIHB]) and the Command Inhibit CMD field (PRES_STATE[CIHB]) in the Present State register before writing to this register. When the CDIHB field in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB field is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as a single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC ignores the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (PRES_STATE[WPSPL] field of Present State register is '1'); otherwise, uSDHC also ignores the command.

If the commands with data transfer do not receive the response in 64 clock cycles, that is, if response time-out happens, uSDHC treats the external device, does not accept the command, and aborts the data transfer. In this scenario, the driver should issue the command again to retry the transfer. It is also possible that for some reason the card responds to the command but uSDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 32-6. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, regarding the Response Type bits as well as the name of the response type.

Table 32-7. Relationship between parameters and the name of the response type

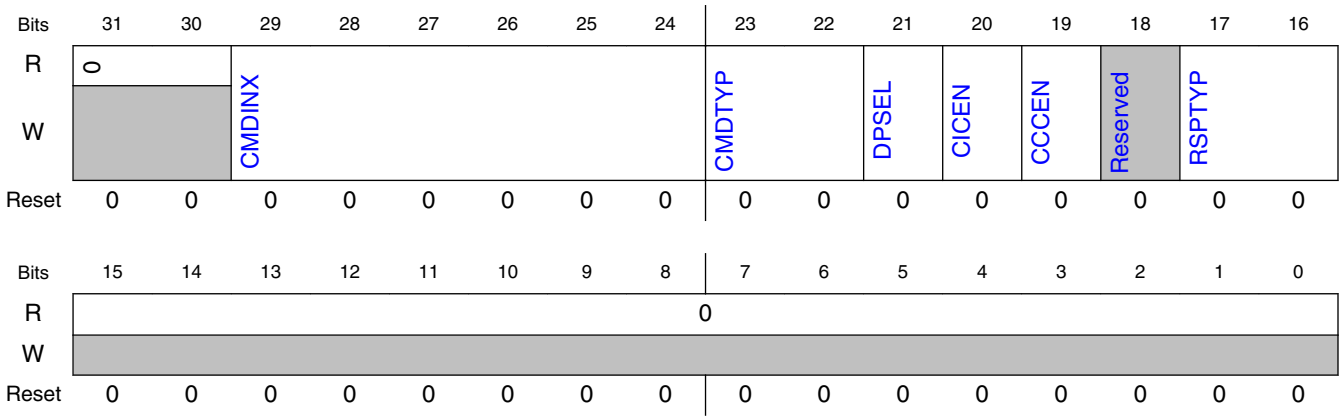
Response type	Index check enable	CRC check enable	Name of response type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification, but R5b is defined in this specification to specify that uSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command is used with R5b.
- The CRC fields for R3 and R4 are expected to be all 1 bits. The CRC check is disabled for these response types.

32.7.1.5.1 Offset

Register	Offset
CMD_XFR_TYP	Ch

32.7.1.5.2 Diagram



32.7.1.5.3 Fields

Field	Description
31-30 —	Reserved
29-24 CMDINX	Command index These fields are set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23-22 CMDTYP	Command type There are three types of special commands: Suspend, Resume, and Abort. These fields are set to 00b for all other commands. • Suspend command: If the Suspend command succeeds, uSDHC assumes that the card bus has been released and that it is possible to issue the next command that uses the DATA line. Because uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the host driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform uSDHC that a Suspend command was successfully issued. See Suspend Resume for more details. After the end bit of command is sent, uSDHC deasserts Read Wait for read transactions and stops checking busy for write transactions. In a 4-bit mode, the interrupt cycle starts. If the Suspend command fails, uSDHC maintains its current state, and the host driver restarts the transfer by setting the Continue Request field in the Protocol Control register. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the Suspend command and then sends the Resume command. The uSDHC module checks for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, uSDHC stops reads to the buffer. If this command is set when executing a write transfer, uSDHC stops driving the DATA line. After issuing the Abort command, the host driver should issue a software reset (Abort Transaction). 00 - Normal other commands 01 - Suspend CMD52 for writing bus suspend in CCCR 10 - Resume CMD52 for writing function select in CCCR 11 - Abort CMD12, CMD52 for writing I/O Abort in CCCR
21	Data present select

Table continues on the next page...

Field	Description
DPSEL	This field is set to 1 to indicate that data is present and is transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b (for example, CMD38)) **NOTE:** In resume command, this field is set, and other bits in this register is set the same as when the transfer was initially launched. When the write protect switch is on, (that is, the WPSPL field is active as '0'), any command with a write operation ignored. When this field is set, while the DTDSEL field is 0, writes to the register Transfer Type are ignored. 0 - No data present 1 - Data present
20 CICEN	Command index check enable If this field is set to 1, uSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this field is set to 0, the Index field is not checked. 0 - Disable command index check 1 - Enables command index check
19 CCCEN	Command CRC check enable If this field is set to 1, uSDHC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this field is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. See RSPTYP[1:0] and Command Transfer Type (CMD_XFR_TYP). 0 - Disables command CRC check 1 - Enables command CRC check
18 —	Reserved
17-16 RSPTYP	Response type select 00 - No response 01 - Response length 136 10 - Response length 48 11 - Response length 48, check busy after response
15-0 —	Reserved

32.7.1.6 Command Response0 (CMD_RSP0)

This register is used to store part 0 of the response bits from the card.

32.7.1.6.1 Offset

Register	Offset
CMD_RSP0	10h

32.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.6.3 Fields

Field	Description
31-0 CMDRSP0	Command response 0 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

32.7.1.7 Command Response1 (CMD_RSP1)

This register is used to store part 1 of the response bits from the card.

32.7.1.7.1 Offset

Register	Offset
CMD_RSP1	14h

32.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.7.3 Fields

Field	Description
31-0 CMDRSP1	Command response 1 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

32.7.1.8 Command Response2 (CMD_RSP2)

This register is used to store part 2 of the response bits from the card.

32.7.1.8.1 Offset

Register	Offset
CMD_RSP2	18h

32.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.8.3 Fields

Field	Description
31-0 CMDRSP2	Command response 2 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

32.7.1.9 Command Response3 (CMD_RSP3)

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD bus to Command Response registers for each response type. In this table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 32-8. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0

Table continues on the next page...

Table 32-8. Response bit definition for each response type (continued)

Response type	Meaning of response	Response field	Response register
R6 (publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

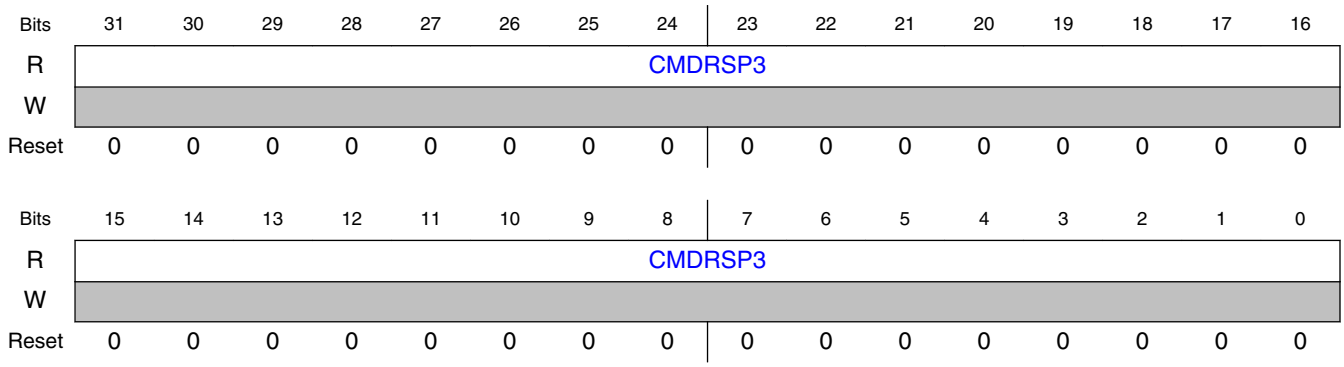
To be able to read the response status efficiently, uSDHC only stores part of the response data in the Command Response registers. This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, uSDHC checks R[47:1], and if the response length is 136 the uSDHC checks R[119:1].

Because uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows uSDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

32.7.1.9.1 Offset

Register	Offset
CMD_RSP3	1Ch

32.7.1.9.2 Diagram



32.7.1.9.3 Fields

Field	Description
31-0 CMDRSP3	Command response 3 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this field for each response type.

32.7.1.10 Data Buffer Access Port (DATA_BUFF_ACC_PORT)

The Buffer Data Port register is for 32-bit data access by the Arm platform. When the internal DMA is enabled, any write to this field is ignored, and any read from this field always yields 0s.

32.7.1.10.1 Offset

Register	Offset
DATA_BUFF_ACC_PORT	20h

32.7.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.10.3 Fields

Field	Description
31-0 DATCONT	Data content This field is used to access the internal buffer.

32.7.1.11 Present State (PRES_STATE)

The host driver can get status of uSDHC from this 32-bit read only register.

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands are issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

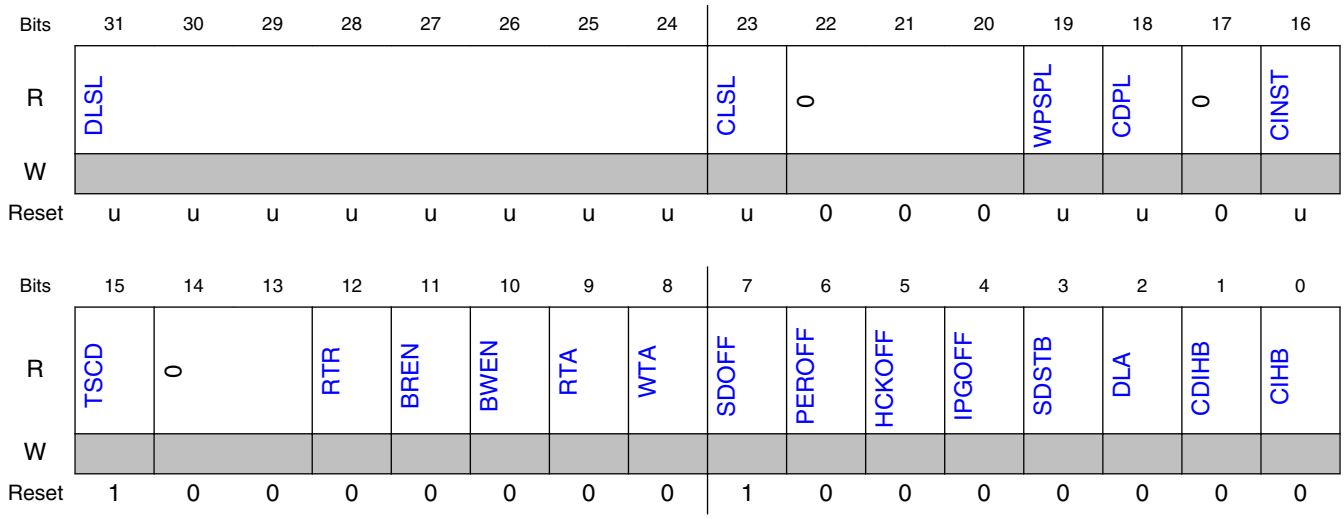
NOTE

The reset value of Present State register depends on board connectivity.

32.7.1.11.1 Offset

Register	Offset
PRES_STATE	24h

32.7.1.11.2 Diagram



32.7.1.11.3 Fields

Field	Description
31-24 DLSL	DATA[7:0] line signal level This field is used to check the DATA line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. 00000000 - Data 0 line signal level 00000001 - Data 1 line signal level 00000010 - Data 2 line signal level 00000011 - Data 3 line signal level 00000100 - Data 4 line signal level 00000101 - Data 5 line signal level 00000110 - Data 6 line signal level 00000111 - Data 7 line signal level
23 CLSL	CMD line signal level This field is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this field after reset is 1'b1, when the command line is pulled up.
22-20 —	Reserved
19 WPSPL	Write protect switch pin level The Write Protect switch is supported for memory and combo cards. This field reflects the inverted value of the WP pin of the card socket. A software reset does not affect this field. The reset value is affected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this field is high and write is enabled.

Table continues on the next page...

Field	Description
	0 - Write protected (WP = 1) 1 - Write enabled (WP = 0)
18 CDPL	Card detect pin level This field reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this field. This field may be valid, but is not guaranteed, because of propagation delay. Use of this field is limited to testing because it must be debounced by software. A software reset does not affect this field. A write to the Force Event Register does not affect this field. The reset value is affected by the external card detection pin. This field shows the value on the CD_B pin (that is, when a card is inserted in the socket, it is 0 on the CD_B input, and consequently, the CDPL reads 1. 0 - No card present (CD_B = 1) 1 - Card present (CD_B = 0)
17 —	Reserved
16 CINST	Card inserted This field indicates whether a card has been inserted. The uSDHC module debounces this signal so that the host driver does not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not affect this field. The Software Reset For All in the System Control register does not affect this field. A software reset does not affect this field. 0 - Power on reset or no card 1 - Card inserted
15 TSCD	Tap select change done This field indicates the delay setting is effective after write CLK_TUNE_CTRL_STATUS register. 0 - Delay cell select change is not finished. 1 - Delay cell select change is finished.
14-13 —	Reserved
12 RTR	Re-Tuning Request (only for SD3.0 SDR104 mode,and EMMC HS200 mode) Host controller may request host driver to execute re-tuning sequence by setting this field when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This field is cleared when a command is issued with setting Execute Tuning field in MIX_CTRL register. Changing of this field from 0 to 1 generates Re-Tuning Event. See Interrupt status registers for more detail. This field isn't set to 1 if Sampling Clock Select in the MIX_CTRL register is set to 0 (using fixed sampling clock). 0 - Fixed or well tuned sampling clock 1 - Sampling clock needs re-tuning
11 BREN	Buffer read enable This status field is used for non-DMA read transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this field is high, valid data greater than the watermark level exist in the buffer. A change of this field from 1 to 0 occurs when some reads from the buffer (read DATPORT (Base + 0x20)) are made and the buffer

Table continues on the next page...

Field	Description
	hasn't valid data greater than the watermark level. A change of this field from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 0 - Read disable 1 - Read enable
10 BWEN	Buffer write enable This status field is used for non-DMA write transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this field is 1, valid data greater than the watermark level can be written to the buffer. A change of this field from 1 to 0 occurs when some writes to the buffer (write DATPORT (Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this field from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 0 - Write disable 1 - Write enable
9 RTA	Read transfer active This status field is used for detecting completion of a read transfer. This field is set for either of the following conditions: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer A transfer complete interrupt is generated when this field changes to 0. This field is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from uSDHC internal buffer. • When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent because of the Stop At Block Gap Request being set to 1. 0 - No valid data 1 - Transferring data
8 WTA	Write transfer active This status field indicates a write transfer is active. If this field is 0, it means no valid write data exists in uSDHC. This field is set in either of the following cases: • After the end field of the write command • When writing 1 to the Continue Request field in the Protocol Control register to restart a write transfer This field is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request During a write transaction, a Block Gap Event interrupt is generated when this field is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the host driver in determining when to issue commands during Write Busy state. 0 - No valid data

Table continues on the next page...

Field	Description
	1 - Transferring data
7 SDOFF	SD clock gated off internally This status field indicates that the SD clock is internally gated off, because of buffer over / under-run or read pause without read wait assertion, or the driver set FRC_SDCLK_ON field is 0 to stop the SD clock in idle status. This field is for the host driver to debug data transaction on the SD bus. 0 - SD clock is active. 1 - SD clock is gated off.
6 PEROFF	IPG_PERCLK gated off internally This status field indicates that the IPG_PERCLK is internally gated off. This field is for the host driver to debug transaction on the SD bus. When IPG_CLK_SOFT_EN is cleared, IPG_PERCLK is gated off, otherwise IPG_PERCLK is always active. 0 - IPG_PERCLK is active. 1 - IPG_PERCLK is gated off.
5 HCKOFF	HCLK gated off internally This status field indicates that the HCLK is internally gated off. This field is for the host driver to debug during a data transfer. 0 - HCLK is active. 1 - HCLK is gated off.
4 IPGOFF	Peripheral clock gated off internally This status field indicates that the peripheral clock is internally gated off. This field is for the host driver to debug. 0 - Peripheral clock is active. 1 - Peripheral clock is gated off.
3 SDSTB	SD clock stable This status field indicates that the internal card clock is stable. This field is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON field in System Control register to remove glitches on the card clock when the frequency is changing. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. 0 - Clock is changing frequency and not stable. 1 - Clock is stable.
2 DLA	Data line active This status field indicates whether one of the DATA lines on the SD bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer This field is cleared in either of the following cases: • When the end field of the last data block is sent from the SD bus to uSDHC. • When the Read Wait state is stopped by a Suspend command and the DATA2 line is released.

Table continues on the next page...

Field	Description
	The uSDHC module waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait to use the suspend / resume function. This field remains 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the write command • When writing to 1 to the Continue Request field in the Protocol Control register to continue a write transfer This field is cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, uSDHC also detects if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, uSDHC assumes the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and because of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This field is cleared when the DATA0 line is released. 0 - DATA line inactive 1 - DATA line active
1 CDIHB	Command Inhibit Data (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / MMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0. 0 - Can issue command that uses the DATA line 1 - Cannot issue command that uses the DATA line
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / MMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0 - Can issue command using only CMD line 1 - Cannot issue command

32.7.1.12 Protocol Control (PROT_CTRL)

This register controls three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether uSDHC issues a Suspend command or the SD card accepts the Suspend command.

- If the host driver does not issue a Suspend command, the Continue request is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card does not accept it, the Continue request is used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver waits for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the host driver clears the Stop At Block Gap Request before or simultaneously.

32.7.1.12.1 Offset

Register	Offset
PROT_CTRL	28h

32.7.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	NON_EXACT_BLK_RD	Reserved			WECRM	WECINS	WECINT	Reserved			RD_DONE_NO_8CLK	IABG	RWCTL	CREQ	SABGREQ
W																
Reset	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DMASEL		CDSS	CDTL	EMODE		D3CD	DTW		Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

32.7.1.12.3 Fields

Field	Description
31 —	Reserved Always write as 0
30 NON_EXACT_BLK_RD	Non-exact block read Current block read is non-exact block read. It is only used for SDIO. 0 - The block read is exact block read. Host driver does not need to issue abort command to terminate this multi-block read. 1 - The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read.
29-27 —	Reserved
26 WECRM	Wakeup event enable on SD card removal This field enables a wakeup event, via a card removal, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Removal Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Removal Status and uSDHC interrupt. 0 - Disables wakeup event enable on SD card removal 1 - Enables wakeup event enable on SD card removal
25 WECINS	Wakeup event enable on SD card insertion This field enables a wakeup event, via a card insertion, in the Interrupt Status register. FN_WUS (Wakeup Support) in CIS does not affect this field. When this field is set, the Card Insertion Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Insertion Status and uSDHC interrupt. 0 - Disable wakeup event enable on SD card insertion 1 - Enable wakeup event enable on SD card insertion
24 WECINT	Wakeup event enable on card interrupt This field enables a wakeup event, via a card interrupt, in the Interrupt Status register. This field can be set to 1 if FN_WUS (Wakeup Support) in CIS is set to 1. When this field is set, the Card Interrupt Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Interrupt Status and uSDHC interrupt. 0 - Disables wakeup event enable on card interrupt 1 - Enables wakeup event enable on card interrupt
23-21 —	Reserved Always write as 3'b100
20 RD_DONE_NO_8CLK	Read performed number 8 clock According to the SD/MMC spec, for read data transaction, 8 clocks are needed after the end field of the last data block. So, by default(RD_DONE_NO_8CLK=0), eight clocks are active after the end field of the last read data transaction. However, these 8 clocks should not be active if user wants to use stop at block gap (include the auto stop at block gap in boot mode) feature for read and the RWCTL field (bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid these 8 clocks. Otherwise, the device might send extra data to uSDHC while uSDHC ignores these data.

Table continues on the next page...

Field	Description
	In a summary, this field should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.
19 IABG	Interrupt at block gap This field is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this field should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card. 0 - Disables interrupt at block gap 1 - Enables interrupt at block gap
18 RWCTL	Read wait control The read wait function provided by this field is optional for SDIO cards. If the card supports read wait, set this field to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise, uSDHC has to stop the SD clock to hold read data, which restricts commands generation. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card. If the card does not support read wait, this field should never be set to 1; otherwise, DATA line conflicts might occur. If this field is set to 0, stop at block gap during read operation is also supported, but uSDHC stops the SD clock to pause reading operation. 0 - Disables read wait control and stop SD clock at block gap when SABGREQ field is set 1 - Enables read wait control and assert read wait without stopping SD clock at block gap when SABGREQ field is set
17 CREQ	Continue request This field is used to restart a transaction which was stopped using the stop at block gap request. When a suspend operation is not accepted by the card, it is also by setting this field to restart the paused transfer. To cancel stop at the block gap, set stop at block gap request to 0 and set this field to 1 to restart the transfer. The uSDHC module automatically clears this field, therefore it is not necessary for the host driver to set this field to 0. If both stop at block gap request and this field are 1, the continue request is ignored. 0 - No effect 1 - Restart
16 SABGREQ	Stop at block gap request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver leaves this field set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC module supports the stop at block gap request for write transfers, but for read transfers it requires that the SDIO card support read wait. Therefore, the host driver does not set this field during read transfers unless the SDIO card supports Read Wait and has set the read wait control to 1; otherwise, uSDHC stops the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the Data Port register, the host driver sets this field after all block data is written. If this field is set to 1, the host driver does not write data to the Data Port register after a block is sent. Once this field is set, the host driver does not clear this field before the Transfer Complete field in Interrupt Status register is set, otherwise uSDHC's behavior is undefined. This field effects read transfer active, write transfer active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 0 - Transfer 1 - Stop
15-10	Reserved

Table continues on the next page...

Field	Description
—	
9-8 DMASEL	DMA select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 - No DMA or simple DMA is selected. 01 - ADMA1 is selected. 10 - ADMA2 is selected. 11 - Reserved
7 CDSS	Card detect signal selection This field selects the source for the card detection. 0 - Card detection level is selected (for normal purpose). 1 - Card detection test level is selected (for test purpose).
6 CDTL	Card detect test level This field is enabled while the card detection signal selection is set to 1 and it indicates card insertion. 0 - Card detect test level is 0, no card inserted 1 - Card detect test level is 1, card inserted
5-4 EMODE	Endian mode This field supports all three endian modes in data transfer. See Data buffer for more details. 00 - Big endian mode 01 - Half word big endian mode 10 - Little endian mode 11 - Reserved
3 D3CD	DATA3 as card detection pin If this field is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 might set the card into the SPI mode, which uSDHC does not support. 0 - DATA3 does not monitor card insertion 1 - DATA3 as card detection pin
2-1 DTW	Data transfer width This field selects the data width of the SD bus for a data transfer. The host driver sets it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00 - 1-bit mode 01 - 4-bit mode 10 - 8-bit mode 11 - Reserved
0 —	Reserved

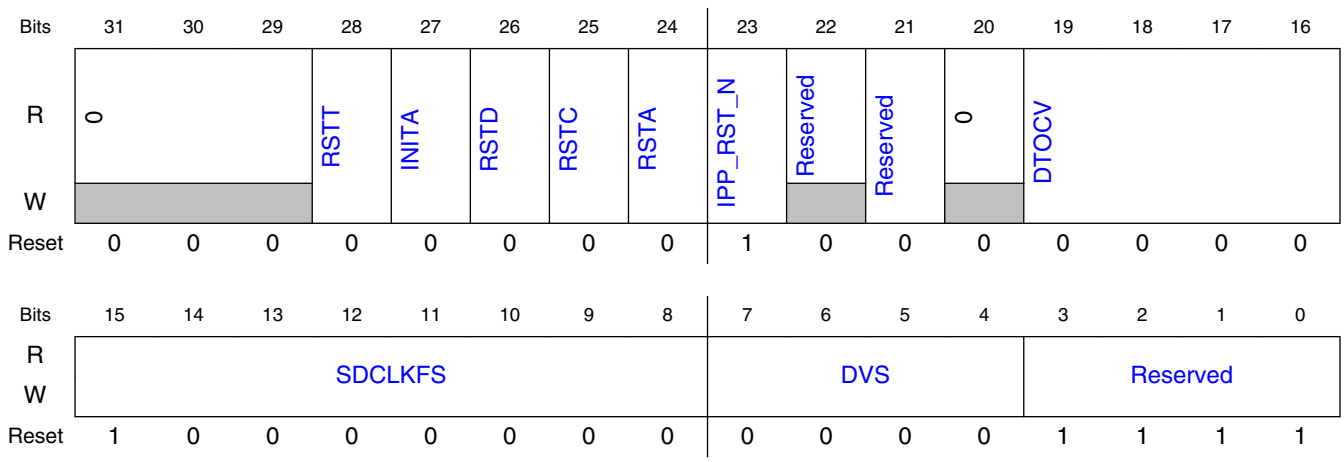
32.7.1.13 System Control (SYS_CTRL)

This register provides control of the system. See detail in the field description.

32.7.1.13.1 Offset

Register	Offset
SYS_CTRL	2Ch

32.7.1.13.2 Diagram



32.7.1.13.3 Fields

Field	Description
31-29 —	Reserved
28 RSTT	Reset tuning When set this field to 1, it resets tuning circuit. After tuning circuits are reset, bit value is 0. Clearing execute_tuning field in AUTOCMD12_ERR_STATUS also sets this field to 1 to reset tuning circuit
27 INITA	Initialization active When this field is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this field is self cleared. This field is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this field is already 1 has no effect. Writing 0 to this field at any time has no effect. When either of the CIHB and CDIHB fields in the Present State register are set, writing 1 to this field is ignored (that is, when command line or data lines are active, write to this field is not allowed). On the other-hand, when this field is set, that is, during initialization active period, it is allowed to issue command, and the command bit stream appears on the CMD pad after all 80 clock cycles are done. So, when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs to send 80 cycles to the card and does not want to wait till this field is self cleared.

Table continues on the next page...

Field	Description

26
RSTD
	Software reset for data line
	Only part of the data circuit is reset. DMA circuit is also reset. After this field is set, the software waits for self-clear.
	The following registers and bits are cleared by this field:
	• Data Port register
	  • Buffer is cleared and initialized
	• Present State register
	  • Buffer read enable
	  • Buffer write enable
	  • Read transfer active
	  • Write transfer active
	  • DATA line active
	  • Command Inhibit (DATA)
	• Protocol Control register
	  • Continue request
	• Interrupt Status register
	  • Buffer read ready
	  • Buffer write ready
	  • DMA interrupt
	  • Block gap event
	  • Transfer complete
	NOTE: When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set.
	0 - No reset
	1 - Reset

25
RSTC
	Software reset for CMD line
	Only part of the command circuit is reset. After this field is set, the software waits for self-clear.
	The following registers and bits are cleared by this field:
	• Present State Register
	  • Command Inhibit (CMD)
	• Interrupt Status Register
	  • Command Complete
	0 - No reset
	1 - Reset

24 RSTA	Software reset for all This reset affects the entire host controller except for the card detection circuit. RSTA resets all the registers that can be reset by RSTC/RSTD. During its initialization, the host driver is set this field to 1 to reset uSDHC. The uSDHC module resets this field to 0 when the capabilities registers are valid and the host driver can read them. Additional use of Software Reset For All does not affect the value of the capabilities registers. After this field is set, it is recommended that the host driver reset the external card and re-initialize it. After this field is set, the software should wait for self-clear. In tuning process, after every CMD19 is finished, this field is set to retest uSDHC. **NOTE:** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0 - No reset 1 - Reset
23	Hardware reset

Table continues on the next page...

Field	Description
IPP_RST_N	This field's value is output to card through pad directly to hardware reset pin of the card if the card supports this feature.
22 —	Reserved
21 —	Reserved
20 —	Reserved
19-16 DTOCV	Data timeout counter value This value determines the interval by which DAT line timeouts are detected. See the Data Timeout Error field in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency is generated by dividing the base clock SDCLK value by this value. The host driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 0000 - SDCLK x 2^{14} 0001 - SDCLK x 2^{15} 0010 - SDCLK x 2^{16} 0011 - SDCLK x 2^{17} 0100 - SDCLK x 2^{18} 0101 - SDCLK x 2^{19} 0110 - SDCLK x 2^{20} 0111 - SDCLK x 2^{21} 1000 - SDCLK x 2^{22} 1001 - SDCLK x 2^{23} 1010 - SDCLK x 2^{24} 1011 - SDCLK x 2^{25} 1100 - SDCLK x 2^{26} 1101 - SDCLK x 2^{27} 1110 - SDCLK x 2^{28} 1111 - SDCLK x 2^{29}
15-8 SDCLKFS	SDCLK frequency select This field is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this field holds the prescaler (of this register) and divisor (next register) of the Base Clock Frequency register. In Single Data Rate mode (DDR_EN field of MIXERCTRL is '0') Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16

Table continues on the next page...

Field	Description
	04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 While in Dual Data Rate mode (DDR_EN field of MIXERCTRL is '1') Only the following settings are allowed: 80h) Base clock divided by 512 40h) Base clock divided by 256 20h) Base clock divided by 128 10h) Base clock divided by 64 08h) Base clock divided by 32 04h) Base clock divided by 16 02h) Base clock divided by 8 01h) Base clock divided by 4 00h) Base clock divided by 2 When the software changes the DDR_EN field, SDCLKFS might need to be changed also. In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h yields 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD clock after reset is 375 kHz. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. If setting SDCLKFS and DVS can generate the same clock frequency,(for example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.), SDCLKFS is highly recommended. For HS400 mode, SDCLKFS must be used so that REF_CLK, generated for STROBE DLL, is at the same frequency as STROBE clock. For HS200 and SDR104 mode, if DLL is to be used, make sure SDCLKFS is used so that REF_CLK, generated for PARTS DLL, is at the same frequency as SD_CLK.
7-4 DVS	Divisor This field is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisors without deterioration of duty cycle. Before changing clock divisor value (SDCLKFS or DVS), Host driver should make sure the SDSTB field is high. The settings are as follows: 0000 - Divide-by-1

Table continues on the next page...

Field	Description
	0001 - Divide-by-2 1110 - Divide-by-15 1111 - Divide-by-16
3-0 —	Reserved Always write as 1.

32.7.1.14 Interrupt Status (INT_STATUS)

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status fields is set to 1. For all fields, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition; otherwise, the CINT field is asserted again.

The table below shows the relationship between the command timeout error and the command complete.

Table 32-9. uSDHC status for command timeout error/command complete bit combinations

Command CRC Error	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the transfer complete and the data timeout error.

Table 32-10. uSDHC status for data timeout error/transfer complete bit combinations

Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC error and command timeout error.

Table 32-11. uSDHC status for command CRC error/command timeout error bit combinations

Command complete	Command timeout error	Meaning of the status
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

32.7.1.14.1 Offset

Register	Offset
INT_STATUS	30h

32.7.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAE	0	TNE	0	AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W				W1C		W1C		W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TP	0	RTE	0			CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W		W1C		W1C				W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

32.7.1.14.3 Fields

Field	Description
31-29 —	Reserved
28 DMAE	DMA error Occurs when an Internal DMA transfer has failed. This field is set to 1, when some error occurs in the data transfer. This error can be caused by either simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs.

Table continues on the next page...

Field	Description
	Because any error corrupts the whole data block, the host driver restarts the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 0 - No error 1 - Error
27 —	Reserved
26 TNE	Tuning error: (only for SD3.0 SDR104 mode and EMMC HS200 mode) This field is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, host driver needs to abort a command executing and perform tuning.
25 —	Reserved
24 AC12E	Auto CMD12 error Occurs when detecting that one of the fields in the Auto CMD12 Error Status register has changed from 0 to 1. This field is set to 1, not only when the errors in Auto CMD12 occur, but also, when the Auto CMD12 is not executed due to the previous command error. 0 - No error 1 - Error
23 —	Reserved
22 DEBE	Data end bit error Occurs either when detecting 0 at the end field position of read data that uses the DATA line, or at the end field position of the CRC. This field is not asserted in tuning process. 0 - No error 1 - Error
21 DCE	Data CRC error Occurs when detecting a CRC error when transferring read data that uses the DATA line, or when detecting the Write CRC status having a value other than 010. This field is not asserted in tuning process. 0 - No error 1 - Error
20 DTOE	Data timeout error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out. This field is not asserted in tuning process. 0 - No error 1 - Time out
19 CIE	Command index error Occurs if a command index error occurs in the command response.

Table continues on the next page...

Field	Description
	This field is not asserted in tuning process. 0 - No error 1 - Error
18 CEBE	Command end bit error Occurs when detecting that the end field of a command response is 0. This field is not asserted in tuning process. 0 - No error 1 - End bit error generated
17 CCE	Command CRC error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this field is set when detecting a CRC error in the command response. • The uSDHC module detects a CMD line conflict by monitoring the CMD line when a command is issued. If uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then uSDHC aborts the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error should also be set to 1 to distinguish CMD line conflict. This field is not asserted in tuning process. 0 - No error 1 - CRC error generated
16 CTOE	Command timeout error Occurs only if no response is returned within 64 SDCLK cycles from the end field of the command. If uSDHC detects a CMD line conflict, in which case a Command CRC Error is also be set (as shown in Interrupt Status (INT_STATUS)), this field is set without waiting for 64 SDCLK cycles. This is because the command is aborted by uSDHC. This field is not asserted in tuning process. 0 - No error 1 - Time out
15 —	Reserved
14 TP	Tuning pass:(only for SD3.0 SDR104 mode and EMMC HS200 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.
13 —	Reserved
12 RTE	Re-tuning event: (only for SD3.0 SDR104 mode and EMMC HS200 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. host controller requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0 - Re-tuning is not required. 1 - Re-tuning should be performed.
11-9 —	Reserved
8	Card interrupt

Table continues on the next page...

Field	Description
CINT	This status field is set when an interrupt signal is detected from the external card. In 1-bit mode, uSDHC detects the Card Interrupt without the SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the host system. Writing this field to 1 can clear this field, but as the interrupt source from the SDIO card does not clear, this field is set again. to clear this field, it is required to reset the interrupt source from the external card followed by a writing 1 to this field. When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset the interrupt sources in the SDIO card and the interrupt signal might not be asserted), write 1 to clear this field, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0 - No card interrupt 1 - Generate card interrupt
7 CRM	Card removal This status field is set if the Card Inserted field in the Present State register changes from 1 to 0. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if no card is inserted. to leave it cleared, clear the Card Removal Status Enable field in Interrupt Status Enable register. 0 - Card state unstable or inserted 1 - Card removed
6 CINS	Card insertion This status field is set if the Card Inserted field in the Present State register changes from 0 to 1. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if a card is inserted. to leave it cleared, clear the Card Inserted Status Enable field in Interrupt Status Enable register. 0 - Card state unstable or removed 1 - Card inserted
5 BRR	Buffer read ready This status field is set if the Buffer Read Enable field, in the Present State register, changes from 0 to 1. See the Buffer Read Enable field in the Present State register for additional information. This field indicates that cmd19 is finished in tuning process. 0 - Not ready to read buffer 1 - Ready to read buffer
4 BWR	Buffer write ready This status field is set if the Buffer Write Enable field, in the Present State register, changes from 0 to 1. See the Buffer Write Enable field in the Present State register for additional information. 0 - Not ready to write buffer 1 - Ready to write buffer
3 DINT	DMA interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this field does not be set. Instead, the DMAE field is set. Either Simple DMA or ADMA finishes data transferring, this field is set.

Table continues on the next page...

Field	Description
	0 - No DMA interrupt 1 - DMA interrupt is generated.
2 BGE	Block gap event If the Stop At Block Gap Request field in the Protocol Control register is set, this field is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this field is not set to 1. In the case of a Read Transaction: This field is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD bus timing). The Read Wait must be supported to use this function. In the case of Write Transaction: This field is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 0 - No block gap event 1 - Transaction stopped at block gap
1 TC	Transfer complete This field is set when a read or write transfer is completed. In the case of a Read Transaction: This field is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the host system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request field in the Protocol Control register (after valid data has been read to the host system). In the case of a Write Transaction: This field is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request field in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this field is set when busy is deasserted. This field is not asserted in tuning process. 0 - Transfer does not complete 1 - Transfer complete
0 CC	Command complete This field is set when you receive the end field of the command response (except auto CMD12). See the Command Inhibit (CMD) in the Present State register. This field is not asserted in tuning process. 0 - Command not complete 1 - Command complete

32.7.1.15 Interrupt Status Enable (INT_STATUS_EN)

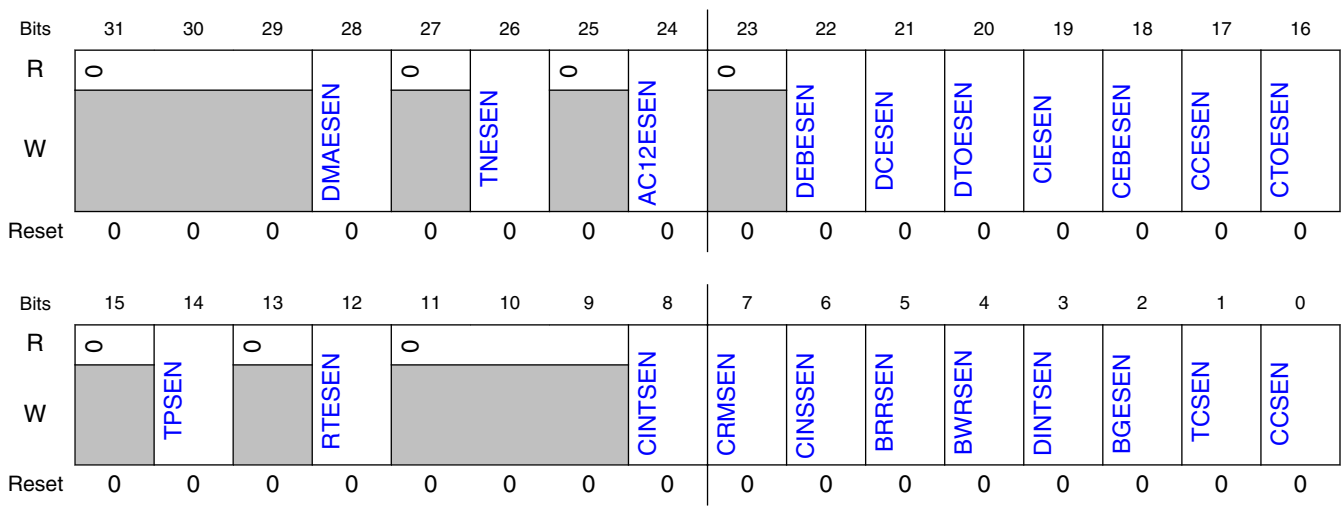
Setting any bit in this register to 1 enables the corresponding interrupt status be requested to the system. If any bit is set to 0, the corresponding interrupt request gets blocked.

- Depending on IABG field setting, uSDHC might be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There are some delays on the Card Interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

32.7.1.15.1 Offset

Register	Offset
INT_STATUS_EN	34h

32.7.1.15.2 Diagram



32.7.1.15.3 Fields

Field	Description
31-29 —	Reserved
28 DMAESEN	DMA error status enable 0 - Masked 1 - Enabled
27 —	Reserved
26	Tuning error status enable

Table continues on the next page...

Field	Description
TNESEN	0 - Masked 1 - Enabled
25 —	Reserved
24 AC12ESEN	Auto CMD12 error status enable 0 - Masked 1 - Enabled
23 —	Reserved
22 DEBESEN	Data end bit error status enable 0 - Masked 1 - Enabled
21 DCESEN	Data CRC error status enable 0 - Masked 1 - Enabled
20 DTOESEN	Data timeout error status enable 0 - Masked 1 - Enabled
19 CIESEN	Command index error status enable 0 - Masked 1 - Enabled
18 CEBESEN	Command end bit error status enable 0 - Masked 1 - Enabled
17 CCESEN	Command CRC error status enable 0 - Masked 1 - Enabled
16 CTOESEN	Command timeout error status enable 0 - Masked 1 - Enabled
15 —	Reserved
14 TPSEN	Tuning pass status enable 0 - Masked 1 - Enabled
13 —	Reserved
12 RTESEN	Re-tuning event status enable 0 - Masked

Table continues on the next page...

Field	Description
	1 - Enabled
11-9 —	Reserved
8 CINTSEN	Card interrupt status enable If this field is set to 0, uSDHC clears the interrupt request to the system. The Card Interrupt detection is stopped when this field is cleared and restarted when this field is set to 1. The host driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this field again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0 - Masked 1 - Enabled
7 CRMSEN	Card removal status enable 0 - Masked 1 - Enabled
6 CINSSEN	Card insertion status enable 0 - Masked 1 - Enabled
5 BRRSEN	Buffer read ready status enable 0 - Masked 1 - Enabled
4 BWRSEN	Buffer write ready status enable 0 - Masked 1 - Enabled
3 DINTSEN	DMA interrupt status enable 0 - Masked 1 - Enabled
2 BGESEN	Block gap event status enable 0 - Masked 1 - Enabled
1 TCSEN	Transfer complete status enable 0 - Masked 1 - Enabled
0 CCSEN	Command complete status enable 0 - Masked 1 - Enabled

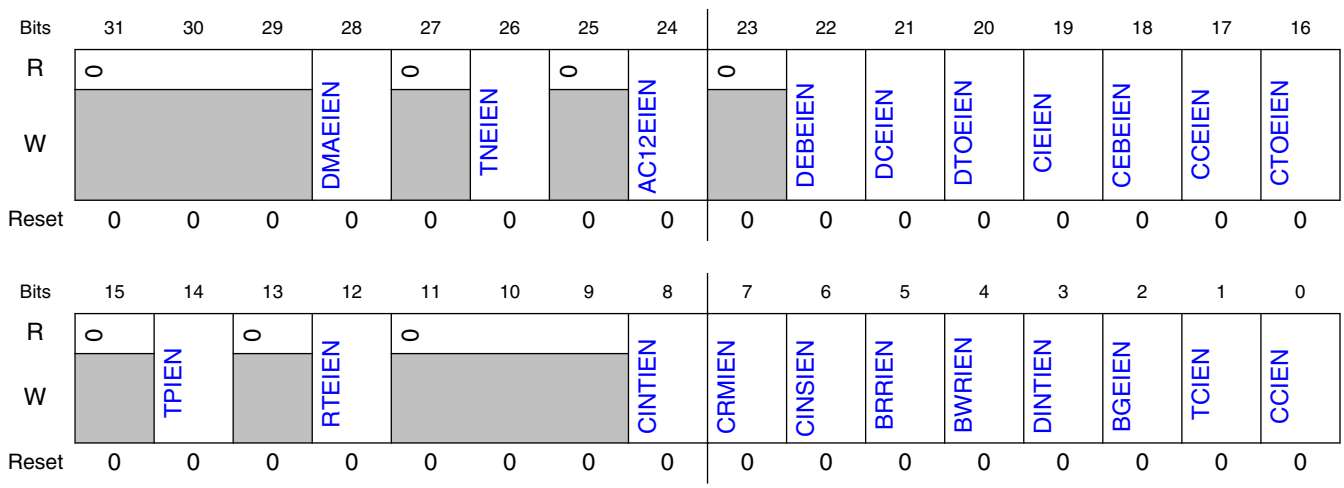
32.7.1.16 Interrupt Signal Enable (INT_SIGNAL_EN)

This register is used to select which interrupt status is indicated to the host system as the interrupt. These status fields all share the same interrupt lines. Setting any of these fields to 1 enables interrupt generation. The corresponding Status register field generates an interrupt when the corresponding interrupt signal enable field is set.

32.7.1.16.1 Offset

Register	Offset
INT_SIGNAL_EN	38h

32.7.1.16.2 Diagram



32.7.1.16.3 Fields

Field	Description
31-29 —	Reserved
28 DMAEIEN	DMA error interrupt enable 0 - Masked 1 - Enable
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0 - Masked 1 - Enabled
25	Reserved

Table continues on the next page...

Field	Description
—	
24 AC12EIEN	Auto CMD12 error interrupt enable 0 - Masked 1 - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable 0 - Masked 1 - Enabled
21 DCEIEN	Data CRC error interrupt enable 0 - Masked 1 - Enabled
20 DTOEIEN	Data timeout error interrupt enable 0 - Masked 1 - Enabled
19 CIEIEN	Command index error interrupt enable 0 - Masked 1 - Enabled
18 CEBEIEN	Command end bit error interrupt enable 0 - Masked 1 - Enabled
17 CCEIEN	Command CRC error interrupt enable 0 - Masked 1 - Enabled
16 CTOEIEN	Command timeout error interrupt enable 0 - Masked 1 - Enabled
15 —	Reserved
14 TPIEN	Tuning Pass interrupt enable 0 - Masked 1 - Enabled
13 —	Reserved
12 RTEIEN	Re-tuning event interrupt enable 0 - Masked 1 - Enabled
11-9 —	Reserved

Table continues on the next page...

Field	Description
8 CINTIEN	Card interrupt enable 0 - Masked 1 - Enabled
7 CRMIEN	Card removal interrupt enable 0 - Masked 1 - Enabled
6 CINSIEN	Card insertion interrupt enable 0 - Masked 1 - Enabled
5 BRRIEN	Buffer read ready interrupt enable 0 - Masked 1 - Enabled
4 BWRIEN	Buffer write ready interrupt enable 0 - Masked 1 - Enabled
3 DINTIEN	DMA interrupt enable 0 - Masked 1 - Enabled
2 BGEIEN	Block gap event interrupt enable 0 - Masked 1 - Enabled
1 TCIEN	Transfer complete interrupt enable 0 - Masked 1 - Enabled
0 CCIEN	Command complete interrupt enable 0 - Masked 1 - Enabled

32.7.1.17 Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)

When the Auto CMD12 Error Status field in the Status register is set, the host driver checks this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in field 04-01. This register is valid only when the Auto CMD12 Error status field is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 32-12. Relationship between command CRC error and command timeout error for auto CMD12

Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

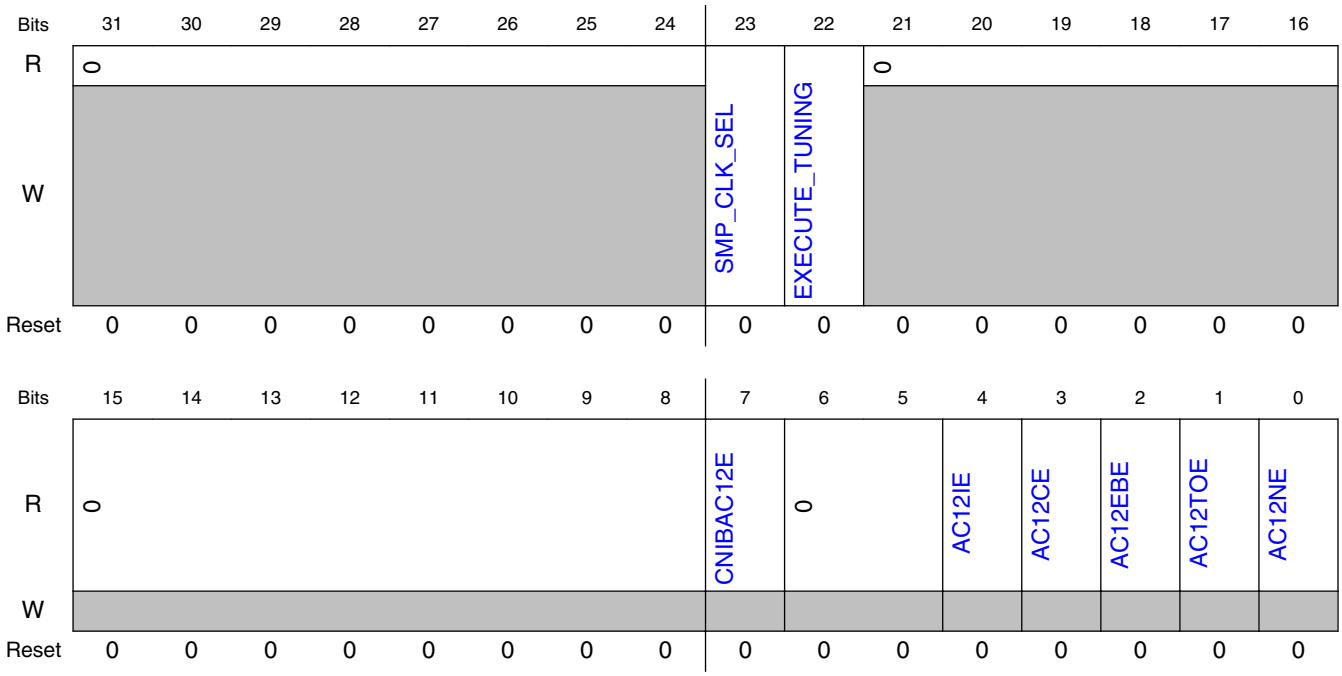
- When uSDHC is going to issue an Auto CMD12
  - Set field 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
  - Set field 0 to 0 if the Auto CMD12 is issued
- At the end field of an Auto CMD12 response
  - Check errors correspond to fields 1-4
  - Set fields 1-4 corresponding to detected errors
  - Clear fields 1-4 corresponding to detected errors
- Before reading the Auto CMD12 Error Status field 7
  - Set field 7 to 1 if there is a command that can't be issued
  - Clear field 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, field 7 is sampled when the driver is not writing to the Command register. So, it is suggested to read this register only when the AC12E field in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error fields (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

32.7.1.17.1 Offset

Register	Offset
AUTOCMD12_ERR_STATUS	3Ch

32.7.1.17.2 Diagram



32.7.1.17.3 Fields

Field	Description
31-24 —	Reserved
23 SMP_CLK_SEL	Sample clock select This field is used to select sampling clock to receive CMD and DATA. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is controlled by re-tuning procedure and is valid after the tuning procedure is complete (AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=0). A value of '1' means that tuning has been completed successfully and a value of '0' implies the tuning has failed. Writing '1' to this field is meaningless and ignored when the tuning has failed. The tuning circuit is reset by writing '0' to this field. This field can be cleared by setting AUTOCMD12_ERR_STATUS[EXECUTE_TUNING]=1. Once the tuning circuit is reset, it takes time to complete tuning sequence. Therefore, host driver should keep this field as 1 to perform the re-tuning sequence in a short time. This field cannot be changed while the Host controller is receiving response or a read data block. 0 - Fixed clock is used to sample data 1 - Tuned clock is used to sample data
22 EXECUTE_TUNING	Execute tuning This field is used to start the tuning procedure. This field is functional only when TUNING_CTRL[STD_TUNING_EN] is set. Otherwise, this field is reserved. This field is set to start tuning procedure and automatically cleared when tuning procedure is completed. The result of tuning is indicated to AUTOCMD12_ERR_STATUS[SMP_CLK_SEL]. 0 - Tuning procedure is aborted 1 - Start tuning procedure

Table continues on the next page...

Field	Description
21-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error Setting this field to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register. 0 - No error 1 - Not issued
6-5 —	Reserved
4 AC12IE	Auto CMD12 / 23 index error Occurs if the command index error occurs in response to a command. 0 - No error 1 - Error, the CMD index in response is not CMD12/23
3 AC12CE	Auto CMD12 / 23 CRC error Occurs when detecting a CRC error in the command response. 0 - No CRC error 1 - CRC error met in Auto CMD12/23 response
2 AC12EBE	Auto CMD12 / 23 end bit error Occurs when detecting that the end field of command response is 0 which should be 1. 0 - No error 1 - End bit error generated
1 AC12TOE	Auto CMD12 / 23 timeout error Occurs if no response is returned within 64 SDCLK cycles from the end field of the command. If this field is set to1, the other error status fields (2-4) have no meaning. 0 - No error 1 - Time out
0 AC12NE	Auto CMD12 not executed If memory multiple block data transfer is not started, due to a command error, this field is not set because it is not necessary to issue an Auto CMD12. Setting this field to 1 means uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this field is set to 1, other error status fields (1-4) have no meaning. 0 - Executed 1 - Not executed

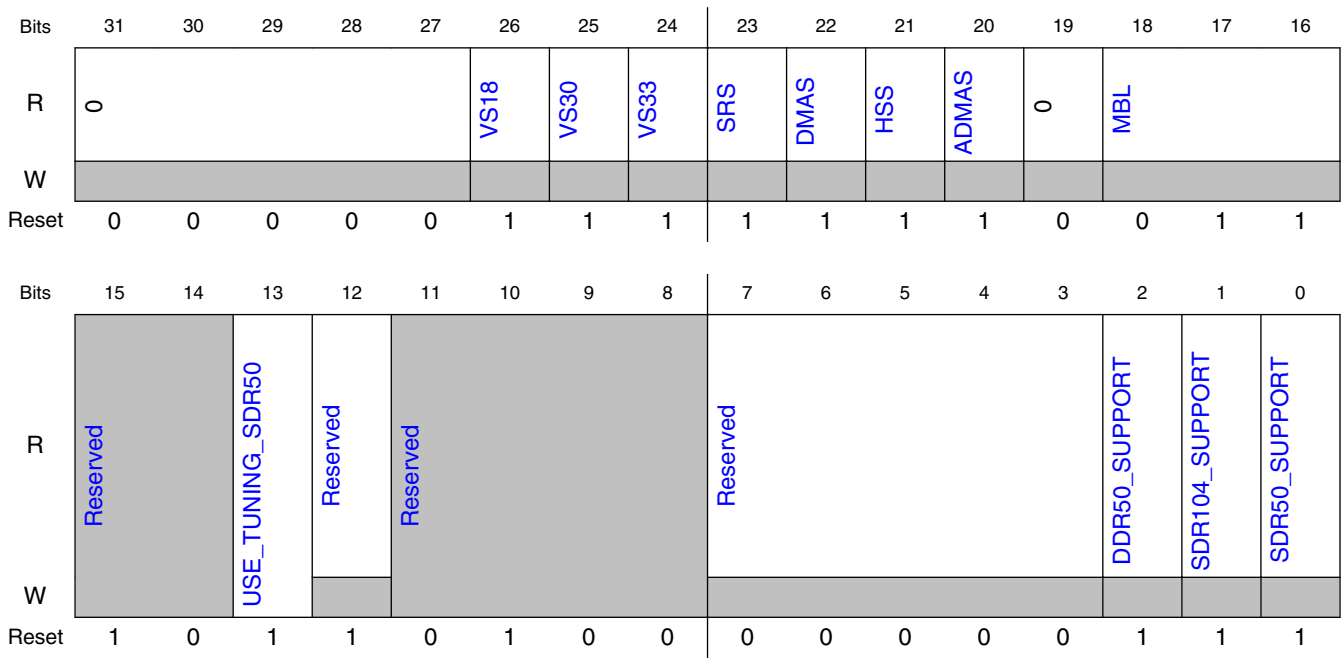
32.7.1.18 Host Controller Capabilities (HOST_CTRL_CAP)

This register provides the host driver with information specific to uSDHC implementation. The value in this register is the power-on-reset value and does not change with a software reset.

32.7.1.18.1 Offset

Register	Offset
HOST_CTRL_CAP	40h

32.7.1.18.2 Diagram



32.7.1.18.3 Fields

Field	Description
31-27 —	Reserved
26 VS18	Voltage support 1.8 V This field depends on the host system ability. 0 - 1.8 V not supported 1 - 1.8 V supported
25	Voltage support 3.0 V

Table continues on the next page...

Field	Description
VS30	This field depends on the host system ability. 0 - 3.0 V not supported 1 - 3.0 V supported
24 VS33	Voltage support 3.3 V This field depends on the host system ability. 0 - 3.3 V not supported 1 - 3.3 V supported
23 SRS	Suspend / resume support This field indicates whether uSDHC supports Suspend / Resume functionality. If this field is 0, the Suspend and Resume mechanism, as well as the read wait, are not supported, and the host driver does not issue either Suspend or Resume commands. 0 - Not supported 1 - Supported
22 DMAS	DMA support This field indicates whether uSDHC can use the internal DMA to transfer data between system memory and the data buffer directly. 0 - DMA not supported 1 - DMA supported
21 HSS	High speed support This field indicates whether uSDHC supports High Speed mode and the host system can supply a SD clock frequency from 25 MHz to 50 MHz. 0 - High speed not supported 1 - High speed supported
20 ADMAS	ADMA support This field indicates whether uSDHC supports the ADMA feature. 0 - Advanced DMA not supported 1 - Advanced DMA supported
19 —	Reserved
18-16 MBL	Max block length This field indicates the maximum block size that the host driver can read and write to the buffer in uSDHC. The buffer transfers block size without wait cycles. 000 - 512 bytes 001 - 1024 bytes 010 - 2048 bytes 011 - 4096 bytes
15-14 —	Reserved
13 USE_TUNING_SDR50	Use Tuning for SDR50 This field is used to enable tuning for SDR50 mode 0 - SDR50 does not support tuning

Table continues on the next page...

Field	Description
	1 - SDR50 supports tuning
12 —	Reserved
11-8 —	Reserved
7-3 —	Reserved
2 DDR50_SUPPORT	DDR50 support This field indicates support of DDR50 mode.
1 SDR104_SUPPORT	SDR104 support This field indicates support of SDR104 mode.
0 SDR50_SUPPORT	SDR50 support This field indicates support of SDR50 mode.

32.7.1.19 Watermark Level (WTMK_LVL)

This register indicates configurability of write and read watermark levels (FIFO threshold). Their value can range from 1 to 128 words.

32.7.1.19.1 Offset

Register	Offset
WTMK_LVL	44h

32.7.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			Reserved					WR_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			Reserved					RD_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

32.7.1.19.3 Fields

Field	Description
31-29 —	Reserved
28-24 —	Reserved
23-16 WR_WML	Write watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also, the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.When using internal DMA, WR_WML can be set any value between 1 and 128. To get max AXI throughput, we recommend to use integer multiple of 16.
15-13 —	Reserved
12-8 —	Reserved
7-0 RD_WML	Read watermark level This field indicates the number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also, the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.When using internal DMA mode, RD_WML must be set to the value less than 16. To get max AXI throughput, recommend to use 16.

32.7.1.20 Mixer Control (MIX_CTRL)

This register is used to DMA and data transfer. To prevent data loss, the software should check if data transfer is active before writing this register. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

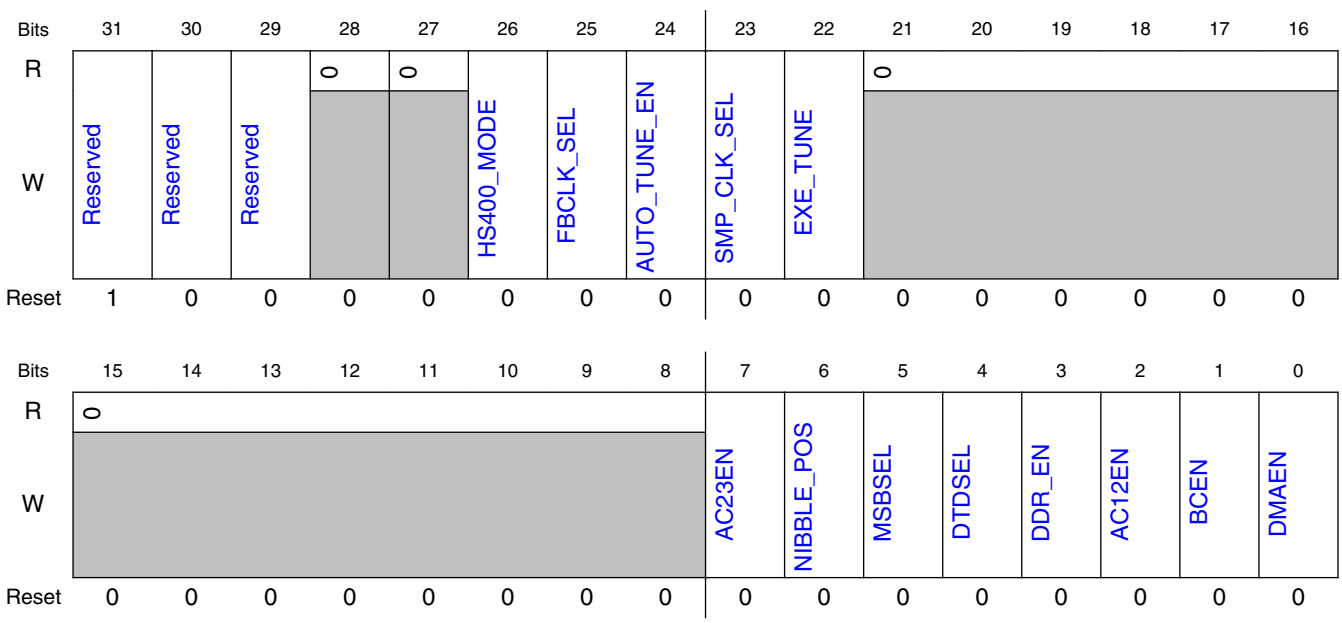
Table 32-13. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

32.7.1.20.1 Offset

Register	Offset
MIX_CTRL	48h

32.7.1.20.2 Diagram



32.7.1.20.3 Fields

Field	Description
31	Reserved
—	Always write as 1.
30	Reserved
—	Always write as 0.

Table continues on the next page...

Field	Description
29 —	Reserved Always write as 0.
28 —	Reserved
27 —	Reserved
26 HS400_MODE	Enable HS400 mode HS400 Enable
25 FBCLK_SEL	Feedback clock source selection (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) 0 - Feedback clock comes from the loopback CLK 1 - Feedback clock comes from the ipp_card_clk_out
24 AUTO_TUNE_EN	Auto tuning enable (Only used for SD3.0, SDR104 mode and and EMMC HS200 mode) 0 - Disable auto tuning 1 - Enable auto tuning
23 SMP_CLK_SEL	Clock selection When TUNING_CTRL[STD_TUNING_EN] is 0, this field is used to select Tuned clock or Fixed clock to sample data / cmd (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) 0 - Fixed clock is used to sample data / cmd 1 - Tuned clock is used to sample data / cmd
22 EXE_TUNE	Execute tuning: (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) When TUNING_CTRL[STD_TUNING_EN] is 0, this field is set to 1 to indicate the host driver is starting tuning procedure. Tuning procedure is aborted by writing 0. 0 - Not tuned or tuning completed 1 - Execute tuning
21-8 —	Reserved
7 AC23EN	Auto CMD23 enable When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	Nibble position indication This field indicates the nibble position in the DDR 4-bit mode. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single block select This field enables multiple block DATA line data transfers. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0 - Single block 1 - Multiple blocks
4 DTDSEL	Data transfer direction select This field defines the direction of DATA line data transfers. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands.

Table continues on the next page...

Field	Description
	If a CMD with the response data size cannot be divided by 64 bytes, is given without a STOP CMD, and is followed by a software reset, then you must write a 0 to DTDSEL before inserting a software reset. This is necessary to avoid any issue during the SD card reinitialization. Read multiple blocks will not be affected as it is required to have a STOP CMD. 0 - Write (Host to card) 1 - Read (Card to host)
3 DDR_EN	Dual data rate mode selection
2 AC12EN	Auto CMD12 enable Multiple block transfers for memory require a CMD12 to stop the transaction. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0 - Disable 1 - Enable
1 BCEN	Block count enable This field is used to enable the Block Count register, which is only relevant for multiple block transfers. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0 - Disable 1 - Enable
0 DMAEN	DMA enable This field enables DMA functionality. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0 - Disable 1 - Enable

32.7.1.21 Force Event (FORCE_EVENT)

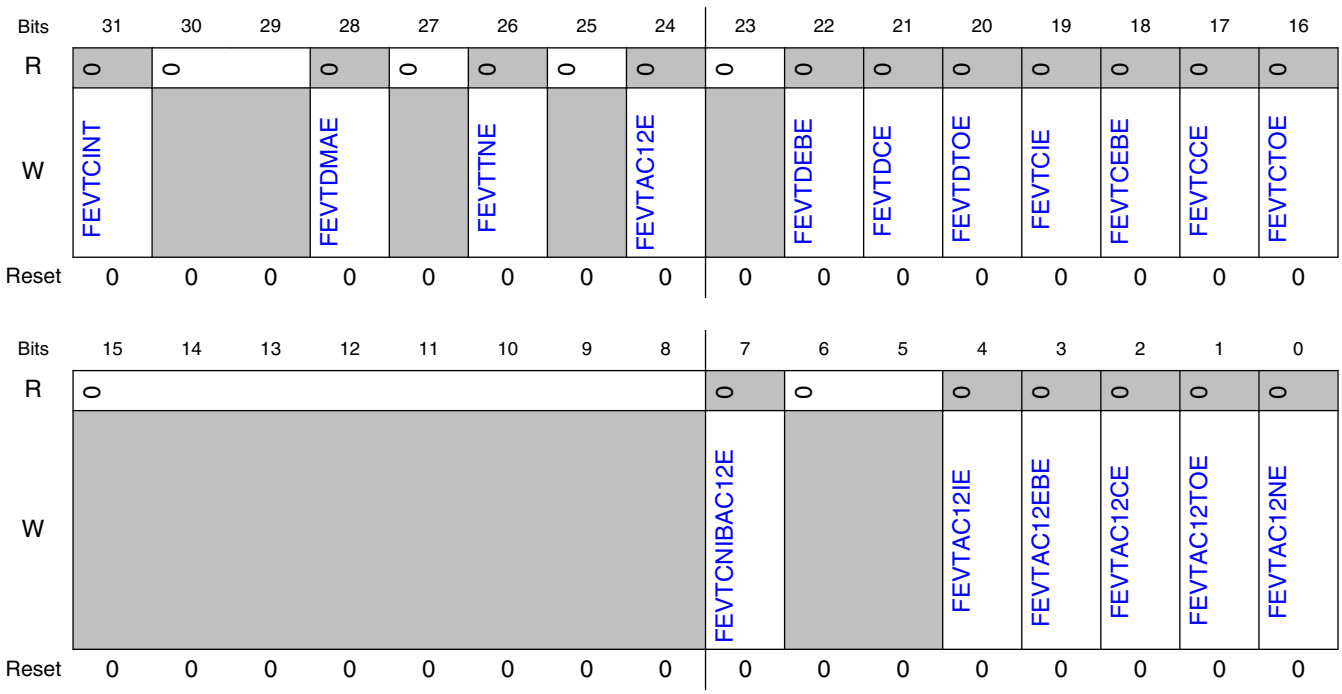
This register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding field of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register sets the corresponding field of Interrupt Status register. A read from this register always results in 0's. to change the corresponding status fields in the Interrupt Status register, make sure to set IPGEN field in System Control Register so that peripheral clock is always active.

Forcing a card interrupt generates a short pulse on the DATA1 line, and the driver might treat this interrupt as a normal interrupt. The interrupt service routine might skip polling the card interrupt factor as the interrupt is self cleared.

32.7.1.21.1 Offset

Register	Offset
FORCE_EVENT	50h

32.7.1.21.2 Diagram



32.7.1.21.3 Fields

Field	Description
31 FEVTCINT	Force event card interrupt Writing 1 to this field generates a short low-level pulse on the internal DATA1 line, as if a self-clearing interrupt was received from the external card. If enabled, the CINT field is set and the interrupt service routine might treat this interrupt as a normal interrupt from the external card.
30-29 —	Reserved
28	Force event DMA error

Table continues on the next page...

Field	Description
FEVTDMAE	Forces the DMAE field of Interrupt Status register to be set
27 —	Reserved
26 FEVTTNE	Force tuning error Forces the TNE field of Interrupt Status register to be set
25 —	Reserved
24 FEVTAC12E	Force event Auto Command 12 error Forces the AC12E field of Interrupt Status register to be set
23 —	Reserved
22 FEVTDEBE	Force event data end bit error Forces the DEBE field of Interrupt Status register to be set
21 FEVTDCE	Force event data CRC error Forces the DCE field of Interrupt Status register to be set
20 FEVTDTOE	Force event data time out error Force the DTOE field of Interrupt Status register to be set
19 FEVTCIE	Force event command index error Forces the CCE field of Interrupt Status register to be set
18 FEVTCEBE	Force event command end bit error Forces the CEBE field of Interrupt Status register to be set
17 FEVTCCE	Force event command CRC error Forces the CCE field of Interrupt Status register to be set
16 FEVTCTOE	Force event command time out error Forces the CTOE field of Interrupt Status register to be set
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto Command 12 error Forces the CNIBAC12E field in the Auto Command12 Error Status register to be set
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto Command 12 index error Forces the AC12IE field in the Auto Command12 Error Status register to be set
3 FEVTAC12EBE	Force event Auto Command 12 end bit error Forces the AC12EBE field in the Auto Command12 Error Status register to be set
2 FEVTAC12CE	Force event auto command 12 CRC error Forces the AC12CE field in the Auto Command12 Error Status register to be set
1 FEVTAC12TOE	Force event auto command 12 time out error Forces the AC12TOE field in the Auto Command12 Error Status register to be set

Table continues on the next page...

Field	Description
0 FEVTAC12NE	Force event auto command 12 not executed Forces the AC12NE field in the Auto Command12 Error Status register to be set

32.7.1.22 ADMA Error Status (ADMA_ERR_STATUS)

When an ADMA Error Interrupt has occurred, the ADMA error sates field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

In case of a write operation, the host driver should use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data might exist in the host controller.

The host controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The host driver can distinguish this error by reading the Valid field of the error descriptor.

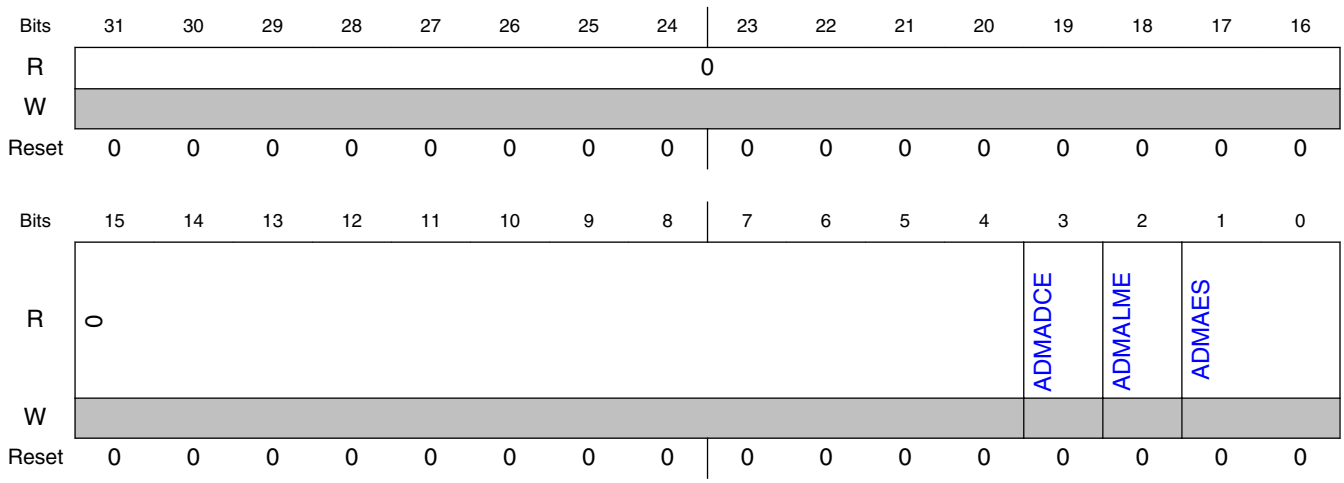
Table 32-14. ADMA error state coding

D01-D00	ADMA error state (when error has occurred)	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (Fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (Change address)	No ADMA error is generated
11	ST_TFR (Transfer data)	Holds the address of the next executable descriptor command

32.7.1.22.1 Offset

Register	Offset
ADMA_ERR_STATUS	54h

32.7.1.22.2 Diagram



32.7.1.22.3 Fields

Field	Description
31-4 —	Reserved
3 ADMADCE	ADMA descriptor error This error occurs when invalid descriptor fetched by ADMA. 0 - No error 1 - Error
2 ADMALME	ADMA length mismatch error This error occurs in the following two cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length cannot be divided by the block length. 0 - No error 1 - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred) This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. See ADMA Error Status (ADMA_ERR_STATUS) for more details.

32.7.1.23 ADMA System Address (ADMA_SYS_ADDR)

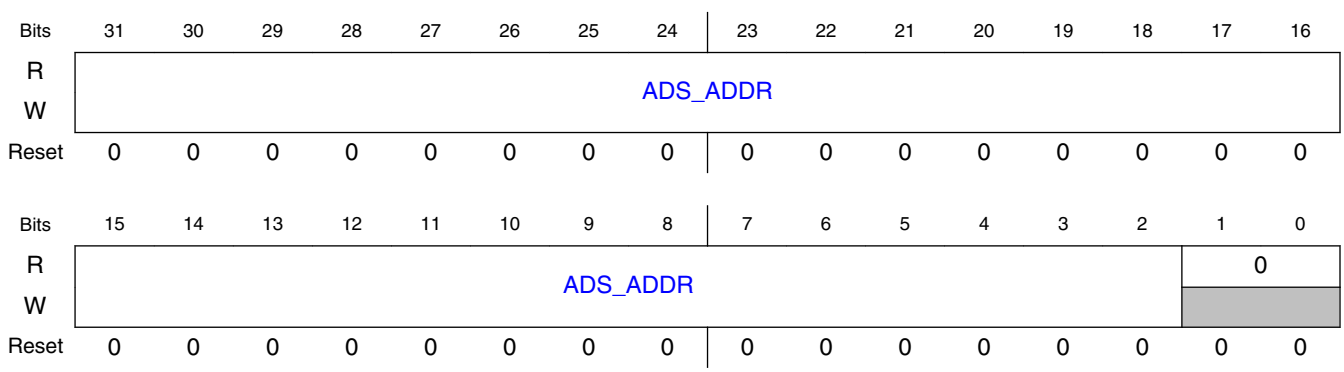
This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the host driver sets the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned.

Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions.

32.7.1.23.1 Offset

Register	Offset
ADMA_SYS_ADDR	58h

32.7.1.23.2 Diagram



32.7.1.23.3 Fields

Field	Description
31-2	ADMA system address

Table continues on the next page...

Field	Description
ADS_ADDR	This field contains the physical system memory address used for ADMA transfers.
1-0 —	Reserved

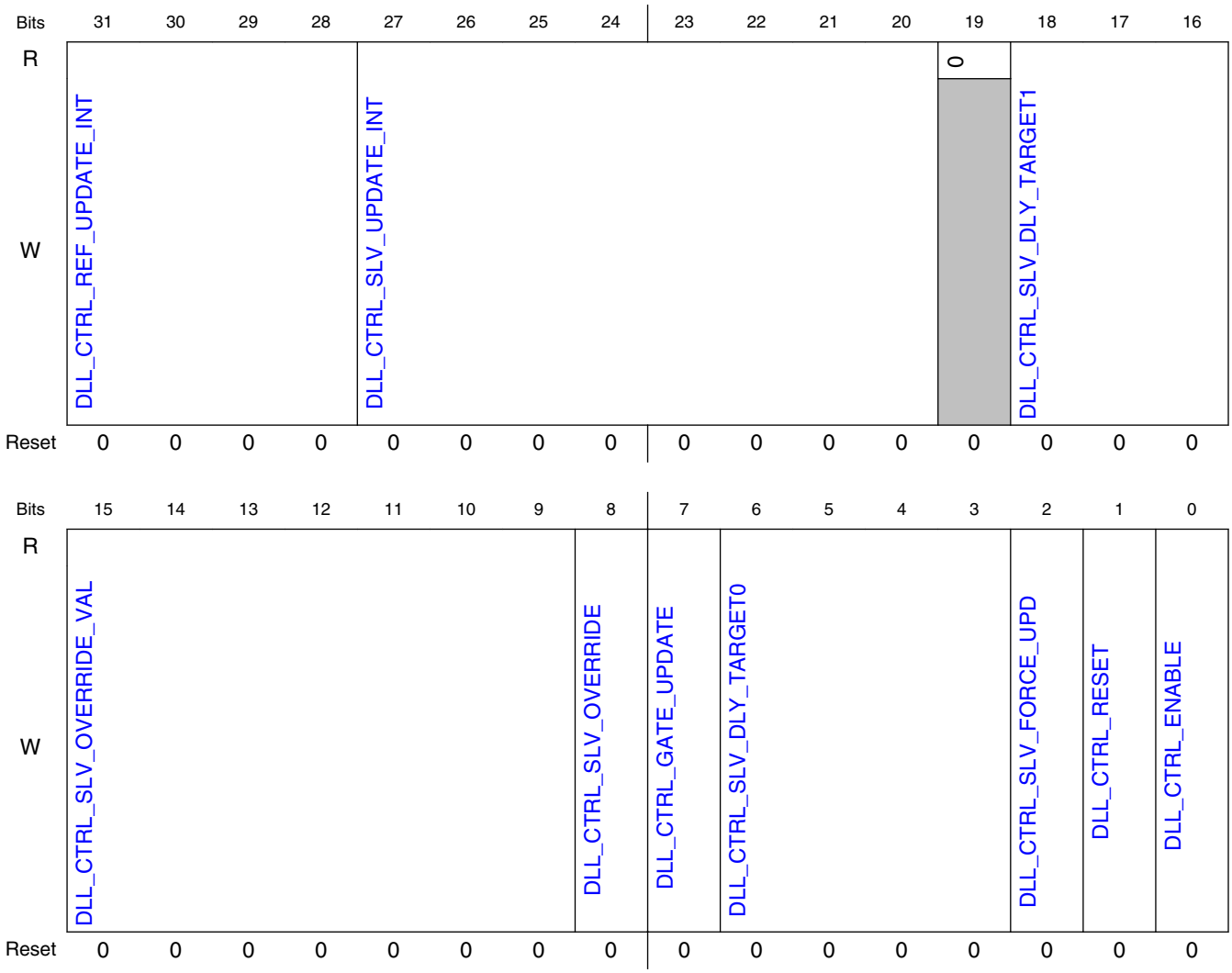
32.7.1.24 DLL (Delay Line) Control (DLL_CTRL)

This register contains control fields for DLL.

32.7.1.24.1 Offset

Register	Offset
DLL_CTRL	60h

32.7.1.24.2 Diagram



32.7.1.24.3 Fields

Field	Description
31-28 DLL_CTRL_REF_UPDATE_INT	DLL control loop update interval The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop updates every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 DLL_CTRL_SLV_UPDATE_INT	Slave delay line update interval If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19	Reserved

Table continues on the next page...

Field	Description
—	
18-16 DLL_CTRL_SLV_DLY_TARGET1	DLL slave delay target1 See DLL_CTRL_SLV_DLY_TARGET0 below.
15-9 DLL_CTRL_SLV_OVERRIDE_VAL	DLL slave override val When SLV_OVERRIDE = 1, this field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_SLV_OVERRIDE	DLL slave override Set this field to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_GATE_UPDATE	DLL gate update Set this field to 1 to prevent the DLL from updating (because when clock_in exists, glitches might appear during DLL updates). This field might be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6-3 DLL_CTRL_SLV_DLY_TARGET0	DLL slave delay target0 The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an ref_clock half-period. The delay is (({DLL_CTRL_SLV_DLY_TARGET1,DLL_CTRL_SLV_DLY_TARGET0} +1)* REF_CLOCK / 2) / 16 So the input read-clock can be delayed relative input data from (REF_CLOCK / 2) / 16 to REF_CLOCK * 4. **NOTE:** For the restrictions of delay cell implementation, the delay target must be set between REF_CLOCK/16 and REF_CLOCK*2 when REF_CLOCK is running at 200 MHz. When REF_CLOCK frequency is slower than 100 MHz, the maximum delay target might not reach REF_CLOCK*2.
2 DLL_CTRL_SLV_FORCE_UPD	DLL slave delay line Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line updates automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 DLL_CTRL_RESET	DLL reset Setting this field to 1 force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, so to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ENABLE	DLL and delay chain Set this field to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

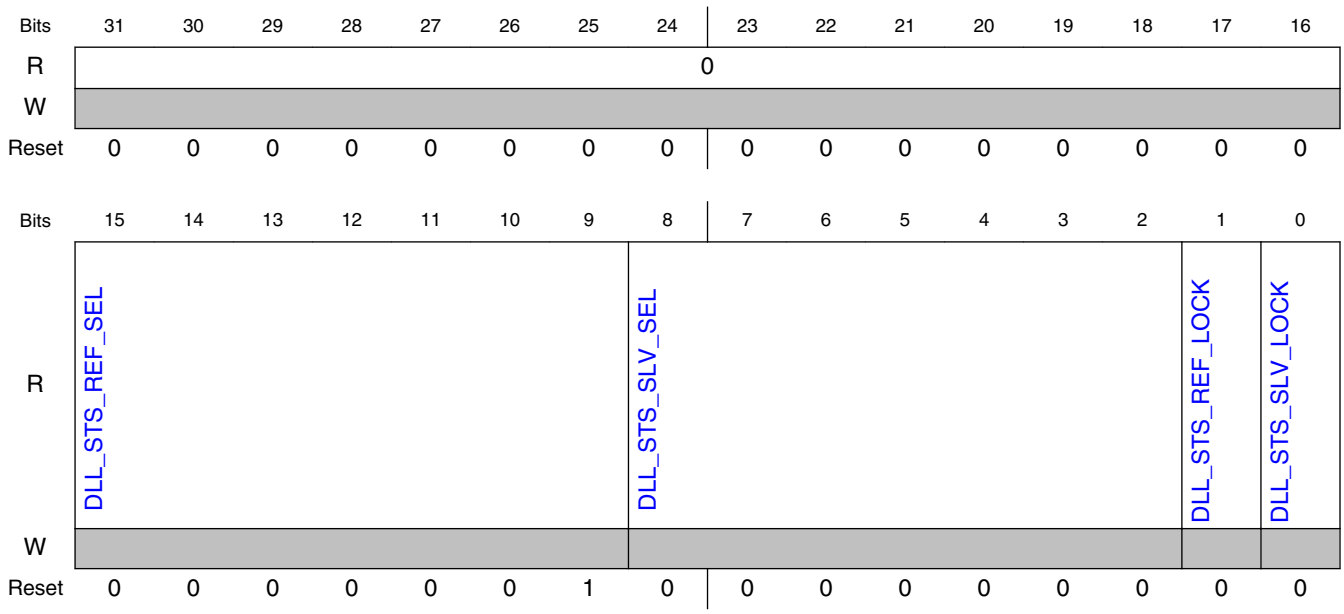
32.7.1.25 DLL Status (DLL_STATUS)

This register contains the DLL status information. All fields are read only and reads the same as the power-reset value.

32.7.1.25.1 Offset

Register	Offset
DLL_STATUS	64h

32.7.1.25.2 Diagram



32.7.1.25.3 Fields

Field	Description
31-16 —	Reserved
15-9 DLL_STS_REF_SEL	Reference delay line select taps This is encoded by 7 fields for 127 taps.
8-2 DLL_STS_SLV_SEL	Slave delay line select status This is the instant value generated from reference chain. Because the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1	Reference DLL lock status

Table continues on the next page...

Field	Description
DLL_STS_REF_LOCK	This signifies that the DLL has detected and locked to a half-phase ref_clock shift, allowing the slave delay-line to perform programmed clock delays
0	Slave delay-line lock status
DLL_STS_SLV_LOCK	This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

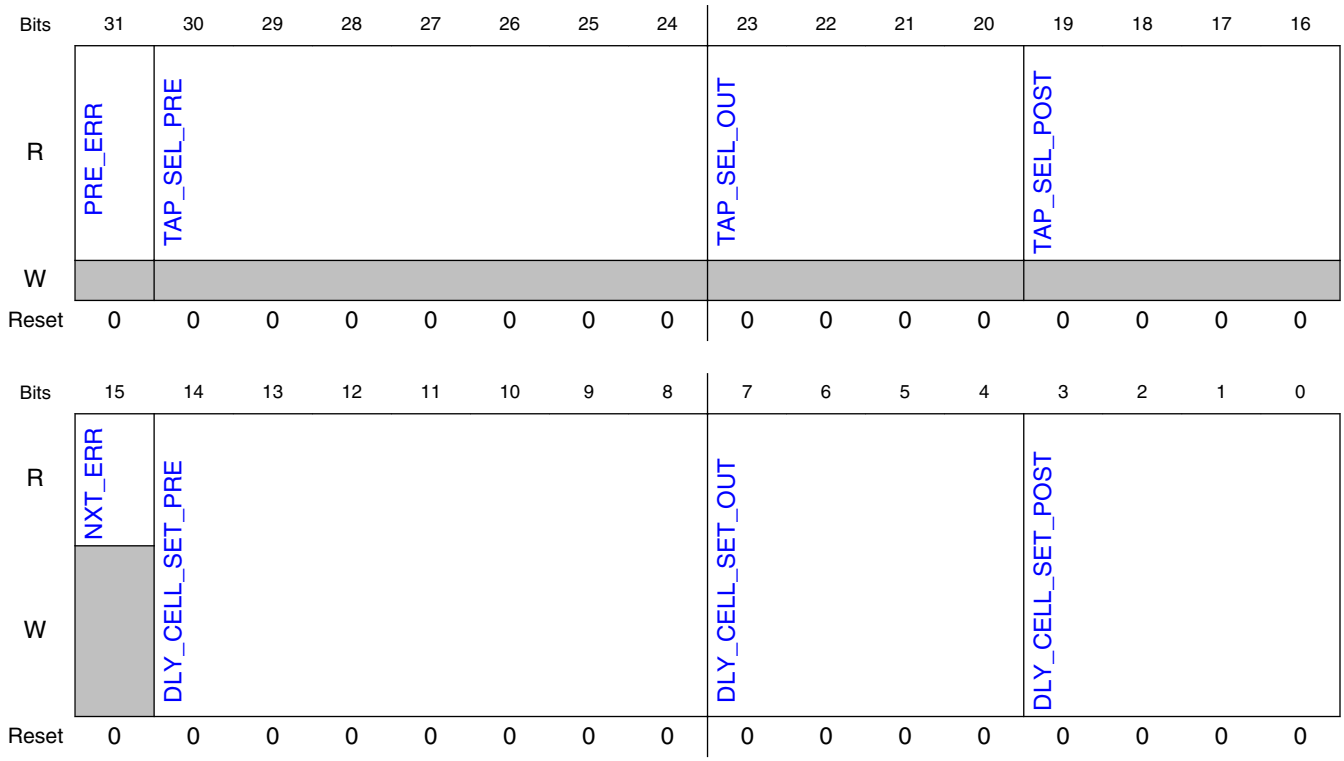
32.7.1.26 CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)

This register contains the Clock Tuning Control status information. All fields are read only and reads the same as the power-reset value. This register is added to support SD3.0 UHS-I SDR104 mode and EMMC HS200 mode.

32.7.1.26.1 Offset

Register	Offset
CLK_TUNE_CTRL_STATUS	68h

32.7.1.26.2 Diagram



32.7.1.26.3 Fields

Field	Description
31 PRE_ERR	PRE error PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
30-24 TAP_SEL_PRE	TAP_SEL_PRE Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE is updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23-20 TAP_SEL_OUT	Delay cells added on the feedback clock between CLK_PRE and CLK_OUT Reflects the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19-16 TAP_SEL_POST	Delay cells added on the feedback clock between CLK_OUT and CLK_POST Reflects the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.

Table continues on the next page...

Field	Description
14-8 DLY_CELL_SET_PRE	delay cells on the feedback clock between the feedback clock and CLK_PRE Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7-4 DLY_CELL_SET_OUT	Delay cells on the feedback clock between CLK_PRE and CLK_OUT Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.
3-0 DLY_CELL_SET_POST	Delay cells on the feedback clock between CLK_OUT and CLK_POST Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

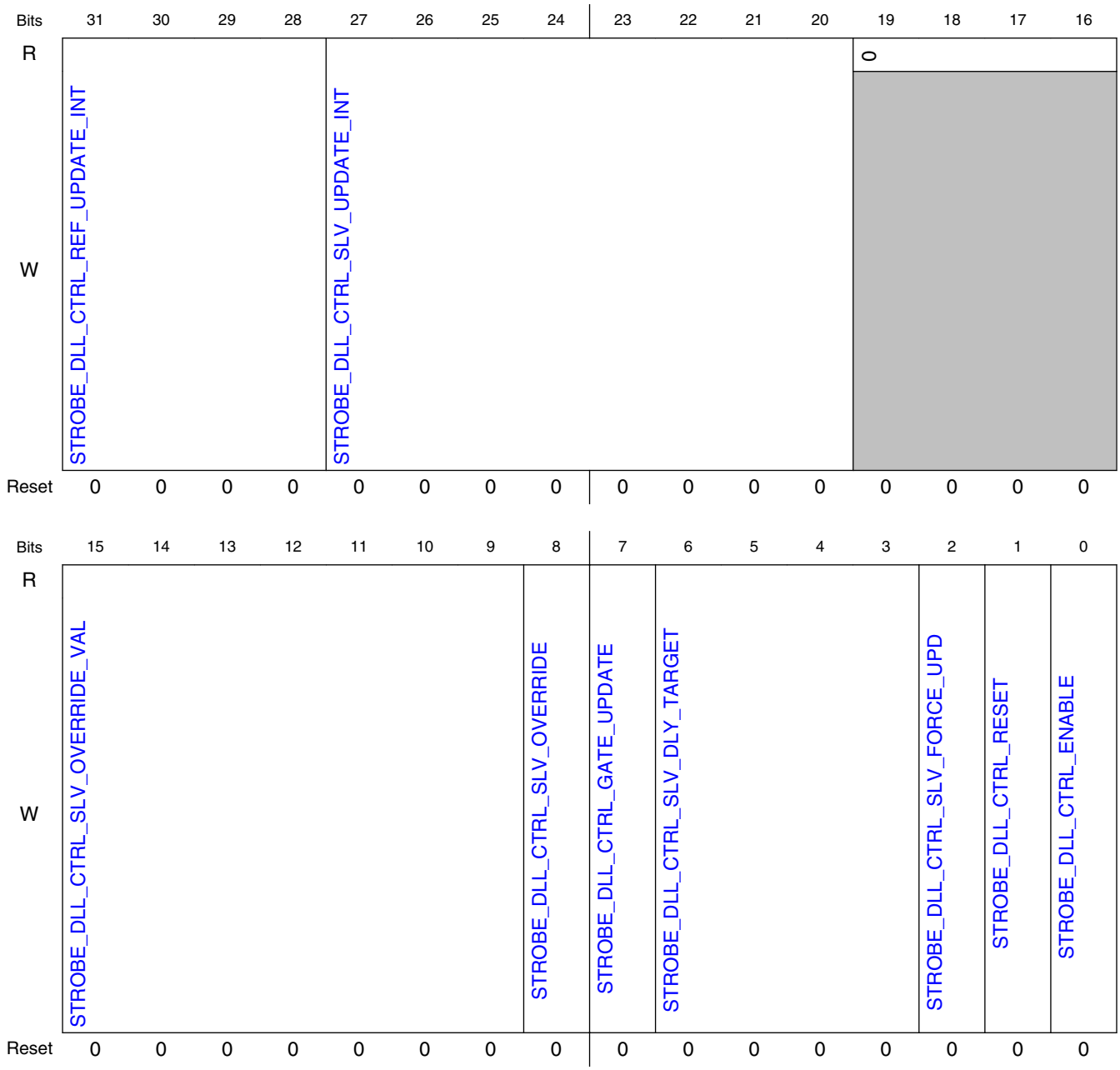
32.7.1.27 Strobe DLL control (STROBE_DLL_CTRL)

This register contains the strobe DLL control information.

32.7.1.27.1 Offset

Register	Offset
STROBE_DLL_CTRL	70h

32.7.1.27.2 Diagram



32.7.1.27.3 Fields

Field	Description
31-28 STROBE_DLL_CTRL_REF_UPDATE_INT	Strobe DLL control reference update interval The interval cycle is (2 + STROBE_REF_UPDATE_INT) * STROBE_REF_CLOCK. By default, the DLL control loop updates every two STROBE_REF_CLOCK cycles.

Table continues on the next page...

Field	Description
	NOTE: Increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 STROBE_DLL_CTRL_SLV_UPDATE_INT	Strobe DLL control slave update interval Slave delay line update interval. If default 0 is used, it means 256 cycles of STROBE_REF_CLOCK. A value of 0x0F results in 15 cycles and so on. **NOTE:** software can always cause an update of the slave-delay line using the STROBE_SLV_FORCE_UPDATE register. The slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_CTRL_SLV_OVERRIDE_VAL	Strobe DLL control slave Override value When STROBE_SLV_OVERRIDE = 1, this field is used to manually select one of 128 physical taps. A value of 0 selects tap 1, and a value of 0x7F selects tap 128.
8 STROBE_DLL_CTRL_SLV_OVERRIDE	Strobe DLL control slave override Set this field to 1 to Enable manual override for slave delay chain using STROBE_SLV_OVERRIDE_VAL; set this field to 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if STROBE_SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0.
7 STROBE_DLL_CTRL_GATE_UPDATE	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
6-3 STROBE_DLL_CTRL_SLV_DLY_TARGET	Strobe DLL Control Slave Delay Target The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an STROBE_REF_CLOCK half-period. The delay is ((STROBE_DLL_CTRL_SLV_DLY_TARGET + 1) * STROBE_REF_CLOCK / 2) / 16, So the input read-clock can be delayed relative input data from (STROBE_REF_CLOCK / 2) / 16 to (STROBE_REF_CLOCK *4)/16. **NOTE:** For the restrictions of delay cell implementation, the minimum delay target that can be set is REF_CLOCK/16.
2 STROBE_DLL_CTRL_SLV_FORCE_UPD	Strobe DLL control slave force updated Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line should automatically update the STROBE_SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if STROBE_SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 STROBE_DLL_CTRL_RESET	Strobe DLL control reset Setting this field to 1 to force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, to create a subsequent reset, RESET must be taken low and then asserted again.
0 STROBE_DLL_CTRL_ENABLE	Strobe DLL control enable Set this field to 1 to enable the DLL and delay chain; otherwise, set to 0 to bypasses DLL. **NOTE:** Using the slave delay line override feature with STROBE_SLV_OVERRIDE and STROBE_SLV_OVERRIDE VAL, the DLL does not need to be enabled.

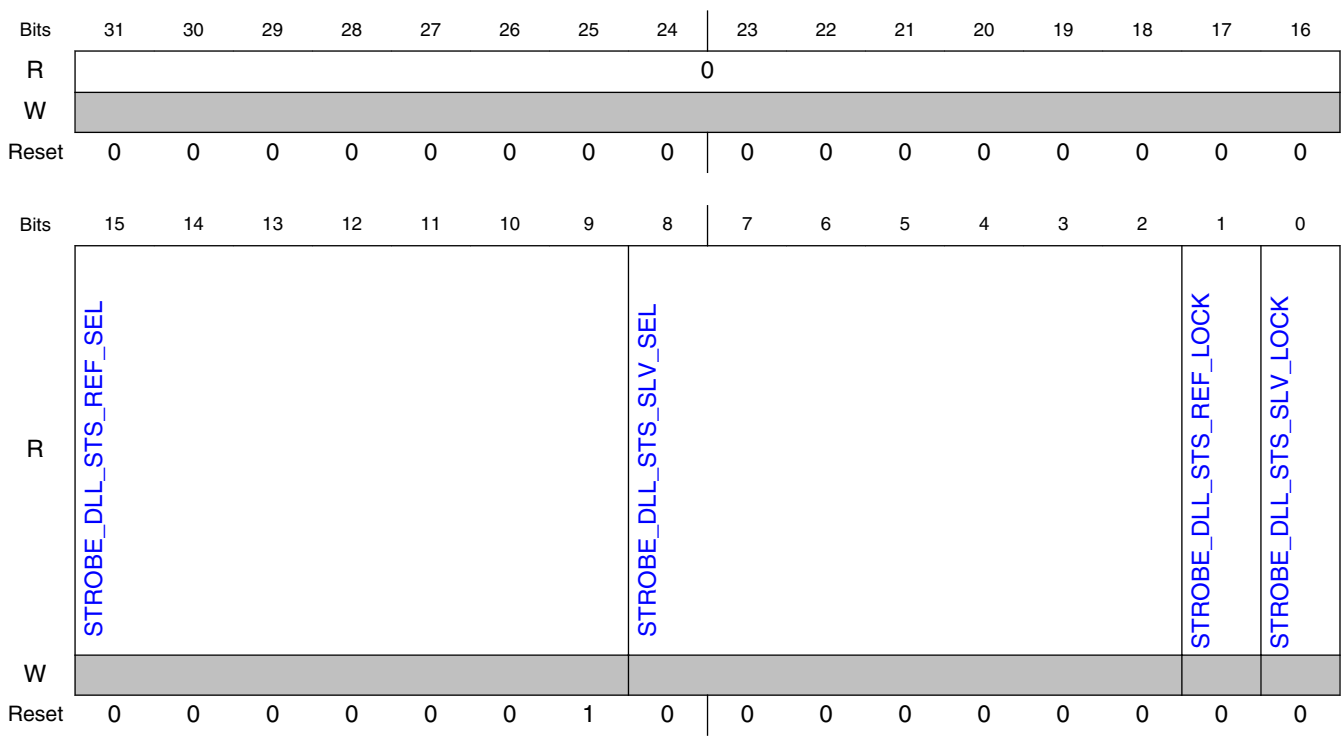
32.7.1.28 Strobe DLL status (STROBE_DLL_STATUS)

This register contains the strobe DLL status information. All fields are read only and read the same as the power-reset value.

32.7.1.28.1 Offset

Register	Offset
STROBE_DLL_STATUS	74h

32.7.1.28.2 Diagram



32.7.1.28.3 Fields

Field	Description
31-16	This field is reserved.
—	This read-only field is reserved and always has the value 0.

Table continues on the next page...

Field	Description
15-9 STROBE_DLL_STS_REF_SEL	Strobe DLL status reference select Reference delay line select taps. This is encoded by 7 fields for 127 taps.
8-2 STROBE_DLL_STS_SLV_SEL	Strobe DLL status slave select Slave delay line select status. This is the instant value generated from reference chain. Because the reference chain can only be updated when STROBE_REF_CLOCK is detected, this value can be updated with the right value when the reference is locked.
1 STROBE_DLL_STS_REF_LOCK	Strobe DLL status reference lock This signifies that the DLL has detected and locked to a half-phase REF_CLOCK shift, it allows the slave delay-line to perform programmed clock delays.
0 STROBE_DLL_STS_SLV_LOCK	Strobe DLL status slave lock Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line, and the slave-delay line is implementing the programmed delay value.

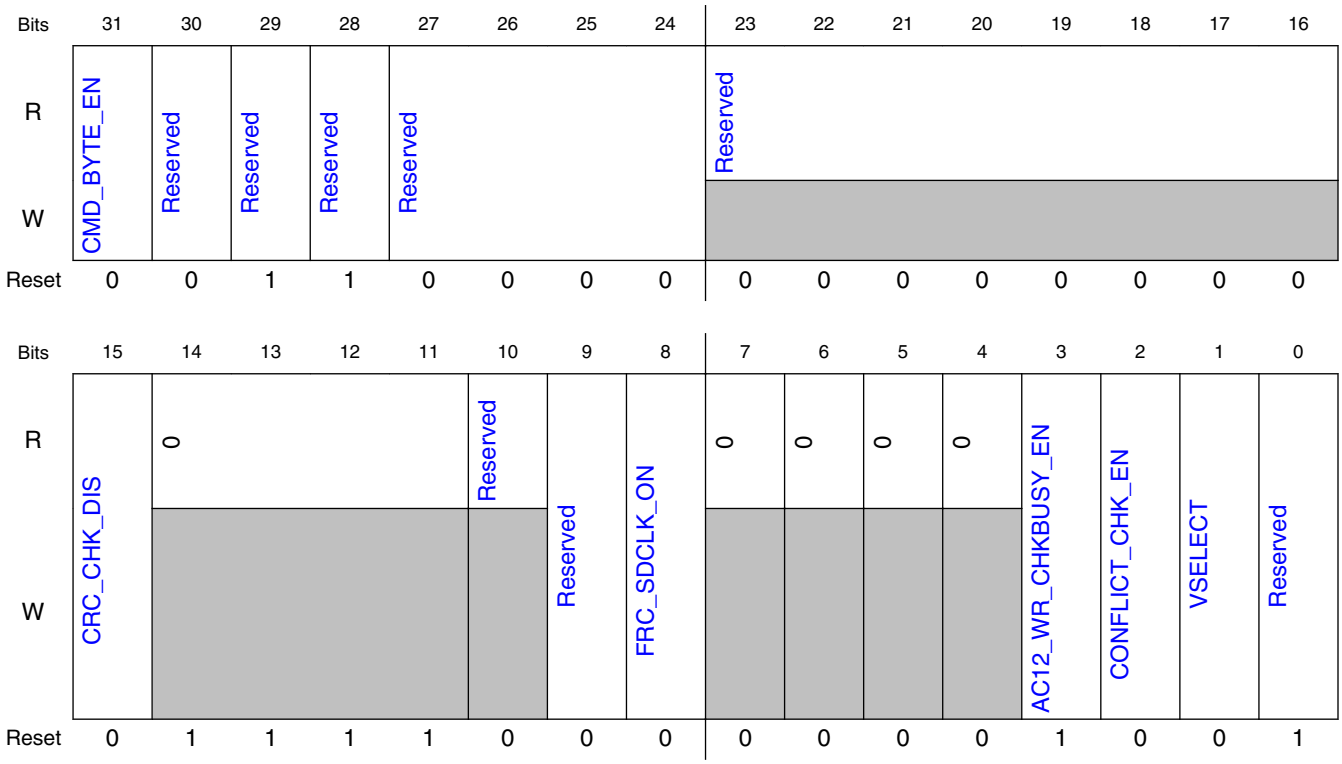
32.7.1.29 Vendor Specific Register (VEND_SPEC)

This register contains the vendor specific control/status register.

32.7.1.29.1 Offset

Register	Offset
VEND_SPEC	C0h

32.7.1.29.2 Diagram



32.7.1.29.3 Fields

Field	Description
31 CMD_BYTE_EN	Byte access This field controls the byte access. 0 - Disable 1 - Enable
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 1
28 —	Reserved Always write as 0.
27-24 —	Reserved Always write as 4'b0000.
23-16 —	Reserved Always write as 8'h00.
15 CRC_CHK_DIS	CRC Check Disable 0 - Check CRC16 for every read data packet and check CRC fields for every write data packet

Table continues on the next page...

Field	Description
	1 - Ignore CRC16 check for every read data packet and ignore CRC fields check for every write data packet
14-11	Reserved
—	Reserved
10	Reserved
—	Always write as 0.
9	Reserved
—	Always write as 0.
8	Force CLK
FRC_SDCLK_ON	Force CLK output active
	Do not set this bit to 1 unless it is necessary. Also, make sure that this bit is cleared when uSDHC's clock is about to be changed (frequency change, clock source change, or delay chain tuning).
	0 - CLK active or inactive is fully controlled by the hardware.
	1 - Force CLK active
7	Reserved
—	
6	Reserved
—	
5	Reserved
—	
4	Reserved
—	
3	Check busy enable
AC12_WR_CHKBUSY_EN	Check busy enable after auto CMD12 for write data packet
	0 - Do not check busy after auto CMD12 for write data packet
	1 - Check busy after auto CMD12 for write data packet
2	Conflict check enable
CONFLICT_CHK_EN	It is not implemented in uSDHC IP.
	0 - Conflict check disable
	1 - Conflict check enable
1	Voltage selection
VSELECT	Change the value of output signal VSELECT to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads.
	0 - Change the voltage to high voltage range, around 3.0 V
	1 - Change the voltage to low voltage range, around 1.8 V
0	Reserved
—	Always write as 0.

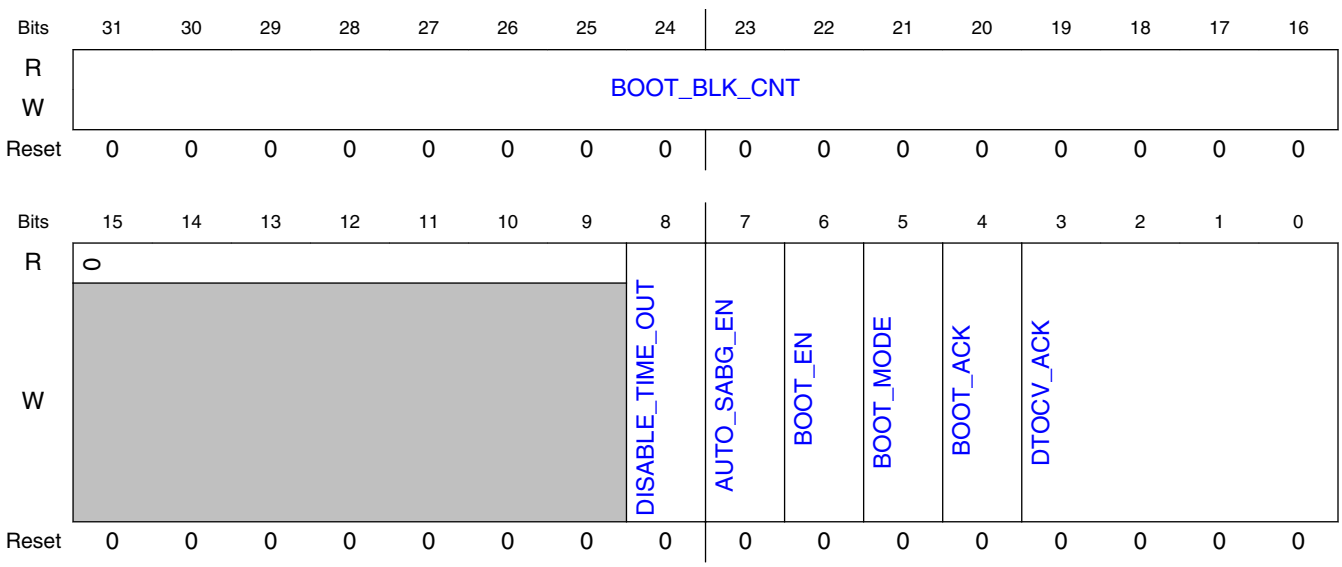
32.7.1.30 MMC Boot (MMC_BOOT)

This register contains the MMC Fast Boot control register.

32.7.1.30.1 Offset

Register	Offset
MMC_BOOT	C4h

32.7.1.30.2 Diagram



32.7.1.30.3 Fields

Field	Description
31-16 BOOT_BLK_CNT	Stop At Block Gap value of automatic mode The value defines the Stop At Block Gap value of automatic mode. When received, card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Attributes Register, field 31 - 16 of 0x04.
15-9 —	Reserved
8 DISABLE_TIME_OUT	Time out **NOTE:** When this field is set, there is no timeout check no matter whether BOOT_EN is set or not. 0 - Enable time out

Table continues on the next page...

Field	Description
	1 - Disable time out
7 AUTO_SABG_EN	Auto stop at block gap During boot, enable auto stop at block gap function. This function is triggered, and host stops at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot enable Boot mode enable 0 - Fast boot disable 1 - Fast boot enable
5 BOOT_MODE	Boot mode Boot mode select 0 - Normal boot 1 - Alternative boot
4 BOOT_ACK	BOOT ACK Boot ACK mode select 0 - No ack 1 - Ack
3-0 DTOCV_ACK	Boot ACK time out Boot ACK time out counter value. 0000 - SDCLK x 2^14 0001 - SDCLK x 2^15 0010 - SDCLK x 2^16 0011 - SDCLK x 2^17 0100 - SDCLK x 2^18 0101 - SDCLK x 2^19 0110 - SDCLK x 2^20 0111 - SDCLK x 2^21 1110 - SDCLK x 2^28 1111 - SDCLK x 2^29

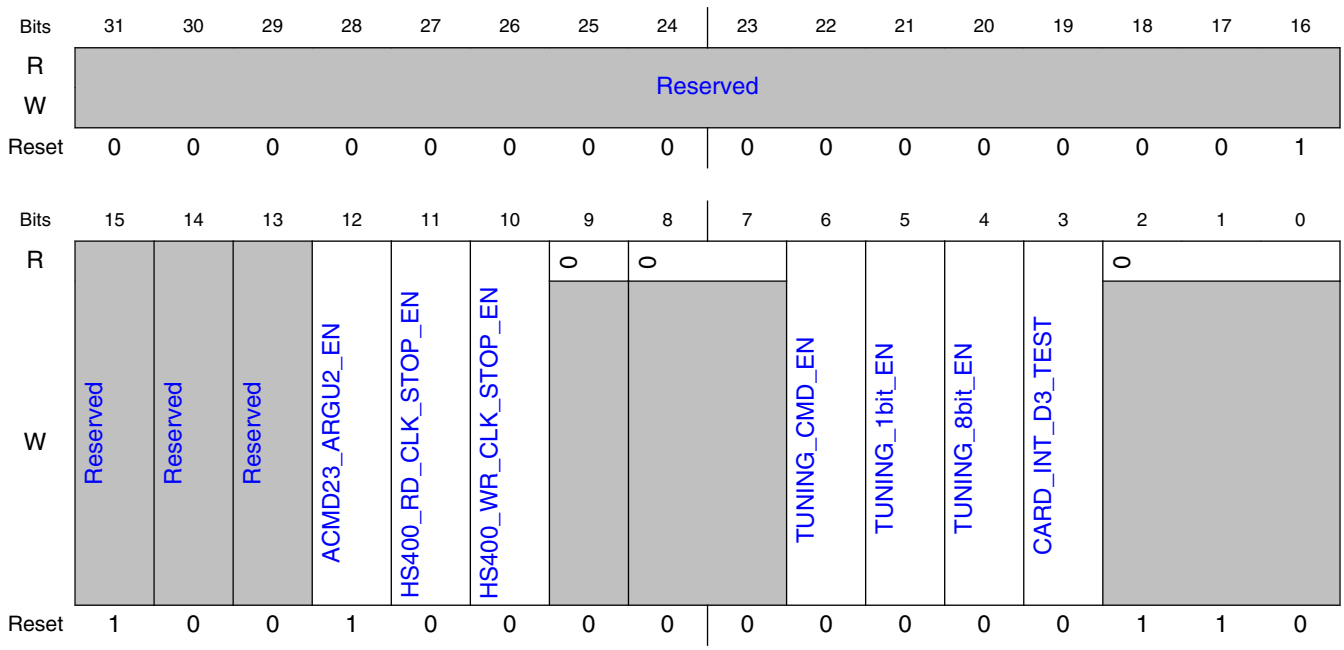
32.7.1.31 Vendor Specific 2 Register (VEND_SPEC2)

This register contains the vendor specific control 2 register.

32.7.1.31.1 Offset

Register	Offset
VEND_SPEC2	C8h

32.7.1.31.2 Diagram



32.7.1.31.3 Fields

Field	Description
31-15 —	Reserved
14 —	Reserved
13 —	Reserved
12 ACMD23_ARGU2_EN	Argument2 register enable for ACMD23 0 - Disable 1 - Argument2 register enable for ACMD23 sharing with SDMA system address register. Default is enabled.
11 HS400_RD_CLK_STOP_EN	HS400 read clock stop enable Only stop clock at read block gap.

Table continues on the next page...

Field	Description
10 HS400_WR_CLK_STOP_EN	HS400 write clock stop enable Only stop clock at write block gap.
9 —	Reserved
8-7 —	Reserved
6 TUNING_CMD_EN	Tuning command enable Enable the auto tuning circuit to check the CMD line. 0 - Auto tuning circuit does not check the CMD line. 1 - Auto tuning circuit checks the CMD line.
5 TUNING_1bit_EN	Tuning 1bit enable Enable the auto tuning circuit to check the DATA0 only. It is used with the TUNING_8bit_EN together.
4 TUNING_8bit_EN	Tuning 8bit enable Enable the auto tuning circuit to check the DATA[7:0]. It is used with the TUNING_1bit_EN together. 0b00 - Tuning circuit only checks the DATA[3:0] 0b01 - Tuning circuit only checks the DATA0 0b10 - Tuning circuit checks the whole DATA[7:0] 0b11 - Invalid **NOTE:** The format of these two fields are [TUNNING_8bit_EN:TUNNING_1bit_EN].
3 CARD_INT_D3_TEST	Card interrupt detection test This field is used only for debugging. 0 - Check the card interrupt only when DATA3 is high. 1 - Check the card interrupt by ignoring the status of DATA3.
2-0 —	Reserved

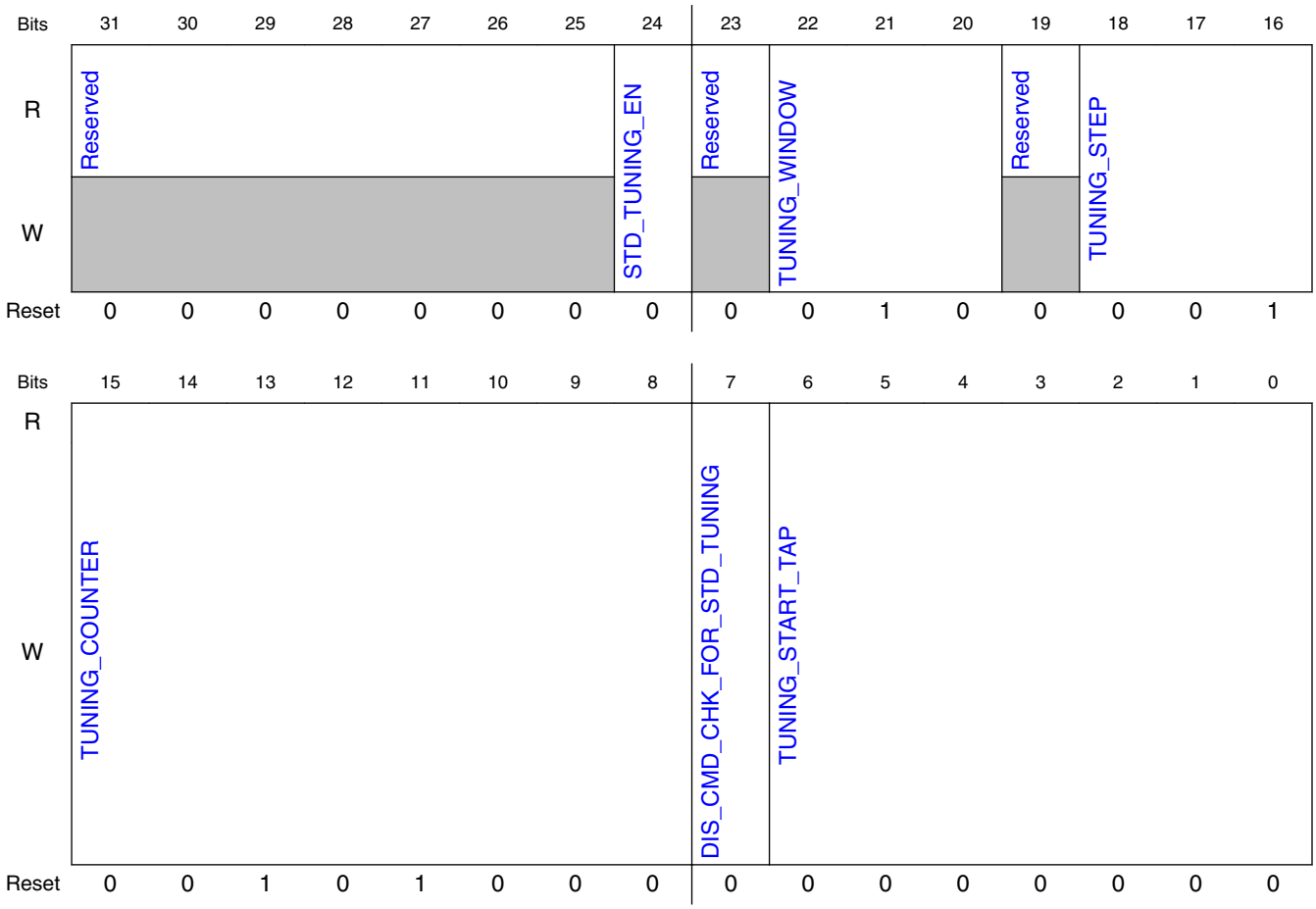
32.7.1.32 Tuning Control (TUNING_CTRL)

The register contains configuration of tuning circuit.

32.7.1.32.1 Offset

Register	Offset
TUNING_CTRL	CCh

32.7.1.32.2 Diagram



32.7.1.32.3 Fields

Field	Description
31-25 —	Reserved
24 STD_TUNING_EN	Standard tuning circuit and procedure enable This field is used to enable standard tuning circuit and procedure.
23 —	Reserved
22-20 TUNING_WINDOW	Data window Select data window value for auto tuning
19 —	Reserved
18-16	TUNING_STEP

Table continues on the next page...

Field	Description
TUNING_STEP	The increasing delay cell steps in tuning procedure.
15-8 TUNING_COUNTER	Tuning counter The MAX repeat CMD19 times in tuning procedure.
7 DIS_CMD_CHK_FOR_STD_TUNING	Disable command check for standard tuning Writing 1 to this field disables command check (command CRC, CMD end bit error, and CMD index error or CMD timeout error) for standard tuning flow after each tuning command is sent.
6-0 TUNING_START_TAP	Tuning start The start delay cell point when send first CMD19 in tuning procedure.

Chapter 33
ARM Cortex M7 Platform (M7)

33.1 Chip-specific Arm Cortex M7 information

Table 33-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments
System Debug	-	System Debug

The Arm Cortex-M7 core revision is CORTEX-M7-r1p2-00rel0. The CoreSight™ revision is CoreSight-SoC-TM100-r3p2. The list shows related Arm documents:

- DDI0403E_B_armv7m_arm.pdf
- DDI0489F_cortex_m7_r1p2_trm.pdf
- DUI0644E_cortex_m7_r1p2_ugrm.pdf
- DDI0480G_coresight_soc_r3p2_trm.pdf
- DUI0563G_coresight_soc_r3p2_ug.pdf

For this chip, the Cortex-M7 core configurations are listed in the following table.

Table 33-2. Cortex-M7 Core Configuration

Feature	Description	Configuration
FPU	Floating point unit (FPU) configuration	2
ICACHE	I-cache configuration	1
DCACHE	D-cache configuration	1

Table continues on the next page...

Table 33-2. Cortex-M7 Core Configuration (continued)

CACHEECC	Cache ECC configuration	1
MPU	MPU region configuration	16
IRQNUM	Number of IRQs	220
IRQLVL	IRQ priority width configuration	4
DBGLVL	Debug (breakpoint/watchpoint) configuration	1
TRC	Internal Trace support	1
ETM	ETM support	1
CTI	CTI support	1
WIC	WIC support	0
LOCKSTEP	Dual-redundant (lock-step) CPU functionality	0
RAR	Reset-all-registers	1
DW	Use Synopsys DesignWare	1
ICACHESIZE	I-cache Size (in k Bytes)	32
DCACHESIZE	D-cache Size (in k Bytes)	32

For the Cross Trigger Interface (CTI) integration in this chip, see the following diagram.

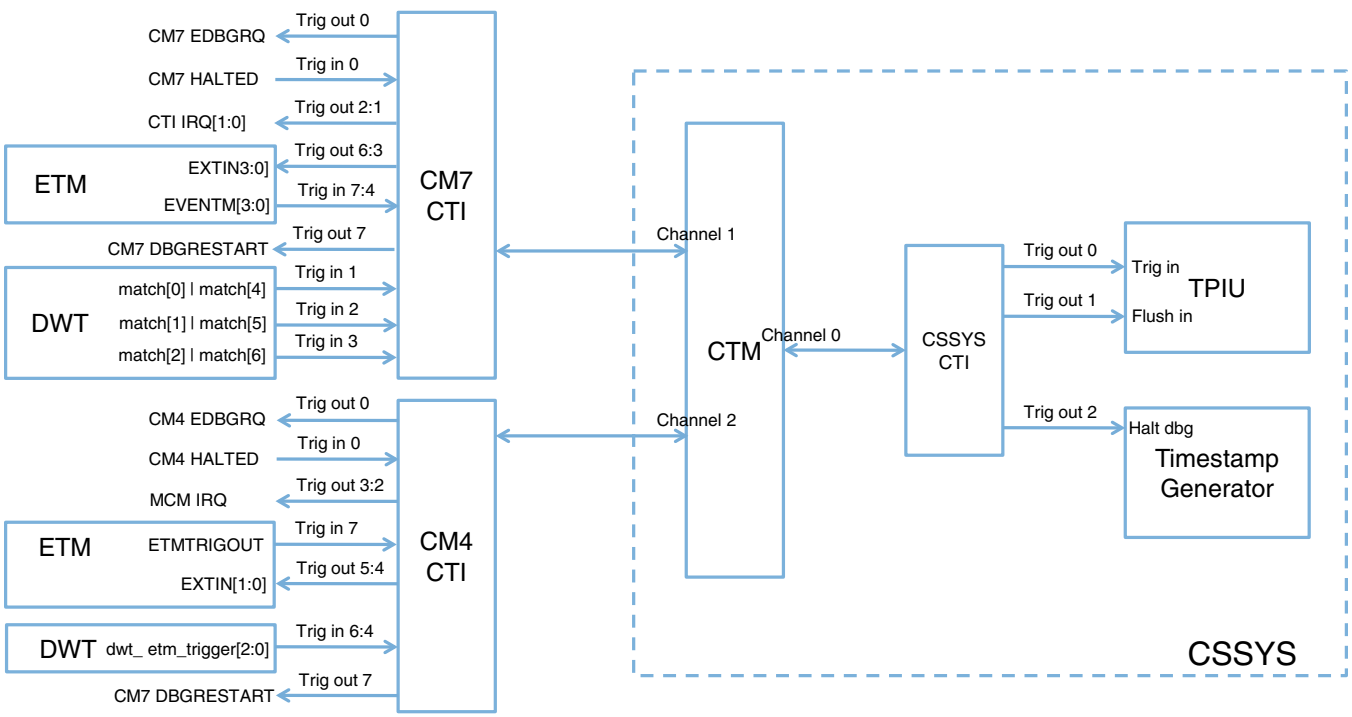


Figure 33-1. CTI inter-connections

33.2 Overview

The Cortex-M7 platform features a single Arm®Cortex®-M7 processor in this chip. The Arm®Cortex®-M7 processor is a highly efficient, high-performance, embedded processor that features low interrupt latency, low-cost debug, and has backwards compatibility with existing Cortex-M profile processors. The processor has an in-order super-scalar pipeline by which many instructions can be dual-issued, including load/load and load/store instruction pairs because of multiple memory interfaces.

33.2.1 Block Diagram

A block diagram for the Cortex-M7 is given below:

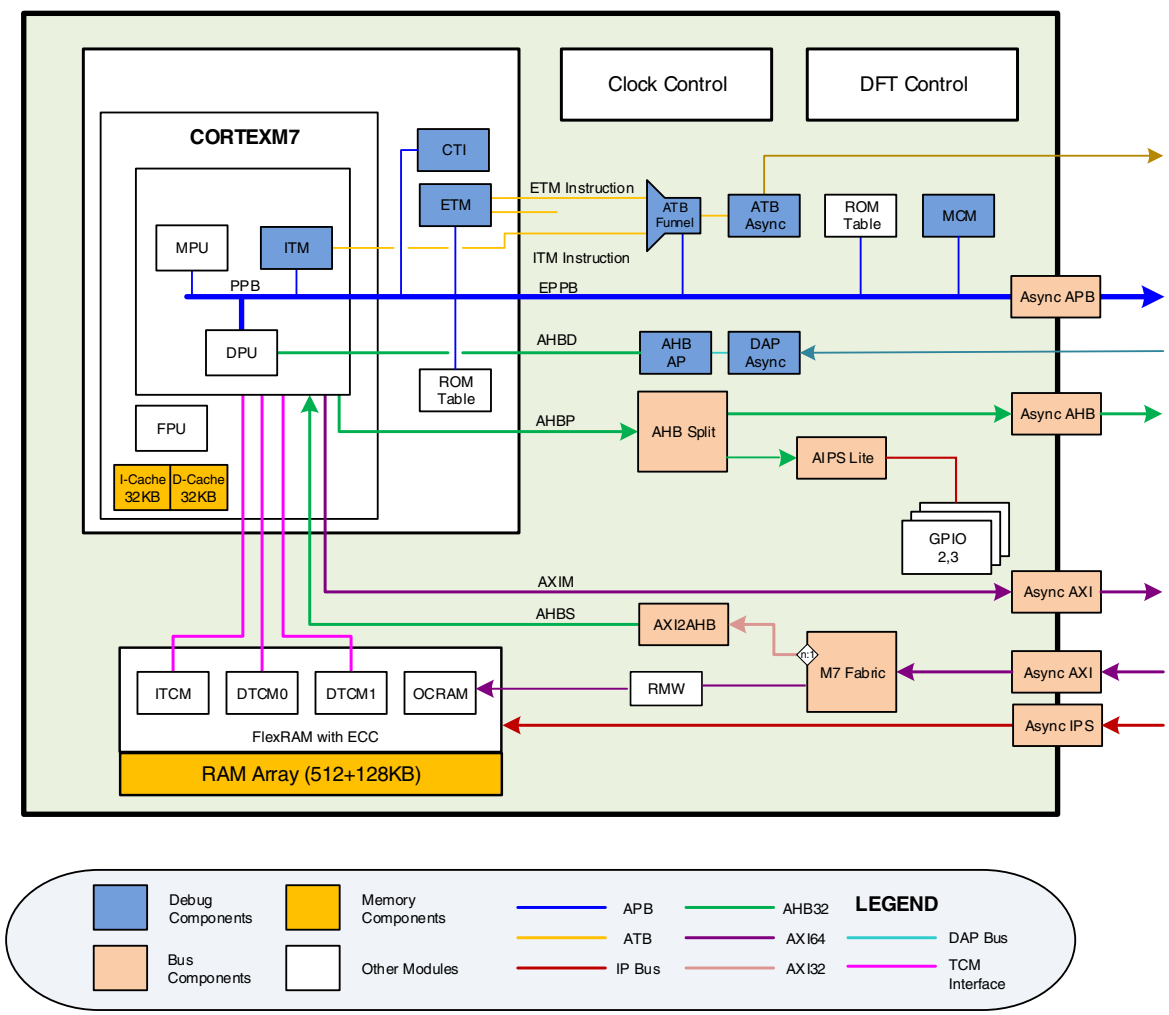


Figure 33-2. Cortex-M7 Platform Block Diagram

33.2.2 Features

The Cortex-M7 platform in this chip has the following features.

- 1× Cortex M7 processor: architecture Arm v7E-M
- floating point unit (FPU) FPv5 including single and double precision
- bus interface 64-bit AMBA4 AXI, 32-bit AHB peripheral port, 32-bit AMBA AHB slave port, AMBA APB interface for CoreSight debug components
- 32 KB instruction / 32 KB data L1 caches with ECC
- configurable size instruction and data TCM (default 256kB ITCM /256kB DTCM) with ECC interface
- 16 regions of MPU
- non-maskable (NMI) + 240 IRQs
- 16 interrupt priority levels (4-bits)
- sleep (WAIT) and deep sleep (STOP) power modes with integrated WFI, WFE, SLEEPONEXIT
- SWD and JTAG, with up to 8 breakpoints and 4 watchpoints
- dual core lock-step support DCLS

33.3 Functional Description

33.3.1 CM7_MCM sub-module

CM7_MCM is a miscellaneous control module for ARM Cortex-M7. It is connected onto the Cortex-M7 PPB bus. See the register descriptions for all its features.

33.3.2 Clocks

Table 33-3. Arm Cortex-M7 Clocks

Clock Name	Description
ipg_clk	Peripheral clock
axi_clk	Bus clock
main_clk	Arm core clock
trace_clk_in	Clock signal

33.3.3 Reset

The CM7_MCM module uses Cortex-M7's reset. Cold or soft resetting the CM7 core would also reset this module.

33.3.4 Interrupts

The CM7_MCM module has one interrupt request: mcm_irq.

The mcm_irq will be asserted when any of the interrupts in the register ISCR occurs.

33.4 External Signals

Table 33-4. External Signals

Signal	Description	Direction
TRACEn	Debug Trace Signal	O
SWO	Single Wire Output Signal	O

33.5 Memory Map and register definition

This section includes the CM7_MCM module memory map and detailed descriptions of all registers.

33.5.1 CM7_MCM register descriptions

33.5.1.1 CM7_MCM memory map

MCM base address: E008_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
10	Interrupt Status and Control Register (ISCR)	32	RW	0000_0000

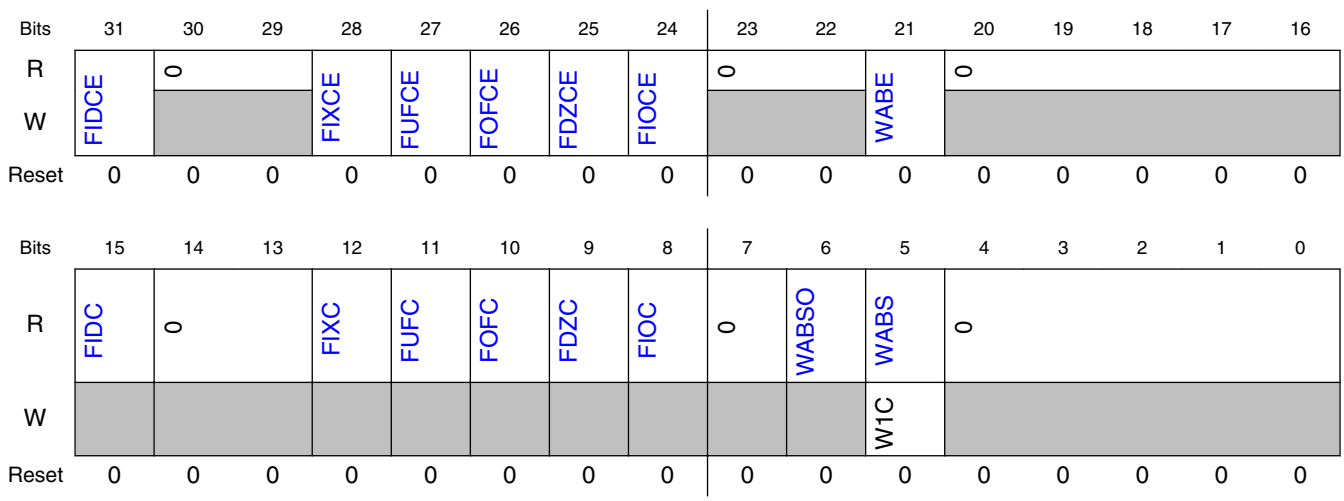
33.5.1.2 Interrupt Status and Control Register (ISCR)

The ISCR register defines the configuration and reports status for a number of core-related interrupt exception conditions. It includes the enable and status bits associated with the core's floating-point exceptions, and bus errors on TCM. The individual event indicators are first qualified with their exception enables and then logically summed to form an interrupt request sent to the core's NVIC. Bits 15-8 are read-only indicator flags based on the processor's FPSCR register. Attempted writes to these bits are ignored. Once set, the flags remain asserted until software clears the corresponding FPSCR bit.

33.5.1.2.1 Offset

Register	Offset
ISCR	10h

33.5.1.2.2 Diagram



33.5.1.2.3 Fields

Field	Description
31 FIDCE	FPU Input Denormal Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt

Table continues on the next page...

Field	Description
30-29 —	Reserved
28 FIXCE	FPU Inexact Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt
27 FUFCE	FPU Underflow Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt
26 FOFCE	FPU Overflow Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt
25 FDZCE	FPU Divide-by-Zero Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt
24 FIOCE	FPU Invalid Operation Interrupt Enable 0 - Disable interrupt 1 - Enable interrupt
23-22 —	Reserved
21 WABE	TCM Write Abort Interrupt enable 0 - Disable interrupt 1 - Enable interrupt
20-16 —	Reserved
15 FIDC	FPU Input Denormal Interrupt Status This read-only bit is a copy of the core's FPSCR[IDC] bit and signals input denormalized number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IDC] bit. 0 - No interrupt 1 - Interrupt occured
14-13 —	Reserved
12 FIXC	FPU Inexact Interrupt Status This read-only bit is a copy of the core's FPSCR[IXC] bit and signals an inexact number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IXC] bit. 0 - No interrupt 1 - Interrupt occured
11 FUFC	FPU Underflow Interrupt Status This read-only bit is a copy of the core's FPSCR[UFC] bit and signals an underflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[UFC] bit.

Table continues on the next page...

Field	Description
	0 - No interrupt 1 - Interrupt occured
10 FOFC	FPU Overflow interrupt status This read-only bit is a copy of the core's FPSCR[OFC] bit and signals an overflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[OFC] bit. 0 - No interrupt 1 - Interrupt occured
9 FDZC	FPU Divide-by-Zero Interrupt Status This read-only bit is a copy of the core's FPSCR[DZC] bit and signals an divide-by-zero operation has been detected in the processor's FPU. Once set, this bit remains set until software clears FPSCR[DZC] bit. 0 - No interrupt 1 - Interrupt occured
8 FIOC	FPU Invalid Operation interrupt Status This read-only bit is a copy of the core's FPSCR[IOC] bit and signals an illegal operation has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IOC] bit. 0 - No interrupt 1 - Interrupt occured
7 —	Reserved
6 WABSO	Write Abort on Slave Overrun This read-only bit indicates if another write abort is detected before system software has retrieved all the information from the original event. When the WABS is cleared, this bit is also cleared. 0 - No write abort overrun 1 - Write abort overrun occurred
5 WABS	Write Abort on Slave This bit indicates when a write abort has occurred on AHBS interface. 0 - No abort 1 - Abort
4-0 —	Reserved

Chapter 34
ARM Cortex M4 Platform (M4)

34.1 Chip-specific Arm Cortex M4 information

Table 34-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments
System Debug	-	System Debug

NOTE

In CM4 core, write error responses will be suppressed for buffered writes. To improve performance, the master can issue buffered writes to allow the master to move on to the next operations in its pipeline. When the buffered write finally reaches its destination slave, the master has no way to determine which of the earlier transfers requires an error response. If a write error response is essential for certain memory spaces, the target space needs to be configured to a strong ordered type in the MPU.

34.2 Overview

The Cortex-M4 implements the ARMv7- ME instruction set architecture (ISA) and adds significant capabilities with DSP and SIMD extensions. The ARM Cortex-M4 core provides additional general processing capability to the SoC with lower power and fast interrupt response time.

The Cortex-M4 also includes a single-precision floating-point unit (FPU) and two 32-bit system bus interfaces. The Cortex-M4 implementation includes two tightly coupled local memories, two cache memories connected to bus interfaces, 64-bit system bus interconnect, and supports a 32-byte cache line size.

34.2.1 Block Diagram

The block diagram details the ARM Cortex-M4 core and platform.

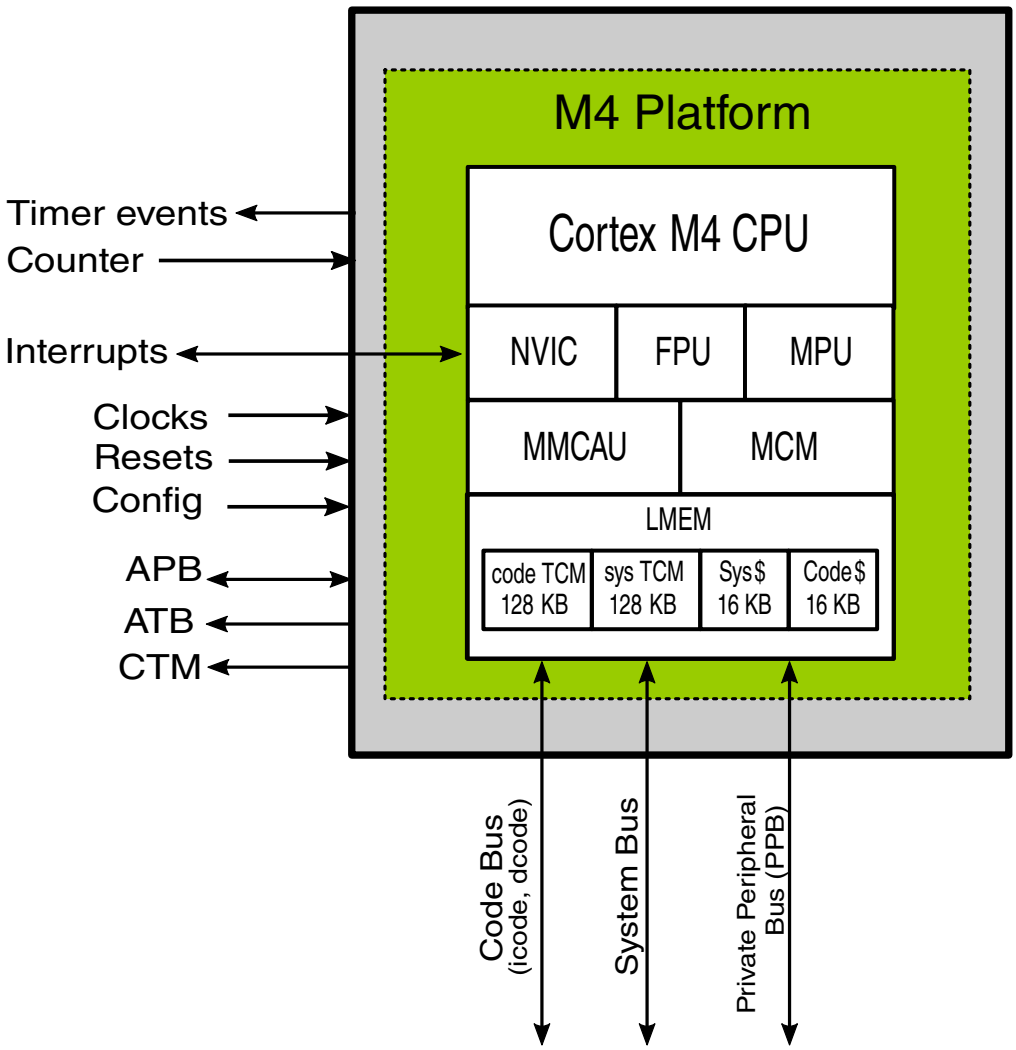


Figure 34-1. Cortex-M4 Platform Block Diagram

34.2.2 Features

The features of the Cortex-M4 platform are detailed below:

- 1x Cortex-M4 processor
- AHB LMEM (Local Memory Controller) including controllers for TCM and cache memories
- 256 KB TCM (128 KB TCMU, 128 KB TCML)
- 16 KB Code Bus Cache
- 16 KB System Bus Cache
- ECC for TCM memories and parity for code and system caches (see LMEM and MCM for more information)
- Integrated Nested Vector Interrupt Controller (NVIC)
- Wakeup Interrupt Controller (WIC)
- FPU (Floating Point Unit)

- Core MPU (Memory Protection Unit)
- Support for exclusive access on the system bus
- MMCAU (Crypto Acceleration Unit)
- MCM (Miscellaneous Control Module)

34.3 Configuration

The configuration of the bus, core, and memory are detailed in the features and sections below.

The Core revision number is Cortex-M4 r0p1.

Table 34-2. Cortex-M4 Configuration

Parameter	Description	Setting
NUM_IRQ	Number of Interrupts	218
LVL_WIDTH (translates to 2^n levels)	Number of interrupt priority bits	4
MPU_PRESENT	Exclude (0) or include (1) optional ARM MPU	1
DEBUG_LVL	Level of debug support	3
TRACE_LVL	Level of trace support 0 = No Trace 1 = ITM and DWT 2 = ITM, DWT, and ETM 3 = ITM, DWT, ETM, and HTM	2
RESET_ALL_REGS	Select whether all registers are reset (1)	1
JTAG_PRESENT	SWJ-DP (1) SW-DP (0)	0
CLKGATE_PRESENT	Disable (0) or enable (1) instantiation of architectural clock gates at all levels	1
WIC_PRESENT	Exclude (0) or include (1) optional ARM provided WIC	1
WIC_LINES	Select number of interrupts and/or events that WIC is sensitive to	NUM_IRQ+3
FPU_PRESENT	Exclude (0) or include (1) Floating Point Unit (FPU)	1
BB_PRESENT	Exclude (0) or include (1) Bit Banding logic	0
CONST_AHB_CTRL	Specifies whether the external AHB-Lite buses maintain control information during wait stated transfers	0

34.4 Performance

This section will discuss power and clocking performance of the Cortex-M4 Platform.

34.4.1 Power

There are several power states supported by the Cortex-M4. Supported primary core states are listed below:

- **Run** - Normal run state. CM4 Platform, TCM and cache memories on nominal voltage.
- **Sleep** – Wait power state. Depending on power state, CM4 Platform, TCM and cache memories can either be in nominal or low-voltage. State recovery through NVIC. NVIC clock is still running in this state.
- **Deep Sleep** – Stop power state. Depending on power state, CM4 Platform, TCM and cache memories can either be in nominal or low-voltage. State recovery through AWIC. Clocks can be completely stopped in this state.

34.4.2 Clocking

The M4 platform is provided a main processor clock that supplies the component clocks to the cluster components. The maximum frequency targets are specified in the chip datasheet. Please see the datasheet for more information. The clocks are described in the table below:

Table 34-3. CM4 Clocks

Clock Signal	Clock Name	Description
Core gated clock	cm4_hclk	Cortex-M4 core clock.
Core Free-Running Clock	cm4_fclk	CM4 NVIC and timer clock
TCM Controller Clock	cm4_tcmc_hclk	CM4 platform TCM controller clock
MMCAU Clock	cm4_mmcau_hclk	CM4 platform MMCAU module clock
Platform Clock	hclk	CM4 platform AXBS fabric and bus masters. Synchronous to the core clock.
Bus Clock	ipg_clk	Clock for bus slaves and peripherals. Synchronous to the system clock.

34.5 Platform sub-blocks

The sections below discuss the high-level overview of the ARM® Cortex®-M4 Platform components.

34.5.1 ARM Cortex-M4 Processor

The Cortex-M4 processor is a low-power processor that features low gate count, low interrupt latency, and low-cost debug. The Cortex-M4 includes floating point arithmetic functionality. The processor is intended for deeply embedded applications that require fast interrupt response features.

The Cortex-M4 implements the ARMv7-M architecture.

The ARM Cortex-M4 processor consists of:

- A processor core
- A Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low-latency interrupt processing
- Multiple high-performance bus interfaces
- A low-cost debug solution with the optional ability to:
  - Implement breakpoints and code patches
  - Implement watchpoints, tracing, and system profiling
  - Support printf() style debugging
  - Bridge to a Trace Port Analyzer (TPA)
- Memory Protection Unit (MPU)
- Floating Point Unit (FPU)
- Crypto Acceleration Unit (MMCAU)
- Miscellaneous Control Module (MCM)

34.5.2 Nested Vectored Interrupt Controller (NVIC)

The M4 platform includes a Nested Vectored Interrupt Controller (NVIC) that is closely integrated with the processor core to achieve low-latency interrupt processing. The platform supports 16 priority levels for interrupts. The NVIC IPR registers will define 4 bits per IRQ.

34.5.3 Floating Point Unit (FPU)

The Cortex-M4 platform supports Floating Point Unit (FPU). The FPU implements the single precision variant of the ARMv7-M Floating-Point Extension (FPv4-SP), which supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. FPU also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions.

The FPU provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard. The FPU contains 32 single-precision extension registers, which you can also access as 16 doubleword registers for load, store, and move operations.

34.5.4 Memory Protection Unit (MPU)

The Cortex-M4 platform supports Memory Protection Unit (MPU). The MPU enforces privilege rules, separates processes, and enforces access rules to memory, and supports the standard ARMv7 Protected Memory System Architecture model. The MPU provides full support for:

- Protection regions
- Overlapping protection regions, with ascending region priority:
  - 7 = highest priority
  - 0 = lowest priority
- Access permissions
- Exporting memory attributes to the system

MPU mismatches and permission violations invoke the programmable-priority MemManage fault handler. You can use the MPU to:

- Enforce privilege rules
- Separate processes
- Enforce access rules

34.6 Debug

This section gives a brief overview of the modules that are implemented within the Cortex-M4 Core Platform. The debug blocks are part of the overall CoreSight platform debug system, which include the ETR, CTM, CTI, ATB replicator, TPIU and DAP. The CoreSight™ compatible Embedded Trace Macrocell (ETM) enables traces of program flow to be collected, compressed, and fed into the trace infrastructure. The Cross Trigger Interface (CTI) is included to provide a common programming model for use by the debug tools, control the trigger sources, and interface to the Cross Trigger Matrix (CTM). The debug is controlled via an ARM Debug Access Port (DAP). For details of the full CoreSight debug subsystem, see the System Debug chapter.

34.6.1 Embedded Trace Router (ETR)

The Embedded Trace Router (ETR) is a CoreSight trace component which buffers trace data before dumping that data into DRAM through the AXI bus (via AXI master port). The output of the funnel is the ETR, and has the ability to buffer a certain amount before writing the data to DRAM.

34.6.2 Embedded Trace Macrocell (ETM)

The Embedded Trace Macrocell (ETM) is a CoreSight component. The ETM trace unit is a module that performs real-time instruction flow tracing based on the Embedded Trace Macrocell (ETM) architecture ETMv4.

34.6.3 Cross Trigger Matrix (CTM)

The CoreSight CTI channel signals from all the cores are combined using a Cross Trigger Matrix (CTM) block so that a single cross trigger channel interface is presented in the Cortex processor. This module can combine four internal channel interfaces corresponding to each core along with one external channel interface.

In the Cortex processor CTM, the external channel output is driven by the OR output of all internal channel outputs. Each internal channel input is driven by the OR output of internal channel outputs of all other CTIs in addition to the external channel input.

34.6.4 Cross Trigger Interface (CTI)

The Cortex processor has a single external cross trigger channel interface. This external interface is connected to the CoreSight Cross Trigger Interface (CTI) interface corresponding to each core through a Cross Trigger Matrix (CTM). A number of Embedded Cross Trigger (ECT) trigger inputs and trigger outputs are connected between debug components in the Cortex-A processor and CoreSight CTI blocks.

The CTI enables the debug logic, ETM trace unit, and performance monitoring, to interact with each other and with other CoreSight components. This is called cross triggering. For example, you configure the CTI to generate an interrupt when the ETM trace unit trigger event occurs.

34.6.5 Instrumentation Trace Macrocell (ITM)

The Instrumentation Trace Macrocell (ITM) is a application-driven trace source that supports printf style debugging to trace operating system and application events. The ITM generates diagnostic system information as packets. Multiple sources can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. These sources in decreasing order of priority are:

- **Software trace.** Software can write directly to ITM stimulus registers to generate packets.
- **Hardware trace.** The DWT generates these packets, and the ITM outputs them.
- **Time stamping.** Timestamps are generated relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M processor clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.

34.6.6 Data Watchpoint and Trace (DWT)

The Data Watchpoint and Trace (DWT) debug unit provides watchpoints, data tracing, and system profiling for the processor.

34.6.7 Debug Access Port (DAP)

The Debug Access Port (DAP) is a physical port that connects to external debug tools. The DAP is part of the standardized ARM Debug Interface, and provides a bridge between a JTAG pin interface and on-chip memory mapped peripherals via the DAP bus. Transactions generated by the DAP are referred to as External Debugger Accesses.

34.7 Local Memory Controller (LMEM)

34.7.1 Overview

The Local Memory Controller (LMEM) provides the processor with tightly-coupled memory (TCM) and cache and bus paths to all slave memory spaces.

34.7.1.1 Block Diagram

The processor has a modified 32-bit Harvard bus architecture. Using a 32-bit address space, low-order addresses (0x0000_0000 through 0x1FFF_FFFF) use the Processor Code (PC) bus, and high-order addresses (0x2000_0000 through 0xFFFF_FFFF) use the Processor System (PS) bus. As the bus names imply, normal operation has code accesses on the PC bus and data accesses on the PS bus.

This device has been augmented with tightly-coupled RAM and cache for the PC and PS buses to provide zero wait state access to the RAM and cacheable address spaces.

The local memory controller includes the following memory controllers and their attached memories:

- TCM lower (code TCM) controller via the PC bus
- TCM upper (system TCM) controller via the PS bus
- Cache memory controller via the PC bus
- Cache memory controller via the PS bus

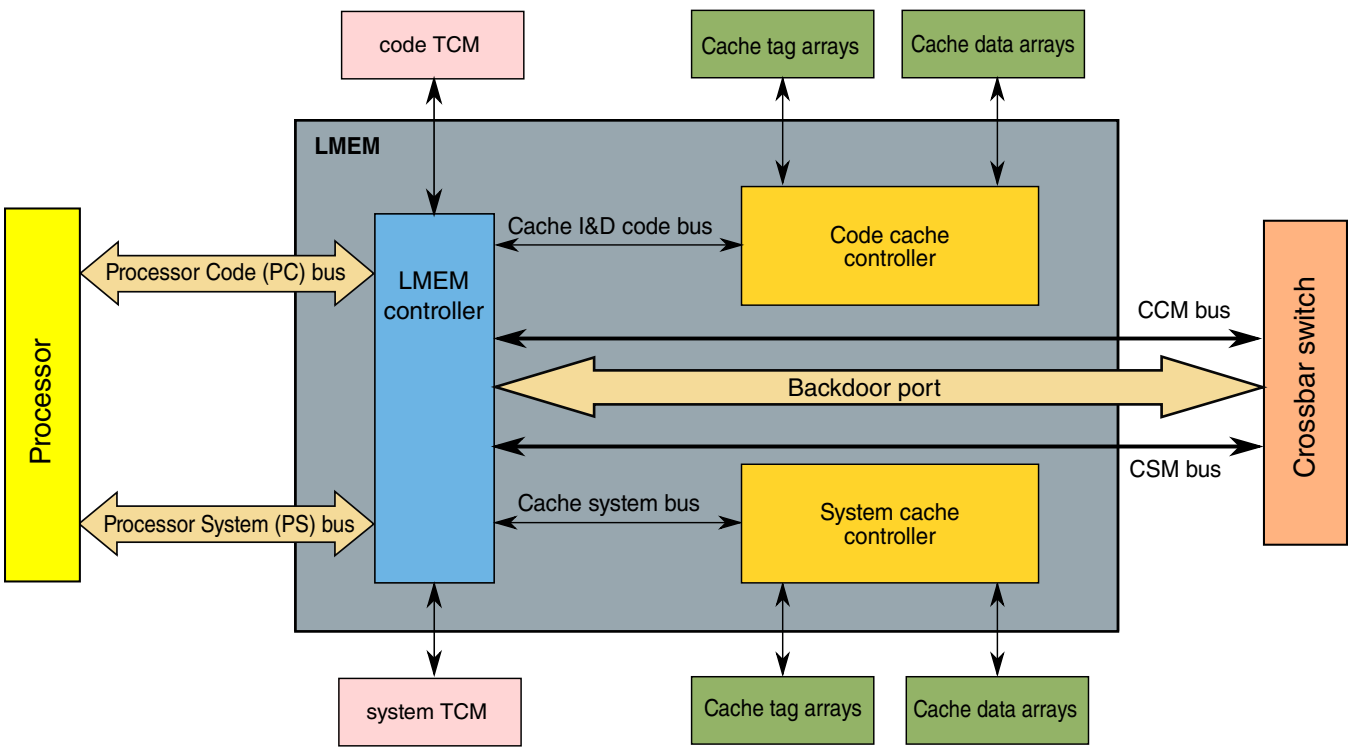


Figure 34-2. Local memory controller block diagram

NOTE

The TCM and cache controllers reside within the LMEM, but the single-port synchronous RAM arrays used by these controllers are external.

The LMEM contains address decode logic for the PC and PS buses. This logic routes the core's accesses to the various system resources. The address spaces are device-specific and are specified in the device's chip-specific information.

34.7.1.2 Cache Features

A cache is a block of high-speed memory locations containing address information (commonly known as a tag) and the associated data. The purpose is to decrease the average time of a memory access. Caches operate on two principles of locality:

- Spatial locality — An access to one location is likely to be followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
- Temporal locality — An access to an area of memory is likely to be repeated within a short time period (for example, execution of a code loop).

To minimize the quantity of control information stored, the spatial locality property is used to group several locations together under the same tag. This logical block is commonly known as a cache line.

Temporal locality is achieved by keeping recently accessed lines in the cache.

When data is loaded into a cache, access times for subsequent loads and stores are reduced, resulting in overall performance benefits. An access to information already in a cache is known as a cache hit, and other accesses are called cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the processor wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

Caches introduce a number of potential problems, mainly because of:

- memory accesses occurring at times other than when the programmer would normally expect them,
- the existence of multiple physical locations where a data item can be held.

The local memory controller supports the following modes of operation:

1. Write-through — access to address spaces with this cache mode are cacheable.
   - If all cacheable spaces are read-only spaces, the cache will contain read-only data and all write to the cache will fault. See the chip-specific cacheable space information.

- A write-through read miss on the input bus causes a line read on the output bus of a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and is marked as valid and not modified.
- A write-through read hit to a valid cache location returns data from the cache with no output bus access.
- A write-through write miss bypasses the cache and writes to the output bus (no allocate on write miss policy for write-through mode spaces).
- A write-through write hit updates the cache hit line with the write data and writes to the output bus.
- The caches are processor-local and do not support hardware cache coherency. If the processor has accessed write-through regions and an external bus master (such as DMA) then needs update these regions, software must first perform explicit cache clears to any needed cache memory range to ensure all modified cache lines update their associated memories before being modified by external masters and subsequent processor accesses will get the updated memory.

2. Write-back — access to address spaces with this cache mode are cacheable.
   - Code Cache (PC Side Cache) does not support this mode of operation
   - A write-back read miss on the input bus will cause a line read on the output bus of a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and marked as valid and not modified.
   - A write-back read hit to a valid cache location will return data from the cache with no output bus access.
   - A write-back write miss will do a "read-to-write" (allocate on write miss policy for write-back mode spaces). A line read on the output bus of a 32 byte aligned memory address containing the desired write address is performed. This miss data is loaded into the cache and marked as valid and modified; and the write data will then update the appropriate cache data locations.
   - A write-back write hit updates the cache hit line with the write data plus marks the line as modified.
   - The caches are processor-local and do not support hardware cache coherency. If the processor has written to write-back regions and an external bus master (such as DMA) then needs to see these updates, software must perform explicit cache pushes to any needed cache memory range to ensure all modified cache lines update their associated memories before being read by external masters. Likewise, if the processor has accessed write-back or write-through regions and an external bus master (such as DMA) then needs update these regions, software must first perform explicit cache clears to any needed cache memory range to ensure all modified cache lines update their associated memories before being modified by external masters and subsequent processor accesses will get the updated memory.

3. Non-cacheable — access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

34.7.2 Functional Description

34.7.2.1 LMEM Function

The Local Memory Controller receives the following requests:

- Core master bus requests on the Processor Code (PC) bus
- Core master bus requests on the Processor Space (PS) bus
- TCM controller requests from all other bus masters on the backdoor port.

The Local Memory Controller address decode logic routes these accesses and also provides any crossbar switch slave target logic. Finally, the Local Memory controller provides the needed MPU connections for checking all TCM Controller and cacheable accesses.

The programming model for the Code and System Caches is accessed via the core's Private Peripheral Bus (PPB).

34.7.2.1.1 Processor Code accesses

Processor Code accesses are routed to the code TCM if they are mapped to that space. All other PC accesses are routed to the Code Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master0 port.

34.7.2.1.2 Processor System accesses

Processor Space accesses are routed to the system TCM if they are mapped to that space. All other PS accesses are routed to the PS Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CSM bus and the crossbar switch using the Master1 port.

34.7.2.1.3 Backdoor port accesses

All LMEM backdoor port accesses are only for the TCM controller, no access to cache controllers. These accesses go to the code TCM or the system TCM depending on their specific address.

34.7.2.2 TCM Function

34.7.2.2.1 TCM Configuration

The figure below shows how the TCM controller is configured.

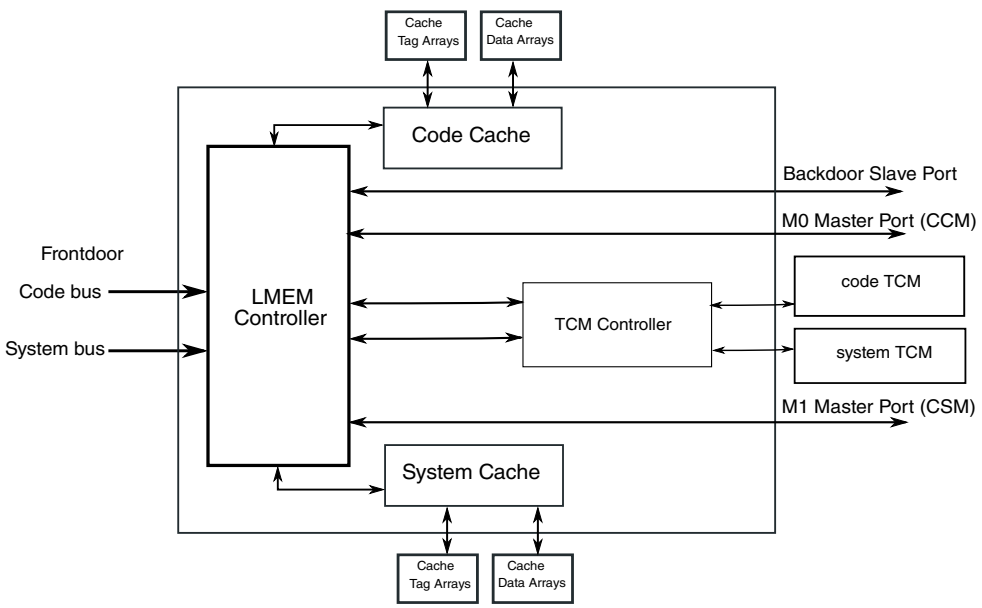


Figure 34-3. LMEM Interface to TCM

34.7.2.2.2 TCM Arrays

The on-chip TCM is split into two logical arrays, code TCM and system TCM.

code TCM size is 128 KBytes.

system TCM size is 128 KBytes.

Valid address ranges for code TCM and system TCM are then defined as:

- code TCM = 0x1FFFE0000 - 0x1FFFFFFF
- system TCM = 0x20000000 - 0x2001FFFF

code TCM and system TCM sizes do not have to be equal.

See the LMEM chip-specific information to confirm the TCM sizes on this device.

34.7.2.2.3 TCM Accesses

The TCM is split into two logical arrays that are 64-bits wide:

- code TCM — Accessible by the 32-bit datapath code bus of the core and by the 64-bit datapath backdoor port.
- system TCM — Accessible by the 32-bit datapath system bus of the core and by the 64-bit datapath backdoor port.

The backdoor port makes the TCM accessible to the non-core bus masters (such as DMA).

The following figure illustrates the TCM accesses within the device.

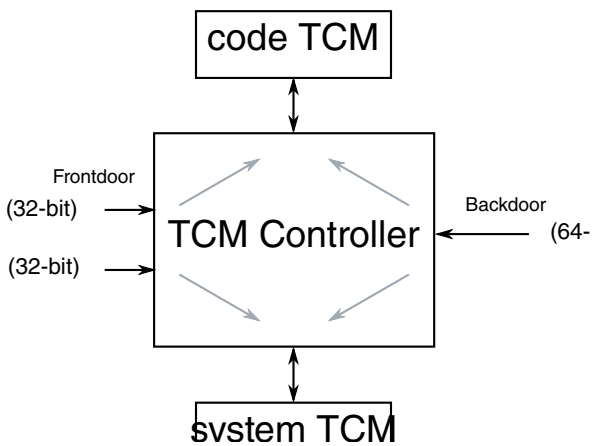


Figure 34-4. TCM access diagram

The following simultaneous accesses can be made to different logical halves of the TCM:

- Core code and core system
- Core code to code TCM and non-core master to system TCM
- Core system to system TCM and non-core master to code TCM

NOTE

Two non-core masters cannot access TCM simultaneously. The required arbitration and serialization is provided by the crossbar switch. The code TCM and system TCM arbitration is controlled by the TCM controller based on the configuration bits in the MCM module.

NOTE

Burst-access cannot occur across the 0x2000_0000 boundary that separates the two TCM arrays. The two arrays should be treated as separate memory ranges for burst accesses.

34.7.2.2.4 TCM ECC Configuration

The TCM Controller ECC is SECDED = Single Error Correct, Double Error Detect. The ECC uses a 7-bit ECC code word per word (32-bits) of data. When ECC is enabled, any write of less than a word (32-bits) is converted to a word read-modify-write sequence. In this way, all TCM writes are 32-bit or 64-bit. All TCM reads are also 32-bit or 64-bit. Each 32-bits of data storage has a related 7-bit ECC code word, written at the same time as the related data and read with the related data. ECC code words are stored in additional RAM storage associated with the main TCM storage and do not subtract from the main RAM size. The TCM Controller ECC does not include any address bits in the ECC code word and does not support error injection.

34.7.2.2.5 TCM ECC Enable

The TCM receives separate control inputs to enable ECC generations on writes and to enable ECC checking on reads. This control occurs outside the TCM from system-level control modules.

The TCM data arrays must first be written with good ECC before being read with ECC checking on reads.

34.7.2.2.6 TCM ECC Function

Since the TCM support byte level writes and the ECC is at a word level, when ECC is enabled all write of less than word size are converted to a word read-modify-write sequence. In this way, correct word level ECC is generated.

When TCM ECC generation on writes is enabled, for TCM writes:

- All TCM write generate the needed ECC word code
- The TCM data and associated data ECC word code are written to the data RAM plus data ECC word code RAM at the same time

When TCM ECC read check is enabled, for TCM data reads:

- For every data word read, the data word and its ECC word code are read in parallel
- The ECC is checked, any single bit error is corrected on-the-fly, any single or multi bit error is flagged

Since all single bit ECC errors are corrected on-the-fly, TCM operations continue as normal when only single bit ECC errors occur.

TCM read accesses that get multi bit ECC errors are handled as follows:

- Any access that gets a multi bit error terminates with a bus fault

34.7.2.2.7 TCM ECC Codes

34.7.2.2.7.1 TCM Word Level ECC Code

The TCM word level ECC uses a Hsiao odd-weight column criteria ECC code for ECC generation and checking. The ECC code covers 32-bit data. The same ECC code is used for the upper and lower 32-bit word of the 64-bit wide TCM datapath.

34.7.2.3 Cache Function

The caches on this device are structured as follows. The PC cache and the PS cache have a 2-way set-associative cache structure with a total size of 16 KBytes. The caches have 32-bit address, 64-bit data paths and a 32-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

For these 16-KByte caches, each cache TAG function uses two 256 x 21-bit RAM arrays and the cache DATA function uses two 1024 x 64-bit RAM arrays. The cache TAG entries store 19 bits of upper address as well as a modified and valid bit per cache line. The cache DATA entries store eight bytes of code or data.

All normal cache accesses use physical addresses. This leads to the following cache address use:

CACHE - 16 KByte size = (256 sets) x (32-byte lines) x (2-way set associative)

TAG:

- address[31:13] used in tag for compare (hit) logic
- address[12:5] used to select 1 of 256 sets
- address[4:0] not used

DATA

- address[31:13] not used
- address[12:5] used to select one of 256 sets
- address[4:3] used to select one of four 64-bit words within a set
- address[2:0] used to select the byte within the 64-bit word

34.7.2.4 Cache Control

The Code and System Caches are disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable the caches, cache commands must be done to clear and initialize the required tag array bits and to configure and enable the caches.

34.7.2.4.1 Cache set commands

The cache set commands may operate on:

- all of way 0,
- all of way 1, or
- all of both ways (complete cache).

Cache set commands are initiated using the upper bits in the PxCCR register. Cache set commands perform their operation on the cache independent of the cache enable bit, PxCCR[ENCACHE].

A cache set command is initiated by setting the PxCCR[GO] bit. This bit also acts as a busy bit for set commands. It stays set while the command is active and is cleared by the hardware when the set command completes.

Supported cache set commands are given in Table 34-4. Set commands work as follows:

- Invalidate – Unconditionally clear valid and modify bits of a cache entry.
- Push – Push a cache entry if it is valid and modified, then clear the modify bit. If entry not valid or not modified, leave as is.
- Clear – Push a cache entry if it is valid and modified, then clear the valid and modify bits. If entry not valid or not modified, clear the valid bit.

Table 34-4. Cache Set Commands

PxCCR[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1

Table continues on the next page...

Table 34-4. Cache Set Commands (continued)

PxCCR[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1; clear all way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, setting PxCCR to 0x8500_0003 will invalidate the cache and enable the cache and write buffer.

34.7.2.4.2 Cache line commands

Cache line commands operate on a single line in the cache at a time. Cache line commands can be performed using a physical or cache address.

- A cache address consists of a set address and a way select. The line command acts on the specified cache line.
- Cache line commands with physical addresses first search both ways of the cache set specified by the following physical address bits. If they hit, the commands perform their action on the hit way:
  - For Code Cache - [12:5]
  - For System Cache - [12:5]

Cache line commands are specified using the upper bits in the PxCLCR register. Cache line commands perform their operation on the cache independent of the cache enable bit (PxCCR[ENCACHE]). Using a cache address, the command can be completely specified using the PxCLCR register. Using a physical address, the command must also use the PxCSAR register to specify the physical address.

A line cache command is initiated by setting the line command go bit (PxCLCR[LGO] or PxCSAR[LGO]). This bit also acts a a busy bit for line commands. It stays set while the command is active and is cleared by the hardware when the command completes.

The PxCLCR[27:24] bits select the line command as follows:

Table 34-5. Cache Line Commands

PxCLCR[27:24]			Command
LACC	LADSEL	LCMD	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

34.7.2.4.2.1 Executing a series of line commands using cache addresses

A series of line commands with incremental cache addresses can be performed by just writing to the PxCLCR.

- Place the command in PxCLCR[27:24],
- Set the way (PxCLCR[WSEL]) and tag/data (PxCLCR[TDSEL]) controls as needed,
- Place the cache address in PxCLCR[CACHEADDR], and
- Set the line command go bit (PxCLCR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the cache address (at bit 3 to step through data or at bit 5 to step through lines), and
- Set the line command go bit (PxCLCR[LGO]).

34.7.2.4.2.2 Executing a series of line commands using physical addresses

Perform a series of line commands with incremental physical addresses using the following steps:

- Write to the PxCLCR.

    - Place the command in PxCLCR[27:24]
    - Set the tag/data (PxCLCR[TDSEL]) control
- Place the physical address in PxCSAR[PHYADDR] and set the line command go bit (PxCSAR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the physical address (at bit 3 to step through data or at bit 5 to step through lines), and
- Set the line command go bit (PxCSAR[LGO]).

The line command go bit is shared between the PxCLCR and PxCSAR registers, so that the above steps can be completed in a single write to the PxCSAR register.

34.7.2.4.2.3 Line command results

At completion of a line command, the PxCLCR register contains information on the initial state of the line targeted by the command. For line commands with cache addresses, this information is read before the line command action is performed from the targeted cache line. For line commands with physical addresses, this information is read on a hit before the line command action is performed from the hit cache line or has initial valid bit cleared if the command misses. In general, if the valid indicator (PxCLCR[LCIVB] is cleared, the targeted line was invalid at the start of the line command and no line operation was performed.

Table 34-6. Line command results

PxCLCR[22:20]			For cache address commands	For physical address commands
LCWAY	LCIMB	LCIVB		
0	0	0	Way 0 line was invalid	No hit
0	0	1	Way 0 valid, not modified	Way 0 valid, not modified
0	1	0	Way 0 line was invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line was invalid	No hit
1	0	1	Way 1 valid, not modified	Way 1 valid, not modified
1	1	0	Way 1 line was invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

At completion of a line command other than a write, the PxCCVR (Cache R/W Value Register) contains information on the initial state of the line tag or data targeted by the command. For line commands, PxCLCR[TDSEL] selects between tag and data. If the line command used a physical address and missed, the data is don't care. For write commands, the PxCCVR holds the write data.

The cache does not have line lock capability. Performance critical code can be loaded in the code TCM controller associated with the low-order address processor code bus.Performance critical data can be loaded in the system TCM controller associated with the upper-order address processor system bus.

34.7.3 Memory Map and Registers

The cache programmer's model provides a variety of registers for configuring and controlling the cache, as well as indirect access paths to all cache tag and data storage.

NOTE

The LMEM registers are accessible in supervisor mode only.

34.7.3.1 LMEM register descriptions

34.7.3.1.1 LMEM memory map

LMEM base address: E008_2000h

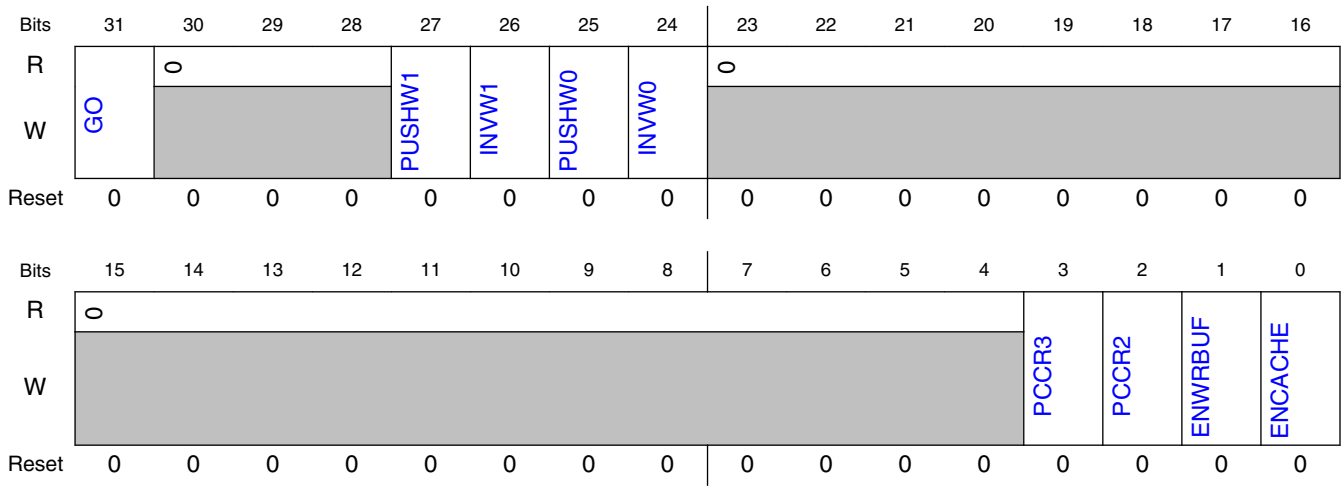
Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PC bus Cache control register (PCCCR)	32	RW	0000_0000
4	PC bus Cache line control register (PCCLCR)	32	RW	0000_0000
8	PC bus Cache search address register (PCCSAR)	32	RW	0000_0000
C	PC bus Cache read/write value register (PCCCVR)	32	RW	0000_0000
800	PS bus Cache control register (PSCCR)	32	RW	0000_0000
804	PS bus Cache line control register (PSCLCR)	32	RW	0000_0000
808	PS bus Cache search address register (PSCSAR)	32	RW	0000_0000
80C	PS bus Cache read/write value register (PSCCVR)	32	RW	0000_0000

34.7.3.1.2 PC bus Cache control register (PCCCR)

34.7.3.1.2.1 Offset

Register	Offset
PCCCR	0h

34.7.3.1.2.2 Diagram



34.7.3.1.2.3 Fields

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 - Write: no effect. Read: no cache command active. 1 - Write: initiate command indicated by bits 27-24. Read: cache command active.
30-28 —	Reserved
27 PUSHW1	Push Way 1 0 - No operation 1 - When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 - No operation 1 - When setting the GO bit, invalidate all lines in way 1
25	Push Way 0

Table continues on the next page...

Field	Description
PUSHW0	0 - No operation 1 - When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 - No operation 1 - When setting the GO bit, invalidate all lines in way 0.
23-4 —	Reserved
3 PCCR3	Forces no allocation on cache misses 0 - Allocation on cache misses 1 - Forces no allocation on cache misses (must also have PCCR2 asserted)
2 PCCR2	Forces all cacheable spaces to write through 0 - Does NOT force all cacheable spaces to write through 1 - Forces all cacheable spaces to write through
1 ENWRBUF	Enable Write Buffer 0 - Write buffer disabled 1 - Write buffer enabled
0 ENCACHE	Cache enable 0 - Cache disabled 1 - Cache enabled

34.7.3.1.3 PC bus Cache line control register (PCCLCR)

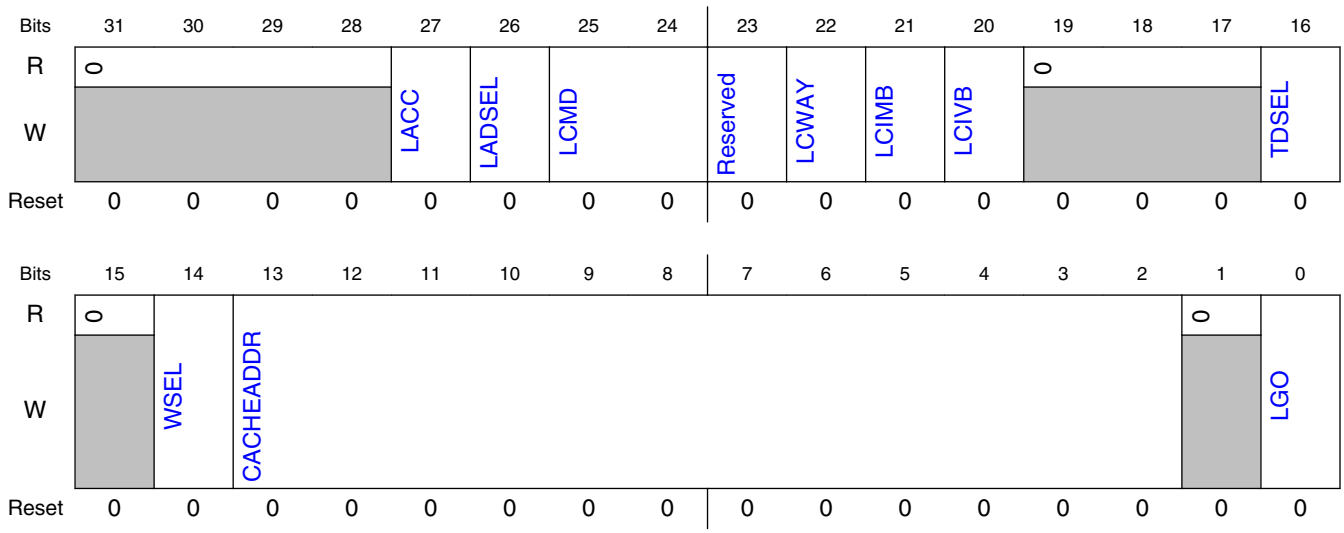
This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

34.7.3.1.3.1 Offset

Register	Offset
PCCLCR	4h

34.7.3.1.3.2 Diagram



34.7.3.1.3.3 Fields

Field	Description
31-28 —	Reserved
27 LACC	Line access type 0 - Read 1 - Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL]. When using the physical address, both ways are searched and the command is performed only if a hit. 0 - Cache address 1 - Physical address
25-24 LCMD	Line Command 00 - Search and read or write 01 - Invalidate 10 - Push 11 - Clear
23 —	Reserved
22 LCWAY	Line Command Way Indicates the way used by the line command. Only applies if valid bit LCIVB = 1.
21	Line Command Initial Modified Bit

Table continues on the next page...

Field	Description
LCIMB	If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19-17 —	Reserved
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 - Data 1 - Tag
15 —	Reserved
14 WSEL	Way select Selects the way for line commands. 0 - Way 0 1 - Way 1
13-2 CACHEADDR	Cache address CLCR [12:5] bits are used to access the tag arrays CLCR [12:3] bits are used to access the data arrays CLCR[2] bit is needed to decide which half of the 64 bits of data to put in the Cache read/write value registers (PCCCVR)
1 —	Reserved
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 - Write: no effect. Read: no line command active. 1 - Write: initiate line command indicated by bits 27-24. Read: line command active.

34.7.3.1.4 PC bus Cache search address register (PCCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

34.7.3.1.4.1 Offset

Register	Offset
PCCSAR	8h

34.7.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHYADDR															LGO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

34.7.3.1.4.3 Fields

Field	Description
31-1 PHYADDR	Physical Address PHYADDR represents bits [31:1] of the system address. CSAR[31:13] bits are used for tag compare CSAR[12:5] bits are used to access the tag arrays CSAR[12:3] bits are used to access the data arrays CSAR[2] bit is needed to decide which half of the 64 bits of data to put in the Cache read/write value registers (PCCCVR) CSAR[1] bit is not used. Writes to this bit are allowed, and the read value will be whatever value was previously written.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO] 0 - Write: no effect. Read: no line command active. 1 - Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

34.7.3.1.5 PC bus Cache read/write value register (PCCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

34.7.3.1.5.1 Offset

Register	Offset
PCCCVR	Ch

34.7.3.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

34.7.3.1.5.3 Fields

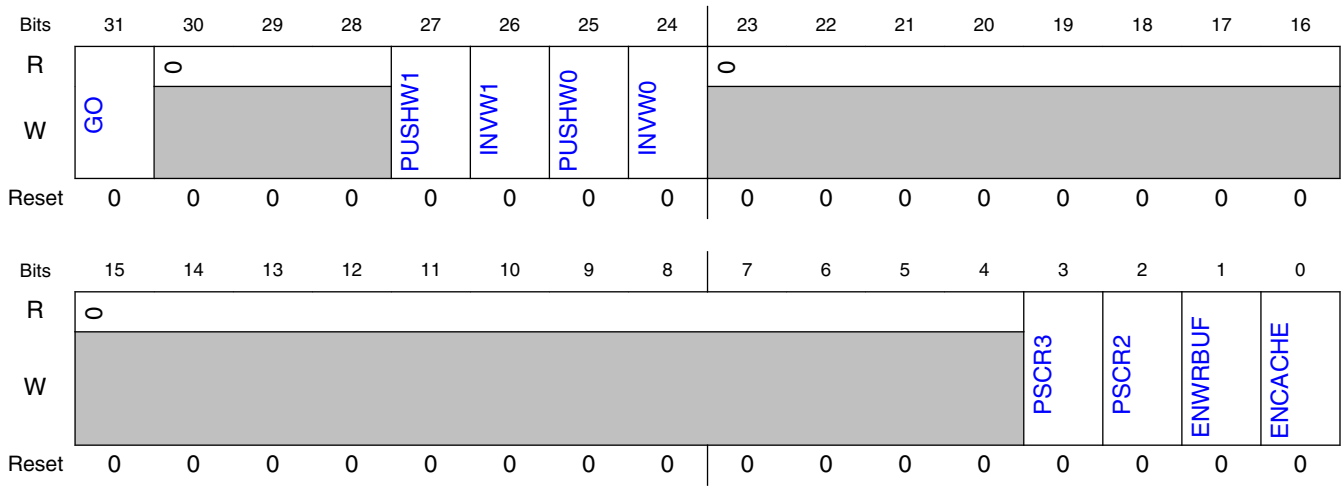
Field	Description
31-0 DATA	Cache read/write Data For tag search, read or write: • CCVR [31:13] bits are used for tag array R/W value • CCVR [12:5] bits are used for tag set address on reads; unused on writes • CCVR[4:2] bits are reserved • CCVR[1] tag modify bit • CCVR[0] tag valid bit For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

34.7.3.1.6 PS bus Cache control register (PSCCR)

34.7.3.1.6.1 Offset

Register	Offset
PSCCR	800h

34.7.3.1.6.2 Diagram



34.7.3.1.6.3 Fields

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 - Write: no effect. Read: no cache command active. 1 - Write: initiate command indicated by bits 27-24. Read: cache command active.
30-28 —	Reserved
27 PUSHW1	Push Way 1 0 - No operation 1 - When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 - No operation 1 - When setting the GO bit, invalidate all lines in way 1
25	Push Way 0

Table continues on the next page...

Field	Description
PUSHW0	0 - No operation 1 - When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 - No operation 1 - When setting the GO bit, invalidate all lines in way 0.
23-4 —	Reserved
3 PSCR3	Forces no allocation on cache misses 0 - Allocation on cache misses 1 - Forces no allocation on cache misses (must also have PSCR2 asserted)
2 PSCR2	Forces all cacheable spaces to write through 0 - Does NOT force all cacheable spaces to write through 1 - Forces all cacheable spaces to write through
1 ENWRBUF	Enable Write Buffer 0 - Write buffer disabled 1 - Write buffer enabled
0 ENCACHE	Cache enable 0 - Cache disabled 1 - Cache enabled

34.7.3.1.7 PS bus Cache line control register (PSCLCR)

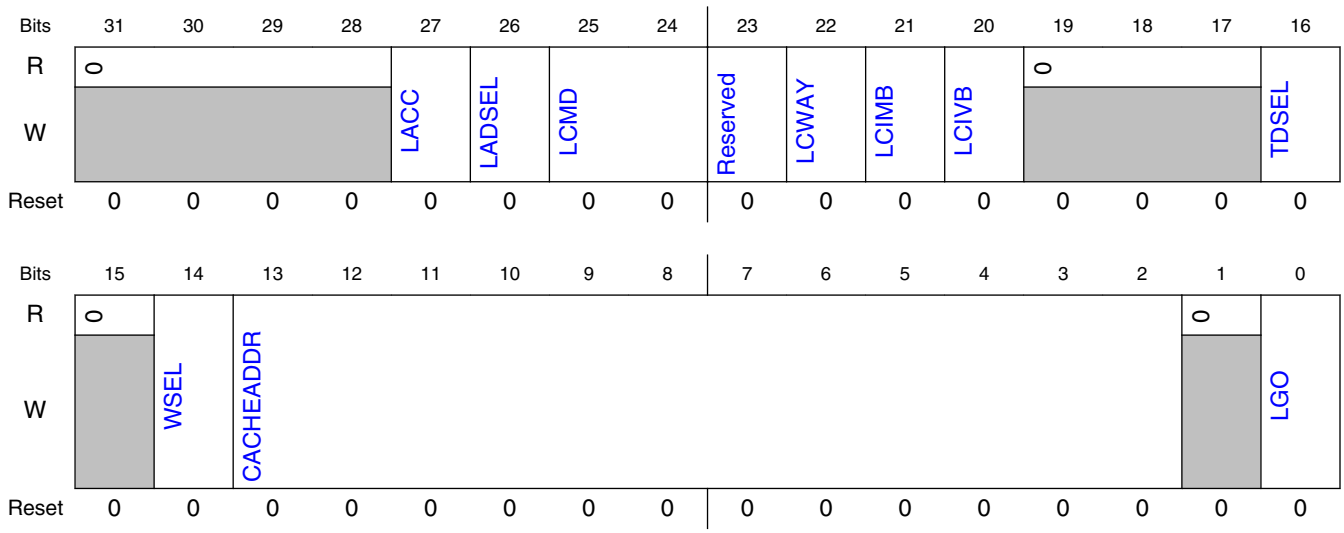
This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

34.7.3.1.7.1 Offset

Register	Offset
PSCLCR	804h

34.7.3.1.7.2 Diagram



34.7.3.1.7.3 Fields

Field	Description
31-28 —	Reserved
27 LACC	Line access type 0 - Read 1 - Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL]. When using the physical address, both ways are searched and the command is performed only if a hit. 0 - Cache address 1 - Physical address
25-24 LCMD	Line Command 00 - Search and read or write 01 - Invalidate 10 - Push 11 - Clear
23 —	Reserved
22 LCWAY	Line Command Way Indicates the way used by the line command. Only applies if valid bit LCIVB = 1.
21	Line Command Initial Modified Bit

Table continues on the next page...

Field	Description
LCIMB	If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19-17 —	Reserved
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 - Data 1 - Tag
15 —	Reserved
14 WSEL	Way select Selects the way for line commands. 0 - Way 0 1 - Way 1
13-2 CACHEADDR	Cache address CLCR [12:5] bits are used to access the tag arrays CLCR [12:3] bits are used to access the data arrays CLCR[2] bit is needed to decide which half of the 64 bits of data to put in the Cache read/write value registers (PSCCVR)
1 —	Reserved
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 - Write: no effect. Read: no line command active. 1 - Write: initiate line command indicated by bits 27-24. Read: line command active.

34.7.3.1.8 PS bus Cache search address register (PSCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

34.7.3.1.8.1 Offset

Register	Offset
PSCSAR	808h

34.7.3.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHYADDR															LGO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

34.7.3.1.8.3 Fields

Field	Description
31-1 PHYADDR	Physical Address PHYADDR represents bits [31:1] of the system address. CSAR [31:13] bits are used for tag compare CSAR [12:5] bits are used to access the tag arrays CSAR [12:3] bits are used to access the data arrays CSAR[2] bit is needed to decide which half of the 64 bits of data to put in the Cache read/write value registers (PSCCVR) CSAR[1] bit is not used. Writes to this bit are allowed, and the read value will be whatever value was previously written.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO] 0 - Write: no effect. Read: no line command active. 1 - Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

34.7.3.1.9 PS bus Cache read/write value register (PSCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

34.7.3.1.9.1 Offset

Register	Offset
PSCCVR	80Ch

34.7.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

34.7.3.1.9.3 Fields

Field	Description
31-0 DATA	Cache read/write Data For tag search, read or write: • CCVR [31:13] bits are used for tag array R/W value • CCVR [12:5] bits are used for tag set address on reads; unused on writes • CCVR[4:2] bits are reserved • CCVR[1] tag modify bit • CCVR[0] tag valid bit For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

34.8 Miscellaneous Control Module (MCM)

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

34.8.1 MCM features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision

34.8.2 MCM Interrupts

The MCM generates the following interrupt requests:

- Non-maskable interrupt
- Normal interrupt

34.8.2.1 Non-maskable interrupt

The MCM's NMI is generated if:

- ISCR[ETBN] is set, when
  - The ETB counter is enabled, ETBCC[CNTEN] = 1
  - The ETB count expires
  - The response to counter expiration is an NMI, MCM_ETBCC[RSPT] = 10

34.8.2.2 Normal interrupt

The MCM's normal interrupt is generated if any of the following is true:

- ISCR[ETBI] is set, when
  - The ETB counter is enabled, ETBCC[CNTEN] = 1
  - The ETB count expires
  - The response to counter expiration is a normal interrupt, ETBCC[RSPT] = 01

34.9 MCM Memory Map/Register Definition

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0000	SoC-defined platform revision (MCM_PLREV)	16	R	0000h	34.9.1/2655
E008_0002	Processor core type (MCM_PCT)	16	R	AC40h	34.9.2/2655
E008_0004	Memory configuration (MCM_MEMCFG)	32	R	Undefined	34.9.3/2656
E008_0008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	001Fh	34.9.4/2657
E008_000A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	001Fh	34.9.5/2658
E008_000C	Control Register (MCM_CR)	32	R/W	0000_0000h	34.9.6/2659
E008_0010	Interrupt Status and Control Register (MCM_ISCR)	32	R	0002_0000h	34.9.7/2661
E008_0020	Fault address register (MCM_FADR)	32	R	Undefined	34.9.8/2664
E008_0024	Fault attributes register (MCM_FATR)	32	R	Undefined	34.9.9/2664
E008_0028	Fault data register (MCM_FDR)	32	R	Undefined	34.9.10/ 2666
E008_0400	Local Memory Descriptor Register (MCM_LMDR0)	32	R/W	See section	34.9.11/ 2667
E008_0404	Local Memory Descriptor Register (MCM_LMDR1)	32	R/W	See section	34.9.11/ 2667
E008_0408	Local Memory Descriptor Register (MCM_LMDR2)	32	R/W	See section	34.9.11/ 2667
E008_040C	Local Memory Descriptor Register (MCM_LMDR3)	32	R/W	See section	34.9.11/ 2667
E008_0480	LMEM Parity & ECC Control Register (MCM_LMPECR)	32	R/W	0030_0003h	34.9.12/ 2671
E008_0488	LMEM Parity & ECC Interrupt Register (MCM_LMPEIR)	32	R/W	0000_0000h	34.9.13/ 2672
E008_0490	LMEM Fault Address Register (MCM_LMFAR)	32	R	0000_0000h	34.9.14/ 2673
E008_0494	LMEM Fault Attribute Register (MCM_LMFATR)	32	R/W	0000_0000h	34.9.15/ 2674
E008_04A0	LMEM Fault Data High Register (MCM_LMFDHR)	32	R	0000_0000h	34.9.16/ 2675
E008_04A4	LMEM Fault Data Low Register (MCM_LMFDLR)	32	R	0000_0000h	34.9.17/ 2676

34.9.1 SoC-defined platform revision (MCM_PLREV)

The PLREV is a 16-bit read-only register specifying an SoC-defined platform revision number. The state of this register is defined by an platform input signal; it can only be read from the IPS programming model. Any attempted write is ignored.

Address: E008_0000h base + 0h offset = E008_0000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PLREV															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_PLREV field descriptions

Field	Description
PLREV	The PLREV[15:0] field is specified by an platform input signal to define a software-visible revision number.

34.9.2 Processor core type (MCM_PCT)

The PCT is a 16-bit read-only register specifying the architecture of the processor core within the platform in the device. The state of this register is defined by a module input signal; it can only be read from the IPS programming model. Any attempted write is ignored.

Address: E008_0000h base + 2h offset = E008_0002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PCT															
Write																
Reset	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0

MCM_PCT field descriptions

Field	Description
PCT	This MCM design supports the ARM Cortex M4 core. The following value identifies this core complex. 0xAC40 ARM Cortex M4

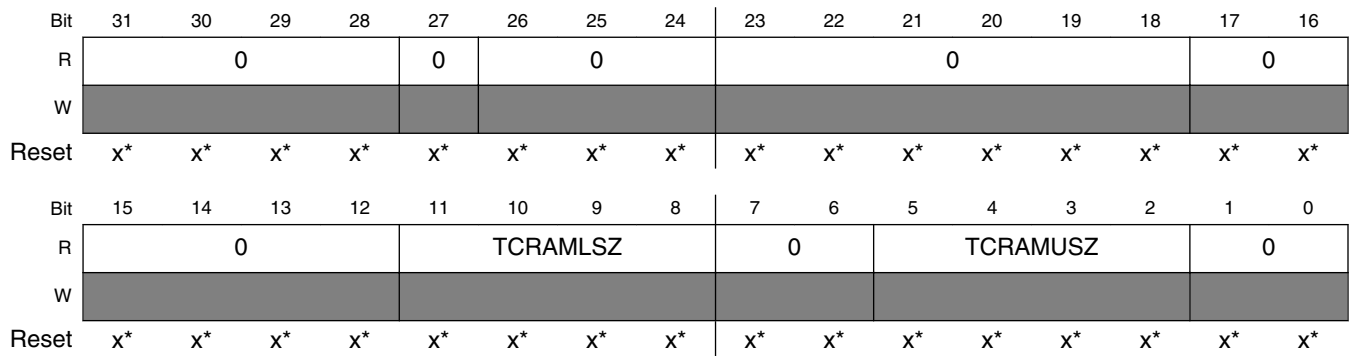
34.9.3 Memory configuration (MCM_MEMCFG)

The CPCFG is a 32-bit read-only register containing information on device memory sizes. The sizes are static and do not change after reset.

NOTE

The reset value depends on the flash and Tightly Coupled Memory (TCM) memory sizes on the specific device.

Address: E008_0000h base + 4h offset = E008_0004h



* Notes:
- x = Undefined at reset.

MCM_MEMCFG field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 TCRAMLSZ	TCRAML size This field indicates the amount of memory available in the lower TCRAM (code TCM) array.

Total TCM size	TCRAMLSZ	TCRAMUSZ
8 KBytes	0101	0101
16 KBytes	0110	0110

Table continues on the next page...

MCM_MEMCFG field descriptions (continued)

Field	Description		
	Total TCM size	**TCRAMLSZ**	**TCRAMUSZ**
	32 KBytes	0111	0111
	64 KBytes	1000	1000
	128 KBytes	1001	1001
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.		
5–2 TCRAMUSZ	TCRAMU size This field indicates the amount of memory available in the upper TCRAM array. See the TCRAMLSZ bit description for encodings.		
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.		

34.9.4 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device's crossbar switch.

Address: E008_0000h base + 8h offset = E008_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent 1 A bus slave connection to AXBS input port *n* is present

34.9.5 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: E008_0000h base + Ah offset = E008_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

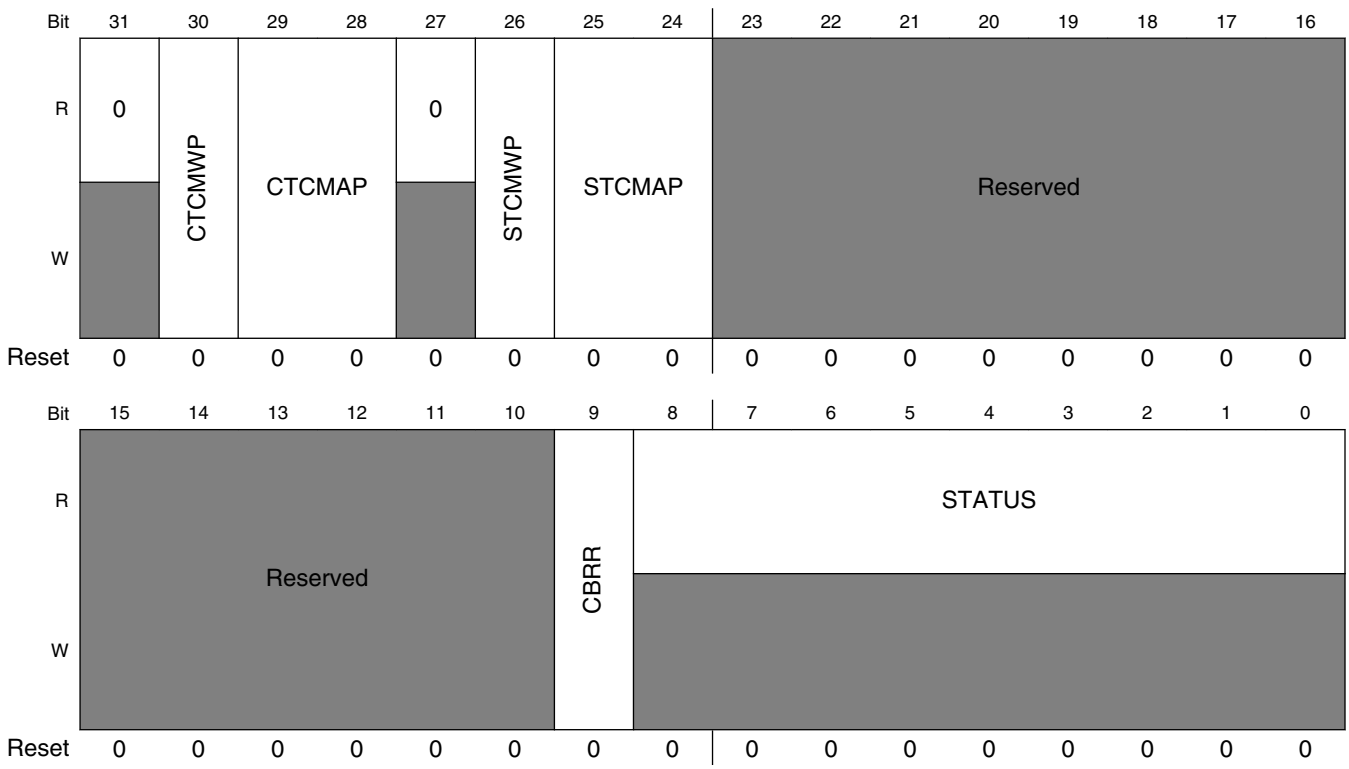
MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port. 0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

34.9.6 Control Register (MCM_CR)

CR defines the arbitration and protection schemes for the two system RAM arrays.

Address: E008_0000h base + Ch offset = E008_000Ch



MCM_CR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 CTCMWP	Code TCM Write Protect When this bit is set, writes to the code TCM array generates a bus error.
29–28 CTCMAP	Code TCM arbitration priority Defines the arbitration scheme and priority for the processor and Tightly Coupled Memory (TCM) backdoor accesses to the code TCM array. 00 Round robin 01 Special round robin (favors TCM backoor accesses over the processor) 10 Fixed priority. Processor has highest, backdoor has lowest 11 Fixed priority. Backdoor has highest, processor has lowest
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_CR field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>26
STCMWP</td><td>System TCM write protect

When this bit is set, writes to system Tightly Coupled Memory (TCM) array generates a bus error.</td></tr>
<tr><td>25–24
STCMAP</td><td>System TCM arbitration priority

Defines the arbitration scheme and priority for the processor and Tightly Coupled Memory (TCM) backdoor accesses to the system TCM array.

00 Round robin
01 Special round robin (favors TCM backoor accesses over the processor)
10 Fixed priority. Processor has highest, backdoor has lowest
11 Fixed priority. Backdoor has highest, processor has lowest</td></tr>
<tr><td>23–10
Reserved</td><td>This field is reserved.</td></tr>
<tr><td>9
CBRR</td><td>Crossbar round-robin arbitration enable

Configures the crossbar slave ports to fixed-priority or round-robin arbitration.

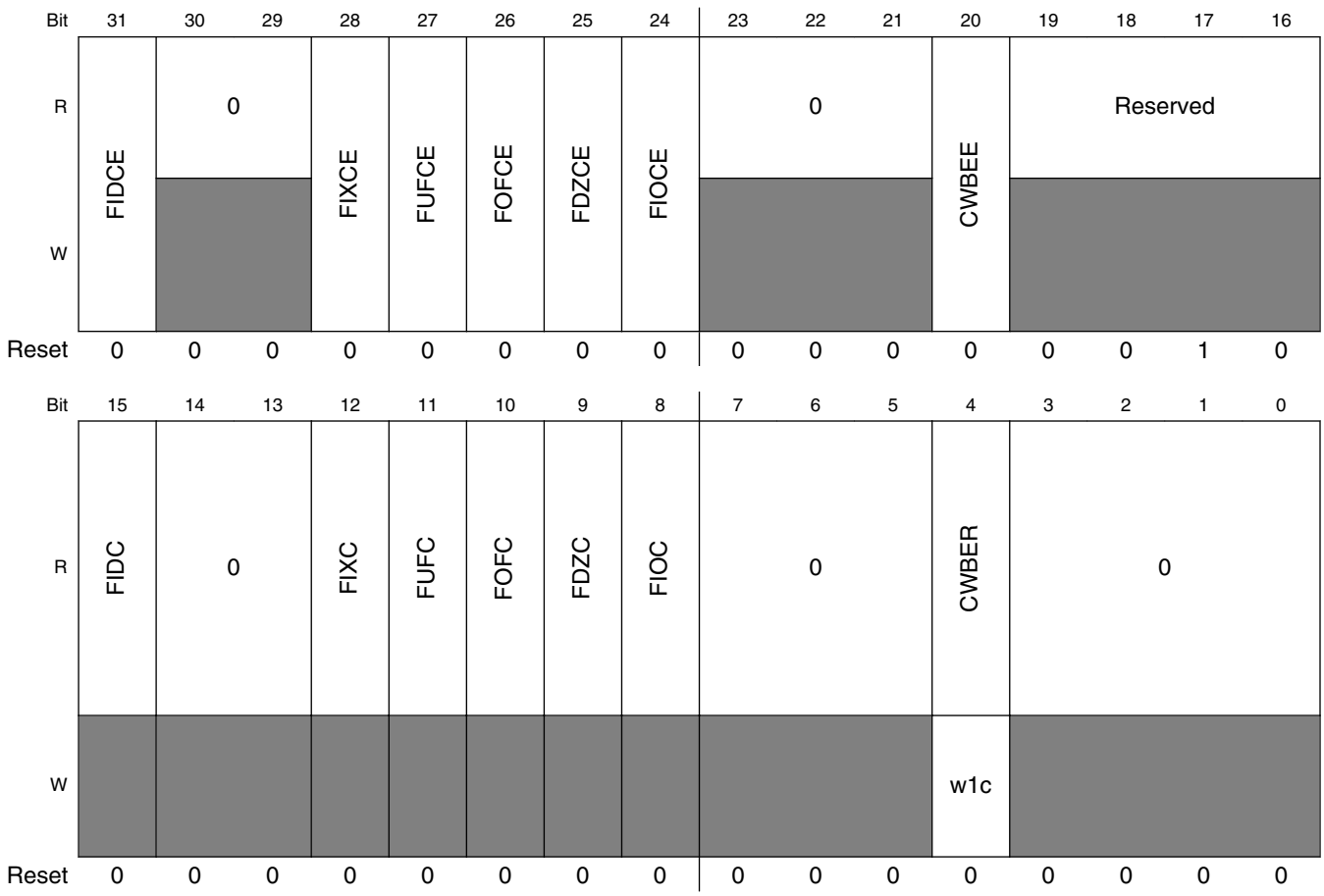
0 Fixed-priority arbitration
1 Round-robin arbitration</td></tr>
<tr><td>STATUS</td><td>Status bits
Information bits that can be used to determine what peripherals are not yet idle during the AXBS stall sequence (for power-down/clock change sequences).

<table>
<tr><th>STATUS bit</th><th>Description</th></tr>
<tr><td>8</td><td>CAU enable</td></tr>
<tr><td>7</td><td>Always 1</td></tr>
<tr><td>6</td><td>AIPS0 idle</td></tr>
<tr><td>5</td><td>Always 1</td></tr>
<tr><td>4</td><td>Always 1</td></tr>
<tr><td>3</td><td>Crossbar idle</td></tr>
<tr><td>2</td><td>Crossbar switch stall request</td></tr>
<tr><td>1-0</td><td>Crossbar switch stall FSM
• 00 - Wait for stall request
• 01 - Wait for all periphs to be idle
• 10 - Platform stalled
• 11 - Reserved</td></tr>
</table>
</td></tr>
</table>

34.9.7 Interrupt Status and Control Register (MCM_ISCR)

The MCM_ISCR register defines the configuration and reports status for a number of core-related interrupt exception conditions. It includes the enable and status bits associated with the core's floating-point exceptions, bus errors associated with the core's cache write buffer, and events associated with the debug ETB module. The individual event indicators are first qualified with their exception enables and then logically summed to form an interrupt request sent to the core's NVIC.

Address: E008_0000h base + 10h offset = E008_0010h



MCM_ISCR field descriptions

Field	Description
31 FIDCE	FPU input denormal interrupt enable 0 Disable interrupt 1 Enable interrupt

Table continues on the next page...

MCM_ISCR field descriptions (continued)

Field	Description
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 FIXCE	FPU inexact interrupt enable 0 Disable interrupt 1 Enable interrupt
27 FUFCE	FPU underflow interrupt enable 0 Disable interrupt 1 Enable interrupt
26 FOFCE	FPU overflow interrupt enable 0 Disable interrupt 1 Enable interrupt
25 FDZCE	FPU divide-by-zero interrupt enable 0 Disable interrupt 1 Enable interrupt
24 FIOCE	FPU invalid operation interrupt enable 0 Disable interrupt 1 Enable interrupt
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 CWBEE	Cache write buffer error enable Enables the generation of an interrupt in response to a bus error termination reported on a system bus transfer initiated from the cache's write buffer. 0 Disable error interrupt 1 Enable error interrupt
19–16 Reserved	This field is reserved.
15 FIDC	FPU input denormal interrupt status This read-only bit is a copy of the core's FPSCR[IDC] bit and signals input denormalized number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IDC] bit. 0 No interrupt 1 Interrupt occurred
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 FIXC	FPU inexact interrupt status This read-only bit is a copy of the core's FPSCR[IXC] bit and signals an inexact number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IXC] bit. 0 No interrupt 1 Interrupt occurred

Table continues on the next page...

MCM_ISCR field descriptions (continued)

Field	Description
11 FUFC	FPU underflow interrupt status This read-only bit is a copy of the core's FPSCR[UFC] bit and signals an underflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[UFC] bit. 0 No interrupt 1 Interrupt occurred
10 FOFC	FPU overflow interrupt status This read-only bit is a copy of the core's FPSCR[OFC] bit and signals an overflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[OFC] bit. 0 No interrupt 1 Interrupt occurred
9 FDZC	FPU divide-by-zero interrupt status This read-only bit is a copy of the core's FPSCR[DZC] bit and signals a divide by zero has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[DZC] bit. 0 No interrupt 1 Interrupt occurred
8 FIOC	FPU invalid operation interrupt status This read-only bit is a copy of the core's FPSCR[IOC] bit and signals an illegal operation has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IOC] bit. 0 No interrupt 1 Interrupt occurred
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 CWBER	Cache write buffer error status Signals a data transfer from the core's cache write buffer was terminated with a bus error. This bit only sets when the corresponding enable bit (CWBEE) is set. The corresponding core fault address, attributes and write data are typically retrieved from the FADR, FATR, and FDR registers during the interrupt service routine before clearing the CWBER flag. 0 No error 1 Error occurred
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.9.8 Fault address register (MCM_FADR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting address is captured in the MCM_FADR register. The MCM logic supports capturing a single cache write buffer bus error event; if a subsequent error is detected before the captured error information has been read from the corresponding registers and the MCM_ISCR[CWBER] indicator cleared, the MCM_FATR[BEOVR] flag is set. However, no additional information is captured.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 20h offset = E008_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FADR field descriptions

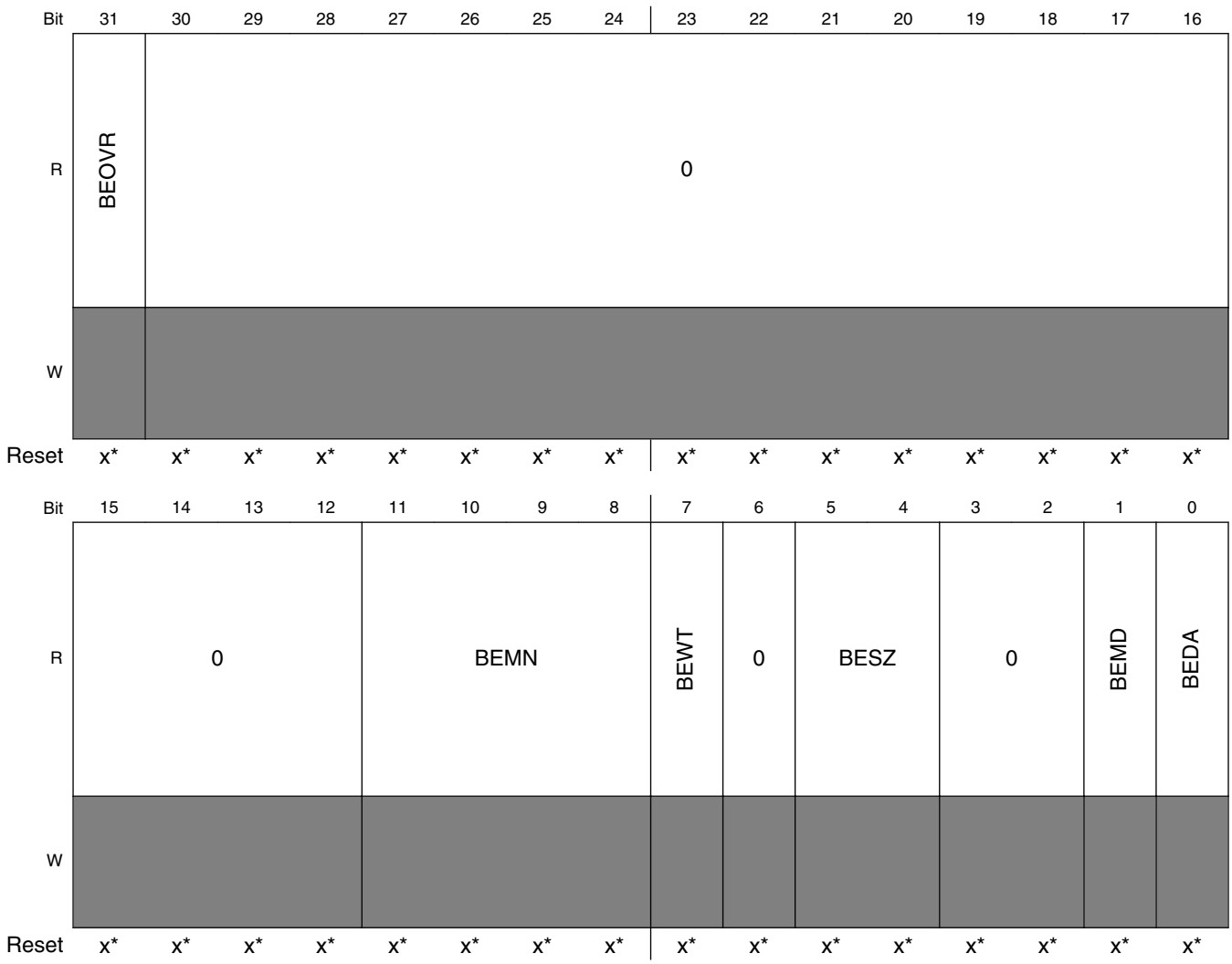
Field	Description
ADDRESS	Fault address

34.9.9 Fault attributes register (MCM_FATR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting attributes are captured in the MCM_FATR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 24h offset = E008_0024h



* Notes:
- x = Undefined at reset.

MCM_FATR field descriptions

Field	Description
31 BEOVR	Bus error overrun Indicates if anothercache write buffer bus error is detected before system software has retrieved all the error information from the original event, this overrun flag is set. The window of time is defined from the detection of the original cache write buffer error termination until the MCM_ISCR[CWBER] is written with a 1 to clear it and rearm the capture logic. This bit is set by the hardware and cleared whenever software writes a 1 to the CWBER bit. 0 No bus error overrun 1 Bus error overrun occurred. The FADR and FDR registers and the other FATR bits are not updated to reflect this new bus error.
30–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_FATR field descriptions (continued)

Field	Description
11–8 BEMN	Bus error master number Crossbar switch bus master number of the captured cache write buffer bus error. For this device, this value is always 0x1.
7 BEWT	Bus error write Indicates the type of system bus access when the error was detected. Since this logic is monitoring data transfers from the cache write buffer, this bit is always a logical one, signaling a write operation. 0 Read access 1 Write access
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 BESZ	Bus error size Indicates the size of the cache write buffer access when the error was detected. 00 8-bit access 01 16-bit access 10 32-bit access 11 Reserved
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 BEMD	Bus error privilege level Indicates the privilege level of the cache write buffer access when the error was detected. 0 User mode 1 Supervisor/privileged mode
0 BEDA	Bus error access type Indicates the type of cache write buffer access when the error was detected. This attribute is always a logical one signaling a data reference. 0 Instruction 1 Data

34.9.10 Fault data register (MCM_FDR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting data is captured in the MCM_FDR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. For byte and halfword writes, only the accessed byte lanes contain valid data; the contents of the other bytes are undefined. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 28h offset = E008_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FDR field descriptions

Field	Description
DATA	Fault data

34.9.11 Local Memory Descriptor Register (MCM_LMDR*n*)

NOTE

The LMDRn registers mapping to the LMEMs is as follows: LMDR0:code TCM, LMDR1:system TCM, LMDR2:PC Cache, LMDR3: PS Cache.

This section of the programming model is an array of 32-bit generic on-chip memory descriptor registers that provide static information on the attached memories as well as configurable controls (where appropriate).

Privileged 32-bit reads from a processor core or the debugger return the appropriate processor information. Reads from any other bus master return all zeroes. Privileged writes from a processor core or the debugger to writeable registers update the appropriate fields. Privileged writes from other bus masters are ignored. Attempted user mode accesses or any access with a size other than 32 bits are terminated with an error.

Address: E008_0000h base + 400h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved	Reserved	LMSZH	LMSZ				WY				DPW			RO
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MT			Reserved	Reserved				CF1				CF0			
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- The reset values are different for the individual LMDR registers. LMDR0: 0x8806_0000; LMDR1: 0x8806_2000; LMDR2: 0x8526_4000; LMDR3: 0x8526_6000. x = Undefined at reset.

MCM_LMDR*n* field descriptions

Field	Description
31 V	Local memory Valid bit. This read-only field defines the validity (presence) of the local memory. 0 LMEMn is not present. 1 LMEMn is present.
30 Reserved	This field is reserved.
29 Reserved	This field is reserved.
28 LMSZH	LMEM Size “Hole”. For local memories that are not fully populated, that is, include a memory “hole” in the upper 25% of the address range, this bit is used. 0 LMEMn is a power-of-2 capacity. 1 LMEMn is not a power-of-2, with a capacity is 0.75 × LMSZ.
27–24 LMSZ	LMEM Size. This read-only field provides an encoded value of the local memory size. The capacity of the memory is expressed as Size [bytes] = $2^{(9+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present. 0000 no LMEMn (0 KB) 0001 1 KB LMEMn 0010 2 KB LMEMn 0011 4 KB LMEMn 0100 8 KB LMEMn 0101 16 KB LMEMn 0110 32 KB LMEMn 0111 64 KB LMEMn 1000 128 KB LMEMn 1001 256 KB LMEMn 1010 512 KB LMEMn 1011 1024 KB LMEMn 1100 2048 KB LMEMn 1101 4096 KB LMEMn 1110 8192 KB LMEMn 1111 16384 KB LMEMn
23–20 WY	Level 1 Cache Ways 0000 No Cache 0010 2-Way Set Associative 0100 4-Way Set Associative
19–17 DPW	LMEM Data Path Width. This read-only field defines the width of the local memory. 000-001 Reserved 010 LMEMn 32-bits wide 011 LMEMn 64-bits wide 100-111 Reserved

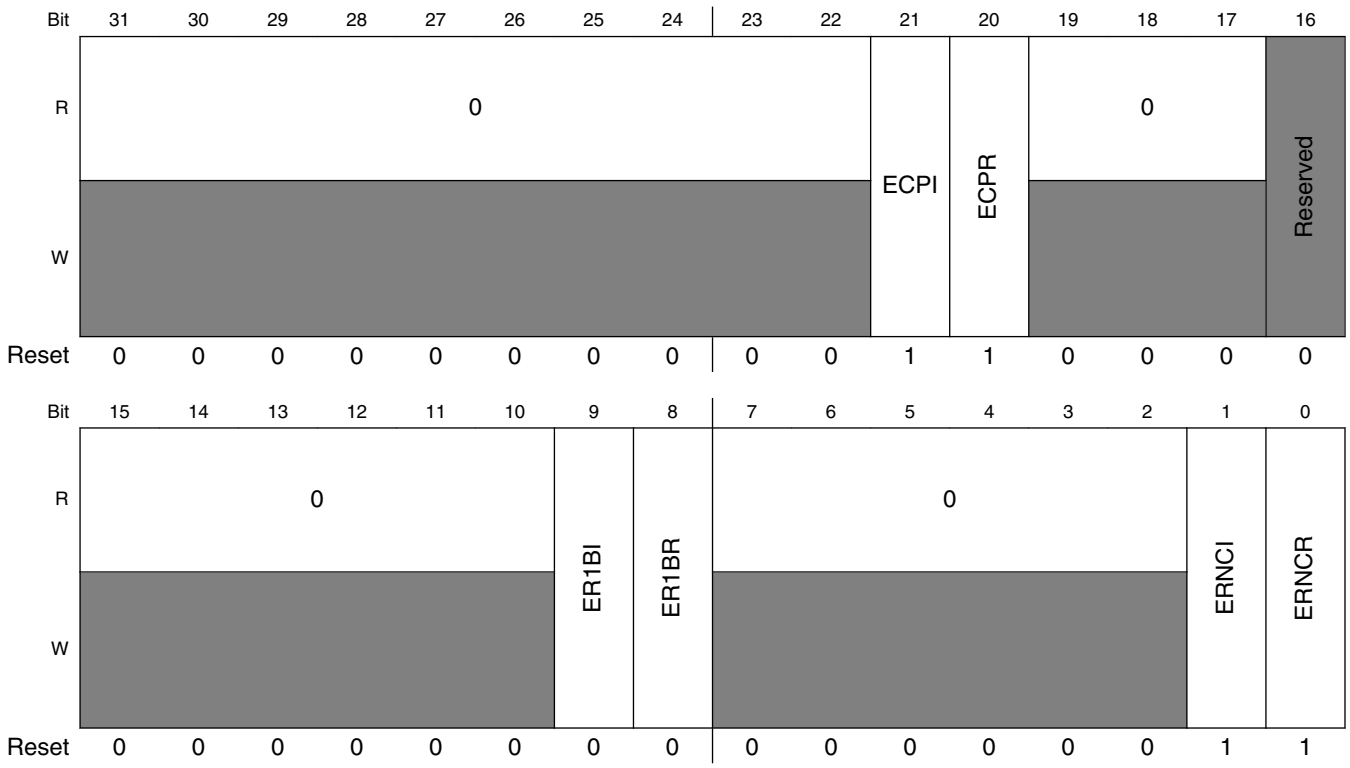
Table continues on the next page...

MCM_LMDR*n* field descriptions (continued)

Field	Description
16 RO	Read-Only. This register bit provides a mechanism to “lock” the configuration state defined by LMDRn[7:0]. Once asserted, attempted writes to the LMDRn[7:0] register are ignored until the next reset clears the flag. 0 Writes to the LMDRn[7:0] are allowed. 1 Writes to the LMDRn[7:0] are ignored.
15–13 MT	Memory Type This field defines the type of the local memory. 000 code TCM 001 system TCM 010 PC Cache 011 PS Cache
12 Reserved	This field is reserved.
11–8 Reserved	This field is reserved.
7–4 CF1	Control Field 1 - for Cache Parity control functions • CF1[3] - PCPFE = PC Parity Fault Enable • CF1[2] - PSPFE = PS Parity Fault Enable • CF1[1] - PCPME = PC Parity Miss Enable • CF1[0] - PSPME = PS Parity Miss Enable
CF0	Control Field 0 - for TCM ECC control functions • CF0[3] - PFE = Parity Fault Enable • CF0[2] - Reserved • CF0[1] - EERC = ECC Enable Read Check • CF0[0] - EEWG = ECC Enable Write Generation

34.9.12 LMEM Parity & ECC Control Register (MCM_LMPECR)

Address: E008_0000h base + 480h offset = E008_0480h



MCM_LMPECR field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 ECPI	Enable Cache Parity IRQ 0 disabled 1 enabled
20 ECPR	Enable Cache Parity Reporting 0 reporting disabled 1 reporting enabled
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	This field is reserved.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_LMPECR field descriptions (continued)

Field	Description
9 ER1BI	Enable RAM ECC 1-bit Interrupt 0 Interrupt is disabled 1 Interrupt is enabled
8 ER1BR	Enable RAM ECC 1-bit Reporting 0 reporting disabled 1 reporting enabled
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ERNCI	Enable RAM Non-correctable ECC Interrupt 0 Interrupt is disabled 1 Interrupt is enabled
0 ERNCR	Enable RAM ECC Non-correctable Reporting 0 reporting disabled 1 reporting enabled

34.9.13 LMEM Parity & ECC Interrupt Register (MCM_LMPEIR)

NOTE

Write 1 to the error bit in MCM_LMPEIR[23:0] can clear the interrupt flag.

Address: E008_0000h base + 488h offset = E008_0488h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	0		PEELOC					PE							
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	E1B								ENC							
W	w1c								w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_LMPEIR field descriptions

Field	Description
31 V	Valid bit This bitfield will be cleared, if Error Flag in bit 23-0 is cleared.

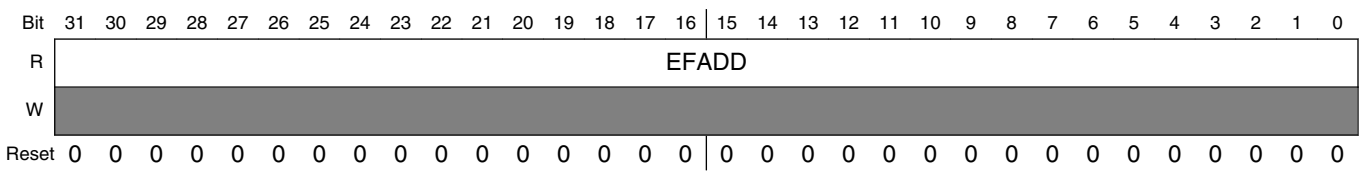
Table continues on the next page...

MCM_LMPEIR field descriptions (continued)

Field	Description
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 PEELOC	Parity or ECC Error Location. This bitfield will be cleared, if Error Flag in bit 23-0 is cleared. • 5'h00 - a non-correctable ECC event from code TCM • 5'h01 - a non-correctable ECC event from system TCM • 5'h08 - a 1-bit correctable ECC event from code TCM • 5'h09 - a 1-bit correctable ECC event from system TCM • 5'h14 - a PC Tag Parity Error • 5'h15 - a PC Data Parity Error
23–16 PE	Parity Error • [23] - PS Data Parity Error • [22] - PS Tag Parity Error • [21] - PC Data Parity Error • [20] - PC Tag Parity Error • [19] - RESERVED • [18] - RESERVED
15–8 E1B	E1Bn = ECC 1-bit Error n • PEIR[15:10] - Reserved • PEIR[9] - 1-bit Error detected on system TCM • PEIR[8] - 1-bit Error detected on code TCM
ENC	ENCn = ECC Non-correctable Error n • PEIR[7:2] - Reserved • PEIR[1] - Non-correctable Error detected on system TCM • PEIR[0] - Non-correctable Error detected on code TCM

34.9.14 LMEM Fault Address Register (MCM_LMFAR)

Address: E008_0000h base + 490h offset = E008_0490h

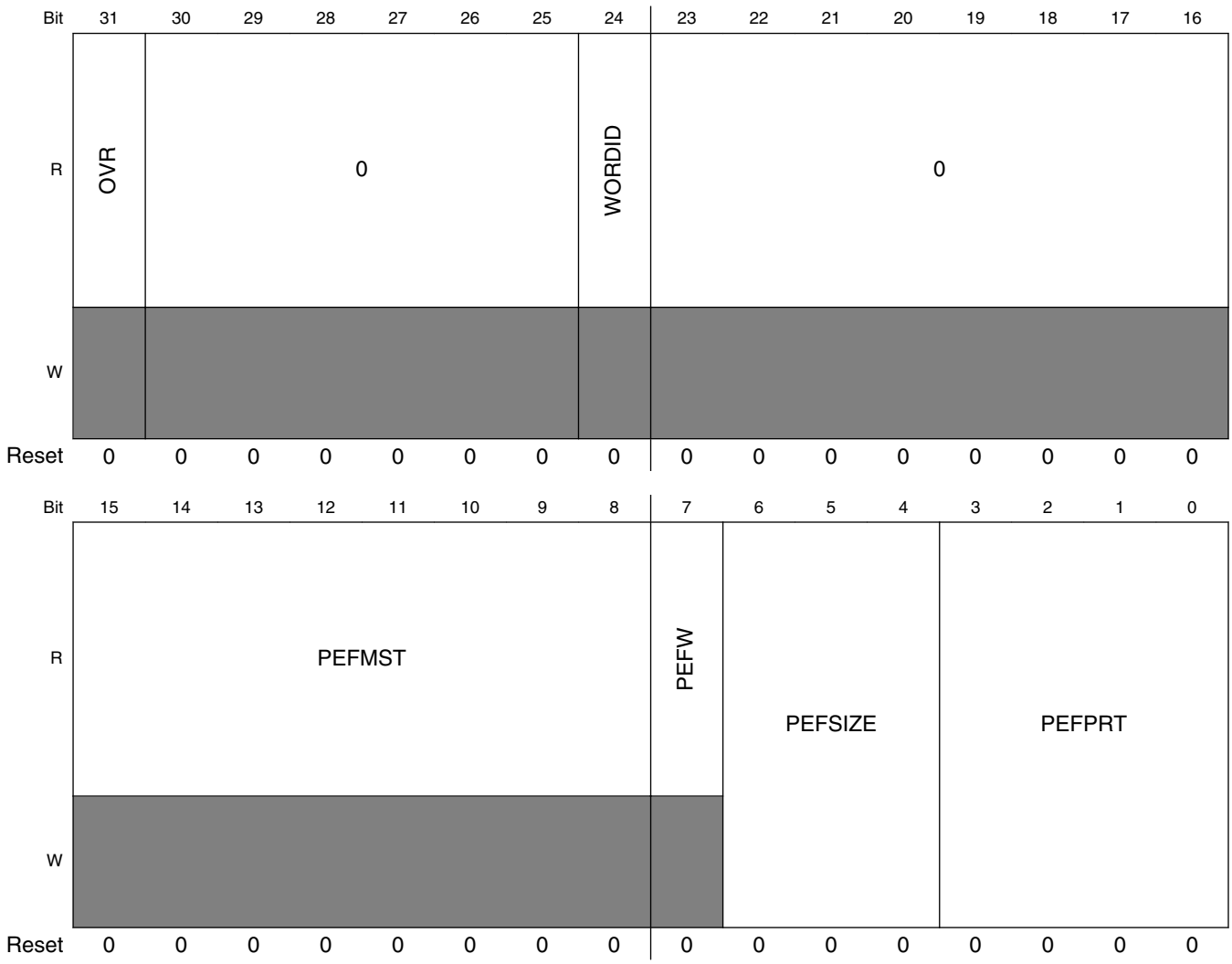


MCM_LMFAR field descriptions

Field	Description
EFADD	ECC Fault Address

34.9.15 LMEM Fault Attribute Register (MCM_LMFATR)

Address: E008_0000h base + 494h offset = E008_0494h



MCM_LMFATR field descriptions

Field	Description
31 OVR	Overrun ECC check error has been happened more than once.
30–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 WORDID	ID of the word which has ECC error

Table continues on the next page...

MCM_LMFATR field descriptions (continued)

Field	Description
23–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 PEFMST	Parity/ECC Fault Master Number
7 PEFW	Parity/ECC Fault Write
6–4 PEFSIZE	Parity/ECC Fault Master Size • 3'b000 = 8-bit access • 3'b001 = 16-bit access • 3'b010 = 32-bit access • 3'b011 = 64-bit access • 3'b1xx = Reserved
PEFPRT	Parity/ECC Fault Protection • FATR[3] is Cacheable: 0=Non-cacheable, 1=Cacheable • FATR[2] is Bufferable: 0=Non-bufferable, 1=Bufferable • FATR[1] is Mode: 0=User mode, 1=Supervisor mode • FATR[0] is Type: 0=I-Fetch, 1=Data

34.9.16 LMEM Fault Data High Register (MCM_LMFDHR)

Address: E008_0000h base + 4A0h offset = E008_04A0h

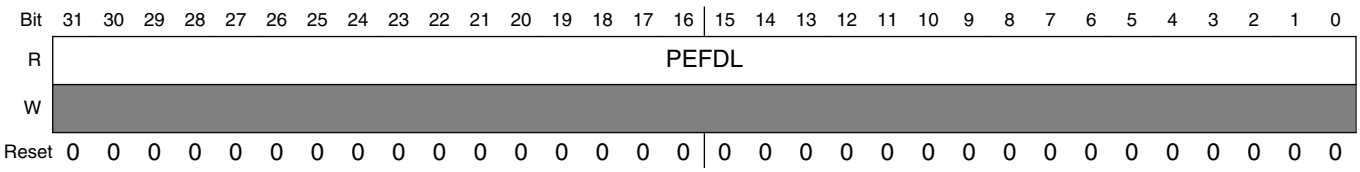
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PEFDH																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_LMFDHR field descriptions

Field	Description
PEFDH	Parity or ECC Fault Data High

34.9.17 LMEM Fault Data Low Register (MCM_LMFDLR)

Address: E008_0000h base + 4A4h offset = E008_04A4h



MCM_LMFDLR field descriptions

Field	Description
PEFDL	Parity or ECC Fault Data Low

Chapter 35
Messaging Unit (MU)

35.1 Chip-specific MU information

Table 35-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments
System Debug	-	System Debug

35.2 Overview

The Messaging Unit (MU) module enables two processors within the SoC to communicate and coordinate by passing messages (e.g. data, status and control) through the MU interface. The MU also provides the ability for one processor to signal the other processor using interrupts.

Because the MU manages the messaging between processors, the MU uses different clocks (from each side of the different peripheral buses). Therefore, the MU must synchronize the accesses from one side to the other. The MU accomplishes synchronization using two sets of matching registers (Processor A-facing, Processor B-facing).

35.2.1 Block Diagram

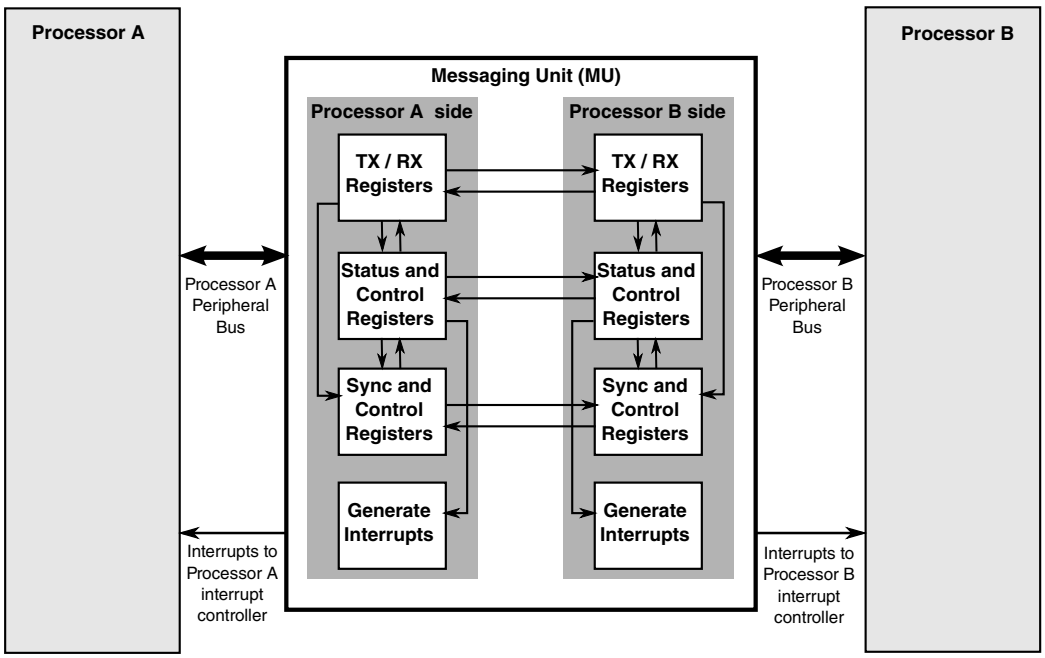


Figure 35-1. MU Block Diagram

35.2.2 Features

The MU includes the following features:

- Messaging control by interrupts or by polling
- Symmetrical processor interfaces with each side supporting the following:
  - Four general-purpose interrupt requests reflected to the other side
  - Three general-purpose flags reflected to the other side [1]
  - Four receive registers with maskable interrupt
  - Four transmit registers with maskable interrupt
- The above interrupts can also be used for waking up the other processor from low-power mode.

NOTE

MU chapter references to Processor A correspond to Core Complex 0 and references to Processor B correspond to Core Complex 1 in this device.

1. For example, MUA (Processor A) can set the flag to MUB (Processor B), and MUB(Processor A) can set the flag to MUA(Processor A).

35.3 Functional Description

Table 35-2. Major Features of the MU

Major Feature	Description
Interprocessor Interrupts	• The MU has 12 interrupt sources on each side (Processor A-side, Processor B-side) that are used for signaling the other processor. The interrupts can be used for notification of RX/TX events and general-purpose signaling between the processors.
MU Reset	• The Processor A can issue a reset to the entire MU, using a control bit (MUR) in the Processor A Control Register (ACR). • The MUR bit is a self-clearing bit.
Status and Control Communications between Core Complexes	• The MU provides a way for the two core complexes to communicate using the status and control registers present on both the Processor B and Processor A sides of the MU. • The status register of one MU side reflects the status of the other MU side. • The control register is used for control operations, such as enabling an interrupt and sending an interrupt to the other processor.
Synchronized Message Transfers between Core Complexes	• The transfer of data messages between core complexes uses transmit empty and receive full flags provided on both sides of the MU. • The update of these transmit and receive flags is accomplished using a synchronization mechanism. There is inherent latency between updating the flag on one side and reflecting its status on other side. For more about latency, see Event Update Timing
Accessing Shared Memory Directly and Avoiding Collisions	• For sending data or messages from one MU-side to the other MU-side, the MU provides 4 transmit registers and 4 receive registers on each side of the MU. • The Processor A or Processor B can access shared memory resources of the SoC directly. However, to avoid simultaneous access to shared memory by both core complexes, the MU provides a method (to prevent simultaneous access) using interrupts and transmit-receive registers for both processors. See Messaging Examples for more information.
Support for Different Clocks in the Two Core Complexes	• The heart of the MU module is the event control mechanism, which synchronizes the access of one MU-side to the other MU-side, because these two MU-sides can operate using different clocks. • Formulated event update latency.
Memory-Mapped Registers	• The MU is connected as a peripheral under the Peripheral bus on both sides—on the Processor A-side, the Processor A Peripheral Bus, and on the Processor B-side, the Processor B Peripheral Bus.

35.3.1 Operating Modes

The MU operates in one of two modes: Run or low power.

35.3.2 Low Power Modes

This section describes the low power operating modes of the MU module.

35.3.2.1 Low Power Clocks and Synchronization

The Processor B and the Processor A clocks operate at different frequencies and from different sources. The MU design does not assume any frequency relationship between the Processor A and the Processor B clocks. Be aware, however, that the frequency relationship affects the MU's throughput performance.

The data buffers and control logic of each MU-side operate with its corresponding clock.

35.3.2.2 Processor Low Power Modes

The Processors have four power modes:

- Run
- WAIT
- STOP
- SUSPEND, can also be called DSM (Deep Sleep Mode)

The Processor can be awakened from a low-power mode by any enabled Processor side MU interrupt, as reflected in the xSR "status" register (RF0–3, TE0–3, GIP0–3 bits are set) and enabled in the xCR control register. Using these bits, the Processor can actively control when to wake the other Processor.

While the Processor is in STOP mode (such that the xSR register bits cannot be updated with events), special logic drives the enabled MU interrupts directly from the other Processor-side (instead of from the xSR register).

While the Processor is in STOP mode, the asynchronous MU interrupt will be asserted to wake the Processor:

- If any transmit data register of the transmitter Processor-side is full, because of a write to it (transmit data register); that is, its "empty" bit in the xSR register is cleared while its corresponding receive interrupt is enabled on the receiver Processor-side.
- If any receive data register of the receiver Processor-side is empty, because of a read on the other Processor -side; that is, its "full" bit in the xSR register is cleared while its corresponding transmit interrupt is enabled on the transmitter Processor-side.
- If any general purpose interrupt bit is set in the xCR register on the other Processor-side while the corresponding interrupt is enabled on the Processor-side.

The logic enables the other Processor to operate independently while the Processor is in any power mode (including STOP). However, the Processor power mode change protocol should be handled with care regarding:

- The interrupts that are enabled on the Processor-side

- The events that could be triggered by the other Processor-side
- The compatibility with the other Processor protocol of entering STOP mode

If the Processor is in STOP mode and an event on the other Processor is triggered, the EP bit (in the xSR register) will remain high until the Processor wakes up.

Before entering STOP mode, the Processor programmer should verify that the EP bit (in the xSR register) is cleared. This check is needed to ensure that all pending updates from the Processor, including the power mode change when STOP or WAIT is executed, will be updated in the xSR register.

- If the other Processor is in STOP mode or DSM mode, the EP bit (in the xSR register) may be stuck high; in this case, the Processor need not check the EP bit before entering STOP mode.

35.3.3 Processor A Side Memory-Mapping

The messaging, control, and status registers of the Processor A-side for the MU are mapped to the Processor A memory as a regular peripheral. The Peripheral bus data is 32 bits wide inside the MU module.

35.3.4 Processor B Side Memory-Mapping

The messaging, control, and status registers of the Processor B-side for the MU are mapped to the Processor B memory as a regular peripheral. The Peripheral bus data is 32 bits wide inside the MU module.

35.3.5 MU Messaging

The MU provides 32-bit status and control registers to the Processor B and Processor A sides for control operations (such as interrupts and reset), and for status checking of the other MU-side.

For messaging, the MU has four, 32-bit write-only transmit registers and four, 32-bit read-only receive registers on the Processor B and Processor A-sides. These registers are used for sending messages to each other. These messages can also be controlled using the 3 general purpose flags provided in the control and status registers of either MU-side.

35.3.5.1 Programmer Model

The messaging logic is used in conjunction with external memory. There are various messaging methods, which can be used to implement a messaging protocol. Some of these messages can mean “a message of N words has just been written, starting at offset X in the memory,” or “a previous data block that was sent has just been read.” Having the messaging logic independent from the memory array does not restrict users to a predefined hardware protocol. On the other hand, the software needed to manage the messaging is short and straightforward.

Most of the messaging mechanisms are symmetric; they are duplicated and are available on both the Processor B-side and the Processor A-side. The messaging mechanisms are:

- Four, 32-bit write-only transmit registers, which are each reflected in four, read-only receive registers in the other processor’s side. Users can use these registers to transfer 32-bit word messages or frame information of messages written to the shared memory (number of words, initial address, and message type code).
- A write to a transmit register on the transmitter side clears a “transmitter empty” bit in the Status Register on the transmitter side, and sets a “receiver full” bit in the Status Register on the receiver side. The setting of the bit at the receiver side can optionally trigger an interrupt at the receiver side (maskable receive interrupt).
- A read of one of the receive registers at the receiver side clears the “receiver full” bit in the Status Register at the receiver side, and sets the “transmitter empty” bit in the Status Register on the transmitter side. The setting of the “transmitter empty” bit can optionally trigger an interrupt at the transmitter side (maskable transmit interrupt).
- Four general purpose flags are reflected in the Status Register on the receiver side
- A read/write access to any reserved location and a write to a read-only register on the Processor A-side of the MU will generate a module transfer error acknowledge to the Processor A.
- A read/write access to any reserved location and write to a read-only register on the Processor B-side of the MU will generate a module transfer error acknowledge to the Processor B.

35.3.5.2 Messaging Examples

The following are messaging examples:

- **Passing short messages:** Transmit register(s) can be used to pass short messages from one to four words in length. For example, when a four-word message is desired, only one of the registers needs to have its corresponding interrupt enable bit set at the receiver side; the message’s first three words are written to the registers whose interrupt is masked, and the fourth word is written to the other register (which triggers an interrupt at the receiver side).

- **Passing frame information:** Transmit registers can be used to pass frame information for long messages written to the shared system memory. Such frame information normally includes a start address, number of words, and perhaps a message type code.
- **Passing event notices and requests:** Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the general interrupts, such as acknowledging that a long message was read from the shared system memory.
- **Passing fixed length data:** Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general interrupt (Processor A or Processor B) to signal the other processor that the data is ready.
- **Passing announcements:** The three flags can be used by a processor to announce its current program state or other billboard messages to the other processor.

Figure 35-2 shows the MU registers schematic.

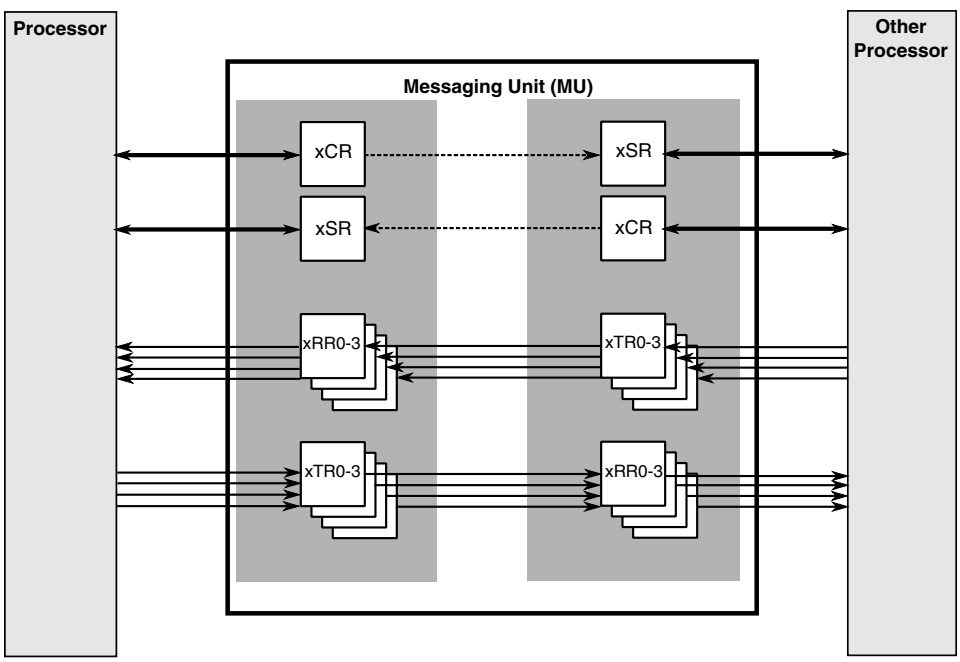


Figure 35-2. MU Registers

35.3.6 Event Update Timing

Each processor's MU messaging side (Processor B or Processor A) has a hardware mechanism to send "event update requests" to the other processor's side. An "event" is considered when any information change should be reflected at the Status Register of the receiving processor. The event update latency is the delay between the event being ready at one processor and the resulting update at the Status Register of the other processor.

- The minimum event latency is "1 clock of the sending side" + "2 1/2 clocks of the receiving side". The minimum case is if there is no event pending when the new event occurs. See clocking chapter for this device for more information about clock sources.
- The maximum event latency is "6 clocks of the sending side" + "6 1/2 clocks of the receiving side." The maximum case is if the event occurred just after a previous event was sent to the other side. The event update latency will vary between the above-mentioned minimum and maximum latencies, depending on the time at which the subsequent event is triggered.

35.3.7 Reset

The MU has two sources of reset, and each reset has a different function from the MU or system perspective.

- One asynchronous system that is connected to both sides of the MU interface.
- One programmable hardware reset (MUR bit) in the ACR register (on the Processor A-side).

Table 35-3. MU programmable resets

Reset	Description
Processor A MU reset	• Processor A MU Reset bit (MUR) of the ACR register • The MUR reset affects the messaging section on both the Processor A and the Processor B sides. The MUR reset causes all control and status registers to return to their default values and all internal states to be cleared. • It is up to the Processor A software to decide whether to use the MUR reset or not. • The instruction immediately following assertion of the MUR bit should not write to MU registers. Such a write may be overwritten by the reset sequence and the register will remain with the reset value. Wait at least one instruction (after assertion of the MUR bit) before attempting a write to MU registers.

After issuing MUR bit reset events, the Processor A programmer can verify that the reset sequence on the Processor B-side has ended, by checking the RS bit in the ASR register.

NOTE

MUR bit assertion is a delicate operation because it affects the other side's registers asynchronously. MUR bit assertion may

cause unpredictable behavior if, for example, the Processor B is concurrently testing an MU register bit (TE bit in Processor B SR register). Before asserting the MUR bit, it should be verified that the Processor B is not presently engaged in an MU signalling activity.

35.3.8 Interrupts

The MU controls the Processor B interrupt requests to the Processor A, and the Processor A interrupt requests to the Processor B. This section describes all the interrupts that the module generates.

35.3.8.1 Interrupts to the Processors

There are 12 interrupt sources from the MU to the Processors:

- Four receive interrupts for each of the receive registers: Asserted when the Processor receive full bits (xSR[RFn]) are set and the xCR[RIEn] bits enabled
- Four transmit interrupts for each of the transmit registers: Asserted when the Processor transmit empty bits (xSR[TEn]) are set and the xCR[TIEn] bits enabled
- Four general purpose interrupts: Asserted when the xSR[GIPn] bits are set and the xCR[GIEn] bits enabled

All the interrupts are maskable in the Processor Control Register (xCR). The MU does not assume any internal priority of these interrupts. Multiple interrupts (for example, Receive 0 and Receive 1 interrupts or any of the transmit and general purpose interrupts) can be asserted at one time. The priority of these interrupts should be resolved by the interrupt controller at the chip level.

The General Purpose Interrupt Pending bits (GIP0, GIP1, GIP2 and GIP3) should be cleared by the software (as part of the interrupt service routine) to de-assert the request to the interrupt controller.

35.3.8.2 General Purpose Interrupt Clearing Sequence

When a Processor writes to the general interrupt bit (GIR), the write event is synchronized to the other Processor clock to set the general interrupt request pending bit (GIP). When the GIP bit is set, and if the general purpose interrupt is enabled on the receiving Processor side (GIE bit is set), then the transmitting Processor general purpose interrupt is issued to the receiving Processor. The receiving Processor clears this interrupt by writing a “1” on the GIP bit. The interrupt is de-asserted as soon as the GIP bit is

written. The write event of the GIP bit is synchronized to the other Processor clock. The synchronized signal clears the GIR bit. The software should not write the GIR bit again until the GIR bit is cleared.

35.3.9 Interrupt Messaging Protocols

35.3.9.1 Messaging Protocols using Interrupts

The example below describes a four-word messaging sequence sent by the Processor to the other Processor.

In this example, the first, second, and third receive interrupts are disabled, and the fourth receive interrupt is enabled. We write registers sequentially for n = 0, 1, 2, 3. For n = 0, 1, 2, the interrupts are disabled, therefore no interrupt will go to the other core complex (although interrupt conditions occur). For n = 3, the interrupt is enabled, and the last Receive Interrupt request is generated.

1. Write Sequence
   - The Processor writes the message information sequentially to its Transmit Registers 0, 1, 2.
   - When the write to the Transmit Register 3 occurs, the RF3 bit of the xSR is set after synchronization, and it immediately trigger the Receive 3 interrupt to the other Processor.
2. Read Sequence
   - The other Processor receives the Receive 3 interrupt and starts reading the message transferred from the receive registers.
   - After Receive Register 3 is read, the interrupt bit is cleared.

Figure 35-3 shows the programmer's model of a messaging protocol using transmit and receive registers. Use Table 35-4 and Figure 35-3 to understand the generalized protocol sequence.

Table 35-4. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A Data write	A data write to the ATRn register by Processor A is immediately reflected in the Processor B BRRn register.
2	Clear Tx Empty bit and Set Rx Full bit	The data write to the ATRn register • Clears the transmitter empty bit (TEn) in the Processor A Transmit Status Register • Sets the receiver full bit (RFn) in the Processor B Receive Status Register

Table continues on the next page...

Table 35-4. Interrupt Messaging Protocol (Generalized) (continued)

Sequence	Action	Description
3	Generate Receive Interrupt request	The setting of the receiver full bit (RFn) in the Receive Status Register generates a Receive Interrupt request to Processor B.
4	Processor B Data read	After receiving the Receive Interrupt request, Processor B performs a data read of the BRRn register.
5	Clear Rx Full bit and Set Tx Empty bit	Reading the data out of the BRRn register • Clears the receiver full bit (RFn) in the Processor B Receive Status Register • Sets the transmitter empty bit (TEn) in the Processor A Transmit Status Register
6	Generate Transmit Interrupt request	The setting of the transmitter empty bit (TEn) in the Transmit Status Register generates a Transmit Interrupt request to Processor A.

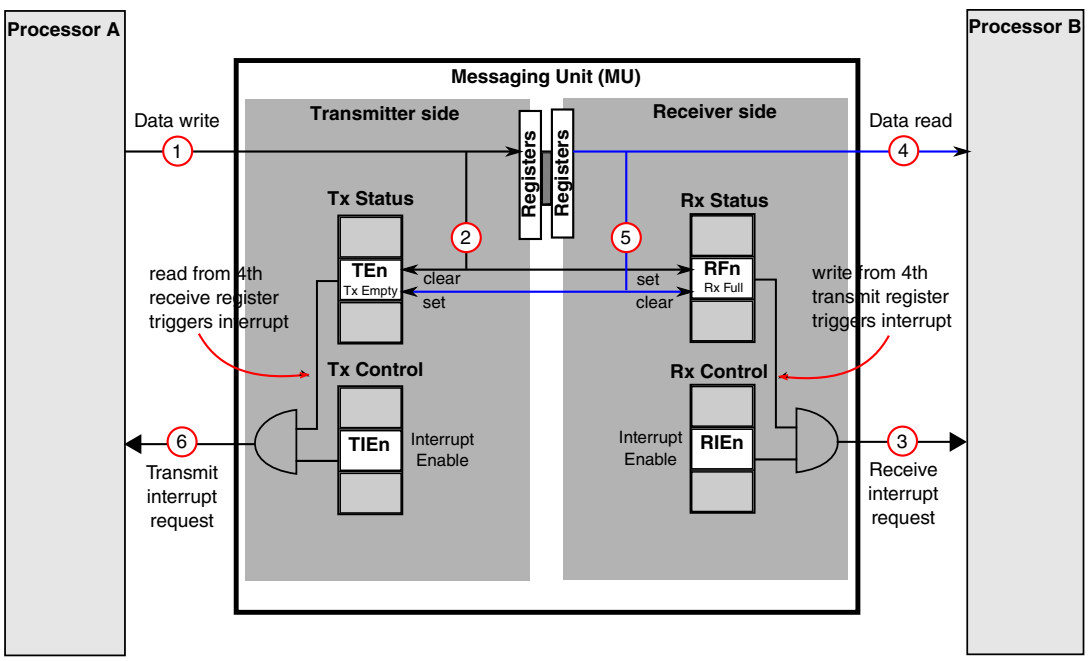


Figure 35-3. Messaging Model Using Transmit and Receive Registers

NOTE

The Transmit registers can be used to pass frame information on long messages written to the shared memory. Such frame information would typically include an initial address, number of words, and perhaps a message type code.

The messaging hardware can be used by software to implement messaging protocols for a wide array of message types. Full support is given for both interrupt and polling management schemes.

35.3.9.2 Messaging Protocols using Event Interrupts

Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the two general interrupts.

Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general purpose interrupt to signal the other processor that the data is ready.

The three flags can be used by a processor to announce to the other processor the program state it is currently in, or to announce similar messages.

Table 35-5 and Figure 35-4 describe the event sequence when the Processor triggers an interrupt.

Table 35-5. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A sets General Interrupt request bit	Processor A sets its associated General Interrupt request bit (GIRn = 1) in the control register (ACR).
2	General Interrupt Request Pending status bit is set	The General Interrupt Request Pending status bit (GIPn) in the status register (BSR) is set to "1"
3	General Interrupt request to Processor B is generated	Setting the GIPn bit generates the General Interrupt request to Processor B (Interrupt Request Enable bit, GIEn, must be set for Processor B)
4	Processor B reads status register	The Processor B reads the GIPn bit in the BSR register.
5	Processor B services the interrupt	-
6	Processor B sets GIPn bit to clear interrupt	The Processor B writes "1" to the corresponding GIPn bit to clear the interrupt
7	GIRn bit is cleared	Setting the GIPn bit to "1" clears the General Interrupt request bit (GIRn) in the Processor A control register (ACR).

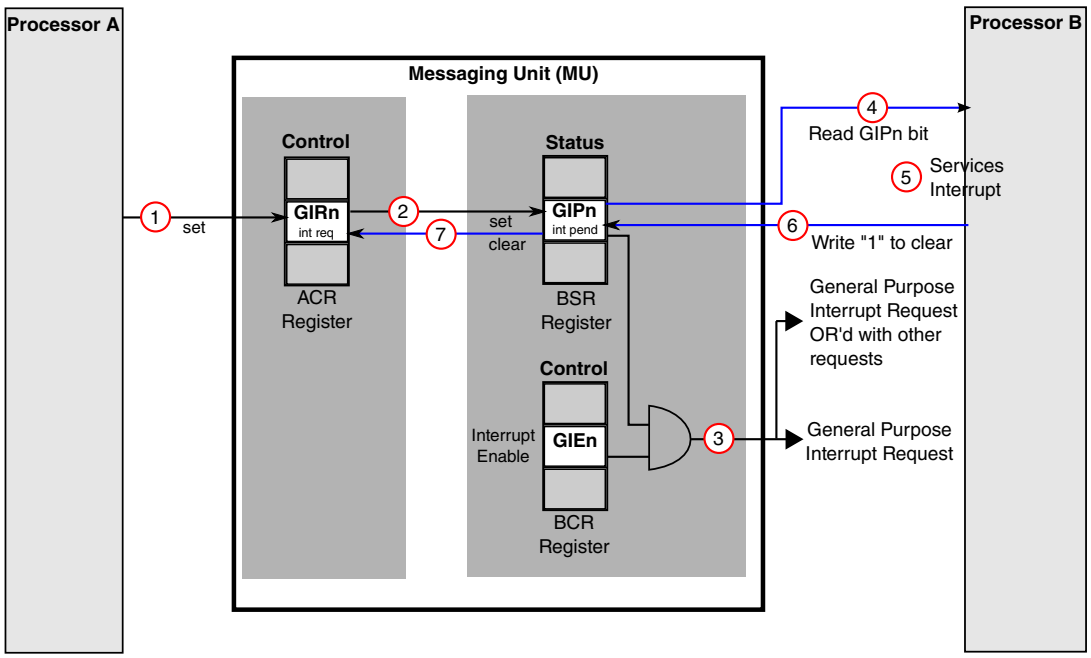


Figure 35-4. Messaging Model Using a General Purpose Interrupt

35.3.10 Exclusive Access to Shared Memory

The MU can be used to signal one processor about its current access to the shared memory, so that the data is not overwritten by the other processor during the exclusive memory access period.

The following tables describe the signaling protocol that the Processor A uses to inform the Processor B about its current access (write) to the shared memory, assuming that the set of bits and registers (GIR0 bit, BRR0 register, BTR0 register, GIR0 bit, ARR0 register, ATR0 register) are reserved to support exclusive access to the shared memory protocol.

Table 35-6. How the Processor A Performs an Exclusive Access to Shared Memory

Sequence	Action	Description
1	Processor A sends GIRn request to Processor B using Processor A control register	When the Processor A wants to perform an exclusive access to the shared memory, the Processor A sends an GIR0 request to the Processor B.
2	Processor A sends an exclusive-access request using a transmit data register (ATRn)	The Processor A will send an exclusive-access request (command, location, and length of target access) to Processor B using a selected transmit data register (ATR0).
3	Processor A waits for a dedicated interrupt from Processor B	The Processor A waits for a dedicated interrupt (as an acknowledgement) triggered by the Processor B before proceeding.

Table continues on the next page...

Table 35-6. How the Processor A Performs an Exclusive Access to Shared Memory (continued)

Sequence	Action	Description
4	Processor A accesses shared memory	After receiving a dedicated interrupt from the Processor B, Processor A proceeds.

Table 35-7. How the Processor B Scans for Transaction Information

Sequence	Action	Description
1	Processor B receives an interrupt from a receive data register (BRRn)	-
2	Processor B reads the receive data register (BRRn)	-
3	Processor B scans the receive data register contents	For transaction information (whether Processor A has requested an exclusive-access)

Table 35-8. How the Processor B Accepts Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B triggers a dedicated interrupt	Processor B acknowledges the Processor A request by triggering a dedicated interrupt (ack) to the Processor A.
2	Processor B sends a code message to Processor A	Along with the acknowledge interrupt, the Processor B sends a code message to the Processor A through the selected transmit register (BTRn). The message informs the Processor A that it can exclusively access the shared memory.

Table 35-9. How the Processor B Rejects Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B ignores Processor A request for exclusive access	If the Processor B does not want to give go-ahead permission to the Processor A, Processor B ignores the exclusive access request.

35.3.11 Packet Data Transfers

The following example describes the packet transfer sequence between the Processor B and Processor A subsystems:

Table 35-10. Packet Data Transfer Sequence

Action	Sequence	Description
Processor B requests DMA	1	The Processor B sends a DMA request to initiate the packet data transfer

Table continues on the next page...

Table 35-10. Packet Data Transfer Sequence (continued)

Action	Sequence	Description
DMA data transfer	2	DMA acknowledges.
	3	DMA starts transferring data from the specified Processor B location to the specified shared memory
	4	DMA interrupts the Processor B to signal that the packet transfer has finished.
Processor B informs Processor A that data is in shared memory	5	Using an MU Processor B-side transmit register, the Processor B sends a packet information message to the Processor A to inform the Processor A of the arrival of new packet data that is stored in shared memory . The message contains the command, location, and length of packet data information.
Processor A receives interrupt	6	The Processor A receives an interrupt (assuming its corresponding Processor A MU-side receive interrupt is enabled), and the pending processing task becomes active and processes packet data from memory.
Processor A reads data, writes data	7	The Processor A reads or processes packet data from shared memory.
	8	The Processor A writes the result from packet processing to a separate buffer.
Processor A informs Processor B that transfer is finished	9	After the processing of the packet data finishes, the Processor A informs the Processor B (using the MU Processor A-side transmit register, ATRn).
Processor A sends interrupt to Processor B (request for more data)	10	The Processor B receives the next interrupt from the Processor A, in which the Processor A requests more packet data.

35.4 External Signals

There are no Messaging Unit signals directly available at the chip boundary of the SoC.

35.5 Initialization

The MU is used for two processors to communicate by passing messages. Below are examples for passing messages:

1. Interrupt mode
    - Processor A:
        - write ATR0;
        - ACR.TIEn[0] = 1;
        - wait for transmit register empty interrupt;
    - Processor B:
        - BCR.RIEn[0] = 1;
        - wait for receive register full interrupt;
        - read BRR0;

2. Polling mode
    - Processor A:
        - write ATR0;
        - while(ASR.TEn[0] != 1) ;
    - Processor B:
        - while(BSR.RFn[0] != 1) ;
        - read BRR0;

See Application Information for details regarding software restrictions.

35.6 Application Information

This section describes certain software restrictions when accessing the MU.

35.6.1 General Restrictions

This section lists the restrictions that apply to both the sides (Processor A, Processor B) of the MU.

35.6.1.1 Write-After-Write to a Transmit Register

A write to a transmit register signals the receiver side that data is ready for retrieval.

- Writing to the transmit register again without verifying that the data was retrieved is prohibited, because the transmitter side has no way of knowing the exact time that the receiver will attempt to retrieve the data.
- Before attempting to write the transmit register again, the transmitter side should wait for a “Transmitter Empty” interrupt, or should poll the “Transmitter Empty” bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being read on the receiver side of the MU.

35.6.1.2 Read-After-Read from a Receive Register

A read of a receive register signals the transmitter side that data can be written to that register. In the same way, the receiver processor should not read a receive register before receiving a “Receiver Full” interrupt or polling the “Receiver Full” bit in the Status Register.

- Reading the receive register again without verifying that the data was written is prohibited, because the receiver side has no way of knowing the exact time that the transmitter will attempt to write the data.
- Before attempting to read the receive register again, the receiver side should wait for a “Receiver Full” interrupt, or should poll the “Receiver Full” bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being written on the transmitter side of the MU.

35.6.2 Processor Restrictions

This section lists the restrictions that apply each side of the processor in the MU.

35.6.2.1 Before Entering Low Power Mode

Before entering Low Power mode, the Processor should verify that the Processor Event Pending (EP) bit in the Status Register is cleared.

- If the Event Pending bit (EP) is still set to “1”, then the Processor should wait and poll the EP bit until it is cleared, before executing the low power mode instruction.
- Note that if the other Processor is in Low Power mode, the EP bit may be stuck high. In this case, the other Processor clock must be turned ON to get the EP bit cleared before the Processor can enter Low Power mode.

35.6.2.2 Before Setting a General Interrupt Request Bit (GIR0–3)

Before setting a General Interrupt Request bit (GIR0–3), verify the GIRn bit is cleared, which means that a general interrupt is not pending. Generally, setting the GIRn bit while the bit is set to “1” will be ignored, but in some cases it may issue a second interrupt. This restriction is meant to prevent this indeterministic behavior.

35.6.2.3 Reset Bit Restrictions

The reset bit (MUR) restrictions are:

- Before asserting the MUR bit in the ACR register, verify that the Processor B-side is not engaged in some MU activity.
- Do not write to an MU register in the instruction immediately after the assertion of the MUR bit in the ACR register, because the written data can be overridden by the reset value.

35.7 MU Processor A-side Memory Map/Register Definition

35.7.1 MUA register descriptions

This section contains the detailed register descriptions for the Processor A-side MU registers.

35.7.1.1 MUA memory map

MUA base address: 40C4_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Processor A Transmit Register 0 (TR0)	32	RW	0000_0000
4	Processor A Transmit Register 1 (TR1)	32	RW	0000_0000
8	Processor A Transmit Register 2 (TR2)	32	RW	0000_0000
C	Processor A Transmit Register 3 (TR3)	32	RW	0000_0000
10	Processor A Receive Register 0 (RR0)	32	RO	0000_0000
14	Processor A Receive Register 1 (RR1)	32	RO	0000_0000
18	Processor A Receive Register 2 (RR2)	32	RO	0000_0000
1C	Processor A Receive Register 3 (RR3)	32	RO	0000_0000
20	Processor A Status Register (SR)	32	W1C	00F0_0080
24	Processor A Control Register (CR)	32	RW	0000_0000

35.7.1.2 Processor A Transmit Register 0 (TR0)

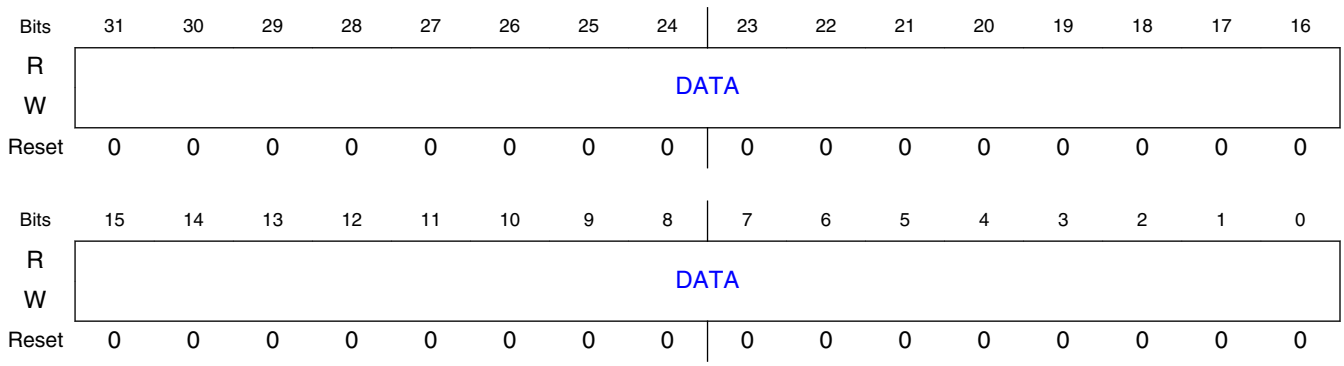
Use Processor A Transmit Register 0 (TR0, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the TR0 register when the TE0 bit in SR register is set to "1".
- Reading the TR0 register returns all zeros.

35.7.1.2.1 Offset

Register	Offset
TR0	0h

35.7.1.2.2 Diagram



35.7.1.2.3 Fields

Field	Description
31-0 DATA	TR0 Processor A Transmit Register 0. (Write-only) • Data written to the MUA.TR0 register is reflected on the Processor B-side in the Processor B Receive Register 0 (MUB.RR0). The MUA.TR0 and MUB.RR0 registers are not double-buffered—a write to the MUA.TR0 register overrides the data readable at the MUB.RR0 register. • A write to the transmit register clears a "transmitter empty" bit (TE0) in the Processor A Status Register (MUA.SR) on the transmitter side, and sets a "receiver full" bit (RF0) in the Processor B Status Register (MUB.SR) on the receiver side (optionally triggering an interrupt 0 on the Processor B-side). • Any write to the TR0 register will update all status information.

35.7.1.3 Processor A Transmit Register 1 (TR1)

Use Processor A Transmit Register 1 (TR1, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the TR1 register when the TE1 bit in SR register is set to "1".
- Reading the TR1 register returns all zeros.

35.7.1.3.1 Offset

Register	Offset
TR1	4h

35.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.3.3 Fields

Field	Description
31-0 DATA	TR1 Processor A Transmit Register 1. (Write-only) • Data written to the MUA.TR1 register is reflected on the Processor B-side in the Processor B Receive Register 1 (MUB.RR1). The TR1 and RR1 registers are not double-buffered—a write to the MUA.TR1 register overrides the data readable at the MUB.RR1 register. • A write to the transmit register clears a "transmitter empty" bit (TE1) in the Processor A Status Register (MUA.SR) on the transmitter side, and sets a "receiver full" bit (RF1) in the Processor B Status Register (MUB.SR) on the receiver side (optionally triggering an interrupt 1 on the Processor B-side). • Any write to the MUA.TR1 register will update all status information.

35.7.1.4 Processor A Transmit Register 2 (TR2)

Use Processor A Transmit Register 2 (TR2, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the TR2 register when the TE2 bit in SR register is set to "1".
- Reading the TR2 register returns all zeros.

35.7.1.4.1 Offset

Register	Offset
TR2	8h

35.7.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.4.3 Fields

Field	Description
31-0 DATA	TR2 Processor A Transmit Register 2. (Write-only) • Data written to the MUA.TR2 register is reflected on the Processor B-side in the Processor B Receive Register 2 (MUB.RR2). The TR2 and RR2 registers are not double-buffered—a write to the MUA.TR2 register overrides the data readable at the MUB.RR2 register. • A write to the transmit register clears a "transmitter empty" bit (TE2) in the Processor A Status Register (MUA.SR) on the transmitter side, and sets a "receiver full" bit (RF2) in the Processor B Status Register (MUB.SR) on the receiver side (optionally triggering an interrupt 2 on the Processor B-side). • Any write to the TR2 register will update all status information.

35.7.1.5 Processor A Transmit Register 3 (TR3)

Use Processor A Transmit Register 3 (TR3, 32-bit, write-only) to transmit a message or data to the Processor B.

- Users can only write to the TR3 register when the TE3 bit in SR register is set to "1".
- Reading the TR3 register returns all zeros.

35.7.1.5.1 Offset

Register	Offset
TR3	Ch

35.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.5.3 Fields

Field	Description
31-0 DATA	TR3 Processor A Transmit Register 3. (Write-only) • Data written to the MUA.TR3 register is reflected on the Processor B-side in the Processor B Receive Register 3 (MUB.RR3). The TR3 and RR3 registers are not double-buffered—a write to the MUA.TR3 register overrides the data readable at the MUB.RR3 register. • A write to the transmit register clears a "transmitter empty" bit (TE3) in the Processor A Status Register (MUA.SR) on the transmitter side, and sets a "receiver full" bit (RF3) in the Processor B Status Register (MUB.SR) on the receiver side (optionally triggering an interrupt 3 on the Processor B-side). • Any write to the TR3 register will update all status information.

35.7.1.6 Processor A Receive Register 0 (RR0)

Use Processor A Receive Register 0 (RR0, 32-bit, read-only) to receive a message or data from the Processor B.

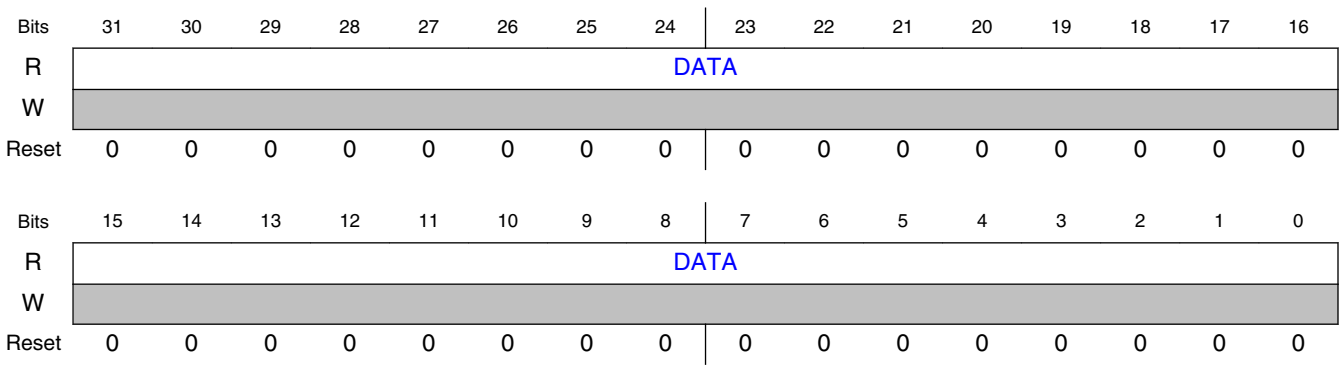
- Data written to the MUB.TR0 register is immediately reflected in the MUA.RR0 register.

- Users can only read the MUA.RR0 register when the RF0 bit in the MUA.SR register is set to "1".
- Writing to the MUA.RR0 register generates an error response to the Processor A.

35.7.1.6.1 Offset

Register	Offset
RR0	10h

35.7.1.6.2 Diagram



35.7.1.6.3 Fields

Field	Description
31-0 DATA	RR0 Processor A Receive Register 0. (Read-only) • Reflects the data written to Processor B Transmit Register 0 (MUB.TR0). • Reading the MUA.RR0 register clears the "receiver full" bit (RF0) in the Processor A Status Register (MUA.SR) on the receiver side, and sets the "transmitter empty" bit (TE0) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor B-side). • Any read of the RR0 register will update all status information.

35.7.1.7 Processor A Receive Register 1 (RR1)

Use Processor A Receive Register 1 (RR1, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the MUB.TR1 register is immediately reflected in the MUA.RR1 register.
- Users can only read the MUA.RR1 register when the RF1 bit in the MUA.SR register is set to "1".
- Writing to the RR1 register generates an error response to the Processor A.

35.7.1.7.1 Offset

Register	Offset
RR1	14h

35.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.7.3 Fields

Field	Description
31-0 DATA	RR1 Processor A Receive Register 1. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (MUB.TR1). • Reading the RR1 register clears the "receiver full" bit (RF1) in the Processor A Status Register (MUA.SR) on the receiver side, and sets the "transmitter empty" bit (TE1) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor B-side). • Any read of the RR1 register will update all status information.

35.7.1.8 Processor A Receive Register 2 (RR2)

Use Processor A Receive Register 2 (RR2, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the MUB.TR2 register is immediately reflected in the MUA.RR2 register.
- Users can only read the MUA.RR2 register when the RF2 bit in the MUA.SR register is set to "1".
- Writing to the RR2 register generates an error response to the Processor A.

35.7.1.8.1 Offset

Register	Offset
RR2	18h

35.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.8.3 Fields

Field	Description
31-0 DATA	RR2 Processor A Receive Register 2. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (MUB.TR2). • Reading the MUA.RR2 register clears the "receiver full" bit (RF2) in the Processor A Status Register (MUA.SR) on the receiver side, and sets the "transmitter empty" bit (TE2) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor B-side). • Any read of the RR2 register will update all status information.

35.7.1.9 Processor A Receive Register 3 (RR3)

Use Processor A Receive Register 3 (RR3, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the MUB.TR3 register is immediately reflected in the MUA.RR3 register.
- Users can only read the MUA.RR3 register when the RF3 bit in the MUA.SR register is set to "1".
- Writing to the RR3 register generates an error response to the Processor A.

35.7.1.9.1 Offset

Register	Offset
RR3	1Ch

35.7.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.7.1.9.3 Fields

Field	Description
31-0 DATA	RR3 Processor A Receive Register 3. (Read-only) • Reflects the data written to Processor B Transmit Register 3 (MUB.TR3). • Reading the MUA.RR3 register clears the "receiver full" bit (RF3) in the Processor A Status Register (MUA.SR) on the receiver side, and sets the "transmitter empty" bit (TE3) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor B-side). • Any read of the RR3 register will update all status information.

35.7.1.10 Processor A Status Register (SR)

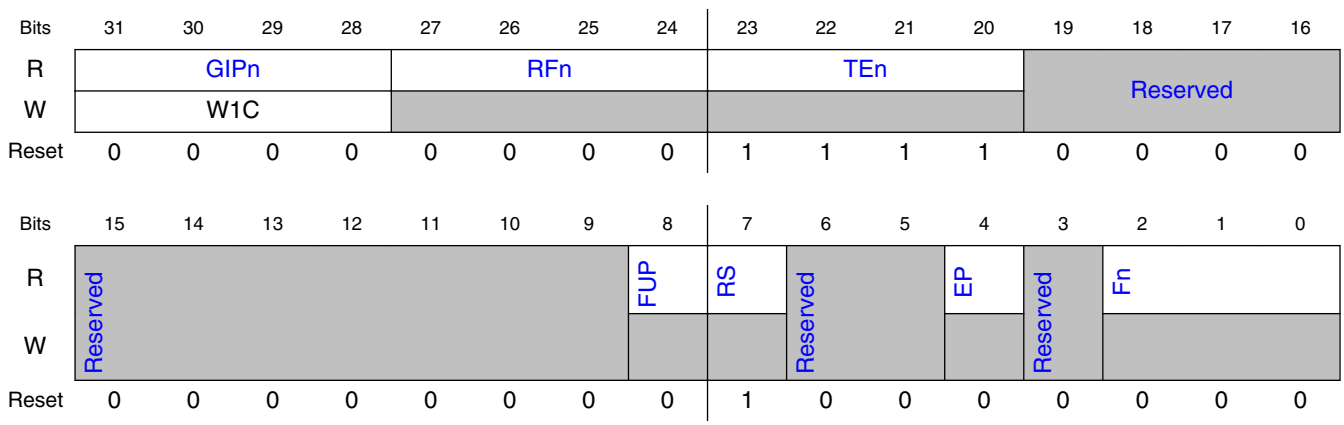
Use the Processor A Status Register (SR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, and to set dual function control-status bits.

- Some dual-purpose bits are set by the MU logic, and cleared by the Processor A-side programmer
- Other dual-purpose bits are set by the Processor A-side programmer, and cleared by the MU logic.

35.7.1.10.1 Offset

Register	Offset
SR	20h

35.7.1.10.2 Diagram



35.7.1.10.3 Fields

Field	Description
31-28 GIPn	GIPn For n = {0, 1, 2, 3} Processor A General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor A that the GIRn bit in the MUB.CR register on the Processor B-side was set from "0" to "1". If the GIEn bit in the MUA.CR register is set to "1", a General Interrupt n request is issued.

Table continues on the next page...

Field	Description
	• The GIPn bit is cleared by writing it back as "1". Writing "0", or writing "1" when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. The proper bit clearing sequence is: clear an Processor A register, set the desired bit in it (Processor A register), and write it to the MUA.SR register, thus clearing the GIPn bit. • GIPn bit is cleared when the MU is reset. 0000 - Processor A general purpose interrupt n is not pending. (default) 0001 - Processor A general purpose interrupt n is pending.
27-24 RFn	RFn For n = {0, 1, 2, 3} Processor A Receive Register n Full. (Read-only) • The RFn bit is set to "1" when the MUB.TRn register is written on the Processor B-side. • After the RFn bit is set to "1", the RFn bit signals the Processor A-side that new data is ready to be read by the Processor A in the MUA.RRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the CR register has been set to "1"). • RFn bit is cleared when the MUA.RRn register is read, and when the MU is reset. 0000 - MUA.RRn register is not full (default). 0001 - MUA.RRn register has received data from MUB.TRn register and is ready to be read by the Processor A.
23-20 TEn	TEn For n = {0, 1, 2, 3} Processor A Transmit Register n Empty. (Read-only) • The TEn bit is set to "1" after the BRRn register is read on the Processor B-side. • After the TEn bit is set to "1", the TEn bit signals the Processor A-side that the MUA.TRn register is ready to be written on the Processor A-side, and a Transmit n interrupt is issued on the Processor A-side (if the TEn bit in the MUA.CR register is set to "1"). • TEn bit is cleared after the MUA.TRn register is written on the Processor A-side. • TEn bit is set to "1" when the MU is reset. 0000 - MUA.TRn register is not empty. 0001 - MUA.TRn register is empty (default).
19-9 —	- Reserved
8 FUP	FUP Processor A Flags Update Pending. (Read-only) • FUP bit is set to "1" when the Processor A-side sends a Flags Update request to the Processor B-side. • A Flags Update request is generated when the ABF[2:0] bits of the CR register change. No flag update changes are allowed while the FUP bit is set to "1". Any write to the ABF[2:0] bits, while the FUP bit is set to "1", will not generate a Flags Update event, and the ABF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor B-side, and during MU reset. 0 - No flags updated, initiated by the Processor A, in progress (default) 1 - Processor A initiated flags update, processing
7 RS	RS Processor B-side Reset State. (Read-only) • RS bit indicates if the Processor B-side of the MU is in a reset state or not. • If the RS bit is set to "1", then the Processor B-side of the MU is still in the reset state. • If the RS bit is cleared, then the Processor B-side of the MU are out of reset.

Table continues on the next page...

Field	Description
	• The RS bit is set to "1" during: a Processor B system reset, or an MU reset (caused by setting the MUR bit at the MUA.CR register). • The RS bit is cleared when the reset sequence on the Processor B-side of the MU ends. After issuing any of the reset events mentioned previously, verify that the RS bit is cleared before starting any accesses. • When Processor A side of MU comes out of reset RS bit has value "1"(default).Then Processor A sees the status of Processor B-side and if Processor B-side has come out of reset then RS bit goes low. This takes 5-6 clock cycles. If the RS bit is read now it shows as low although its reset value was "1" . 0 - The Processor B-side of the MU is not in reset. 1 - The Processor B-side of the MU is in reset.
6-5 —	- Reserved
4 EP	EP Processor A-Side Event Pending. (Read-only) • EP bit is set to "1" when the Processor A-side mechanism sends an event update request to the Processor B-side. • EP bit is cleared when the event update acknowledge is received. An "event" is any hardware message that is reflected in the MUB.SR register on the Processor B-side (for example, "transmit register 0 written"). During normal operations, users do not have to deal with the state of the EP bit because the event update mechanism works automatically. • To ensure events have been posted to Processor B before entering STOP mode, verify that the EP bit is cleared. If EP bit is set to "1", wait and continue to poll it (EP bit) before entering STOP mode. • Reading the SR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOP or WAIT modes; otherwise, the EP bit may be set by subsequent additional actions. • The EP bit is cleared when the MU resets. 0 - The Processor A-side event is not pending (default). 1 - The Processor A-side event is pending.
3 —	- Reserved
2-0 Fn	Fn For n = {0, 1, 2} Processor A-Side Flag n. (Read-only) • Fn bit is the Processor A-side flag that reflects the values written to the BAFn bit in the Processor B control register. • Every time that the BAFn bit is written, the BAFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor B and the Processor A. 000 - BAFn bit in MUB.CR register is written 0 (default). 001 - BAFn bit in MUB.CR register is written 1.

35.7.1.11 Processor A Control Register (CR)

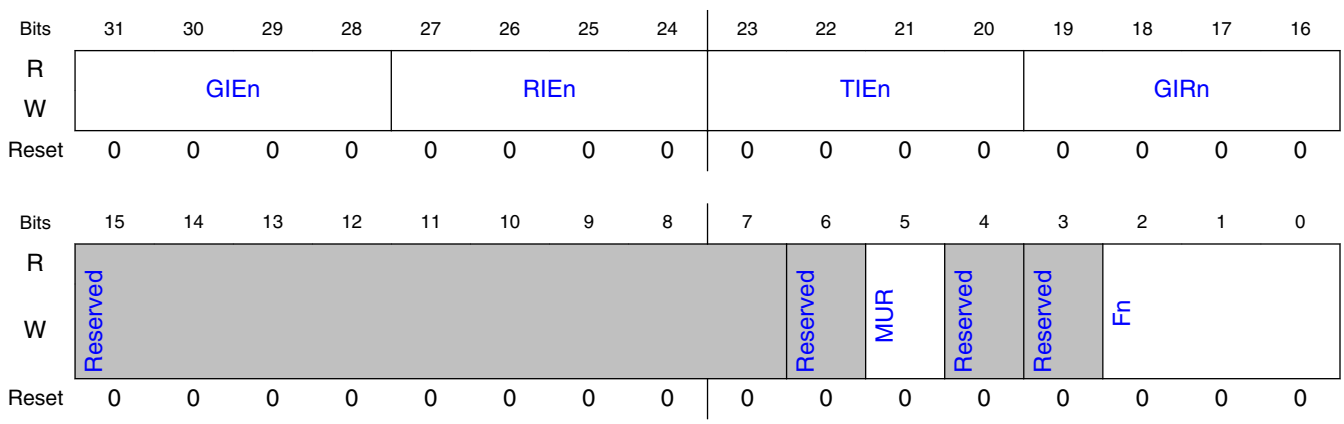
Use the Processor A Control Register (CR, 32-bit, read-write) to enable the MU interrupts on the Processor A-side, and trigger events and interrupts on the Processor B-side (general purpose interrupt, flag update).

For the fields GIEn, RIEn, TIEn and GIRn, n=0 corresponds to the high order bit and n=3 corresponds to the low order bit.

35.7.1.11.1 Offset

Register	Offset
CR	24h

35.7.1.11.2 Diagram



35.7.1.11.3 Fields

Field	Description
31-28 GIEn	GIEn For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Enable n. (Read-Write) When GIEn=0 corresponds to the high order bit and GIE3 corresponds to the low order bit. • GIEn bit enables Processor A General Interrupt n. • If GIEn bit is set to "1" (enabled), then a General Interrupt n request is issued when the GIPn bit in the MUA.SR register is set to "1". • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0000 - Disables Processor A General Interrupt n. (default) 0001 - Enables Processor A General Interrupt n.
27-24	RIEn

Table continues on the next page...

Field	Description
RIEn	For n = {0, 1, 2, 3} Processor A Receive Interrupt Enable n. (Read-Write) • RIEn bit enables Processor A Receive Interrupt n. • If RIEn bit is set to "1" (enabled), then an Processor A Receive Interrupt n request is issued when the RFn bit in the MUA.SR register is set to "1". • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor A Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0000 - Disables Processor A Receive Interrupt n. (default) 0001 - Enables Processor A Receive Interrupt n.
23-20 TIEn	TIEn For n = {0, 1, 2, 3} Processor A Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor A Transmit Interrupt n. • If TIEn bit is set to "1" (enabled), then an Processor A Transmit Interrupt n request is issued when the TEn bit in the MUA.SR register is set to "1". • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor A Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0000 - Disables Processor A Transmit Interrupt n. (default) 0001 - Enables Processor A Transmit Interrupt n.
19-16 GIRn	GIRn For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Request n. (Read-Write) • Writing "1" to the GIRn bit sets the GIPn bit in the MUB.SR register on the Processor B-side. If the GIEn bit in the MUB.CR register is set to "1" on the Processor B-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the MUB.SR register on the Processor B-side) is cleared by writing it (GIPn bit) as "1", thereby signalling the Processor A that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as "0" on the Processor A-side. • To ensure proper operations, verify that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0000 - Processor A General Interrupt n is not requested to the Processor B (default). 0001 - Processor A General Interrupt n is requested to the Processor B.
15-7 —	- Reserved
6 —	- Reserved
5 MUR	MUR Processor A MU Reset. • Setting MUR bit to "1" resets both the Processor B and the Processor A sides of the MU module, forcing all control and status registers to return to their default values (except the HR bit in the MUA.CR register and HRM bit in MUB.CR register), and all internal states to be cleared. • Before setting the MUR bit to "1", it is advisable to interrupt the Processor B, because setting the MUR bit may affect the ongoing Processor B program. • After setting the MUR bit, monitor the value of the RS bit in the MUA.SR register to know when the reset sequence on the Processor B-side has ended. • MUR bit can only be written as "1". • MUR bit is always read as "0". • MUR bit is cleared during the MU reset sequence.

Table continues on the next page...

Field	Description
	0 - N/A. Self clearing bit (default). 1 - Asserts the Processor A MU reset.
4 —	- Reserved
3 —	- Reserved
2-0 Fn	Fn For n = {0, 1, 2} Processor A to Processor B Flag n. (Read-Write) • Fn bit is a read-write flag that is reflected in Fn bit in the MUB.SR register on the Processor B-side. • Fn bit is cleared when the MU resets. 000 - N/A. Self clearing bit (default). 001 - Asserts the Processor A MU reset.

35.8 MU Processor B-side Memory Map/Register Definition

35.8.1 MUB register descriptions

This section contains the detailed register descriptions for the Processor B-side MU registers.

35.8.1.1 MUB memory map

MUB base address: 40C4_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Processor B Transmit Register 0 (TR0)	32	RW	0000_0000
4	Processor B Transmit Register 1 (TR1)	32	RW	0000_0000
8	Processor B Transmit Register 2 (TR2)	32	RW	0000_0000
C	Processor B Transmit Register 3 (TR3)	32	RW	0000_0000
10	Processor B Receive Register 0 (RR0)	32	RO	0000_0000
14	Processor B Receive Register 1 (RR1)	32	RO	0000_0000
18	Processor B Receive Register 2 (RR2)	32	RO	0000_0000
1C	Processor B Receive Register 3 (RR3)	32	RO	0000_0000
20	Processor B Status Register (SR)	32	W1C	00F0_0080

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
24	Processor B Control Register (CR)	32	RW	0000_0000

35.8.1.2 Processor B Transmit Register 0 (TR0)

Use Processor B Transmit Register 0 (TR0, 32-bit, write-only) to transmit a message or data to the Processor A.

- The TR0 register can only be written when the TE0 bit in SR register is set to "1".
- Reading the TR0 register returns all zeros.

35.8.1.2.1 Offset

Register	Offset
TR0	0h

35.8.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.8.1.2.3 Fields

Field	Description
31-0 DATA	TR0 Processor B Transmit Register 0. (Write-only) • Data written to the MUB.TR0 register is reflected on the Processor A-side in the Processor A Receive Register 0 (MUA.RR0). The MUB.TR0 and MUA.RR0 registers are not double-buffered—a write to the MUB.TR0 register overrides the data readable at the MUA.RR0 register.

Field	Description
	• A write to the transmit register clears a "transmitter empty" bit (TE0) in the Processor B Status Register (MUB.SR) on the transmitter side, and sets a "receiver full" bit (RF0) in the Processor A Status Register (MUA.SR) on the receiver side (optionally triggering an interrupt 0 on the Processor A-side). • Any write to the TR0 register will update all status information.

35.8.1.3 Processor B Transmit Register 1 (TR1)

Use Processor B Transmit Register 1 (TR1, 32-bit, write-only) to transmit a message or data to the Processor A.

- The TR1 register can only be written when the TE1 bit in SR register is set to "1".
- Reading the TR1 register returns all zeros.

35.8.1.3.1 Offset

Register	Offset
TR1	4h

35.8.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.8.1.3.3 Fields

Field	Description
31-0 DATA	TR1 Processor B Transmit Register 1. (Write-only)

Field	Description
	• Data written to the MUB.TR1 register is reflected on the Processor A-side in the Processor A Receive Register 1 (MUA.RR1). The MUB.TR1 and MUA.RR1 registers are not double-buffered—a write to the MUB.TR1 register overrides the data readable at the MUA.RR1 register. • A write to the transmit register clears a "transmitter empty" bit (TE1) in the Processor B Status Register (MUB.SR) on the transmitter side, and sets a "receiver full" bit (RF1) in the Processor A Status Register (MUA.SR) on the receiver side (optionally triggering an interrupt 1 on the Processor A-side). • Any write to the TR1 register will update all status information.

35.8.1.4 Processor B Transmit Register 2 (TR2)

Use Processor B Transmit Register 2 (TR2, 32-bit, write-only) to transmit a message or data to the Processor A.

- The TR2 register can only be written when the TE2 bit in SR register is set to "1".
- Reading the TR2 register returns all zeros.

35.8.1.4.1 Offset

Register	Offset
TR2	8h

35.8.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.8.1.4.3 Fields

Field	Description
31-0	TR2

Field	Description
DATA	Processor B Transmit Register 2. (Write-only) • Data written to the MUB.TR2 register is reflected on the Processor A-side in the Processor A Receive Register 2 (MUA.RR2). The MUB.TR2 and MUA.RR2 registers are not double-buffered—a write to the MUB.TR2 register overrides the data readable at the MUA.RR2 register. • A write to the transmit register clears a "transmitter empty" bit (TE2) in the Processor B Status Register (MUB.SR) on the transmitter side, and sets a "receiver full" bit (RF2) in the Processor A Status Register (MUA.SR) on the receiver side (optionally triggering an interrupt 2 on the Processor A-side). • Any write to the TR2 register will update all status information.

35.8.1.5 Processor B Transmit Register 3 (TR3)

Use Processor B Transmit Register 3 (TR3, 32-bit, write-only) to transmit a message or data to the Processor A.

- The TR3 register can only be written when the TE3 bit in SR register is set to "1".
- Reading the TR3 register returns all zeros.

35.8.1.5.1 Offset

Register	Offset
TR3	Ch

35.8.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.8.1.5.3 Fields

Field	Description
31-0	TR3

Field	Description
DATA	Processor B Transmit Register 3. (Write-only) • Data written to the TR3 register is reflected on the Processor A-side in the Processor A Receive Register 3 (MUA.RR3). The MUB.TR3 and MUA.RR3 registers are not double-buffered—a write to the TR3 register overrides the data readable at the MUA.RR3 register. • A write to the transmit register clears a "transmitter empty" bit (TE3) in the Processor B Status Register (MUB.SR) on the transmitter side, and sets a "receiver full" bit (RF3) in the Processor A Status Register (MUA.SR) on the receiver side (optionally triggering an interrupt 3 on the Processor A-side). • Any write to the TR3 register will update all status information.

35.8.1.6 Processor B Receive Register 0 (RR0)

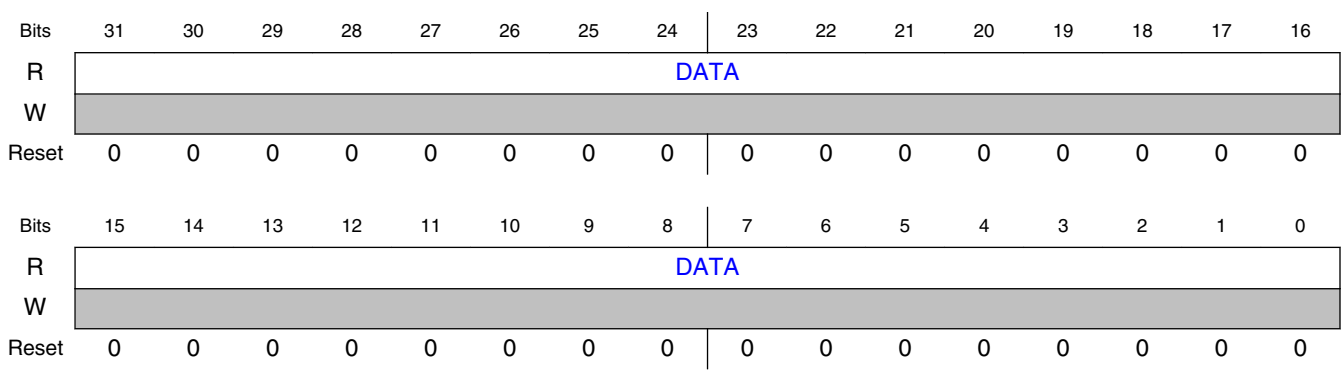
Use Processor B Receive Register 0 (RR0, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the MUA.TR0 register is immediately reflected in the MUB.RR0 register.
- The MUB.RR0 register can only be read when the RF0 bit in the MUB.SR register is set to "1".
- Writing to the MUB.RR0 register generates an error response to the Processor B.

35.8.1.6.1 Offset

Register	Offset
RR0	10h

35.8.1.6.2 Diagram



35.8.1.6.3 Fields

Field	Description
31-0 DATA	RR0 Processor B Receive Register 0. (Read-only) • Reflects the data written to Processor A Transmit Register 0 (MUA.TR0). • Reading the MUB.RR0 register clears the "receiver full" bit (RF0) in the Processor B Status Register (MUB.SR) on the receiver side, and sets the "transmitter empty" bit (TE0) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor A-side). • Any read of the RR0 register will update all status information.

35.8.1.7 Processor B Receive Register 1 (RR1)

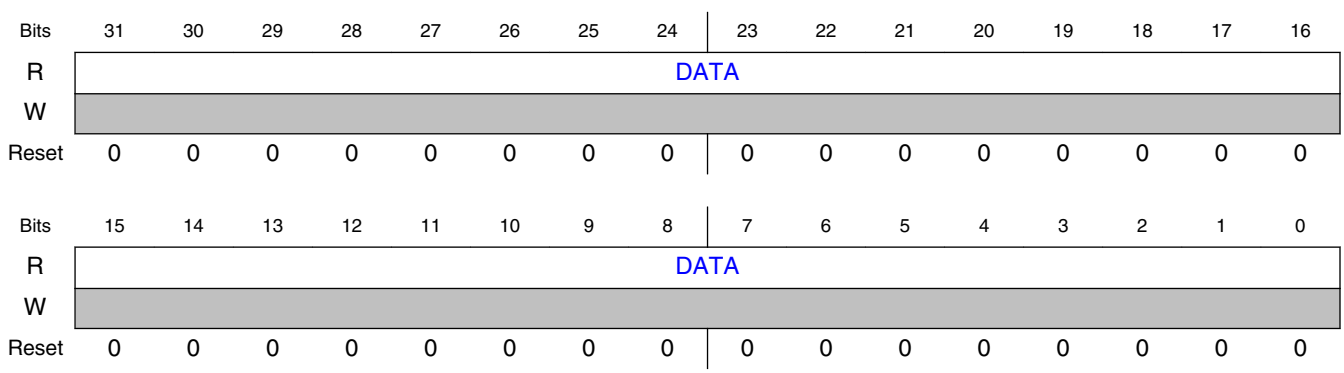
Use Processor B Receive Register 1 (RR1, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the MUA.TR1 register is immediately reflected in the MUB.RR1 register.
- The MUB.RR1 register can only be read when the RF1 bit in the MUB.SR register is set to "1".
- Writing to the MUB.RR1 register generates an error response to the Processor B.

35.8.1.7.1 Offset

Register	Offset
RR1	14h

35.8.1.7.2 Diagram



35.8.1.7.3 Fields

Field	Description
31-0 DATA	RR1 Processor B Receive Register 1. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (MUA.TR1). • Reading the MUB.RR1 register clears the "receiver full" bit (RF1) in the Processor B Status Register (MUB.SR) on the receiver side, and sets the "transmitter empty" bit (TE1) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor A-side). • Any read of the RR1 register will update all status information.

35.8.1.8 Processor B Receive Register 2 (RR2)

Use Processor B Receive Register 2 (RR2, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the MUA.TR2 register is immediately reflected in the MUB.RR2 register.
- The MUB.RR2 register can only be read when the RF2 bit in the MUB.SR register is set to "1".
- Writing to the MUB.RR2 register generates an error response to the Processor B.

35.8.1.8.1 Offset

Register	Offset
RR2	18h

35.8.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35.8.1.8.3 Fields

Field	Description
31-0 DATA	RR2 Processor B Receive Register 2. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (MUA.TR2). • Reading the MUB.RR2 register clears the "receiver full" bit (RF2) in the Processor B Status Register (MUB.SR) on the receiver side, and sets the "transmitter empty" bit (TE2) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor A-side). • Any read of the RR2 register will update all status information.

35.8.1.9 Processor B Receive Register 3 (RR3)

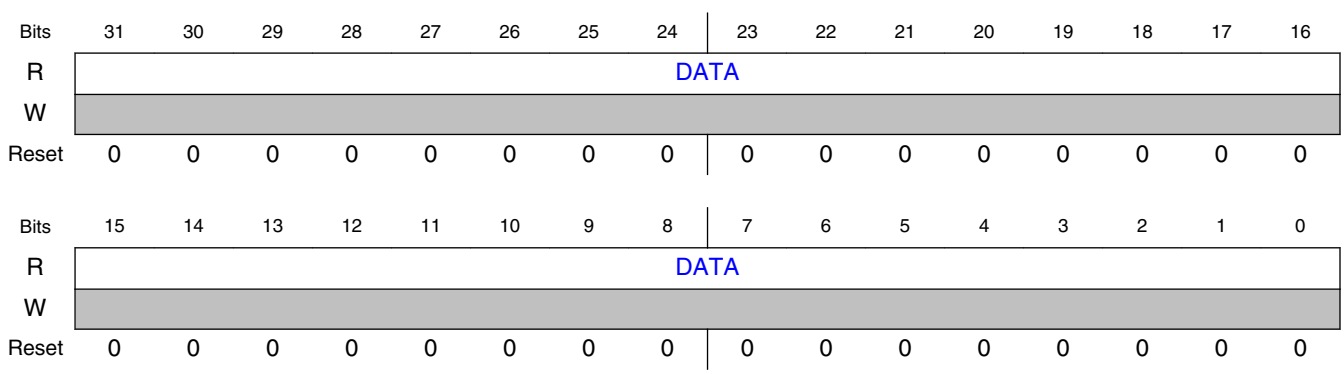
Use Processor B Receive Register 3 (RR3, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the MUA.TR3 register is immediately reflected in the MUB.RR3 register.
- The MUB.RR3 register can only be read when the RF3 bit in the MUB.SR register is set to "1".
- Writing to the MUB.RR3 register generates an error response to the Processor B.

35.8.1.9.1 Offset

Register	Offset
RR3	1Ch

35.8.1.9.2 Diagram



35.8.1.9.3 Fields

Field	Description
31-0 DATA	RR3 Processor B Receive Register 3. (Read-only) • Reflects the data written to Processor A Transmit Register 3 (MUA.TR3). • Reading the MUB.RR3 register clears the "receiver full" bit (RF3) in the Processor B Status Register (MUB.SR) on the receiver side, and sets the "transmitter empty" bit (TE3) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor A-side). • Any read of the RR3 register will update all status information.

35.8.1.10 Processor B Status Register (SR)

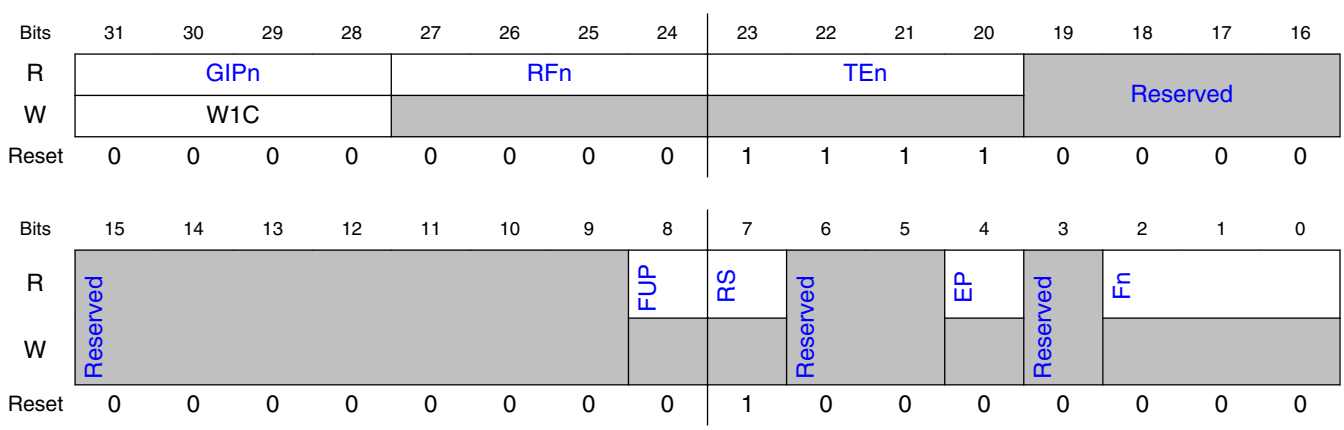
Use the Processor B Status Register (SR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, and to set dual function control-status bits.

- Dual-purpose bits are set by the Processor B-side programmer, and cleared by the MU logic.

35.8.1.10.1 Offset

Register	Offset
SR	20h

35.8.1.10.2 Diagram



35.8.1.10.3 Fields

Field	Description
31-28 GIPn	GIPn For n = {0, 1, 2, 3} Processor B General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor B that the GIRn bit in the MUA.CR register on the Processor A-side was set from "0" to "1". If the GIEn bit in the MUB.CR register is set to "1", a General Interrupt n request is issued. • The GIPn bit is cleared by writing it back as "1". Writing "0", or writing "1" when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. • GIPn bit is cleared when the MU is reset. 0000 - Processor B general purpose interrupt n is not pending. (default) 0001 - Processor B general purpose interrupt n is pending.
27-24 RFn	RFn For n = {0, 1, 2, 3} Processor B Receive Register n Full. (Read-only) • RFn bit signals to the Processor B-side that new data was written by the Processor A to the MUA.TRn register, and is ready to be read by the Processor B in the MUB.RRn register. • The RFn bit is set to "1" when the MUA.TRn register is written on the Processor A-side. • After the RFn bit is set to "1", the RFn bit signals the Processor B-side that new data is ready to be read by the Processor B in the MUB.RRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the MUB.CR register has been set to "1"). • RFn bit is cleared when the MUB.RRn register is read, and when the MU is reset. 0000 - MUB.RRn register is not full (default). 0001 - MUB.RRn register has received data from MUA.TRn register and is ready to be read by the Processor B.
23-20 TEn	TEn For n = {0, 1, 2, 3} Processor B Transmit Register n Empty. (Read-only) • When TEn = "1", it signals to the Processor B-side that the MUB.TRn register is ready to be written on the Processor B-side. • The TEn bit is set to "1" after the MUA.RRn register is read on the Processor A-side. • Setting TEn bit will issue a transmit n interrupt on the Processor B-side (if the TIEn bit in the MUB.CR register is set to "1". • TEn bit is cleared after the MUB.TRn register is written on the Processor B-side. • TEn bit is set to "1" when the MU is reset. 0000 - MUB.TRn register is not empty. 0001 - MUB.TRn register is empty (default).
19-9 —	Reserved
8 FUP	FUP Processor B Flags Update Pending. (Read-only) • FUP bit is set to "1" when the Processor B-side sends a Flags Update request to the Processor A-side. • A Flags Update request is generated when the BAF[2:0] bits of the MUB.CR register change. No flag update changes are allowed while the FUP bit is set to "1". Any write to the BAF[2:0] bits, while the FUP bit is set to "1", will not generate a Flags Update event, and the BAF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor A-side, and during MU reset.

Table continues on the next page...

Field	Description
	0 - No flags updated, initiated by the Processor B, in progress (default) 1 - Processor B initiated flags update, processing
7 RS	RS Processor A Reset State. (Read-only) • RS bit indicates if the Processor A-side of the MU is in a reset state or not. • If the RS bit is set to "1", then the Processor A-side of the MU is still in the reset state. • If the RS bit is cleared, then both the Processor A and the Processor A-side of the MU are out of reset. • The RS bit is set to "1" during: a Processor A system reset, or an MU reset (caused by setting the MUR bit at the MUB.CR register). • The RS bit is cleared when the reset sequence on the Processor A-side of the MU ends. After issuing any of the three reset events mentioned previously, it should be verified that the RS bit is cleared before starting any accesses. • When Processor B side of MU comes out of reset RS bit has value "1"(default).Then Processor B sees the status of Processor A side and if Processor A has come out of reset then RS bit goes low. This takes 5-6 clock cycles. If the RS bit is read now it is low although its reset value was "1" . 0 - The Processor A or the Processor A-side of the MU is not in reset. 1 - The Processor A or the Processor A-side of the MU is in reset.
6-5 —	- Reserved
4 EP	EP Processor B-Side Event Pending. (Read-only) • EP bit is set to "1" when the Processor B-side mechanism sends an event update request to the Processor A-side. • EP bit is cleared when the event update acknowledge is received. An "event" is any hardware message that is reflected in the MUA.SR register on the Processor A-side (for example, "transmit register 0 written"). During normal operations, the state of the EP bit does not need to be updated manually because the event update mechanism works automatically. • To ensure events have been posted to Processor A before entering STOP mode, it should be verified that the EP bit is cleared. If EP bit is set to "1", continue to poll it (EP bit) before entering STOP mode. • Reading the SR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOP mode; otherwise, the EP bit may be set by subsequent additional actions. • Due to Processor B pipeline effects, three NOP operations (or their timing equivalent) should be given after an instruction that sets an event before the EP bit can reflect this event. • The EP bit is cleared when the MU resets. 0 - The Processor B-side event is not pending (default). 1 - The Processor B-side event is pending.
3 —	Reserved
2-0 Fn	Fn For n = {0, 1, 2} Processor B-Side Flag n. (Read-only) • Fn bit is the Processor B-side flag that reflects the values written to the ABFn bit in the Processor A control register. • Every time that the ABFn bit is written, the ABFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor A and the Processor B.

Field	Description
	000 - ABFn bit in MUA.CR register is written 0 (default). 001 - ABFn bit in MUA.CR register is written 1.

35.8.1.11 Processor B Control Register (CR)

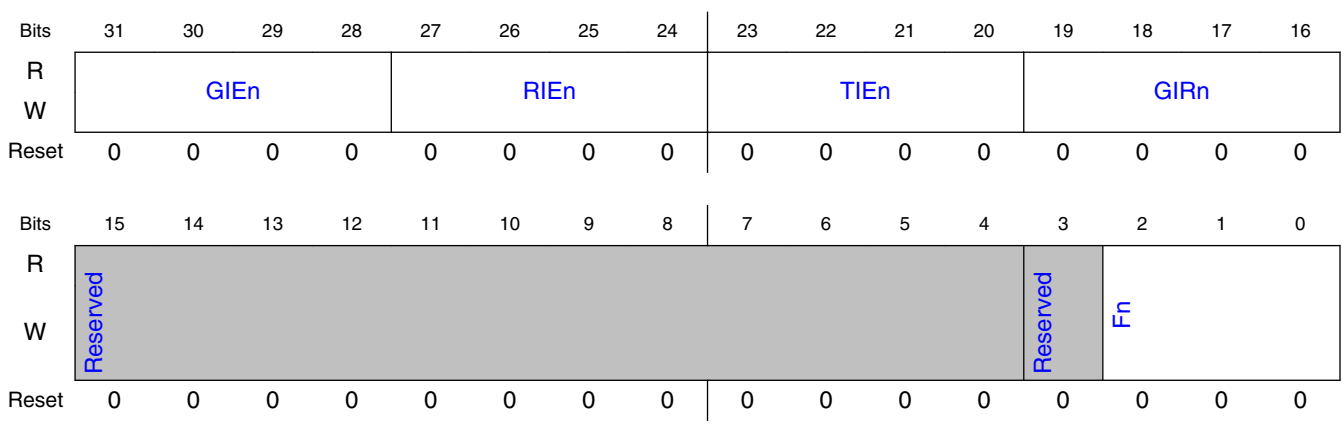
Use the Processor B Control Register (CR, 32-bit, read-write) to enable the MU interrupts on the Processor B-side, and trigger events and interrupts on the Processor A-side (wake from STOP, hardware reset, flag update).

For the fields GIEn, RIEn, TIEn and GIRn, n=0 corresponds to the high order bit and n=3 corresponds to the low order bit.

35.8.1.11.1 Offset

Register	Offset
CR	24h

35.8.1.11.2 Diagram



35.8.1.11.3 Fields

Field	Description
31-28 GIEn	GIEn For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Enable n. (Read-Write)

Table continues on the next page...

Field	Description
	• GIEn bit enables Processor B General Interrupt n. • If GIEn bit is set to "1" (enabled), then a General Interrupt n request is issued when the GIPn bit in the MUB.SR register is set to "1". • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0000 - Disables Processor B General Interrupt n. (default) 0001 - Enables Processor B General Interrupt n.
27-24 RIEn	RIEn For n = {0, 1, 2, 3} Processor B Receive Interrupt Enable n. (Read-Write) • RIEn bit enables Processor B Receive Interrupt n. • If RIEn bit is set to "1" (enabled), then an Processor B Receive Interrupt n request is issued when the RFn bit in the MUB.SR register is set to "1". • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor B Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0000 - Disables Processor B Receive Interrupt n. (default) 0001 - Enables Processor B Receive Interrupt n.
23-20 TIEn	TIEn For n = {0, 1, 2, 3} Processor B Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor B Transmit Interrupt n. • If TIEn bit is set to "1" (enabled), then an Processor B Transmit Interrupt n request is issued when the TEn bit in the MUB.SR register is set to "1". • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor B Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0000 - Disables Processor B Transmit Interrupt n. (default) 0001 - Enables Processor B Transmit Interrupt n.
19-16 GIRn	GIRn For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Request n. (Read-Write) • Writing "1" to the GIRn bit sets the GIPn bit in the MUA.SR register on the Processor A-side. If the GIEn bit in the ACR register is set to "1" on the Processor A-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the MUA.SR register on the Processor A-side) is cleared by writing it (GIPn bit) as "1", thereby signalling the Processor B that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as "0" on the Processor B-side. • To ensure proper operations, it must be verified that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0000 - Processor B General Interrupt n is not requested to the Processor A (default). 0001 - Processor B General Interrupt n is requested to the Processor A.
15-4 —	Reserved
3 —	Reserved
2-0	Fn

Field	Description
Fn	For n = {0, 1, 2} Processor B to Processor A Flag n. (Read-Write) • Fn bit is a read-write flag that is reflected in Fn bit in the MUA.SR register on the Processor A-side. • Fn bit is cleared when the MU resets. 000 - Clears the Fn bit in the MUA.SR register. 001 - Sets the Fn bit in the MUA.SR register.

Chapter 36
Network Interconnect Bus System (NIC-301)

36.1 Chip-specific NIC-301 information

Table 36-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

36.2 Overview

This section provides an overview of the

- NIC-301 (Network Inter-Connect) AXI arbiter IP

  The NIC-301 (by Arm Ltd.) is a configurable AXI arbiter between several masters and slaves. The NIC-301 IP is designed so that many configuration options are selected at the hardware design stage, determined by SoC characteristics and needs, while several other configuration options are software-controlled.

This chapter covers in brief the NIC-301 while providing configuration details on the NIC-301 instances used in the chip. For complete details on the NIC-301 design, see the Arm specification, *AMBA® Network Interconnect (NIC-301) Technical Reference Manual, version r2p3*.

NOTE

The NIC-301 default settings, in most cases, should not be modified by the customer. The default settings have gone through exhaustive testing during the validation of the part, and have proven to work well for the part's intended target applications. Changes to the default settings may result in a degradation in system performance.

36.2.1 NIC-301 Main Features

Key features of the NIC-301 module include the following:

- Address space memory mapping.
- Programmer's view, for software-configured parameters, via internal "GPV" ports.
- Support for cross-clock domain synchronization.

36.2.2 Modes and Operations

The NIC-301 supports a normal functional mode only.

36.3 External Signals

There are no external I/O interfaces for NIC-301.

36.4 Memory Map and Register Definition

The bus system is composed of the several instances that have GPV registers, which are helpful for bus arbitration and performance. For detailed descriptions of these registers, see the Arm document: *DDI0397I_corelink_network_interconnect_nic301_r2p3_trm.pdf*.

1. **SIM_DISP registers**

   The SIM_DISP GPV base address is GPV0_BASE = 0x41000000. The following registers are implemented in this NIC.

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
read_qos	m_a_0	GC355	GPV0_BASE + 0x42000 + 0x100	6	Set the priority of master's read. The priority level would be used when the master's read transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
read_qos	m_a_1	PXP	GPV0_BASE + 0x45000 + 0x100	6	Same as above
read_qos	m_a_2	LCDIF	GPV0_BASE + 0x46000 + 0x100	12	Same as above
read_qos	m_a_3	LCDIFv2	GPV0_BASE + 0x47000 + 0x100	12	Same as above
read_qos	m_a_4	CSI	GPV0_BASE + 0x48000 + 0x100	14	Same as above
write_qos	m_a_0	GC355	GPV0_BASE + 0x42000 + 0x104	6	Set the priority of master's write. The priority level would be used when the master's write transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
write_qos	m_a_1	PXP	GPV0_BASE + 0x45000 + 0x104	6	Same as above
write_qos	m_a_2	LCDIF	GPV0_BASE + 0x46000 + 0x104	12	Same as above
write_qos	m_a_3	LCDIFv2	GPV0_BASE + 0x47000 + 0x104	12	Same as above
write_qos	m_a_4	CSI	GPV0_BASE + 0x48000 + 0x104	14	Same as above
fn_mod	m_a_0	GC355	GPV0_BASE + 0x42000 + 0x108	0	Issuing functionality modification register. This register sets the block issuing capability to one outstanding transaction. Legal programmable values are 0, 1, 2 and 3. NOTE: It is recommended to keep its reset value to obtain the best performance. 0: Default setting 1: Set read issuing capability to one outstanding transaction 2: Set write issuing capability to one outstanding transaction 3: Set both read and write issuing capability to one outstanding transaction
fn_mod	m_a_1	PXP	GPV0_BASE + 0x45000 + 0x108	0	Same as above
fn_mod	m_a_2	LCDIF	GPV0_BASE + 0x46000 + 0x108	0	Same as above

Table continues on the next page...

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
fn_mod	m_a_3	LCDIFv2	GPV0_BASE + 0x47000 + 0x108	0	Same as above
fn_mod	m_a_4	CSI	GPV0_BASE + 0x48000 + 0x108	0	Same as above

2. SIM_M registers

The SIM_M GPV base address is GPV1_BASE = 0x41100000. The following registers are implemented in this NIC.

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
fn_mod_ahb	m_c_3	ENET	GPV1_BASE + 0x45000 + 0x028	0	This register has 3 control bits. Bit 0: rd_incr_override. Writting 1 to this bit forces NIC to convert all AHB read transactions to a series of AXI singles. Bit 1: wr_incr_override. Writting 1 to this bit forces NIC to convert all AHB write transactions to a series of AXI singles. Bit 2: lock_override. Writting 1 to this bit forces NIC not to create any AXI lock transactions. NOTE: It is recommended to keep its reset value to obtain the best performance.
read_qos	m_c_0	CAAM	GPV1_BASE + 0x42000 + 0x100	6	Set the priority of master's read. The priority level would be used when the master's read transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
read_qos	m_c_1	ENET1GRX	GPV1_BASE + 0x43000 + 0x100	6	Same as above
read_qos	m_c_2	ENET1GTX	GPV1_BASE + 0x44000 + 0x100	6	Same as above
read_qos	m_c_3	ENET	GPV1_BASE + 0x45000 + 0x100	6	Same as above
read_qos	m_c_4	USBO2	GPV1_BASE + 0x46000 + 0x100	12	Same as above
read_qos	m_c_5	USDHC1	GPV1_BASE + 0x47000 + 0x100	12	Same as above
read_qos	m_c_6	USDHC2	GPV1_BASE + 0x48000 + 0x100	14	Same as above
read_qos	m_c_8	ENET_QOS	GPV1_BASE + 0x4A000 + 0x100	0	Same as above
write_qos	m_c_0	CAAM	GPV1_BASE + 0x42000 + 0x104	6	Set the priority of master's write. The priority level would be used when the master's write

Table continues on the next page...

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
					transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
write_qos	m_c_1	ENET1G RX	GPV1_BASE + 0x43000 + 0x104	6	Same as above
write_qos	m_c_2	ENET1G TX	GPV1_BASE + 0x44000 + 0x104	6	Same as above
write_qos	m_c_3	ENET	GPV1_BASE + 0x45000 + 0x104	6	Same as above
write_qos	m_c_4	USBO2	GPV1_BASE + 0x46000 + 0x104	12	Same as above
write_qos	m_c_5	USDHC1	GPV1_BASE + 0x47000 + 0x104	12	Same as above
write_qos	m_c_6	USDHC2	GPV1_BASE + 0x48000 + 0x104	14	Same as above
write_qos	m_c_8	ENET_QOS	GPV1_BASE + 0x4A000 + 0x104	0	
fn_mod	m_c_0	CAAM	GPV1_BASE + 0x42000 + 0x108	0	Issuing functionality modification register. This register sets the block issuing capability to one outstanding transaction. Legal programmable values are 0, 1, 2 and 3. NOTE: It is recommended to keep its reset value to obtain the best performance. 0: Default setting 1: Set read issuing capability to one outstanding transaction 2: Set write issuing capability to one outstanding transaction 3: Set both read and write issuing capability to one outstanding transaction
fn_mod	m_c_1	ENET1G RX	GPV1_BASE + 0x43000 + 0x108	0	Same as above
fn_mod	m_c_2	ENET1G TX	GPV1_BASE + 0x44000 + 0x108	0	Same as above
fn_mod	m_c_3	ENET	GPV1_BASE + 0x45000 + 0x108	0	Same as above
fn_mod	m_c_4	USBO2	GPV1_BASE + 0x46000 + 0x108	0	Same as above
fn_mod	m_c_5	USDHC1	GPV1_BASE + 0x47000 + 0x108	0	Same as above
fn_mod	m_c_6	USDHC2	GPV1_BASE + 0x48000 + 0x108	0	Same as above
fn_mod	m_c_8	ENET_QOS	GPV1_BASE + 0x4A000 + 0x108	0	Same as above

3. SIM_M7 registers

The SIM_M7 GPV base address is GPV4_BASE = 0x41400000. The following registers are implemented in this NIC.

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
fn_mod_ahb	m_b_5	DMA	GPV4_BASE + 0x47000 + 0x028	0	This register has 3 control bits. Bit 0: rd_incr_override. Writting 1 to this bit forces NIC to convert all AHB read transactions to a series of AXI singles. Bit 1: wr_incr_override. Writting 1 to this bit forces NIC to convert all AHB write transactions to a series of AXI singles. Bit 2: lock_override. Writting 1 to this bit forces NIC not to create any AXI lock transactions. NOTE: It is recommended to keep its reset value to obtain the best performance.
wr_tidemark	m_b_4	LPSRMIX masters	GPV4_BASE + 0x46000 + 0x040	4	The WFIFO depth is 8, so the valid programmable values to this register are 0 to 8. This is a tidemark level that stalls the release of the transaction until: The NIC receives the WLAST beat. The write FIFO becomes full. The number of occupied slots in the write data FIFO exceeds the write tidemark.
read_qos	m_b_0	CORTEXM7	GPV4_BASE + 0x42000 + 0x100	6	Set the priority of master's read. The priority level would be used when the master's read transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
read_qos	m_b_5	DMA	GPV4_BASE + 0x47000 + 0x100	6	Same as above
read_qos	m_b_6	IEE	GPV4_BASE + 0x48000 + 0x100	0	Same as above
write_qos	m_b_0	CORTEXM7	GPV4_BASE + 0x42000 + 0x104	6	Set the priority of master's write. The priority level would be used when the master's write transaction is being arbitrated by the NIC. Legal programmable values are from 0 to 15. Higher number sets higher priority (0: the lowest arbitration priority; 15: the highest priority).
write_qos	m_b_5	DMA	GPV4_BASE + 0x47000 + 0x104	6	Same as above
write_qos	m_b_6	IEE	GPV4_BASE + 0x48000 + 0x104	0	Same as above

Table continues on the next page...

Register Name	Port Name	Module Name	Absolute Address	Reset Value	Descriptions
fn_mod	m_b_0	CORTEXM 7	GPV4_BASE + 0x42000 + 0x108	0	Issuing functionality modification register. This register sets the block issuing capability to one outstanding transaction. Legal programmable values are 0, 1, 2 and 3. NOTE: It is recommended to keep its reset value to obtain the best performance. 0: Default setting 1: Set read issuing capability to one outstanding transaction 2: Set write issuing capability to one outstanding transaction 3: Set both read and write issuing capability to one outstanding transaction
fn_mod	m_b_5	DMA	GPV4_BASE + 0x47000 + 0x108	0	Same as above
fn_mod	m_b_6	IEE	GPV4_BASE + 0x48000 + 0x108	0	Same as above

4. **IB registers**

   There are no IB registers implemented in this chip.

5. **Address region control registers**

   There are not such type of registers implemented in this chip.

6. **Peripheral ID registers**

   The peripheral ID registers are implemented in the SIM_DISP, SIM_M, and SIM_M7. For more details, please see the Arm document: *DDI0397I_corelink_network_interconnect_nic301_r2p3_trm.pdf*.

Chapter 37
On-Chip RAM Memory Controller (OCRAM)

37.1 Chip-specific OCRAM information

Table 37-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

37.2 Overview

NOTE

For the M7 OCRAM (within FlexRAM module), see the FlexRAM chapter for introduction and details.

There are two OCRAM modules (OCRAM1 and OCRAM2), with dedicated 1.125 MB RAM in total, implemented in the chip. For each instance, there is 576 KB RAM[1], including 512 KB general data RAM and 64 KB cipher RAM.

NOTE

The OCRAM minimum size is 128 KB in default.

1. • When ECC enabled, total valid RAM is 512 KB.
   • When ECC disabled, total valid RAM is 576 KB.

The OCRAM block is implemented as an slave module on the 64-bit system AXI bus. Designed as a simple on-chip memory controller, it supports only one AXI port with memory banks. For the AXI port, the read and write transactions are handled by two independent modules. As it is possible to have simultaneous read and write request from the AXI bus, each memory bank has an arbiter with round-robin scheme. After arbitration, the granted read or write access command can then be issued to the memory cell through a read/write MUX.

The memory banks are organized with the lower 2 bits of the address which is the AXI bus address and is 64 bits aligned interleaved. This allows a read access and a write access to be processed at the same time if they are targeted to different memory banks.

Various options are provided for adding a pipeline in a read/write access, in order to ensure flexible timing control at both high and low frequencies.

37.2.1 Block diagram

The OCRAM block diagram is shown in the figure below.

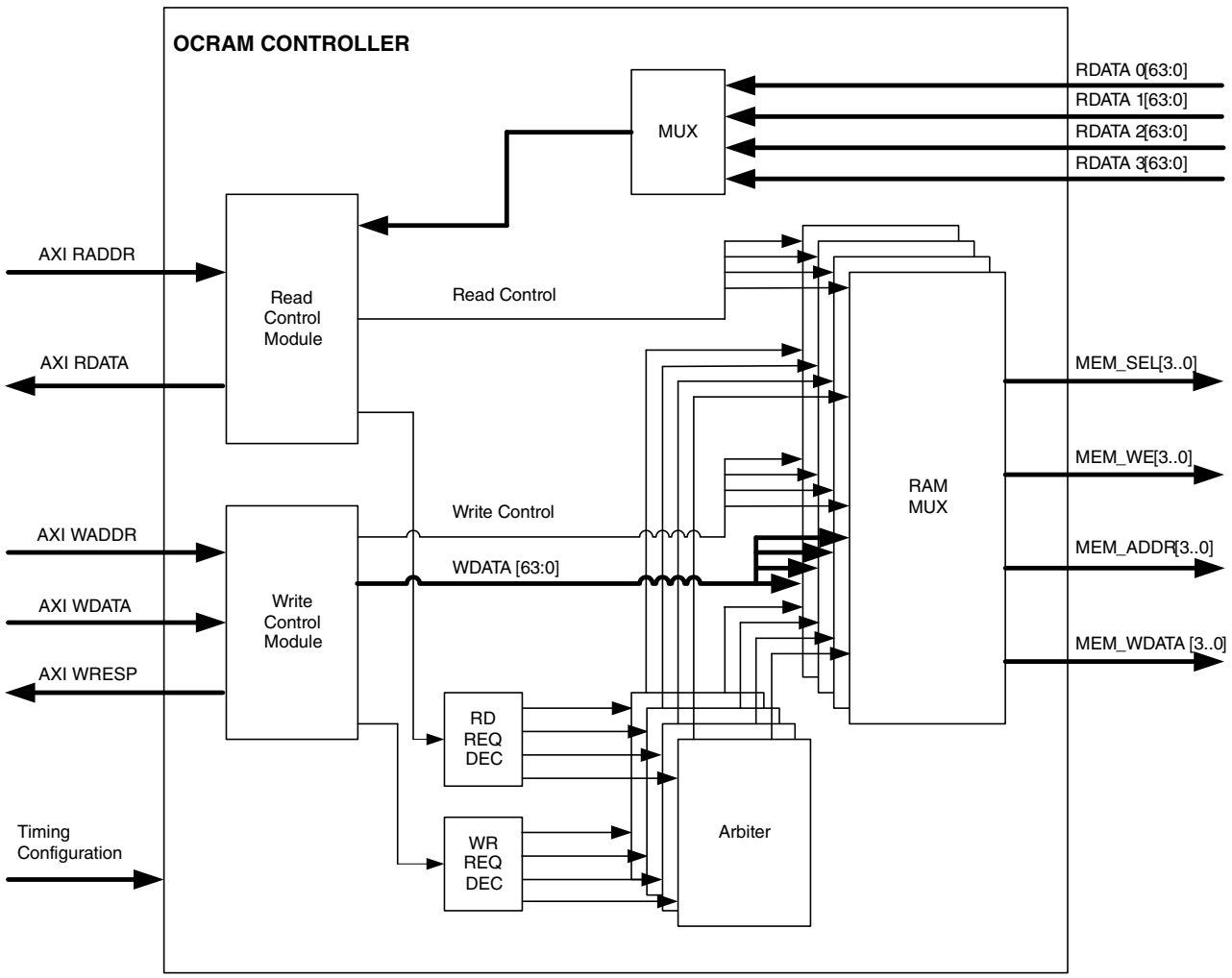


Figure 37-1. OCRAM Block Diagram

37.3 Functional Description

37.3.1 Read/Write Arbitration

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only a read request or a write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.

- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before the next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

37.4 Programmable Registers

There are no programmable registers in this block. However, OCRAM configurable bits can be found in the IOMUX Controller (IOMUXC) general purpose registers as below.

- TrustZone bits: IOMUXC_GPR_GPR10

Chapter 38
MECC64

38.1 Chip-specific MECC information

Table 38-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

ROM code will set the ECC enable bit (PIPE_ECC_EN[ECC_EN]) when MECC fuse is burnt.

38.2 Overview

The Memory ECC Controller (MECC64) module supports Single Error Correction and Double Error Detection (SEC-DED) ECC functions to On-Chip RAM (OCRAM) access. It can generate a 8-bit ECC code by Hsiao Hamming algorithm for 64-bit data. Corresponding OCRAM space is available to store ECC code. When ECC function is disabled, ECC OCRAM can also be used to store data.

38.2.1 Block diagram

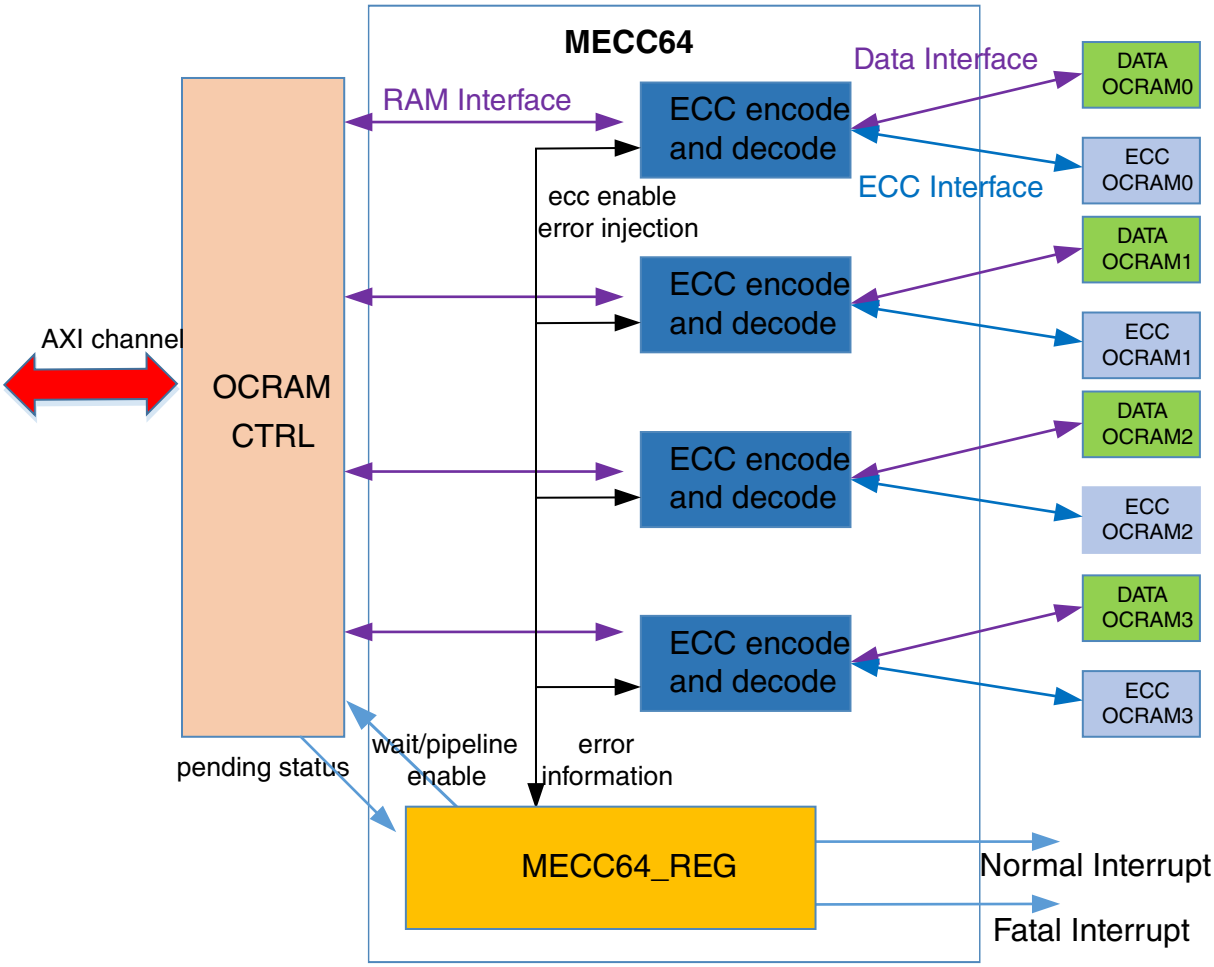


Figure 38-1. MECC64 block diagram

The following list shows detailed register/bitfield correspondance in the block diagram.

- ecc enable:
  - PIPE_ECC_EN[ECC_EN]
- error injection:
  - ERR_DATA_INJ_LOW/HIGH*n*
  - ERR_ECC_INJ*n*
- error information:
  - SINGLE_ERR_ADDR_ECC*n*
  - SINGLE_ERR_DATA_LOW/HIGH*n*
  - SINGLE_ERR_POS_LOW/HIGH*n*
  - MULTI_ERR_ADDR_ECC*n*
  - MULTI_ERR_DATA_LOW/HIGH*n*
- pending status:
  - PENDING_STAT

- wait/pipeline enable:
  - PIPE_ECC_EN[READ_DATA_WAIT_EN]
  - PIPE_ECC_EN[READ_ADDR_PIPE_EN]
  - PIPE_ECC_EN[WRITE_DATA_PIPE_EN]
  - PIPE_ECC_EN[WRITE_ADDR_PIPE_EN]

38.2.2 Features

The MECC64 includes the following features:

- Generates a 8-bit ECC code for 64-bit write data. Write data and ECC code are written into seperate OCRAM banks.
- Supports SEC-DED function for 64-bit read data and 8-bit ECC code.
- Integrate 4 banks OCRAM controller for converting AXI transaction to RAM interface.
- Error injection on write data and ECC code for debug purpose.
- Two different interrupt types: Normal and Fatal.

38.3 Functional description

The MECC64 converts AXI interface to RAM interface signals, and provides ECC encode/decode function for OCRAM write/read. For each data in OCRAM bank, corresponding space in ECC OCRAM banks is available to store ECC code when ECC function is enabled. When ECC function is disabled, ECC OCRAM banks can also be used to store data.

The OCRAM bank address allocation is as below:

- Bank0: ocram_base_address+0x20*i
- Bank1: ocram_base_address+0x20*i+0x8
- Bank2: ocram_base_address+0x20*i+0x10
- Bank3: ocram_base_address+0x20*i+0x18

where i = 0,1,2,3,4,...

38.3.1 Interface Conversion

The MECC64 integrates 4 banks OCRAM controller which converts AXI master interface signals to 4 groups of OCRAM interface signals (see OCRAM chapter for details). The 4 OCRAM banks are organized with lower 2 bits of address interleaved.

This allows a read access and a write access to be processed at the same time if they are targeted to different OCRAM banks. OCRAM controller can support to add pipeline or wait-state in read/write access by register configuration.

38.3.2 ECC encode

When ECC is enabled, the 4 groups of OCRAM write access will enter into seperate ECC encode channels. In a ECC encode channel, it will generate 8 bits ECC code for 64 bits OCRAM write data. Then 64 bits OCRAM write access will split into one 64 bits data write and one 8 bits ECC code write. The 64 bits data will be written into one of the DATA OCRAM banks according to address, and 8 bits ECC code will be written simutaneously into the corresponding ECC OCRAM bank.

When ECC is disabled, 4 ECC OCRAM banks can also be used to store data. The ECC OCRAM banks space address is after DATA OCRAM banks.

38.3.3 Error injection

The MECC64 can inject error in write data and ECC code by configuring ERR_DATA_INJ* and ERR_ECC_INJ* registers. The 64 bits write data and 8 bits ECC code can be inverted if corresponding bits in ERR_DATA_INJ* and ERR_ECC_INJ* registers are set. Each OCRAM bank has separate error injection registers. This feature is mainly used for debug.

38.3.4 ECC decode

When ECC is enabled, the 64 bits OCRAM read access will be split into one 64 bits data read and one 8 bits ECC code read. After getting read data from DATA OCRAM and ECC code from the corresponding ECC OCRAM, ECC decode channel can report single bit error or multiple bits error if ECC check is failed. At the same time, the error information will be saved in the module's error registers for debug. When single bit error is detected on read data, the error bit in read data will be corrected. In this case, AXI master will get read data correctly. When multiple bits error is detected on read data, error read data will not be corrected. In this case, master will get read data in error. When single bit error or multiple bits error happens, interrupt will be asserted if the corresponding enable bits are set in ERROR_STAT_EN and ERROR_SIG_EN registers. Each OCRAM bank has separate error and interrupt registers.

When ECC is disabled, 4 ECC OCRAM banks can also be used to store data. The ECC OCRAM banks can be read by using the address after DATA OCRAM banks.

NOTE

- If read access is 64-bit, the MECC64 will generate single or multi error interrupt once when meeting ECC error.
- If read access is 32-bit (or 16-bit/8-bit), the MECC64 will generate interrupt twice when meeting ECC error, once each in both the low 32 bits and the high 32 bits, even if it has a bit error in only the low 32 bits or the high 32 bits. This is because the ECC function will do ECC code validation on the full 64 bits for one read access.

38.3.5 Interrupts

The MECC64 can generate two kinds of interrupts:

- Normal interrupt can be generated for the following cases:
  - Single bit error is corrected for read;
  - AXI access address exceeds OCRAM address range;
  - AXI write access to OCRAM bank is not full size (wstrb=0xff).
- Fatal interrupt can be generated when 'multiple bits error' is detected for read.

38.3.6 Clocks

There are two clocks in MECC64:

Clock	Description
clk	Main clock
ips_clk	This is for IPS bus, used for register write/read.

There are no special requirements for these two clocks.

38.4 Signals

There are no external signals for MECC.

38.5 Initialization

The steps for initialization process is as below:

1. Enable MECC function (set PIPE_ECC_EN[ECC_EN] = 1).
2. Enable interrupts (set ERR_STAT_EN = 0xFFFF, ERR_SIG_EN = 0xFFFF).

NOTE

- When the MECC function is enabled, the OCRAM memory region can be initialized with 0x00 or 0xFF (initialization data for the first ECC encode), to generate ECC code within OCRAM ECC code region.
  - Initialization is required because the MECC64 needs a complete OCRAM and OCRAM ECC data. When initialized, it generates ECC code depending on initialization data, store data and its ECC code in OCRAM and OCRAM ECC memory.
  - Do such initialization with 64 bits access, with STRD or STMIA instruction. Do not use memset() API, which is STR instruction, in software coding level.
- OCRAM memory with ECC should be non-cacheable, or cacheable with write through, to assure data in OCRAM memory at the write time. It may have different methods to achieve this goal, such as Memory Protection Unit (MPU) or disable cache.

38.6 Application information

There are no application pseudocode examples for MECC64.

38.7 Memory Map and register definition

This section includes the MECC64 module memory map and detailed descriptions of all registers.

38.7.1 MECC64 register descriptions

38.7.1.1 MECC64 memory map

MECC1 base address: 4001_4000h

MECC2 base address: 4001_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Error Interrupt Status Register (ERR_STATUS)	32	W1C	0000_0000
4	Error Interrupt Status Enable Register (ERR_STAT_EN)	32	RW	0000_0000
8	Error Interrupt Enable Register (ERR_SIG_EN)	32	RW	0000_0000
C - 30	Error Injection On LOW 32 bits Of OCRAM Banka Write Data (ERR_DATA_INJ_LOW0 - ERR_DATA_INJ_LOW3)	32	RW	0000_0000
10 - 34	Error Injection On HIGH 32 bits Of OCRAM Banka Write Data (ERR_DATA_INJ_HIGH0 - ERR_DATA_INJ_HIGH3)	32	RW	0000_0000
14 - 38	Error Injection On 8 bits ECC code Of OCRAM Banka Write Data (ERR_ECC_INJ0 - ERR_ECC_INJ3)	32	RW	0000_0000
3C - 78	Single Error Address And ECC code On OCRAM Banka (SINGLE_ERR_ADDR_ECC0 - SINGLE_ERR_ADDR_ECC3)	32	RO	0000_0000
40 - 7C	LOW 32 Bits Single Error Read Data On OCRAM Banka (SINGLE_ERR_DATA_LOW0 - SINGLE_ERR_DATA_LOW3)	32	RO	0000_0000
44 - 80	HIGH 32 Bits Single Error Read Data On OCRAM Banka (SINGLE_ERR_DATA_HIGH0 - SINGLE_ERR_DATA_HIGH3)	32	RO	0000_0000
48 - 84	LOW Single Error Bit Position On OCRAM Banka (SINGLE_ERR_POS_LOW0 - SINGLE_ERR_POS_LOW3)	32	RO	0000_0000
4C - 88	HIGH Single Error Bit Position On OCRAM Banka (SINGLE_ERR_POS_HIGH0 - SINGLE_ERR_POS_HIGH3)	32	RO	0000_0000
8C - B0	Multiple Error Address And ECC code On OCRAM Banka (MULTI_ERR_ADDR_ECC0 - MULTI_ERR_ADDR_ECC3)	32	RO	0000_0000
90 - B4	LOW 32 Bits Multiple Error Read Data On OCRAM Banka (MULTI_ERR_DATA_LOW0 - MULTI_ERR_DATA_LOW3)	32	RO	0000_0000
94 - B8	HIGH 32 Bits Multiple Error Read Data On OCRAM Banka (MULTI_ERR_DATA_HIGH0 - MULTI_ERR_DATA_HIGH3)	32	RO	0000_0000
100	OCRAM Pipeline And ECC Enable (PIPE_ECC_EN)	32	RW	0000_0000
104	Pending Status (PENDING_STAT)	32	RO	0000_0000

38.7.1.2 Error Interrupt Status Register (ERR_STATUS)

Error Interrupt Status Register.

38.7.1.2.1 Offset

Register	Offset
ERR_STATUS	0h

38.7.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR_ERR3	ADDR_ERR2	ADDR_ERR1	ADDR_ERR0	STRB_ERR3	STRB_ERR2	STRB_ERR1	STRB_ERR0	MULTI_ERR3	MULTI_ERR2	MULTI_ERR1	MULTI_ERR0	SINGLE_ERR3	SINGLE_ERR2	SINGLE_ERR1	SINGLE_ERR0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.2.3 Fields

Field	Description
31-16 —	Reserved
15 ADDR_ERR3	OCRAM Access Error On Bank3 When OCRAM access unallocated address on bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - OCRAM access error does not happen on OCRAM bank3. 1 - OCRAM access error happens on OCRAM bank3.
14 ADDR_ERR2	OCRAM Access Error On Bank2 When OCRAM access unallocated address on bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - OCRAM access error does not happen on OCRAM bank2. 1 - OCRAM access error happens on OCRAM bank2.
13 ADDR_ERR1	OCRAM Access Error On Bank1 When OCRAM access unallocated address on bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set.

Table continues on the next page...

Field	Description
	0 - OCRAM access error does not happen on OCRAM bank1. 1 - OCRAM access error happens on OCRAM bank1.
12 ADDR_ERR0	OCRAM Access Error On Bank0 When OCRAM access unallocated address on bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - OCRAM access error does not happen on OCRAM bank0. 1 - OCRAM access error happens on OCRAM bank0.
11 STRB_ERR3	AXI Strobe Error On OCRAM Bank3 When AXI write access to OCRAM bank3 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on OCRAM bank3. 1 - AXI strobe error happens on OCRAM bank3.
10 STRB_ERR2	AXI Strobe Error On OCRAM Bank2 When AXI write access to OCRAM bank2 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on OCRAM bank2. 1 - AXI strobe error happens on OCRAM bank2.
9 STRB_ERR1	AXI Strobe Error On OCRAM Bank1 When AXI write access to OCRAM bank1 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on OCRAM bank1. 1 - AXI strobe error happens on OCRAM bank1.
8 STRB_ERR0	AXI Strobe Error On OCRAM Bank0 When AXI write access to OCRAM bank0 is not full size(wstrb=0xff), this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - AXI strobe error does not happen on OCRAM bank0. 1 - AXI strobe error happens on OCRAM bank0.
7 MULTI_ERR3	Multiple Bits Error On OCRAM Bank3 When detect multiple bits error on read data from OCRAM bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on OCRAM bank3. 1 - Multiple bits error happens on OCRAM bank3.
6 MULTI_ERR2	Multiple Bits Error On OCRAM Bank2 When detect multiple bits error on read data from OCRAM bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on OCRAM bank2. 1 - Multiple bits error happens on OCRAM bank2.
5 MULTI_ERR1	Multiple Bits Error On OCRAM Bank1 When detect multiple bits error on read data from OCRAM bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on OCRAM bank1. 1 - Multiple bits error happens on OCRAM bank1.

Table continues on the next page...

Field	Description
4 MULTI_ERR0	Multiple Bits Error On OCRAM Bank0 When detect multiple bits error on read data from OCRAM bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Multiple bits error does not happen on OCRAM bank0. 1 - Multiple bits error happens on OCRAM bank0.
3 SINGLE_ERR3	Single Bit Error On OCRAM Bank3 When detect single bit error on read data from OCRAM bank3, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on OCRAM bank3. 1 - Single bit error happens on OCRAM bank3.
2 SINGLE_ERR2	Single Bit Error On OCRAM Bank2 When detect single bit error on read data from OCRAM bank2, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on OCRAM bank2. 1 - Single bit error happens on OCRAM bank2.
1 SINGLE_ERR1	Single Bit Error On OCRAM Bank1 When detect single bit error on read data from OCRAM bank1, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on OCRAM bank1. 1 - Single bit error happens on OCRAM bank1.
0 SINGLE_ERR0	Single Bit Error On OCRAM Bank0 When detect single bit error on read data from OCRAM bank0, this bit will be asserted if corresponding status enable bit in ERR_STAT_EN register is set. 0 - Single bit error does not happen on OCRAM bank0. 1 - Single bit error happens on OCRAM bank0.

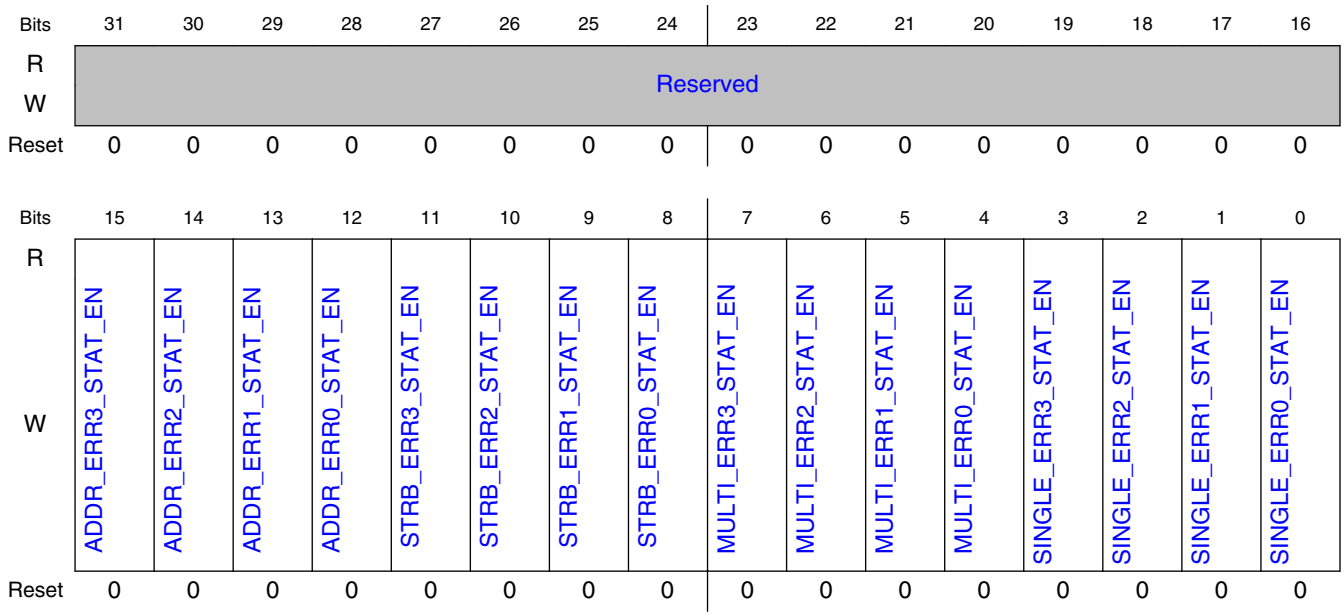
38.7.1.3 Error Interrupt Status Enable Register (ERR_STAT_EN)

Setting the bits in this register to 1 enables the corresponding Error Interrupt Status to be set by the specified event.

38.7.1.3.1 Offset

Register	Offset
ERR_STAT_EN	4h

38.7.1.3.2 Diagram



38.7.1.3.3 Fields

Field	Description
31-16 —	Reserved
15 ADDR_ERR3_STAT_EN	OCRAM Access Error Status Enable On Bank3 0 - Disabled 1 - Enabled
14 ADDR_ERR2_STAT_EN	OCRAM Access Error Status Enable On Bank2 0 - Disabled 1 - Enabled
13 ADDR_ERR1_STAT_EN	OCRAM Access Error Status Enable On Bank1 0 - Disabled 1 - Enabled
12 ADDR_ERR0_STAT_EN	OCRAM Access Error Status Enable On Bank0 0 - Disabled 1 - Enabled
11 STRB_ERR3_STAT_EN	AXI Strobe Error Status Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
10	AXI Strobe Error Status Enable On OCRAM Bank2 0 - Disabled

Table continues on the next page...

Field	Description
STRB_ERR2_STAT_EN	1 - Enabled
9 STRB_ERR1_STAT_EN	AXI Strobe Error Status Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
8 STRB_ERR0_STAT_EN	AXI Strobe Error Status Enable On OCRAM Bank0 0 - Disabled 1 - Enabled
7 MULTI_ERR3_STAT_EN	Multiple Bits Error Status Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
6 MULTI_ERR2_STAT_EN	Multiple Bits Error Status Enable On OCRAM Bank2 0 - Disabled 1 - Enabled
5 MULTI_ERR1_STAT_EN	Multiple Bits Error Status Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
4 MULTI_ERR0_STAT_EN	Multiple Bits Error Status Enable On OCRAM Bank0 0 - Disabled 1 - Enabled
3 SINGLE_ERR3_STAT_EN	Single Bit Error Status Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
2 SINGLE_ERR2_STAT_EN	Single Bit Error Status Enable On OCRAM Bank2 0 - Disabled 1 - Enabled
1 SINGLE_ERR1_STAT_EN	Single Bit Error Status Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
0 SINGLE_ERR0_STAT_EN	Single Bit Error Status Enable On OCRAM Bank0 0 - Disabled 1 - Enabled

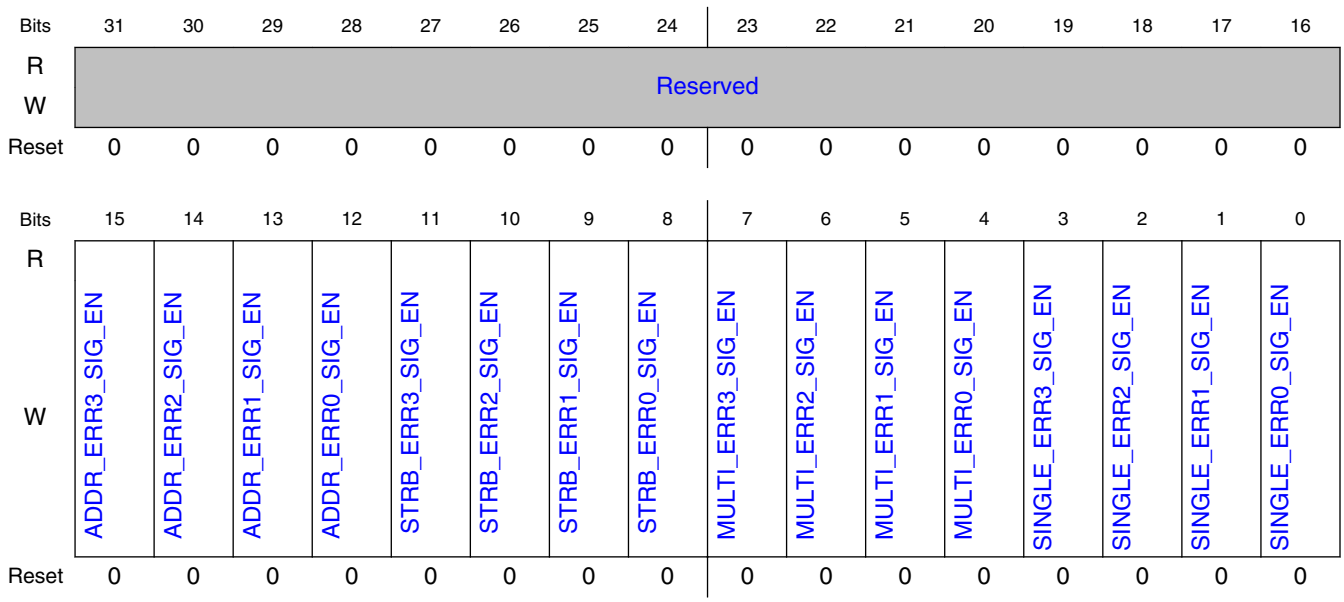
38.7.1.4 Error Interrupt Enable Register (ERR_SIG_EN)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

38.7.1.4.1 Offset

Register	Offset
ERR_SIG_EN	8h

38.7.1.4.2 Diagram



38.7.1.4.3 Fields

Field	Description
31-16 —	Reserved
15 ADDR_ERR3_SIG_EN	OCRAM Access Error Interrupt Enable On Bank3 0 - Disabled 1 - Enabled
14 ADDR_ERR2_SIG_EN	OCRAM Access Error Interrupt Enable On Bank2 0 - Disabled 1 - Enabled
13 ADDR_ERR1_SIG_EN	OCRAM Access Error Interrupt Enable On Bank1 0 - Disabled 1 - Enabled
12	OCRAM Access Error Interrupt Enable On Bank0

Table continues on the next page...

Field	Description
ADDR_ERR0_SIG_EN	0 - Disabled 1 - Enabled
11 STRB_ERR3_SIG_EN	AXI Strobe Error Interrupt Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
10 STRB_ERR2_SIG_EN	AXI Strobe Error Interrupt Enable On OCRAM Bank2 0 - Disabled 1 - Enabled
9 STRB_ERR1_SIG_EN	AXI Strobe Error Interrupt Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
8 STRB_ERR0_SIG_EN	AXI Strobe Error Interrupt Enable On OCRAM Bank0 0 - Disabled 1 - Enabled
7 MULTI_ERR3_SIG_EN	Multiple Bits Error Interrupt Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
6 MULTI_ERR2_SIG_EN	Multiple Bits Error Interrupt Enable On OCRAM Bank2 0 - Disabled 1 - Enabled
5 MULTI_ERR1_SIG_EN	Multiple Bits Error Interrupt Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
4 MULTI_ERR0_SIG_EN	Multiple Bits Error Interrupt Enable On OCRAM Bank0 0 - Disabled 1 - Enabled
3 SINGLE_ERR3_SIG_EN	Single Bit Error Interrupt Enable On OCRAM Bank3 0 - Disabled 1 - Enabled
2 SINGLE_ERR2_SIG_EN	Single Bit Error Interrupt Enable On OCRAM Bank2 0 - Disabled 1 - Enabled
1 SINGLE_ERR1_SIG_EN	Single Bit Error Interrupt Enable On OCRAM Bank1 0 - Disabled 1 - Enabled
0 SINGLE_ERR0_SIG_EN	Single Bit Error Interrupt Enable On OCRAM Bank0 0 - Disabled 1 - Enabled

38.7.1.5 Error Injection On LOW 32 bits Of OCRAM Banka Write Data (ERR_DATA_INJ_LOW0 - ERR_DATA_INJ_LOW3)

LOW 32 bits of OCRAM bank0 write data will be reversed if corresponding bits in this register are set.

38.7.1.5.1 Offset

Register	Offset
ERR_DATA_INJ_LOW0	Ch
ERR_DATA_INJ_LOW1	18h
ERR_DATA_INJ_LOW2	24h
ERR_DATA_INJ_LOW3	30h

38.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.5.3 Fields

Field	Description
31-0 ERR_DATA_INJ	Error Injection On LOW 32 bits Of OCRAM Bank0 Write Data LOW 32 bits of OCRAM bank0 write data will be reversed if corresponding bits in this register are set. For example: • If it is set to 0x00000001, then bit 0 of LOW 32bits of OCRAM bank0 write data will be reversed. • If it is set to 0x00000003, then bit 0 and bit 1 of LOW 32bits of OCRAM bank0 write data will be reversed. • if it is set to 0xFFFFFFFF, all bits of LOW 32bits of OCRAM bank0 write data will be reversed.

38.7.1.6 Error Injection On HIGH 32 bits Of OCRAM Banka Write Data (ERR_DATA_INJ_HIGH0 - ERR_DATA_INJ_HIGH3)

HIGH 32 bits of OCRAM bank0 write data will be reversed if corresponding bits in this register are set.

38.7.1.6.1 Offset

Register	Offset
ERR_DATA_INJ_HIGH0	10h
ERR_DATA_INJ_HIGH1	1Ch
ERR_DATA_INJ_HIGH2	28h
ERR_DATA_INJ_HIGH3	34h

38.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR_DATA_INJ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.6.3 Fields

Field	Description
31-0 ERR_DATA_INJ	Error Injection On HIGH 32 bits Of OCRAM Bank0 Write Data HIGH 32 bits of OCRAM bank0 write data will be reversed if corresponding bits in this register are set. For example: • If it is set to 0x00000001, then bit 0 of HIGH 32bits of OCRAM bank0 write data will be reversed. • If it is set to 0x00000003, then bit 0 and bit 1 of HIGH 32bits of OCRAM bank0 write data will be reversed. • if it is set to 0xFFFFFFFF, all bits of HIGH 32bits of OCRAM bank0 write data will be reversed.

38.7.1.7 Error Injection On 8 bits ECC code Of OCRAM Banka Write Data (ERR_ECC_INJ0 - ERR_ECC_INJ3)

8 bits ECC code of OCRAM bank0 write data will be reversed if corresponding bits in this register are set.

38.7.1.7.1 Offset

Register	Offset
ERR_ECC_INJ0	14h
ERR_ECC_INJ1	20h
ERR_ECC_INJ2	2Ch
ERR_ECC_INJ3	38h

38.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ERR_ECC_INJ							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.7.3 Fields

Field	Description
31-8 —	Reserved
7-0 ERR_ECC_INJ	Error Injection On 8 bits ECC code Of OCRAM Bank0 Write Data 8 bits ECC code of OCRAM bank0 write data will be reversed if corresponding bits in this register are set.

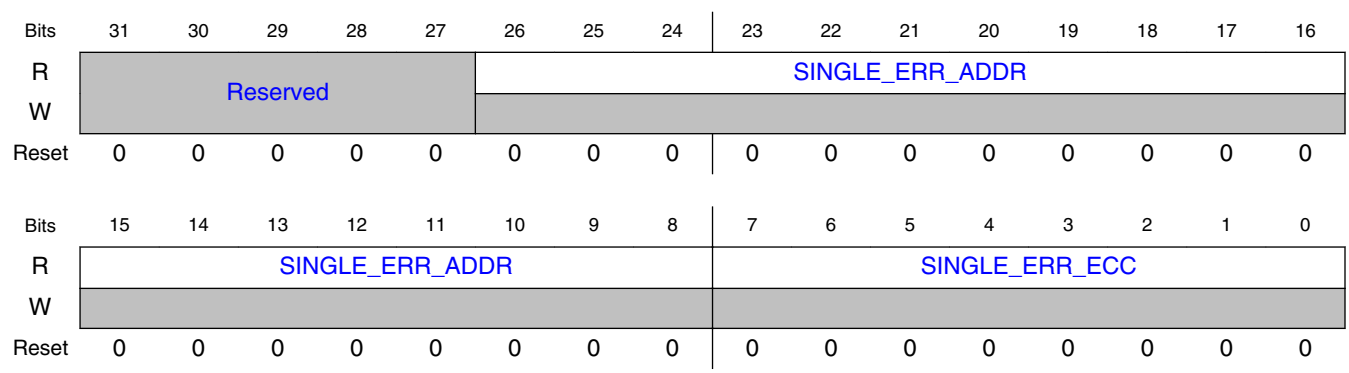
38.7.1.8 Single Error Address And ECC code On OCRAM Banka (SINGLE_ERR_ADDR_ECC0 - SINGLE_ERR_ADDR_ECC3)

When single bit error is corrected in OCRAM bank0 read access, this register stores OCRAM address and ECC code.

38.7.1.8.1 Offset

Register	Offset
SINGLE_ERR_ADDR_ECC0	3Ch
SINGLE_ERR_ADDR_ECC1	50h
SINGLE_ERR_ADDR_ECC2	64h
SINGLE_ERR_ADDR_ECC3	78h

38.7.1.8.2 Diagram



38.7.1.8.3 Fields

Field	Description
31-27 —	Reserved
26-8 SINGLE_ERR_ADDR	Single Error Address On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field stores OCRAM address.
7-0 SINGLE_ERR_ECC	Single Error ECC code On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field stores ECC code.

38.7.1.9 LOW 32 Bits Single Error Read Data On OCRAM Banka (SINGLE_ERR_DATA_LOW0 - SINGLE_ERR_DATA_LOW3)

When single bit error is corrected in OCRAM bank0 read access, this register stores LOW 32 bits uncorrected read data.

38.7.1.9.1 Offset

Register	Offset
SINGLE_ERR_DATA_LOW0	40h
SINGLE_ERR_DATA_LOW1	54h
SINGLE_ERR_DATA_LOW2	68h
SINGLE_ERR_DATA_LOW3	7Ch

38.7.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.9.3 Fields

Field	Description
31-0 SINGLE_ERR_DATA	LOW 32 Bits Single Error Read Data On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field stores LOW 32 bits uncorrected read data.

38.7.1.10 HIGH 32 Bits Single Error Read Data On OCRAM Banka (SINGLE_ERR_DATA_HIGH0 - SINGLE_ERR_DATA_HIGH3)

When single bit error is corrected in OCRAM bank0 read access, this register stores HIGH 32 bits uncorrected read data.

38.7.1.10.1 Offset

Register	Offset
SINGLE_ERR_DATA_HIGH0	44h
SINGLE_ERR_DATA_HIGH1	58h
SINGLE_ERR_DATA_HIGH2	6Ch
SINGLE_ERR_DATA_HIGH3	80h

38.7.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.10.3 Fields

Field	Description
31-0 SINGLE_ERR_DATA	HIGH 32 Bits Single Error Read Data On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field stores HIGH 32 bits uncorrected read data.

38.7.1.11 LOW Single Error Bit Position On OCRAM Banka (SINGLE_ERR_POS_LOW0 - SINGLE_ERR_POS_LOW3)

When single bit error is corrected in OCRAM bank0 read access, this register indicates which bit is corrected in the LOW 32 bits of read data.

38.7.1.11.1 Offset

Register	Offset
SINGLE_ERR_POS_LOW0	48h
SINGLE_ERR_POS_LOW1	5Ch
SINGLE_ERR_POS_LOW2	70h
SINGLE_ERR_POS_LOW3	84h

38.7.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.11.3 Fields

Field	Description
31-0 SINGLE_ERR_POS	LOW Single Error Bit Position On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field indicates which bit is corrected in LOW 32 bits of read data.

38.7.1.12 HIGH Single Error Bit Position On OCRAM Banka (SINGLE_ERR_POS_HIGH0 - SINGLE_ERR_POS_HIGH3)

When single bit error is corrected in OCRAM bank0 read access, this register indicates which bit is corrected in the HIGH 32 bits of read data.

38.7.1.12.1 Offset

Register	Offset
SINGLE_ERR_POS_HIGH0	4Ch
SINGLE_ERR_POS_HIGH1	60h
SINGLE_ERR_POS_HIGH2	74h
SINGLE_ERR_POS_HIGH3	88h

38.7.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SINGLE_ERR_POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

38.7.1.12.3 Fields

Field	Description
31-0 SINGLE_ERR_POS	HIGH Single Error Bit Position On OCRAM Bank0 When single bit error is corrected in OCRAM bank0 read access, this field indicates which bit is corrected in HIGH 32 bits of read data.

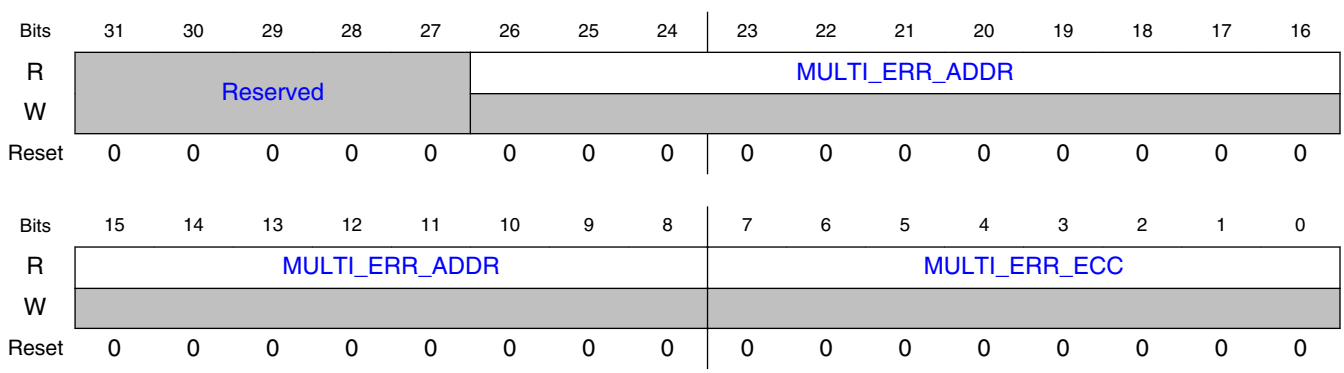
38.7.1.13 Multiple Error Address And ECC code On OCRAM Banka (MULTI_ERR_ADDR_ECC0 - MULTI_ERR_ADDR_ECC3)

When multiple bits error are detected in OCRAM bank0 read access, this register stores OCRAM address and ECC code.

38.7.1.13.1 Offset

Register	Offset
MULTI_ERR_ADDR_ECC0	8Ch
MULTI_ERR_ADDR_ECC1	98h
MULTI_ERR_ADDR_ECC2	A4h
MULTI_ERR_ADDR_ECC3	B0h

38.7.1.13.2 Diagram



38.7.1.13.3 Fields

Field	Description
31-27 —	Reserved
26-8 MULTI_ERR_ADDR	Multiple Error Address On OCRAM Bank0 When multiple bits error are detected in OCRAM bank0 read access, this field stores OCRAM address.
7-0 MULTI_ERR_ECC	Multiple Error ECC code On OCRAM Bank0 When multiple bits error are detected in OCRAM bank0 read access, this field stores ECC code.

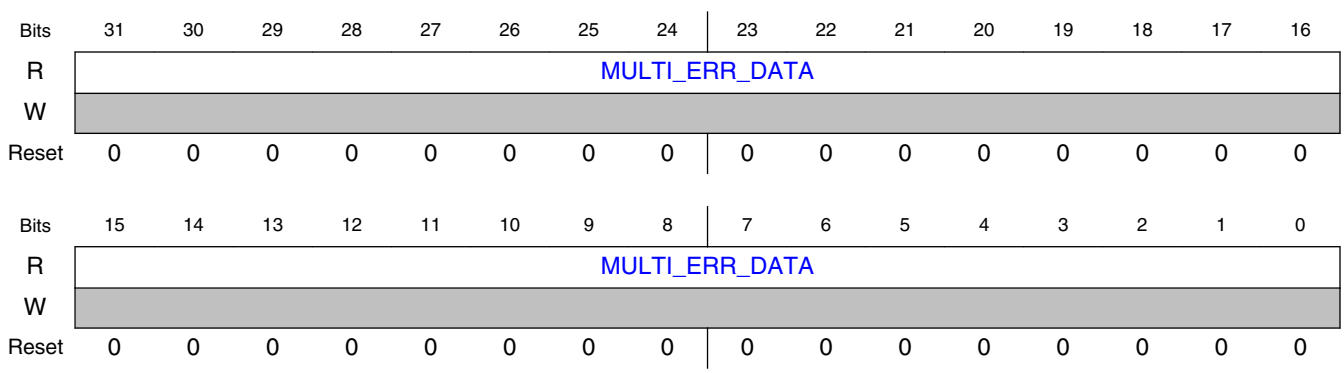
38.7.1.14 LOW 32 Bits Multiple Error Read Data On OCRAM Banka (MULTI_ERR_DATA_LOW0 - MULTI_ERR_DATA_LOW3)

When multiple bits error are detected in OCRAM bank0 read access, this register stores LOW 32 bits read data.

38.7.1.14.1 Offset

Register	Offset
MULTI_ERR_DATA_LOW0	90h
MULTI_ERR_DATA_LOW1	9Ch
MULTI_ERR_DATA_LOW2	A8h
MULTI_ERR_DATA_LOW3	B4h

38.7.1.14.2 Diagram



38.7.1.14.3 Fields

Field	Description
31-0 MULTI_ERR_DATA	LOW 32 Bits Multiple Error Read Data On OCRAM Bank0 When multiple bits error are detected in OCRAM bank0 read access, this field stores LOW 32 bits read data.

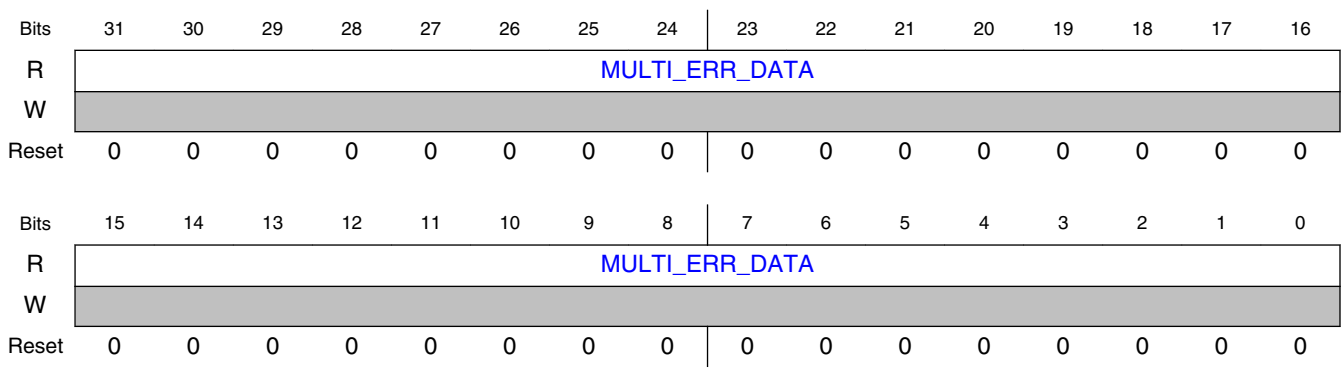
38.7.1.15 HIGH 32 Bits Multiple Error Read Data On OCRAM Banka (MULTI_ERR_DATA_HIGH0 - MULTI_ERR_DATA_HIGH3)

When multiple bits error are detected in OCRAM bank0 read access, this register stores HIGH 32 bits read data.

38.7.1.15.1 Offset

Register	Offset
MULTI_ERR_DATA_HIGH0	94h
MULTI_ERR_DATA_HIGH1	A0h
MULTI_ERR_DATA_HIGH2	ACh
MULTI_ERR_DATA_HIGH3	B8h

38.7.1.15.2 Diagram



38.7.1.15.3 Fields

Field	Description
31-0 MULTI_ERR_DATA	HIGH 32 Bits Multiple Error Read Data On OCRAM Bank0 When multiple bits error are detected in OCRAM bank0 read access, this field stores HIGH 32 bits read data.

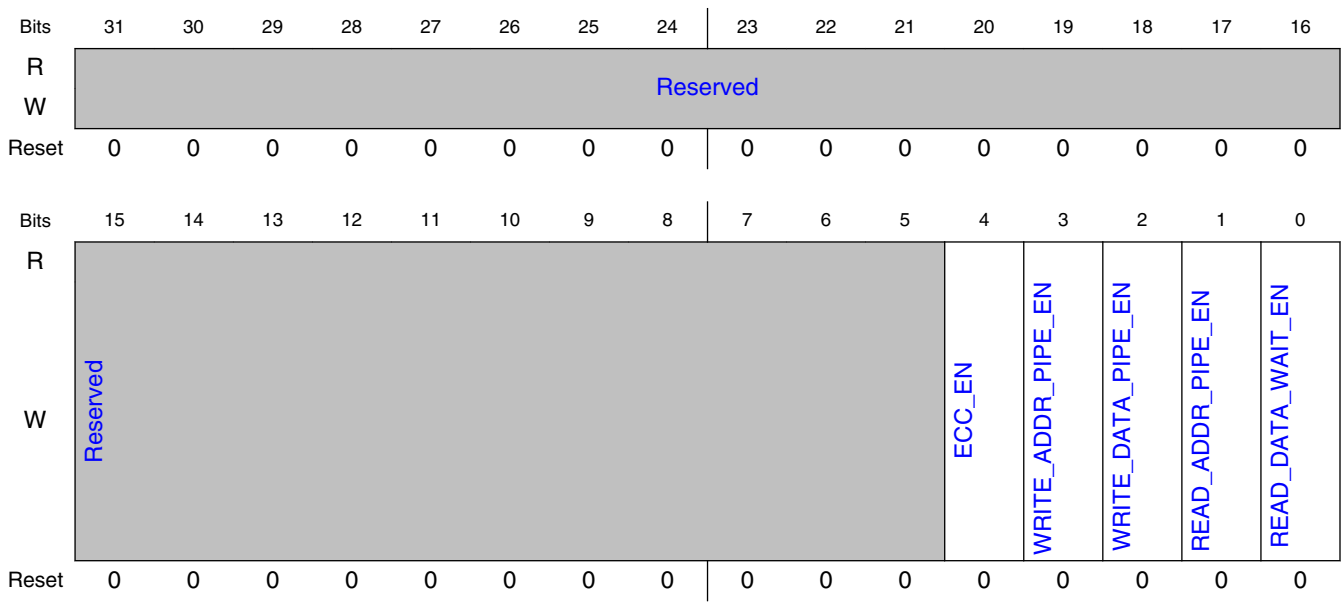
38.7.1.16 OCRAM Pipeline And ECC Enable (PIPE_ECC_EN)

OCRAM pipeline and ECC enable.

38.7.1.16.1 Offset

Register	Offset
PIPE_ECC_EN	100h

38.7.1.16.2 Diagram



38.7.1.16.3 Fields

Field	Description
31-5 —	Reserved
4 ECC_EN	ECC Function Enable Enable ECC64 function for OCRAM access. This register bit reset value comes from IOMUX GPR. 0 - Disable. 1 - Enable.
3 WRITE_ADDR_PIPE_EN	Write Address Pipeline Enable Pipeline enable signal for write address path, when this signal is set to 1, the write address will be registered before can be applied to memory cell.

Table continues on the next page...

Field	Description
	0 - Disable. 1 - Enable.
2 WRITE_DATA_PIPE_EN	Write Data Pipeline Enable Pipeline enable signal for write data path, when this signal is set to 1, the write data will be registered before can be applied to memory cell. 0 - Disable. 1 - Enable.
1 READ_ADDR_PIPE_EN	Read Address Pipeline Enable Pipeline enable signal for read address path, when this signal is set to 1, the read address will be registered before can be applied to memory cell. 0 - Disable. 1 - Enable.
0 READ_DATA_WAIT_EN	Read Data Wait Enable 1-cycle wait state in read access enable signal, when this signal is set to 1, 1-cycle wait state will be inserted into each read access. 0 - Disable. 1 - Enable.

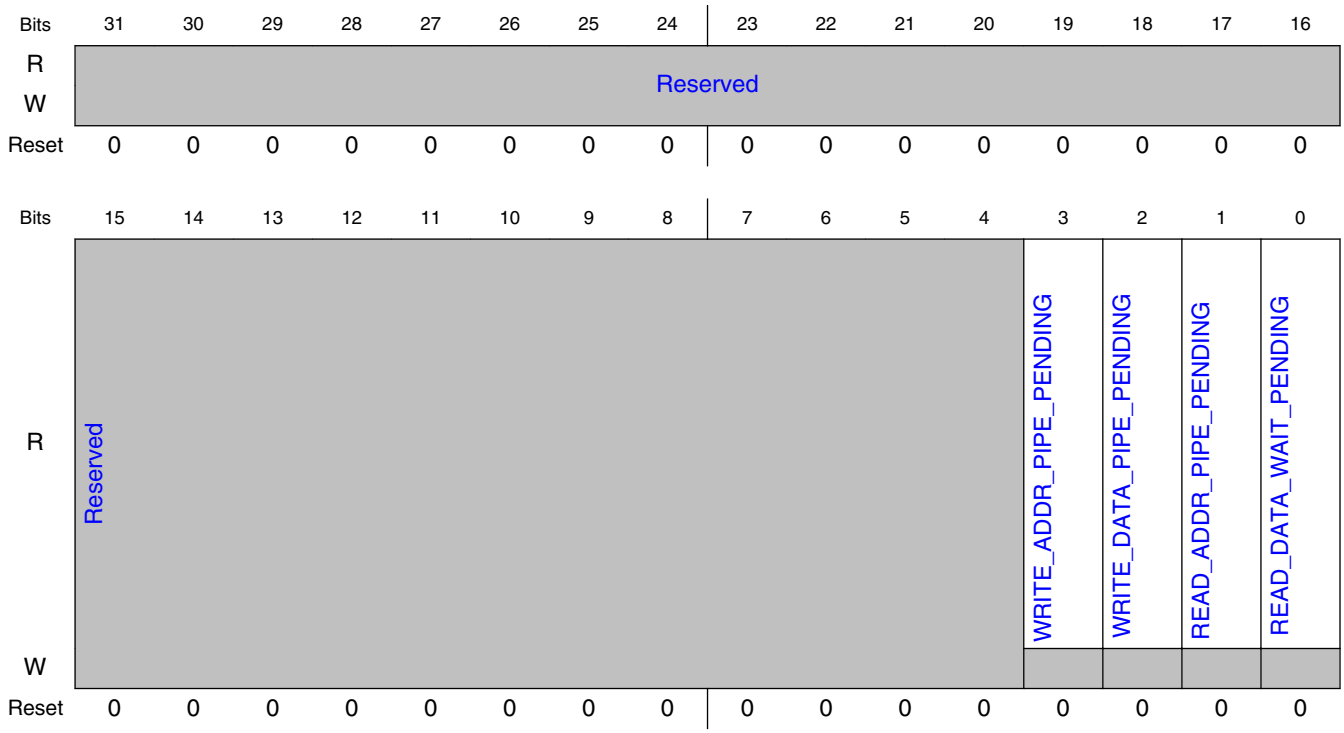
38.7.1.17 Pending Status (PENDING_STAT)

Update pending signal for OCRAM wait and pipeline enable.

38.7.1.17.1 Offset

Register	Offset
PENDING_STAT	104h

38.7.1.17.2 Diagram



38.7.1.17.3 Fields

Field	Description
31-4 —	Reserved
3 WRITE_ADDR_PIPE_PENDING	Write Address Pipeline Pending Update pending status for WRITE_ADDR_PIPE_EN. 0 - No update pending status for WRITE_ADDR_PIPE_EN. 1 - When WRITE_ADDR_PIPE_EN register bit is changed, this register bit will be set until the new setup becomes valid in the controller.
2 WRITE_DATA_PIPE_PENDING	Write Data Pipeline Pending Update pending status for WRITE_DATA_PIPE_EN. 0 - No update pending status for WRITE_DATA_PIPE_EN. 1 - When WRITE_DATA_PIPE_EN register bit is changed, this register bit will be set until the new setup becomes valid in the controller.
1 READ_ADDR_PIPE_PENDING	Read Address Pipeline Pending Update pending status for READ_ADDR_PIPE_EN. 0 - No update pending status for READ_ADDR_PIPE_EN. 1 - When READ_ADDR_PIPE_EN register bit is changed, this register bit will be set until the new setup becomes valid in the controller.

Table continues on the next page...

Field	Description
0 READ_DATA_WAIT_PENDING	Read Data Wait Pending Update pending status for READ_DATA_WAIT_EN. 0 - No update pending status for READ_DATA_WAIT_EN. 1 - When READ_DATA_WAIT_EN register bit is changed, this register bit will be set until the new setup becomes valid in the controller.

Chapter 39
FlexRAM

39.1 Chip-specific FlexRAM information

Table 39-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

For additional information see application note AN12077: Using the i.MX RT FlexRAM. This document describes the flexible memory array available on the i.MX RT MCUs. It includes the following information:

- Configuration of the bank array
- Memory type size definition
- Available memory controllers
- Power domains and clocks
- Interrupt request generation
- Example of FlexRAM configuration usage on a specific application use case

39.2 Overview

The Flexible RAM Controller (FLEXRAM) provides access control, as well as Single-bit Error Correction and Dual-bit Error Detection (SEC-DED) ECC, from TCM/OCRAM to 512KB of on-chip RAM. Another 128KB of on-chip RAM is allocated to store ECC cipher or used by OCRAM to store normal data.

39.2.1 Block diagram

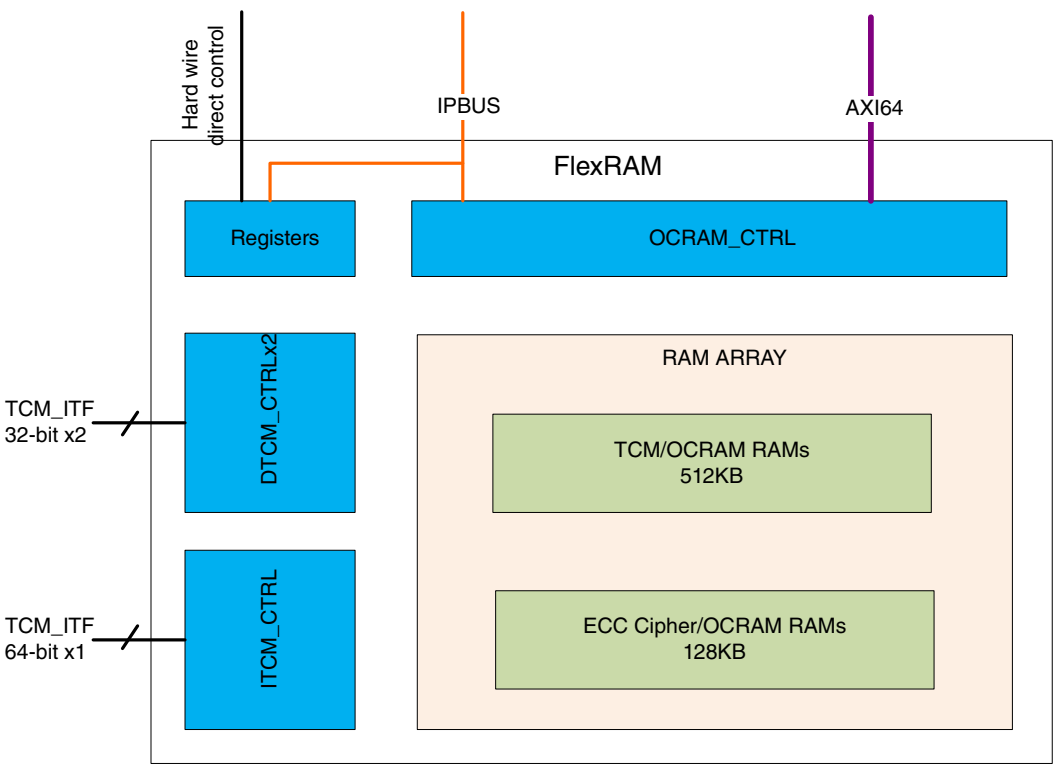


Figure 39-1. FLEXRAM block diagram

39.2.2 Features

FLEXRAM includes the following features:

- Integrated I-TCM and D-TCM RAM controller:
  - 64-bit I-TCM interface and 2x 32-bit D-TCM interfaces
  - Supports 512 KB TCM (for both ITCM and DTCM) space
  - Supports two access modes:

    - Fast mode: RAM access is expected to be finished in 1-cycle for both read and write.
    - Wait mode: RAM access is expected to be finished in 2-cycle. Wait mode for read and write path can be enabled separately.
  - Synchronous interface to the Cortex-M7 Core, run at the same frequency as the core
  - Automatic clock gating control to reduce power consumption
- Integrated OCRAM controller:
  - Single SRAM bank controller
  - Supports 640 KB SRAM size
  - Synchronous to the system bus, runs at the same frequency as bus fabric
- Parametrized RAM array:
  - Supports 640 KB total memory space
  - Support 64 blocks of 32-bit RAM
- RAM array portioning:
  - RAM size of ITCM, DTCM and OCRAM can be configured separately
  - Step of the RAM size partitioning for ITCM, DTCM and OCRAM is (Total RAM SIZE)/16
  - Flexible RAM bank organization include 16 banks with size of 32KB and 16 banks with size of 8KB. The first 16 banks can be allocated for ITCM, DTCM and OCRAM separately for master data storage, while the latter 16 banks (smaller sized) can only store data for OCRAM or store ECC for ITCM, DTCM and OCRAM.
- Flexible power mode:
  - Run mode: All RAM banks are powered on
  - Retention mode: Only one (bank0) out of the 16 RAM banks is on while the rest banks are powered off
  - Partial mode: 16 out of the 32 RAM banks (bank8 to bank15, and bank24 to bank31) can be powered off dynamically
- Bank8 to bank15, and bank24 to bank 31 can be disabled with hardware fuse setting. When the fuse is set, total RAM size will be limit to 320 KB
- Generates interrupt upon TCM, OCRAM access out of the configured RAM range
- Generates interrupt when access to pre-configured OCRAM, ITCM and DTCM address occurs
  - Independent address control for OCRAM, ITCM and DTCM
  - Interrupt generation on read or write access based register settings

39.3 Functional description

FLEXRAM converts AXI and TCM interface signals to RAM interface signals, and implements MUX control for OCRAM , DTCM and ITCM access to on-chip RAM. FLEXRAM supports Single-bit Error Correction and Dual-bit Error Detection ECC (SEC-DED) for both OCRAM and TCM.

39.3.1 Interface Conversion

FLEXRAM integrates OCRAM controller which converts AXI master interface signals to RAM interface signals (Please see the OCRAM chapter for more details). OCRAM controller can support to add pipeline or wait-state in read/write access by setting FLEXRAM_CTRL register.

FLEXRAM also implements TCM controller module to convert DTCM/ITCM interface to RAM interface. TCM read/write access can be extended to 2 cycles by setting TCM_CTRL register. When DTCM and ITCM access unallocated RAM address, TCM controller can send TCM_ERR signa to DTCM/ITCM interface.

If TCM ECC is enabled and ECC Error is detected on a read access, the subsequent valid access in the error detection cycle would be re-aligned. TCM_RETRY signal would be sent to ITCM/DTCM, and if the Error is a single-bit ECC error, TCM controller would scrub the address with corrected data.

39.3.2 RAM Bank Allocation

There are two parts of RAM can be accessed through FlexRAM, 512 KB data RAM and 128 KB ECC RAM.

Software can configure data RAM array size for OCRAM, DTCM and ITCM by GPR16 and GPR17 in IOMUX. The RAM size step is 32KB.

ECC RAM array can be configured by GPR16 and GPR17 in IOMUX, FLEXRAM_CTRL and two fuse bits (fuse_tcm_ECC_disable and fuse_ocram_ECC_disable).

The allocated RAM bank cannot be overlapped for OCRAM, DTCM and ITCM.

The following table shows an example of a 128KB ITCM, 128KB DTCM and 256KB OCRAM configuration for data RAM.

Table 39-2. FLEXRAM Partitioning Example

	ITCM	DTCM	OCRAM
Bank0	0	0	1
Bank1	0	0	1
Bank2	0	0	1
Bank3	0	0	1
Bank4	1	0	0
Bank5	1	0	0
Bank6	0	1	0
Bank7	0	1	0
Bank8	0	0	1
Bank9	0	0	1
Bank10	0	0	1
Bank11	0	0	1
Bank12	1	0	0
Bank13	1	0	0
Bank14	0	1	0
Bank15	0	1	0

Each RAM bank has 2 bits bank configuration signal (e.g. from IOMUX GPR17).

Table 39-3. RAM Bank Configuration

	RAM Bank Configuration
00	Not used
01	OCRAM
10	DTCM
11	ITCM

The left 128KB RAM, taken as bank16 to bank31, can be used as data RAM for OCRAM or as ECC RAM shared by OCRAM, ITCM and DTCM based on the configuration of FLEXRAM_CTRL, GPR (ram_bank_config) and fuse bits (fuse_tcm_ECC_disable and fuse_ocram_ECC_disable). If the fuse bit is set, then corresponding ECC function would be disabled regardless of the setting in FLEXRAM_CTRL register. While if fuse bit is 0, FLEXRAM_CTRL setting determines whether ECC function is turned on. Generally, there would be 4 scenarios:

- TCM ECC enabled & OCRAM ECC enabled
- TCM ECC enabled & OCRAM ECC disabled
- TCM ECC disabled & OCRAM ECC enabled
- TCM ECC disabled & OCRAM ECC disabled

If both TCM ECC and OCRAM ECC are enabled, then the 128KB would be allocated just similarly as the 512KB data RAM, but with step size 8KB and used as ECC RAM for OCRAM and TCM. The following table shows the corresponding ECC RAM bank allocation in this case: bank16~19 and bank24~27 are allocated as OCRAM ECC RAM, bank20,21,28,29 (32KB) are allocates as ITCM ECC RAM, and bank22,23,30,31 (32KB) are allocated as DTCM ECC RAM.

Table 39-4. FLEXRAM ECC RAM Partitioning Example

	ITCM ECC	DTCM ECC	OCRAM ECC
Bank16	0	0	1
Bank17	0	0	1
Bank18	0	0	1
Bank19	0	0	1
Bank20	1	0	0
Bank21	1	0	0
Bank22	0	1	0
Bank23	0	1	0
Bank24	0	0	1
Bank25	0	0	1
Bank26	0	0	1
Bank27	0	0	1
Bank28	1	0	0
Bank29	1	0	0
Bank30	0	1	0
Bank31	0	1	0

If TCM ECC is enabled and OCRAM ECC is disabled, RAM partition would also be aligned with corresponding data RAM bank. If configured for OCRAM, they would be used as OCRAM data RAM addressed after normal OCRAM data RAM. Take the table above as an example, the RAM partition for the 128KB would be just similar as shown in the table: only bank16~19 and bank24~27 would be used as OCRAM data RAM. This 64K space can be addressed after bank11.

If TCM ECC is disabled and OCRAM ECC is enabled, partition of the 128KB RAM would be decided by which bank of bank0~15 is configured for OCRAM. Take the table above as an example, under such configuration, bank16~19 and bank24~27 would be used as OCRAM ECC RAM, while bank20~23 and bank28~31 cannot be addressed.

If both OCRAM ECC and TCM ECC are disabled, then the whole 128KB RAM would be used for OCRAM data RAM, no matter what value the GPR (ram_bank_config) is configured.

39.3.3 Low power modes

FLEXRAM has input fuse control signal to control OCRAM bank8~15 and bank24~31 interface access. When the fuse bit is programmed, bank8~15 and bank24~31 cannot be accessed again and can be powered down.

39.3.4 Clocks

The following table lists the clock sources of the FlexRAM module.

Table 39-5. FlexRAM Clocks

Clock name	Description
core_clk	Arm core clock
axi_clk	AXI bus clock
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock for register accesses

39.3.5 Reset

FLEXRAM has 2 reset signals. One is used for OCRAM interface and in always-on power domain. The other one is used for TCM interface and in core power domain.

39.3.6 Interrupts

FLEXRAM can generate interrupt in the following cases:

- OCRAM, DTCM or ITCM access the RAM address which is not allocated.
- OCRAM, DTCM or ITCM access the RAM address which is assigned in *_MAGIC_ADDR register.
- ECC error is detected in a read access from OCRAM, DTCM or ITCM. Multi-bit ECC error and single-bit ECC error interrupt would be asserted separately.
- Partial write is launched by OCRAM, and the corresponding ECC_EN is set to 1'b1.

39.3.7 Lookup Table

Syndrome for single-bit ECC can be used to locate the Error bit. For OCRAM or ITCM single-bit ECC Error Information Register, see "Lookup table for 64-bit data ECC syndrome". For D0TCM or D1TCM single-bit ECC Error Information Register, see "Lookup table for 32-bit data ECC syndrome".

Table 39-6. Lookup table for 64-bit data ECC syndrome

Syndrome value	Error bit
8'hb0	63
8'h23	62
8'h70	61
8'h62	60
8'h85	59
8'h13	58
8'h45	57
8'h52	56
8'h2a	55
8'h8a	54
8'h0b	53
8'h0e	52
8'hf8	51
8'h25	50
8'hd9	49
8'ha1	48
8'h54	47
8'ha7	46
8'ha8	45
8'h92	44
8'hc8	43
8'h07	42
8'h34	41
8'h32	40
8'h68	39
8'h89	38
8'h98	37
8'h49	36
8'h61	35
8'h86	34
8'h91	33
8'h46	32

Table continues on the next page...

Table 39-6. Lookup table for 64-bit data ECC syndrome (continued)

Syndrome value	Error bit
8'h58	31
8'h4f	30
8'h38	29
8'h75	28
8'hc4	27
8'h0d	26
8'ha4	25
8'h37	24
8'h64	23
8'h16	22
8'h94	21
8'h29	20
8'hea	19
8'h26	18
8'h1a	17
8'h19	16
8'hd0	15
8'hc2	14
8'h2c	13
8'h51	12
8'he0	11
8'ha2	10
8'h1c	9
8'h31	8
8'h8c	7
8'h4a	6
8'h4c	5
8'h15	4
8'h83	3
8'h9e	2
8'h43	1
8'hc1	0

Table 39-7. Lookup table for 32-bit data ECC syndrome

Syndrome value	Error bit
7'h49	31
7'h0d	30
7'h0e	29

Table continues on the next page...

Table 39-7. Lookup table for 32-bit data ECC syndrome (continued)

Syndrome value	Error bit
7'h38	28
7'h4c	27
7'h1c	26
7'h58	25
7'h0b	24
7'h54	23
7'h15	22
7'h16	21
7'h34	20
7'h25	19
7'h26	18
7'h64	17
7'h2c	16
7'h1a	15
7'h23	14
7'h2a	13
7'h32	12
7'h46	11
7'h4a	10
7'h52	9
7'h62	8
7'h13	7
7'h29	6
7'h31	5
7'h43	4
7'h45	3
7'h19	2
7'h51	1
7'h61	0

39.4 Memory Map and Register Definition

This section includes the FLEXRAM module memory map and detailed descriptions of all registers.

39.4.1 FLEXRAM register descriptions

39.4.1.1 FLEXRAM memory map

FLEXRAM base address: 4002_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	TCM CRTL Register (TCM_CTRL)	32	RW	0000_0000h
4h	OCRAM Magic Address Register (OCRAM_MAGIC_ADDR)	32	RW	0000_0000h
8h	DTCM Magic Address Register (DTCM_MAGIC_ADDR)	32	RW	0000_0000h
Ch	ITCM Magic Address Register (ITCM_MAGIC_ADDR)	32	RW	0000_0000h
10h	Interrupt Status Register (INT_STATUS)	32	RW	0000_0000h
14h	Interrupt Status Enable Register (INT_STAT_EN)	32	RW	0000_0000h
18h	Interrupt Enable Register (INT_SIG_EN)	32	RW	0000_0000h
1Ch	OCRAM single-bit ECC Error Information Register (OCRAM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000h
20h	OCRAM single-bit ECC Error Address Register (OCRAM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000h
24h	OCRAM single-bit ECC Error Data Register (OCRAM_ECC_SINGLE_ERROR_DATA_LSB)	32	R	0000_0000h
28h	OCRAM single-bit ECC Error Data Register (OCRAM_ECC_SINGLE_ERROR_DATA_MSB)	32	R	0000_0000h
2Ch	OCRAM multi-bit ECC Error Information Register (OCRAM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000h
30h	OCRAM multi-bit ECC Error Address Register (OCRAM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000h
34h	OCRAM multi-bit ECC Error Data Register (OCRAM_ECC_MULTI_ERROR_DATA_LSB)	32	R	0000_0000h
38h	OCRAM multi-bit ECC Error Data Register (OCRAM_ECC_MULTI_ERROR_DATA_MSB)	32	R	0000_0000h
3Ch	ITCM single-bit ECC Error Information Register (ITCM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000h
40h	ITCM single-bit ECC Error Address Register (ITCM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000h
44h	ITCM single-bit ECC Error Data Register (ITCM_ECC_SINGLE_ERROR_DATA_LSB)	32	R	0000_0000h
48h	ITCM single-bit ECC Error Data Register (ITCM_ECC_SINGLE_ERROR_DATA_MSB)	32	R	0000_0000h
4Ch	ITCM multi-bit ECC Error Information Register (ITCM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000h
50h	ITCM multi-bit ECC Error Address Register (ITCM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
54h	ITCM multi-bit ECC Error Data Register (ITCM_ECC_MULTI_ERROR_DATA_LSB)	32	R	0000_0000h
58h	ITCM multi-bit ECC Error Data Register (ITCM_ECC_MULTI_ERROR_DATA_MSB)	32	R	0000_0000h
5Ch	D0TCM single-bit ECC Error Information Register (D0TCM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000h
60h	D0TCM single-bit ECC Error Address Register (D0TCM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000h
64h	D0TCM single-bit ECC Error Data Register (D0TCM_ECC_SINGLE_ERROR_DATA)	32	R	0000_0000h
68h	D0TCM multi-bit ECC Error Information Register (D0TCM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000h
6Ch	D0TCM multi-bit ECC Error Address Register (D0TCM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000h
70h	D0TCM multi-bit ECC Error Data Register (D0TCM_ECC_MULTI_ERROR_DATA)	32	R	0000_0000h
74h	D1TCM single-bit ECC Error Information Register (D1TCM_ECC_SINGLE_ERROR_INFO)	32	R	0000_0000h
78h	D1TCM single-bit ECC Error Address Register (D1TCM_ECC_SINGLE_ERROR_ADDR)	32	R	0000_0000h
7Ch	D1TCM single-bit ECC Error Data Register (D1TCM_ECC_SINGLE_ERROR_DATA)	32	R	0000_0000h
80h	D1TCM multi-bit ECC Error Information Register (D1TCM_ECC_MULTI_ERROR_INFO)	32	R	0000_0000h
84h	D1TCM multi-bit ECC Error Address Register (D1TCM_ECC_MULTI_ERROR_ADDR)	32	R	0000_0000h
88h	D1TCM multi-bit ECC Error Data Register (D1TCM_ECC_MULTI_ERROR_DATA)	32	R	0000_0000h
8Ch	OCRAM ECC Error Injection Register (OCRAM_ECC_ERROR_INJEC)	32	RW	0000_0000h
90h	ITCM ECC Error Injection Register (ITCM_ECC_ERROR_INJEC)	32	RW	0000_0000h
94h	D0TCM ECC Error Injection Register (D0TCM_ECC_ERROR_INJEC)	32	RW	0000_0000h
98h	D1TCM ECC Error Injection Register (D1TCM_ECC_ERROR_INJEC)	32	RW	0000_0000h
108h	FlexRAM feature Control register (FLEXRAM_CTRL)	32	RW	0000_0000h
10Ch	OCRAM Pipeline Status register (OCRAM_PIPELINE_STATUS)	32	R	0000_0000h

39.4.1.2 TCM CRTL Register (TCM_CTRL)

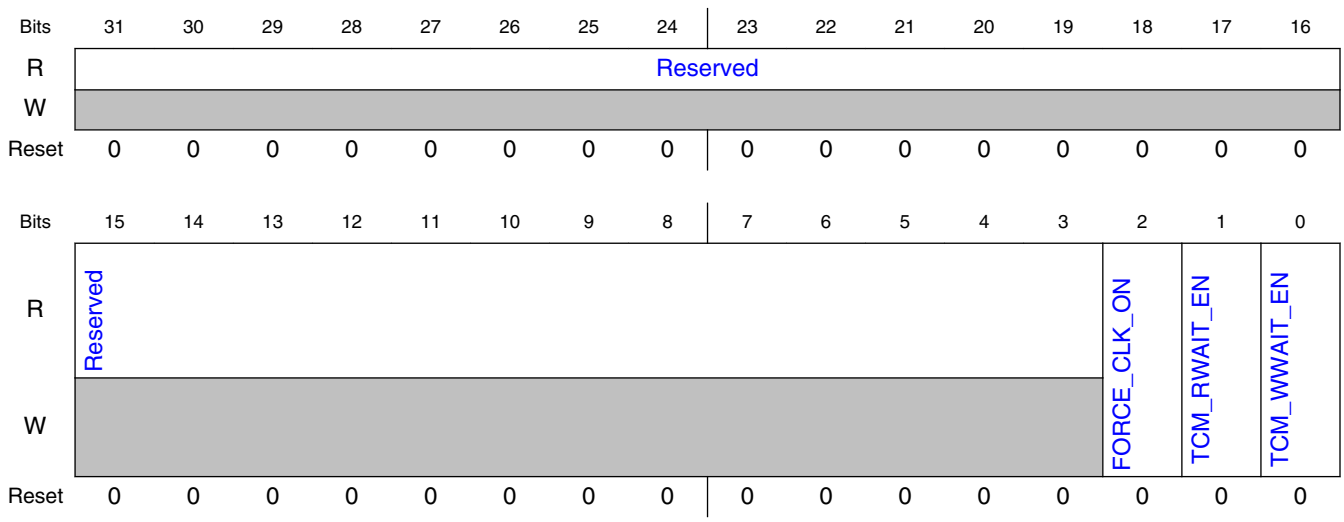
39.4.1.2.1 Offset

Register	Offset
TCM_CTRL	0h

39.4.1.2.2 Function

.

39.4.1.2.3 Diagram



39.4.1.2.4 Fields

Field	Function
31-3 Reserved	Reserved
2 FORCE_CLK_ON	Force RAM Clock Always On
1 TCM_RWAIT_EN	TCM Read Wait Mode Enable 0b - TCM read fast mode: Read RAM accesses are expected to be finished in 1-cycle. 1b - TCM read wait mode: Read RAM accesses are expected to be finished in 2-cycles.
0 TCM_WWAIT_EN	TCM Write Wait Mode Enable 0b - TCM write fast mode: Write RAM accesses are expected to be finished in 1-cycle. 1b - TCM write wait mode: Write RAM accesses are expected to be finished in 2-cycles.

39.4.1.3 OCRAM Magic Address Register (OCRAM_MAGIC_ADDR)

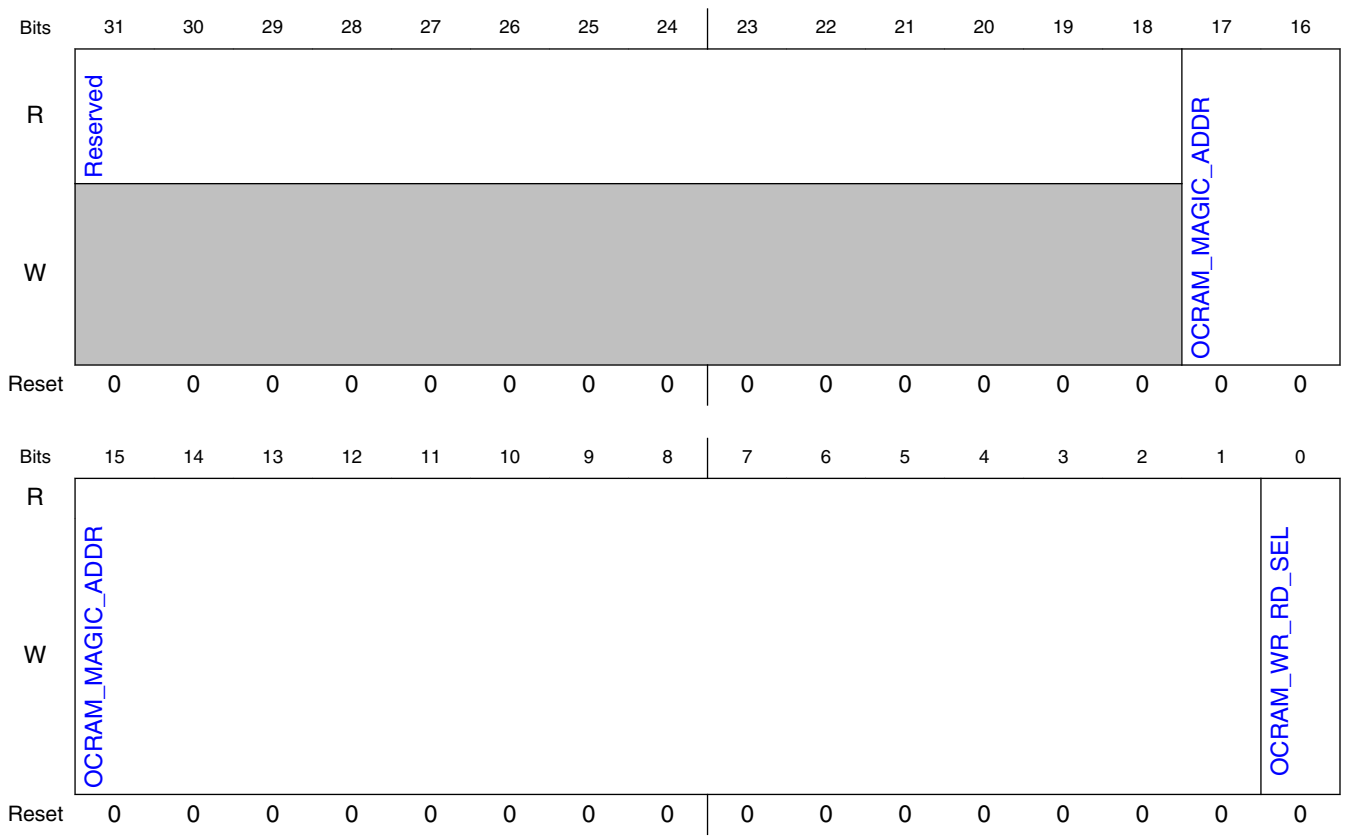
39.4.1.3.1 Offset

Register	Offset
OCRAM_MAGIC_ADDR	4h

39.4.1.3.2 Function

.

39.4.1.3.3 Diagram



39.4.1.3.4 Fields

Field	Function
31-18	Reserved

Table continues on the next page...

Field	Function
Reserved	
17-1 OCRAM_MAGIC_ADDR	OCRAM Magic Address When OCRAM access hits Magic Address, it will generate interrupt if access type is match with ocram_wr_rd_sel.
0 OCRAM_WR_RD_SEL	OCRAM Write Read Select 0b - When OCRAM read access hits magic address, it will generate interrupt. 1b - When OCRAM write access hits magic address, it will generate interrupt.

39.4.1.4 DTCM Magic Address Register (DTCM_MAGIC_ADDR)

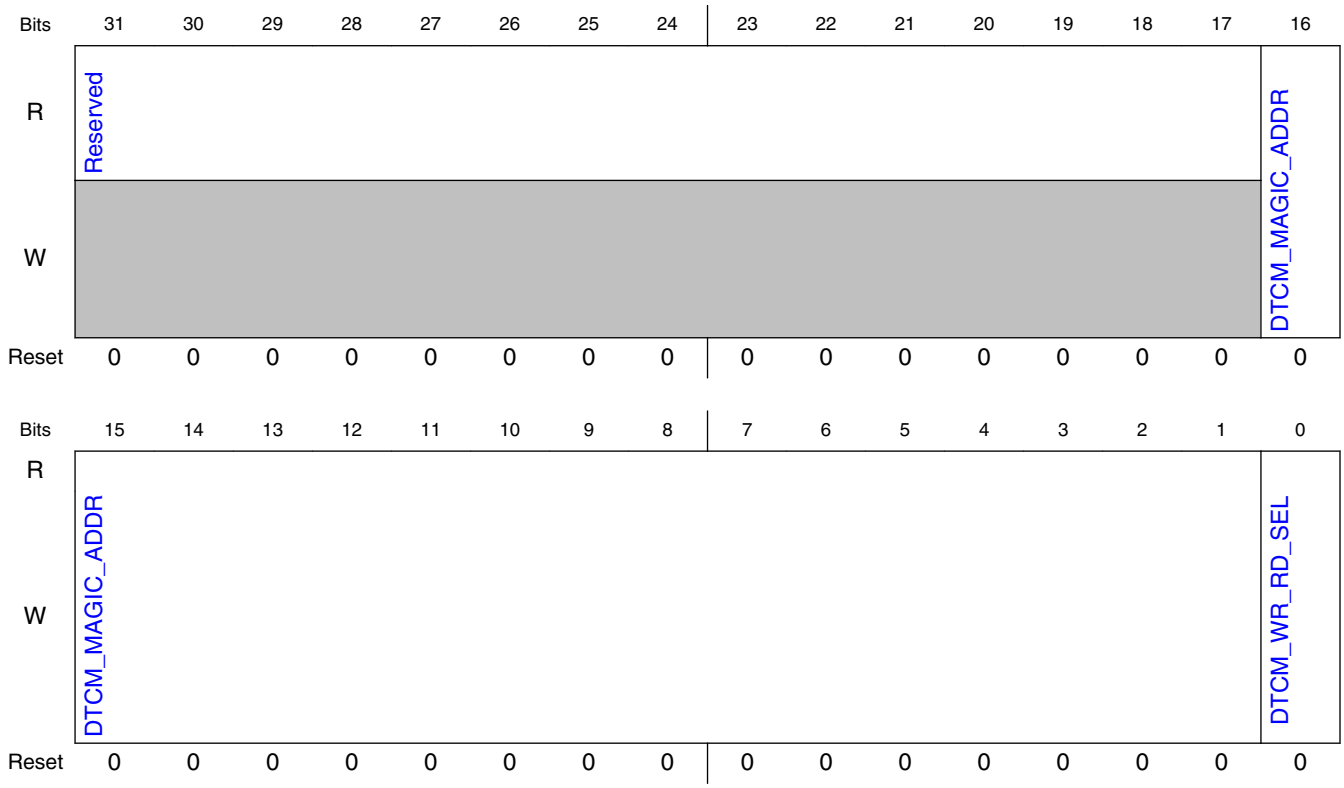
39.4.1.4.1 Offset

Register	Offset
DTCM_MAGIC_ADDR	8h

39.4.1.4.2 Function

.

39.4.1.4.3 Diagram



39.4.1.4.4 Fields

Field	Function
31-17 Reserved	Reserved
16-1 DTCM_MAGIC_ADDR	DTCM Magic Address When DTCM access hits Magic Address, it will generate interrupt if access type is match with dtcm_wr_rd_sel.
0 DTCM_WR_RD_SEL	DTCM Write Read Select 0b - When DTCM read access hits magic address, it will generate interrupt. 1b - When DTCM write access hits magic address, it will generate interrupt.

39.4.1.5 ITCM Magic Address Register (ITCM_MAGIC_ADDR)

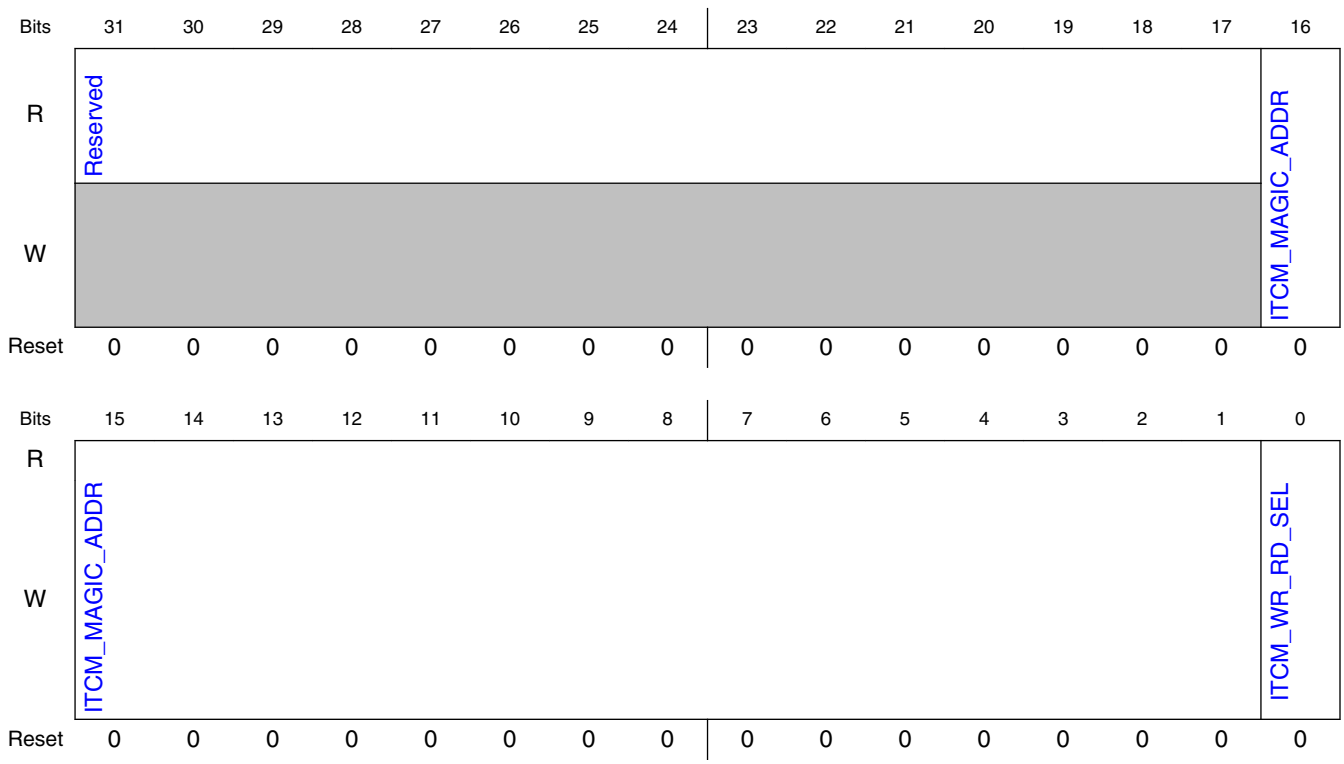
39.4.1.5.1 Offset

Register	Offset
ITCM_MAGIC_ADDR	Ch

39.4.1.5.2 Function

.

39.4.1.5.3 Diagram



39.4.1.5.4 Fields

Field	Function
31-17 Reserved	Reserved
16-1 ITCM_MAGIC_ADDR	ITCM Magic Address When ITCM access hits Magic Address, it will generate interrupt if access type is match with itcm_wr_rd_sel.
0	ITCM Write Read Select 0b - When ITCM read access hits magic address, it will generate interrupt.

Field	Function
ITCM_WR_RD_SEL	1b - When ITCM write access hits magic address, it will generate interrupt.

39.4.1.6 Interrupt Status Register (INT_STATUS)

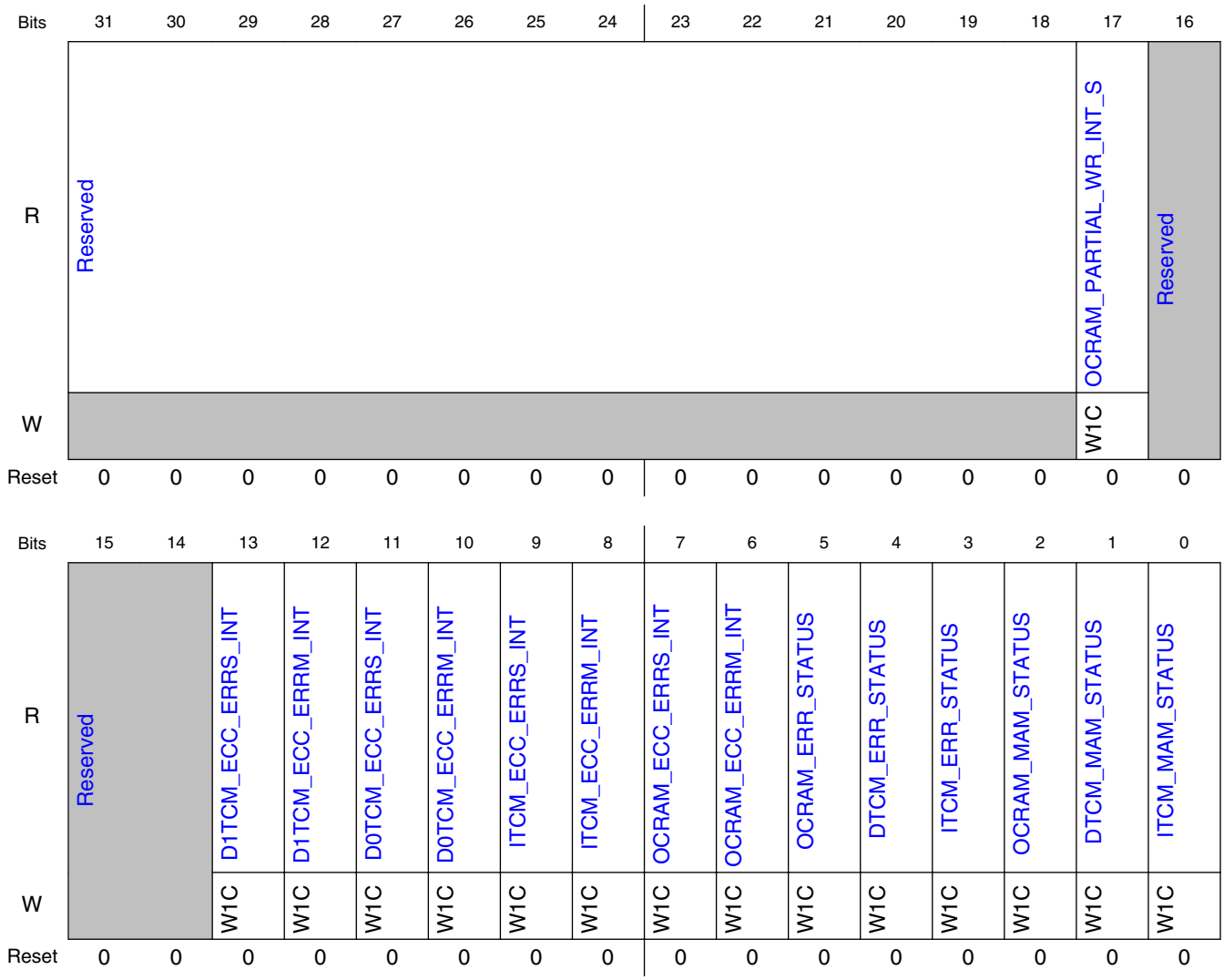
39.4.1.6.1 Offset

Register	Offset
INT_STATUS	10h

39.4.1.6.2 Function

.

39.4.1.6.3 Diagram



39.4.1.6.4 Fields

Field	Function
31-18 Reserved	Reserved
17 OCRAM_PARTIAL_WR_INT_S	OCRAM Partial Write Interrupt Status In OCRAM ECC enable mode,if OCRAM write access was lauched with we!=8'hFF and corresponding status enable bit in INT_STAT_EN register is set, this bit will be set. 0b - OCRAM Partial Write does not happen 1b - OCRAM Partial Write happens.
16-14 —	Reserved
13	D1TCM access single-bit ECC Error Interrupt Status

Table continues on the next page...

Field	Function
D1TCM_ECC_ERRS_INT	When D1TCM read data was detected with single-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - D1TCM single-bit ECC error does not happen 1b - D1TCM single-bit ECC error happens.
12 D1TCM_ECC_ERRM_INT	D1TCM access multi-bit ECC Error Interrupt Status When D1TCM read data was detected with multi-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - D1TCM multi-bit ECC error does not happen 1b - D1TCM multi-bit ECC error happens.
11 D0TCM_ECC_ERRS_INT	D0TCM access single-bit ECC Error Interrupt Status When D0TCM read data was detected with single-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - D0TCM single-bit ECC error does not happen 1b - D0TCM single-bit ECC error happens.
10 D0TCM_ECC_ERRM_INT	D0TCM access multi-bit ECC Error Interrupt Status When D0TCM read data was detected with multi-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - D0TCM multi-bit ECC error does not happen 1b - D0TCM multi-bit ECC error happens.
9 ITCM_ECC_ERRS_INT	ITCM access single-bit ECC Error Interrupt Status When ITCM read data was detected with single-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - ITCM single-bit ECC error does not happen 1b - ITCM single-bit ECC error happens.
8 ITCM_ECC_ERRM_INT	ITCM Access multi-bit ECC Error Interrupt Status When ITCM read data was detected with multi-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - ITCM multi-bit ECC error does not happen 1b - ITCM multi-bit ECC error happens.
7 OCRAM_ECC_ERRS_INT	OCRAM access single-bit ECC Error Interrupt Status When OCRAM read data was detected with single-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - OCRAM single-bit ECC error does not happen 1b - OCRAM single-bit ECC error happens.
6 OCRAM_ECC_ERRM_INT	OCRAM access multi-bit ECC Error Interrupt Status When OCRAM read data was detected with multi-bit ECC error, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - OCRAM multi-bit ECC error does not happen 1b - OCRAM multi-bit ECC error happens.
5 OCRAM_ERR_STATUS	OCRAM Access Error Status When OCRAM access unallocated address, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - OCRAM access error does not happen 1b - OCRAM access error happens.
4 DTCM_ERR_STATUS	DTCM Access Error Status When DTCM access unallocated address, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - DTCM access error does not happen 1b - DTCM access error happens.

Table continues on the next page...

Field	Function
3 ITCM_ERR_STATUS	ITCM Access Error Status When ITCM access unallocated address, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - ITCM access error does not happen 1b - ITCM access error happens.
2 OCRAM_MAM_STATUS	OCRAM Magic Address Match Status When OCRAM access hits Magic Address and access type is also matched, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - OCRAM did not access magic address. 1b - OCRAM accessed magic address.
1 DTCM_MAM_STATUS	DTCM Magic Address Match Status When DTCM access hits Magic Address and access type is also matched, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - DTCM did not access magic address. 1b - DTCM accessed magic address.
0 ITCM_MAM_STATUS	ITCM Magic Address Match Status When ITCM access hits Magic Address and access type is also matched, this bit will be asserted if corresponding status enable bit in INT_STAT_EN register is set. 0b - ITCM did not access magic address. 1b - ITCM accessed magic address.

39.4.1.7 Interrupt Status Enable Register (INT_STAT_EN)

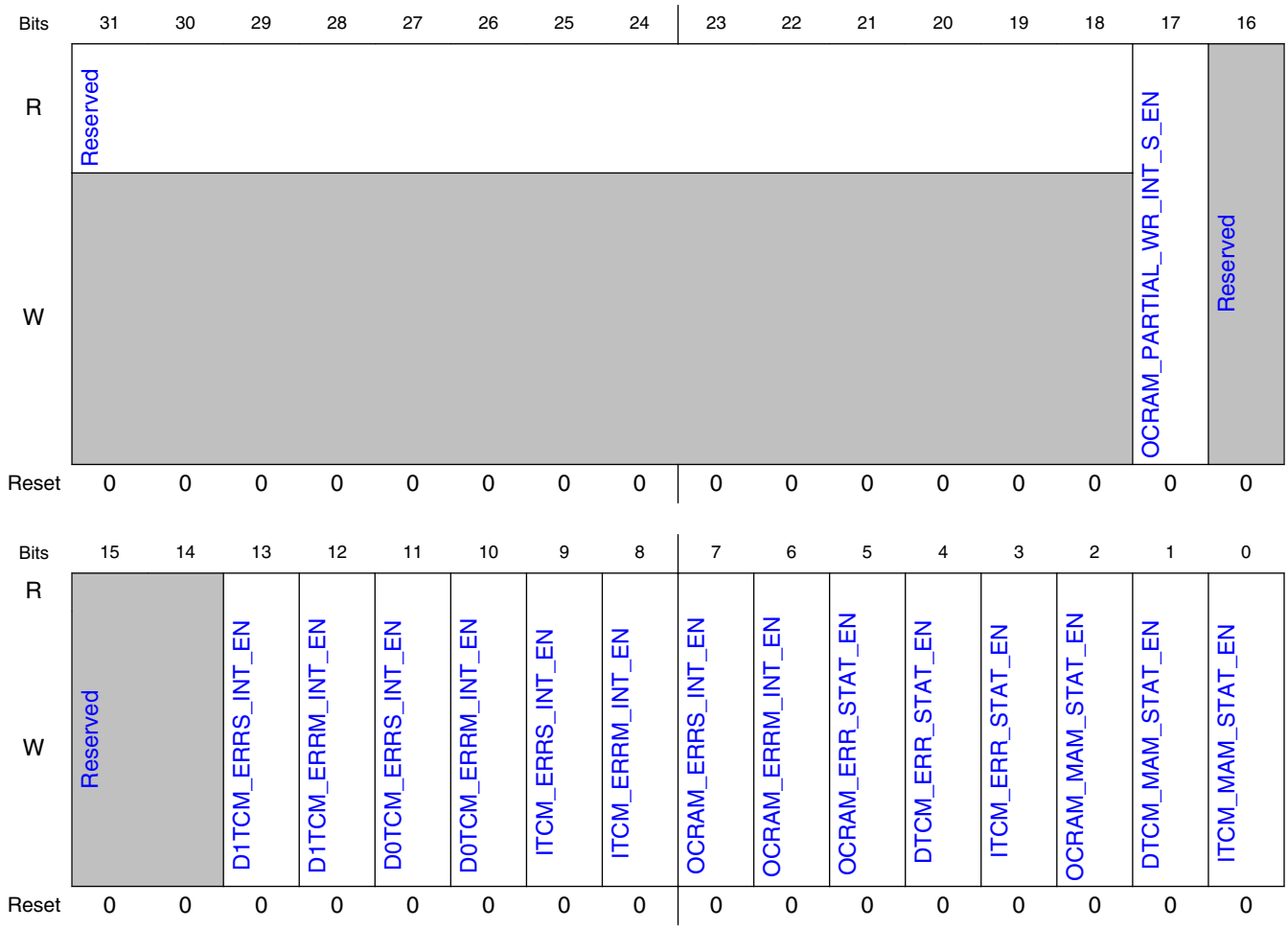
39.4.1.7.1 Offset

Register	Offset
INT_STAT_EN	14h

39.4.1.7.2 Function

Setting the bits in this register to 1 enables the corresponding Interrupt Status to be set by the specified event.

39.4.1.7.3 Diagram



39.4.1.7.4 Fields

Field	Function
31-18 Reserved	Reserved
17 OCRAM_PARTIAL_WR_INT_S_EN	OCRAM Partial Write Interrupt Status Enable 0b - Masked 1b - Enabled
16-14 —	Reserved
13 D1TCM_ERRS_INT_EN	D1TCM Access single-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
12	D1TCM Access multi-bit ECC Error Interrupt Status Enable

Table continues on the next page...

Field	Function
D1TCM_ERRM_INT_EN	0b - Masked 1b - Enabled
11 D0TCM_ERRS_INT_EN	D0TCM Access single-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
10 D0TCM_ERRM_INT_EN	D0TCM Access multi-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
9 ITCM_ERRS_INT_EN	ITCM Access single-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
8 ITCM_ERRM_INT_EN	ITCM Access multi-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
7 OCRAM_ERRS_INT_EN	OCRAM Access single-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
6 OCRAM_ERRM_INT_EN	OCRAM Access multi-bit ECC Error Interrupt Status Enable 0b - Masked 1b - Enabled
5 OCRAM_ERR_STAT_EN	OCRAM Access Error Status Enable 0b - Masked 1b - Enabled
4 DTCM_ERR_STAT_EN	DTCM Access Error Status Enable 0b - Masked 1b - Enabled
3 ITCM_ERR_STAT_EN	ITCM Access Error Status Enable 0b - Masked 1b - Enabled
2 OCRAM_MAM_STAT_EN	OCRAM Magic Address Match Status Enable 0b - Masked 1b - Enabled
1 DTCM_MAM_STAT_EN	DTCM Magic Address Match Status Enable 0b - Masked 1b - Enabled
0 ITCM_MAM_STAT_EN	ITCM Magic Address Match Status Enable 0b - Masked 1b - Enabled

39.4.1.8 Interrupt Enable Register (INT_SIG_EN)

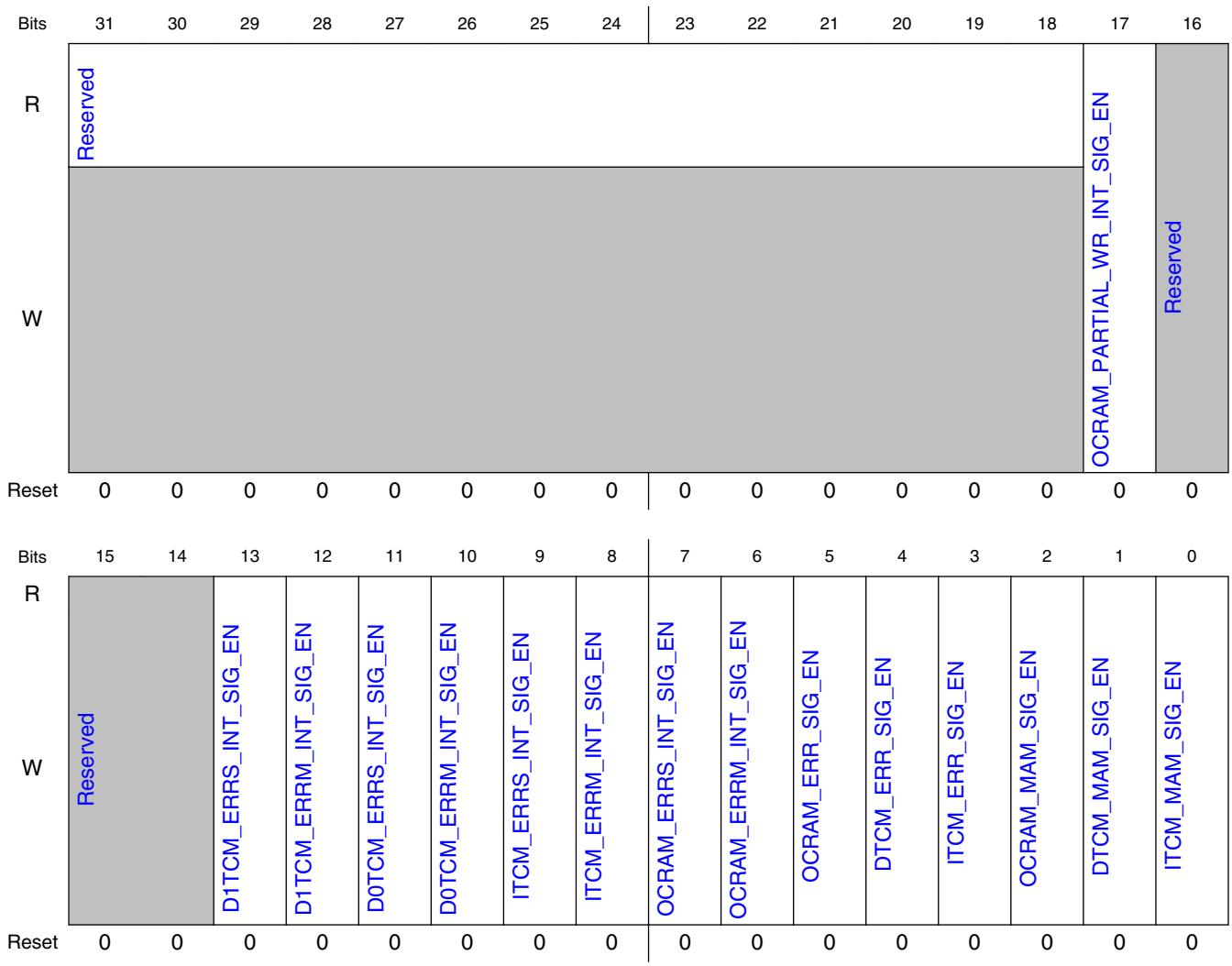
39.4.1.8.1 Offset

Register	Offset
INT_SIG_EN	18h

39.4.1.8.2 Function

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

39.4.1.8.3 Diagram



39.4.1.8.4 Fields

Field	Function
31-18 Reserved	Reserved
17 OCRAM_PARTIAL_WR_INT_SIG_EN	OCRAM Partial Write Interrupt Signal Enable 0b - Masked 1b - Enabled
16-14 —	Reserved
13 D1TCM_ERRS_INT_SIG_EN	D1TCM Access single-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
12 D1TCM_ERRM_INT_SIG_EN	D1TCM Access multi-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
11 D0TCM_ERRS_INT_SIG_EN	D0TCM Access single-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
10 D0TCM_ERRM_INT_SIG_EN	D0TCM Access multi-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
9 ITCM_ERRS_INT_SIG_EN	ITCM Access single-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
8 ITCM_ERRM_INT_SIG_EN	ITCM Access multi-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
7 OCRAM_ERRS_INT_SIG_EN	OCRAM Access single-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
6 OCRAM_ERRM_INT_SIG_EN	OCRAM Access multi-bit ECC Error Interrupt Signal Enable 0b - Masked 1b - Enabled
5 OCRAM_ERR_SIG_EN	OCRAM Access Error Interrupt Enable 0b - Masked 1b - Enabled
4 DTCM_ERR_SIG_EN	DTCM Access Error Interrupt Enable 0b - Masked 1b - Enabled
3 ITCM_ERR_SIG_EN	ITCM Access Error Interrupt Enable 0b - Masked 1b - Enabled
2	OCRAM Magic Address Match Interrupt Enable

Table continues on the next page...

Field	Function
OCRAM_MAM_SIG_EN	0b - Masked 1b - Enabled
1 DTCM_MAM_SIG_EN	DTCM Magic Address Match Interrupt Enable 0b - Masked 1b - Enabled
0 ITCM_MAM_SIG_EN	ITCM Magic Address Match Interrupt Enable 0b - Masked 1b - Enabled

39.4.1.9 OCRAM single-bit ECC Error Information Register (OCRAM_ECC_SINGLE_ERROR_INFO)

39.4.1.9.1 Offset

Register	Offset
OCRAM_ECC_SINGLE_ERROR_INFO	1Ch

39.4.1.9.2 Function

This register is used to store detail error information for corresponding OCRAM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCS_ERRED_SYN								OCRAM_ECCS_ERRED_ECC							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.9.4 Fields

Field	Function
31-16 Reserved	Reserved
15-8 OCRAM_ECCS_ERRED_SYN	corresponding ECC syndrome of OCRAM single-bit ECC error ECC syndrome can be used to locate the Error bit using a look-up table (see "Lookup table for 64-bit data ECC syndrome")
7-0 OCRAM_ECCS_ERRED_ECC	corresponding ECC cipher of OCRAM single-bit ECC error

39.4.1.10 OCRAM single-bit ECC Error Address Register (OCRAM_ECC_SINGLE_ERROR_ADDR)

39.4.1.10.1 Offset

Register	Offset
OCRAM_ECC_SINGLE_ERROR_ADDR	20h

39.4.1.10.2 Function

This register is used to store the errored address for corresponding OCRAM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCRAM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.10.4 Fields

Field	Function
31-0 OCRAM_ECCS_ERRED_ADDR	OCRAM single-bit ECC error address

39.4.1.11 OCRAM single-bit ECC Error Data Register (OCRAM_ECC_SINGLE_ERROR_DATA_LSB)

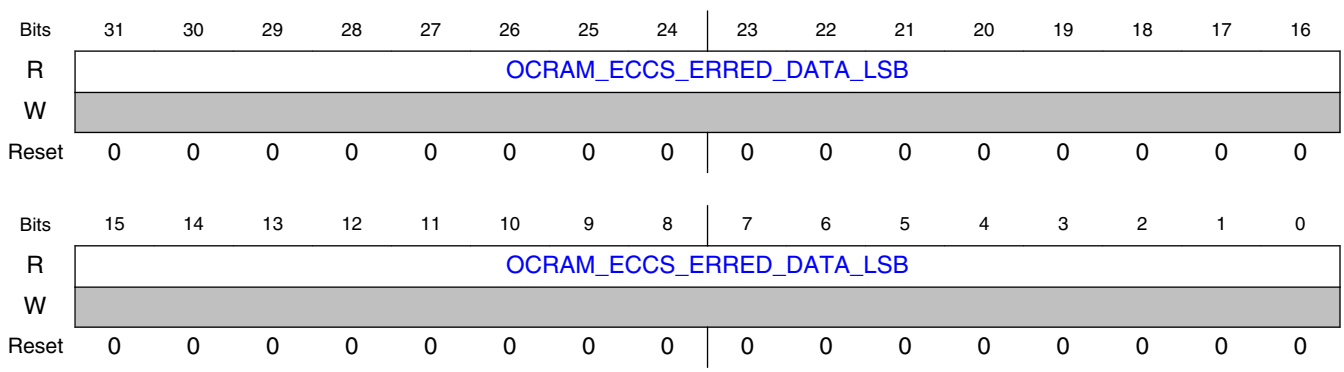
39.4.1.11.1 Offset

Register	Offset
OCRAM_ECC_SINGLE_ERROR_DATA_LSB	24h

39.4.1.11.2 Function

This register is used to store LSB of errored data for corresponding OCRAM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.11.3 Diagram



39.4.1.11.4 Fields

Field	Function
31-0	OCRAM single-bit ECC error data [31:0]

Field	Function
OCRAM_ECCS_ERRED_DATA_LSB	

39.4.1.12 OCRAM single-bit ECC Error Data Register (OCRAM_ECC_SINGLE_ERROR_DATA_MSB)

39.4.1.12.1 Offset

Register	Offset
OCRAM_ECC_SINGLE_ERROR_DATA_MSB	28h

39.4.1.12.2 Function

This register is used to store MSB of errored data for corresponding OCRAM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCRAM_ECCS_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCS_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.12.4 Fields

Field	Function
31-0 OCRAM_ECCS_ERRED_DATA_MSB	OCRAM single-bit ECC error data [63:32]

39.4.1.13 OCRAM multi-bit ECC Error Information Register (OCRAM_ECC_MULTI_ERROR_INFO)

39.4.1.13.1 Offset

Register	Offset
OCRAM_ECC_MULTI_ERROR_INFO	2Ch

39.4.1.13.2 Function

This register is used to store detail error information for corresponding OCRAM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								OCRAM_ECCM_ERRED_ECC							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.13.4 Fields

Field	Function
31-8 Reserved	Reserved
7-0 OCRAM_ECCM_ERRED_ECC	OCRAM multi-bit ECC error corresponding ECC value

39.4.1.14 OCRAM multi-bit ECC Error Address Register (OCRAM_ECC_MULTI_ERROR_ADDR)

39.4.1.14.1 Offset

Register	Offset
OCRAM_ECC_MULTI_ERROR_ADDR	30h

39.4.1.14.2 Function

This register is used to store the errored address for corresponding OCRAM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCRAM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.14.4 Fields

Field	Function
31-0 OCRAM_ECCM_ERRED_ADDR	OCRAM multi-bit ECC error address

39.4.1.15 OCRAM multi-bit ECC Error Data Register (OCRAM_ECC_MULTI_ERROR_DATA_LSB)

39.4.1.15.1 Offset

Register	Offset
OCRAM_ECC_MULTI_ERROR_DATA_LSB	34h

39.4.1.15.2 Function

This register is used to store LSB of errored data for corresponding OCRAM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCRAM_ECCM_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCM_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.15.4 Fields

Field	Function
31-0 OCRAM_ECCM_ERRED_DATA_LSB	OCRAM multi-bit ECC error data [31:0]

39.4.1.16 OCRAM multi-bit ECC Error Data Register (OCRAM_ECC_MULTI_ERROR_DATA_MSB)

39.4.1.16.1 Offset

Register	Offset
OCRAM_ECC_MULTI_ERROR_DATA_MSB	38h

39.4.1.16.2 Function

This register is used to store MSB of errored data for corresponding OCRAM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OCRAM_ECCM_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OCRAM_ECCM_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.16.4 Fields

Field	Function
31-0 OCRAM_ECCM_ERRED_DATA_MSB	OCRAM multi-bit ECC error data [63:32]

39.4.1.17 ITCM single-bit ECC Error Information Register (ITCM_ECC_SINGLE_ERROR_INFO)

39.4.1.17.1 Offset

Register	Offset
ITCM_ECC_SINGLE_ERROR_INFO	3Ch

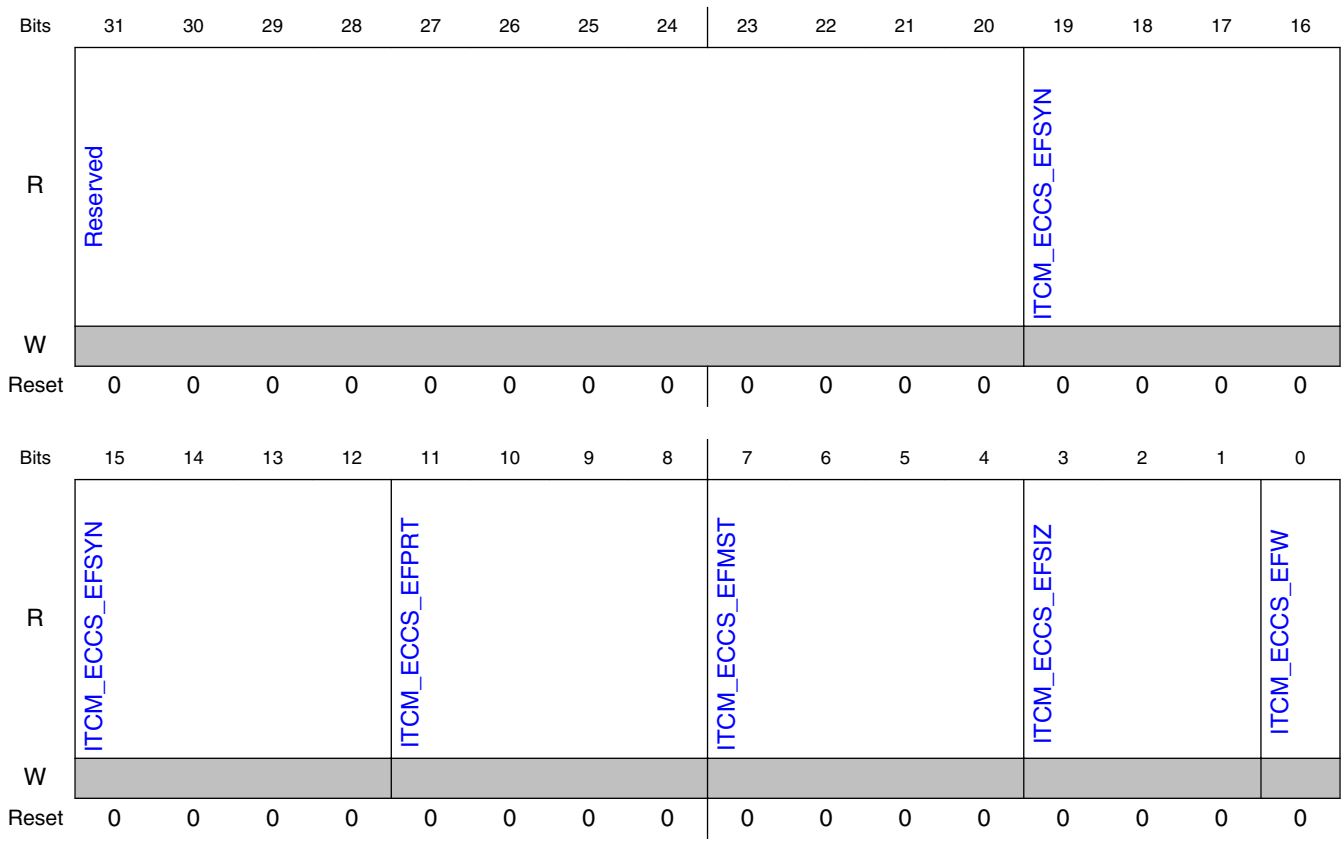
39.4.1.17.2 Function

This register is used to store detail error information for corresponding ITCM single-bit ECC interrupt if the interrupt is enabled.

NOTE

For TCM_PRIV, TCM_WR and TCM_MASTER, they are interface signals defined by Arm TCM bus protocol. ITCM, D0TCM and D1TCM have their respective sets of these signals.

39.4.1.17.3 Diagram



39.4.1.17.4 Fields

Field	Function
31-20 Reserved	Reserved
19-12 ITCM_ECCS_EFSYN	ITCM single-bit ECC error corresponding syndrome ECC syndrome can be used to locate the Error bit using a look-up table (see "Lookup table for 64-bit data ECC syndrome")
11-8 ITCM_ECCS_EFPRT	ITCM single-bit ECC error corresponding TCM_PRIV. TCM_PRIV is to tell the privilege level of access.
7-4	ITCM single-bit ECC error corresponding TCM_MASTER.

Table continues on the next page...

Field	Function
ITCM_ECCS_EFMST	TCM_MASTER is to tell the requester of the current access. For TCM_MASTER, the following list shows the encoding of the access requestor. • 4'b0000 Instruction fetch • 4'b0001 Data access • 4'b0010 Vector fetch on automated exception entry • 4'b0011 AHB slave access • 4'b0100 Debugger access • 4'b0101 MBIST access • 4'b1001 Software data access from sequence queue • 4'b1011 AHB slave access from sequence queue • 4'b1100 Debugger access from sequence queue
3-1 ITCM_ECCS_EFSIZ	ITCM single-bit ECC error corresponding TCM size For ITCM, this value should be 3 (64bit).
0 ITCM_ECCS_EFW	ITCM single-bit ECC error corresponding TCM_WR value. TCM_WR is to tell whether it is a write access(1'b1) or a read access(1'b0).

39.4.1.18 ITCM single-bit ECC Error Address Register (ITCM_ECC_SINGLE_ERROR_ADDR)

39.4.1.18.1 Offset

Register	Offset
ITCM_ECC_SINGLE_ERROR_ADDR	40h

39.4.1.18.2 Function

This register is used to store the errored address for corresponding ITCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ITCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.18.4 Fields

Field	Function
31-0 ITCM_ECCS_ERRED_ADDR	ITCM single-bit ECC error address

39.4.1.19 ITCM single-bit ECC Error Data Register (ITCM_ECC_SINGLE_ERROR_DATA_LSB)

39.4.1.19.1 Offset

Register	Offset
ITCM_ECC_SINGLE_ERROR_DATA_LSB	44h

39.4.1.19.2 Function

This register is used to store LSB of errored data for corresponding ITCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITCM_ECCS_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ITCM_ECCS_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.19.4 Fields

Field	Function
31-0 ITCM_ECCS_ERRED_DATA_LSB	ITCM single-bit ECC error data [31:0]

39.4.1.20 ITCM single-bit ECC Error Data Register (ITCM_ECC_SINGLE_ERROR_DATA_MSB)

39.4.1.20.1 Offset

Register	Offset
ITCM_ECC_SINGLE_ERROR_DATA_MSB	48h

39.4.1.20.2 Function

This register is used to store MSB of errored data for corresponding ITCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITCM_ECCS_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ITCM_ECCS_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.20.4 Fields

Field	Function
31-0 ITCM_ECCS_ERRED_DATA_MSB	ITCM single-bit ECC error data [63:32]

39.4.1.21 ITCM multi-bit ECC Error Information Register (ITCM_ECC_MULTI_ERROR_INFO)

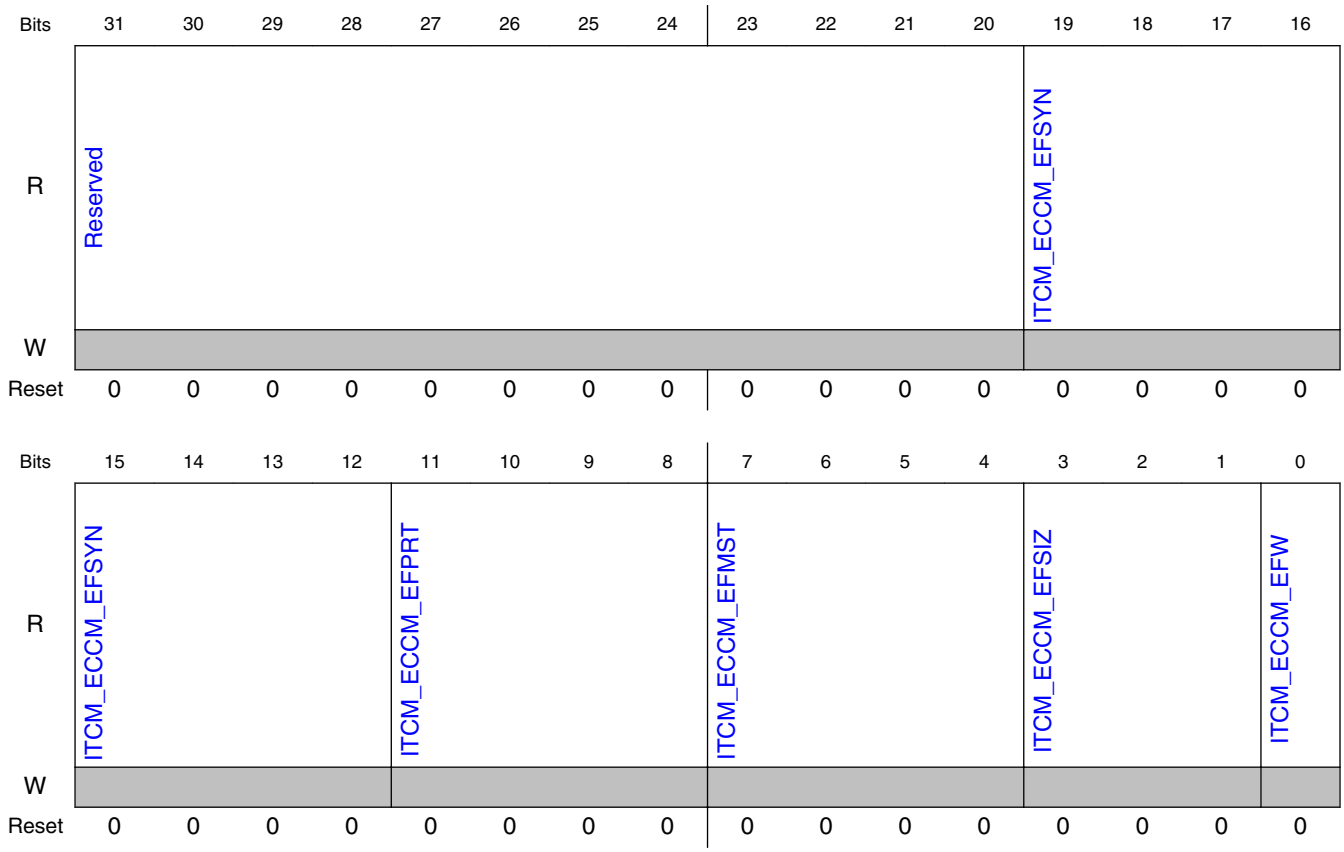
39.4.1.21.1 Offset

Register	Offset
ITCM_ECC_MULTI_ERROR_INFO	4Ch

39.4.1.21.2 Function

This register is used to store detail error information for corresponding ITCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.21.3 Diagram



39.4.1.21.4 Fields

Field	Function
31-20 Reserved	Reserved
19-12 ITCM_ECCM_EFSYN	ITCM multi-bit ECC error corresponding syndrome Be noted that syndrome for multi-bit ECC cannot be used to locate the Error bit
11-8 ITCM_ECCM_EFPRT	ITCM multi-bit ECC error corresponding TCM_PRIV TCM_PRIV is to tell the privilege level of access.
7-4 ITCM_ECCM_EFMST	ITCM multi-bit ECC error corresponding TCM_MASTER TCM_MASTER is to tell the requester of the current access. See the list in ITCM_ECCS_EFMST for the encoding of the access requestor.
3-1 ITCM_ECCM_EFSIZ	ITCM multi-bit ECC error corresponding tcm access size For ITCM, this value should be 3 (64bit).
0	ITCM multi-bit ECC error corresponding TCM_WR value

Field	Function
ITCM_ECCM_EFW	TCM_WR is to tell whether it is a write access(1'b1) or a read access(1'b0).

39.4.1.22 ITCM multi-bit ECC Error Address Register (ITCM_ECC_MULTI_ERROR_ADDR)

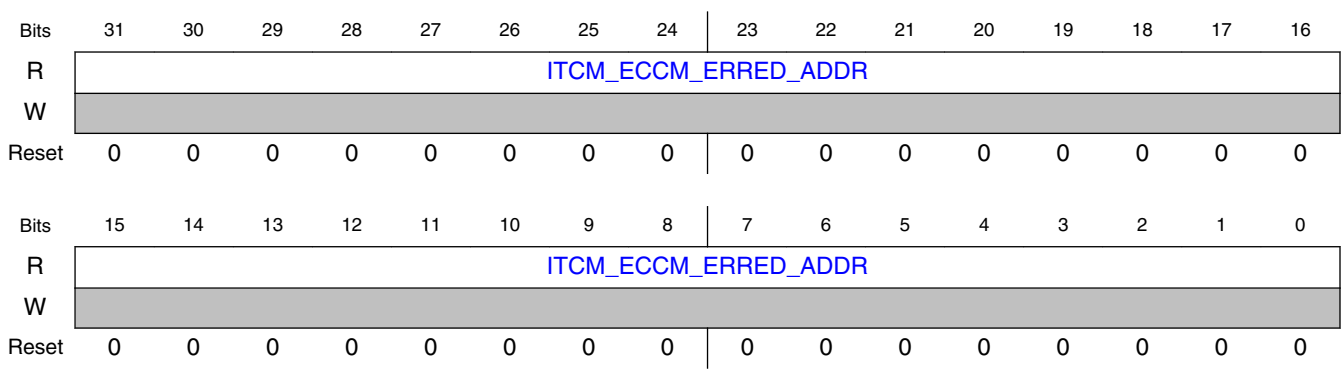
39.4.1.22.1 Offset

Register	Offset
ITCM_ECC_MULTI_ERROR_ADDR	50h

39.4.1.22.2 Function

This register is used to store the errored address for corresponding ITCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.22.3 Diagram



39.4.1.22.4 Fields

Field	Function
31-0 ITCM_ECCM_ERRED_ADDR	ITCM multi-bit ECC error address

39.4.1.23 ITCM multi-bit ECC Error Data Register (ITCM_ECC_MULTI_ERROR_DATA_LSB)

39.4.1.23.1 Offset

Register	Offset
ITCM_ECC_MULTI_ERROR_DATA_LSB	54h

39.4.1.23.2 Function

This register is used to store LSB of errored data for corresponding ITCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITCM_ECCM_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ITCM_ECCM_ERRED_DATA_LSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.23.4 Fields

Field	Function
31-0 ITCM_ECCM_ERRED_DATA_LSB	ITCM multi-bit ECC error data [31:0]

39.4.1.24 ITCM multi-bit ECC Error Data Register (ITCM_ECC_MULTI_ERROR_DATA_MSB)

39.4.1.24.1 Offset

Register	Offset
ITCM_ECC_MULTI_ERROR_DATA_MSB	58h

39.4.1.24.2 Function

This register is used to store MSB of errored data for corresponding ITCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITCM_ECCM_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ITCM_ECCM_ERRED_DATA_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.24.4 Fields

Field	Function
31-0 ITCM_ECCM_ERRED_DATA_MSB	ITCM multi-bit ECC error data [63:32]

39.4.1.25 D0TCM single-bit ECC Error Information Register (D0TCM_ECC_SINGLE_ERROR_INFO)

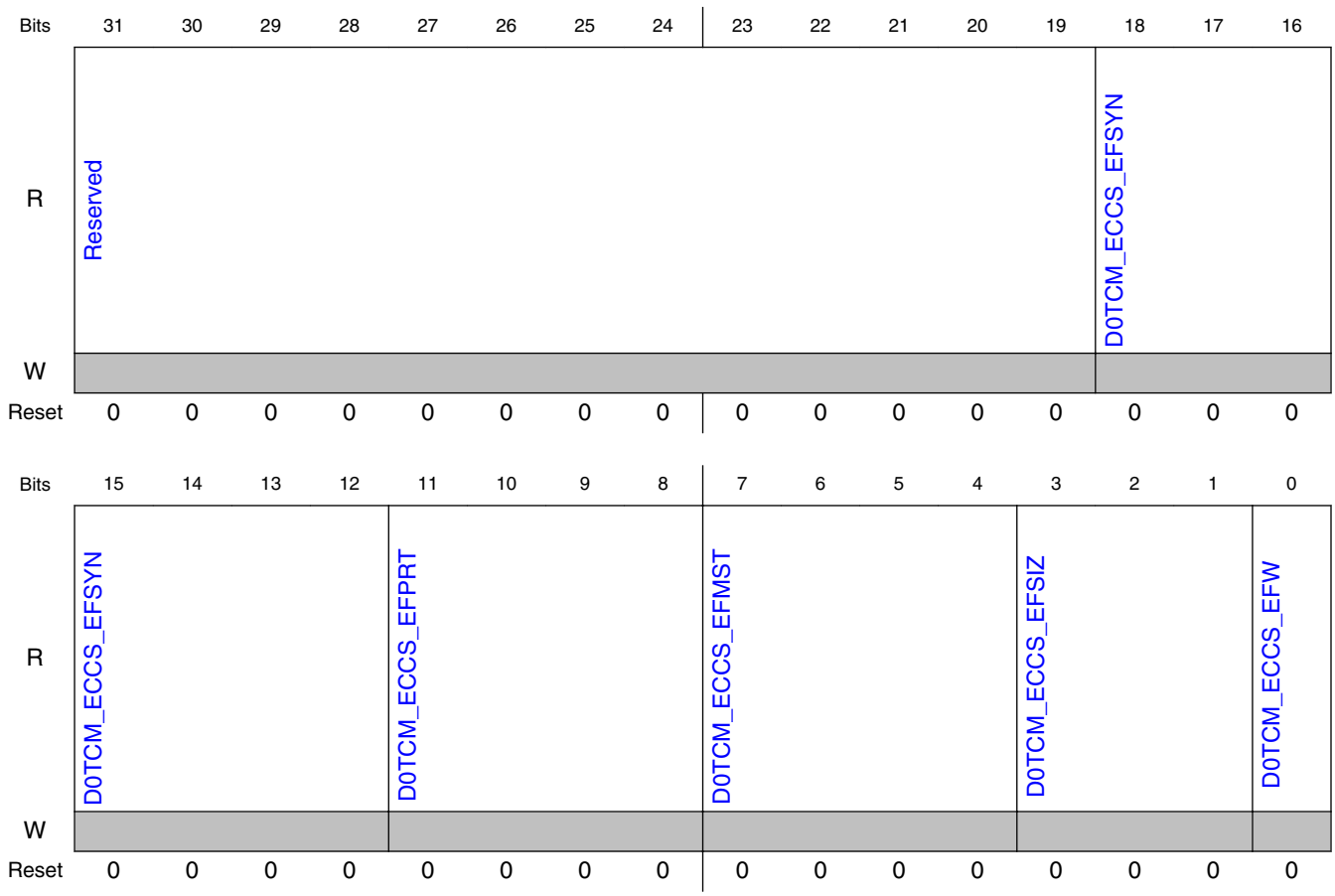
39.4.1.25.1 Offset

Register	Offset
D0TCM_ECC_SINGLE_ERROR_INFO	5Ch

39.4.1.25.2 Function

This register is used to store detail error information for corresponding D0TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.25.3 Diagram



39.4.1.25.4 Fields

Field	Function
31-19	Reserved

Table continues on the next page...

Field	Function
Reserved	
18-12 D0TCM_ECCS_EFSYN	D0TCM single-bit ECC error corresponding syndrome ECC syndrome can be used to locate the Error bit using a look-up table (see "Lookup table for 32-bit data ECC syndrome")
11-8 D0TCM_ECCS_EFPRT	D0TCM single-bit ECC error corresponding TCM_PRIV TCM_PRIV is to tell the privilege level of access.
7-4 D0TCM_ECCS_EFMST	D0TCM single-bit ECC error corresponding TCM_MASTER TCM_MASTER is to tell the requester of the current access. See the list in ITCM_ECCS_EFMST for the encoding of the access requestor.
3-1 D0TCM_ECCS_EFSIZ	D0TCM single-bit ECC error corresponding tcm access size For D0TCM, this value should be 2 (32bit).
0 D0TCM_ECCS_EFW	D0TCM single-bit ECC error corresponding TCM_WR value TCM_WR is to tell whether it is a write access(1’b1) or a read access(1’b0).

39.4.1.26 D0TCM single-bit ECC Error Address Register (D0TCM_ECC_SINGLE_ERROR_ADDR)

39.4.1.26.1 Offset

Register	Offset
D0TCM_ECC_SINGLE_ERROR_ADDR	60h

39.4.1.26.2 Function

This register is used to store the errored address for corresponding D0TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D0TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D0TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.26.4 Fields

Field	Function
31-0 D0TCM_ECCS_ERRED_ADDR	D0TCM single-bit ECC error address

39.4.1.27 D0TCM single-bit ECC Error Data Register (D0TCM_ECC_SINGLE_ERROR_DATA)

39.4.1.27.1 Offset

Register	Offset
D0TCM_ECC_SINGLE_ERROR_DATA	64h

39.4.1.27.2 Function

This register is used to store errored data for corresponding D0TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D0TCM_ECCS_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D0TCM_ECCS_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.27.4 Fields

Field	Function
31-0 D0TCM_ECCS_ERRED_DATA	D0TCM single-bit ECC error data

39.4.1.28 D0TCM multi-bit ECC Error Information Register (D0TCM_ECC_MULTI_ERROR_INFO)

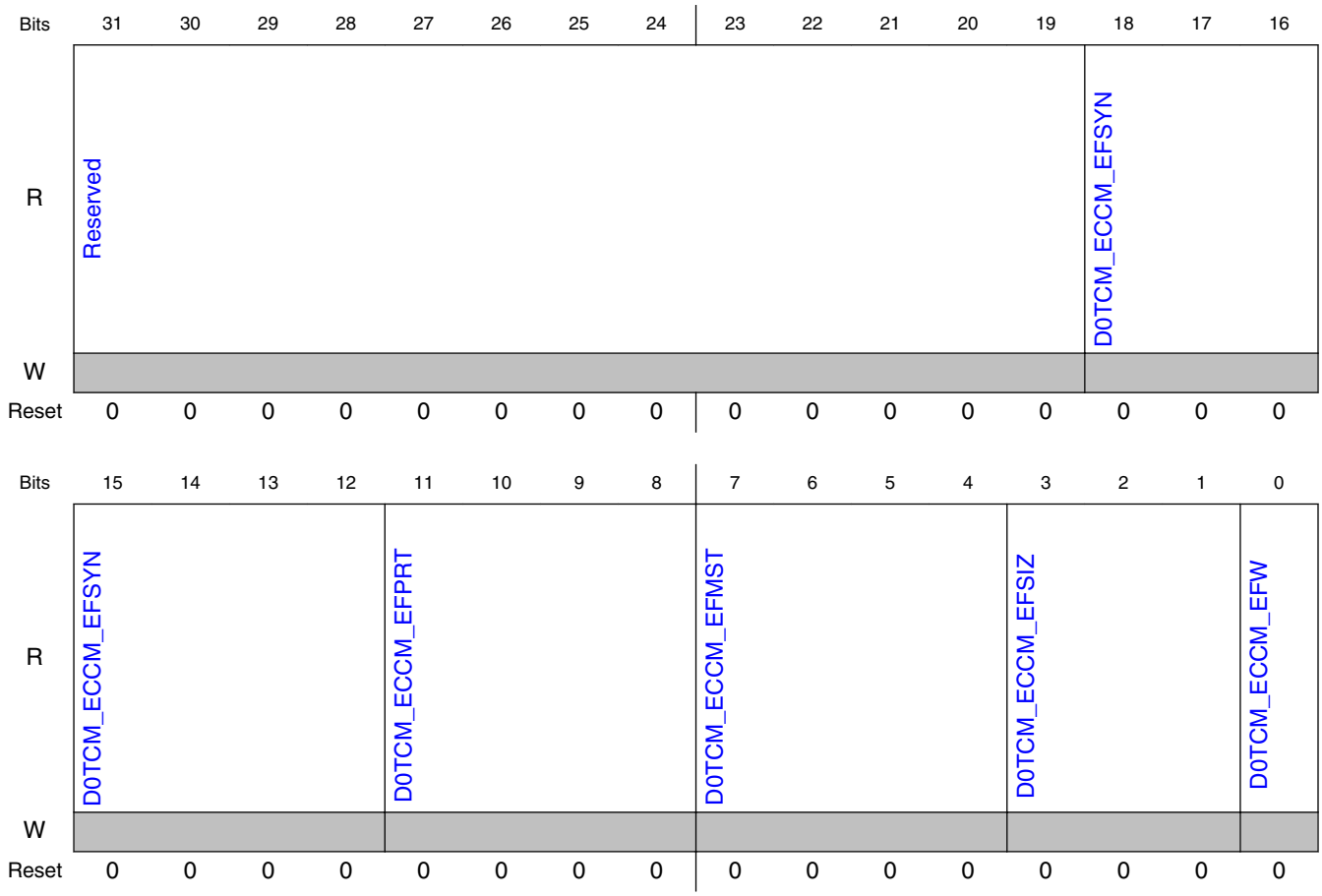
39.4.1.28.1 Offset

Register	Offset
D0TCM_ECC_MULTI_ERROR_INFO	68h

39.4.1.28.2 Function

This register is used to store detail error information for corresponding D0TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.28.3 Diagram



39.4.1.28.4 Fields

Field	Function
31-19 Reserved	Reserved
18-12 D0TCM_ECCM_EFSYN	D0TCM multi-bit ECC error corresponding syndrome Be noted that syndrome for multi-bit ECC cannot be used to locate the Error bit
11-8 D0TCM_ECCM_EFPRT	D0TCM multi-bit ECC error corresponding TCM_PRIV TCM_PRIV is to tell the privilege level of access.
7-4 D0TCM_ECCM_EFMST	D0TCM multi-bit ECC error corresponding TCM_MASTER TCM_MASTER is to tell the requester of the current access. See the list in ITCM_ECCS_EFMST for the encoding of the access requestor.
3-1 D0TCM_ECCM_EFSIZ	D0TCM multi-bit ECC error corresponding tcm access size For D0TCM, this value should be 2 (32bit).

Table continues on the next page...

Field	Function
0 D0TCM_ECCM_EFW	D0TCM multi-bit ECC error corresponding TCM_WR value TCM_WR is to tell whether it is a write access(1'b1) or a read access(1'b0).

39.4.1.29 D0TCM multi-bit ECC Error Address Register (D0TCM_ECC_MULTI_ERROR_ADDR)

39.4.1.29.1 Offset

Register	Offset
D0TCM_ECC_MULTI_ERROR_ADDR	6Ch

39.4.1.29.2 Function

This register is used to store the errored address for corresponding D0TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D0TCM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D0TCM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.29.4 Fields

Field	Function
31-0 D0TCM_ECCM_ERRED_ADDR	D0TCM multi-bit ECC error address

39.4.1.30 D0TCM multi-bit ECC Error Data Register (D0TCM_ECC_MULTI_ERROR_DATA)

39.4.1.30.1 Offset

Register	Offset
D0TCM_ECC_MULTI_ERROR_DATA	70h

39.4.1.30.2 Function

This register is used to store errored data for corresponding D0TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D0TCM_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D0TCM_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.30.4 Fields

Field	Function
31-0 D0TCM_ECCM_ERRED_DATA	D0TCM multi-bit ECC error data

39.4.1.31 D1TCM single-bit ECC Error Information Register (D1TCM_ECC_SINGLE_ERROR_INFO)

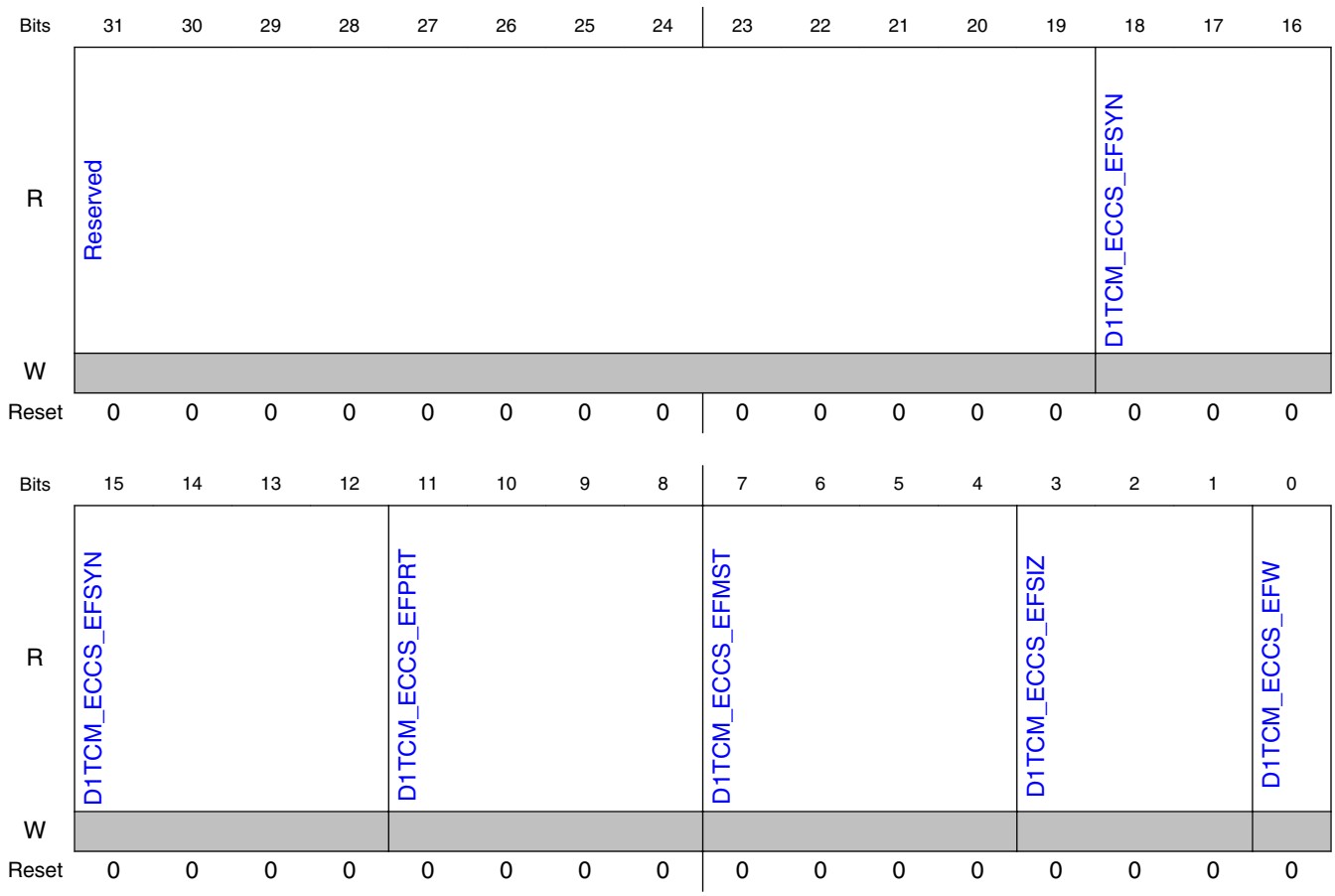
39.4.1.31.1 Offset

Register	Offset
D1TCM_ECC_SINGLE_ERROR_INFO	74h

39.4.1.31.2 Function

This register is used to store detail error information for corresponding D1TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.31.3 Diagram



39.4.1.31.4 Fields

Field	Function
31-19	Reserved

Table continues on the next page...

Field	Function
Reserved	
18-12 D1TCM_ECCS_EFSYN	D1TCM single-bit ECC error corresponding syndrome ECC syndrome can be used to locate the Error bit using a look-up table (see "Lookup table for 32-bit data ECC syndrome")
11-8 D1TCM_ECCS_EFPRT	D1TCM single-bit ECC error corresponding TCM_PRIV TCM_PRIV is to tell the privilege level of access.
7-4 D1TCM_ECCS_EFMST	D1TCM single-bit ECC error corresponding TCM_MASTER TCM_MASTER is to tell the requester of the current access. See the list in ITCM_ECCS_EFMST for the encoding of the access requestor.
3-1 D1TCM_ECCS_EFSIZ	D1TCM single-bit ECC error corresponding tcm access size For D1TCM, this value should be 2 (32bit).
0 D1TCM_ECCS_EFW	D1TCM single-bit ECC error corresponding TCM_WR value TCM_WR is to tell whether it is a write access(1’b1) or a read access(1’b0).

39.4.1.32 D1TCM single-bit ECC Error Address Register (D1TCM_ECC_SINGLE_ERROR_ADDR)

39.4.1.32.1 Offset

Register	Offset
D1TCM_ECC_SINGLE_ERROR_ADDR	78h

39.4.1.32.2 Function

This register is used to store the errored address for corresponding D1TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D1TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D1TCM_ECCS_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.32.4 Fields

Field	Function
31-0 D1TCM_ECCS_ERRED_ADDR	D1TCM single-bit ECC error address

39.4.1.33 D1TCM single-bit ECC Error Data Register (D1TCM_ECC_SINGLE_ERROR_DATA)

39.4.1.33.1 Offset

Register	Offset
D1TCM_ECC_SINGLE_ERROR_DATA	7Ch

39.4.1.33.2 Function

This register is used to store errored data for corresponding D1TCM single-bit ECC interrupt if the interrupt is enabled.

39.4.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D1TCM_ECCS_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D1TCM_ECCS_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.33.4 Fields

Field	Function
31-0 D1TCM_ECCS_ERRED_DATA	D1TCM single-bit ECC error data

39.4.1.34 D1TCM multi-bit ECC Error Information Register (D1TCM_ECC_MULTI_ERROR_INFO)

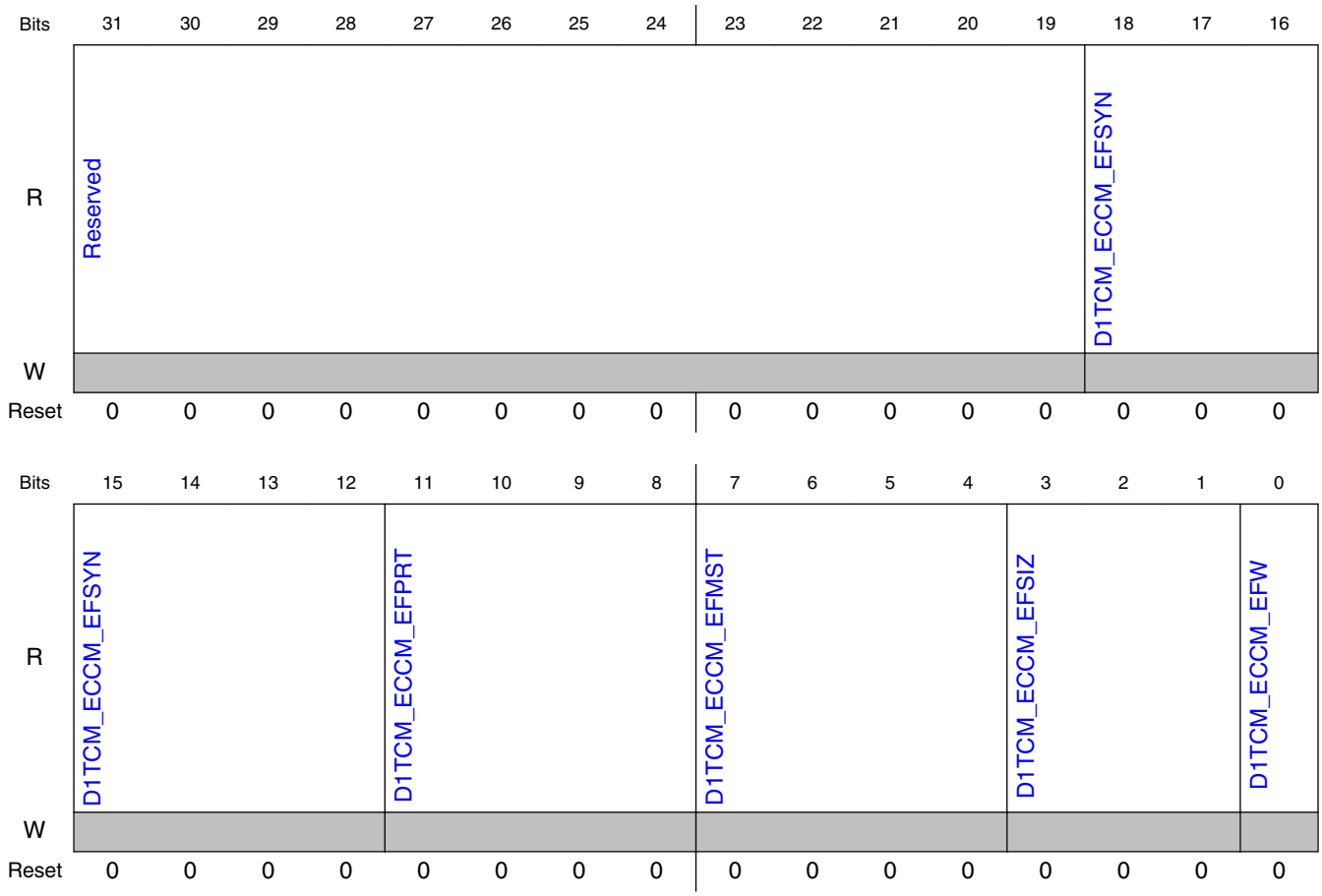
39.4.1.34.1 Offset

Register	Offset
D1TCM_ECC_MULTI_ERROR_INFO	80h

39.4.1.34.2 Function

This register is used to store detail error information for corresponding D1TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.34.3 Diagram



39.4.1.34.4 Fields

Field	Function
31-19 Reserved	Reserved
18-12 D1TCM_ECCM_EFSYN	D1TCM multi-bit ECC error corresponding syndrome Be noted syndrome for multi-bit ECC cannot be used to locate the Error bit
11-8 D1TCM_ECCM_EFPRT	D1TCM multi-bit ECC error corresponding TCM_PRIV TCM_PRIV is to tell the privilege level of access.
7-4 D1TCM_ECCM_EFMST	D1TCM multi-bit ECC error corresponding TCM_MASTER TCM_MASTER is to tell the requester of the current access. See the list in ITCM_ECCS_EFMST for the encoding of the access requestor.
3-1 D1TCM_ECCM_EFSIZ	D1TCM multi-bit ECC error corresponding tcm access size For D1TCM, this value should be 2 (32bit).

Table continues on the next page...

Field	Function
0 D1TCM_ECCM_EFW	D1TCM multi-bit ECC error corresponding TCM_WR value TCM_WR is to tell whether it is a write access(1'b1) or a read access(1'b0).

39.4.1.35 D1TCM multi-bit ECC Error Address Register (D1TCM_ECC_MULTI_ERROR_ADDR)

39.4.1.35.1 Offset

Register	Offset
D1TCM_ECC_MULTI_ERROR_ADDR	84h

39.4.1.35.2 Function

This register is used to store the errored address for corresponding D1TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D1TCM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D1TCM_ECCM_ERRED_ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.35.4 Fields

Field	Function
31-0 D1TCM_ECCM_ERRED_ADDR	D1TCM multi-bit ECC error address

39.4.1.36 D1TCM multi-bit ECC Error Data Register (D1TCM_ECC_MULTI_ERROR_DATA)

39.4.1.36.1 Offset

Register	Offset
D1TCM_ECC_MULTI_ERROR_DATA	88h

39.4.1.36.2 Function

This register is used to store errored data for corresponding D1TCM multi-bit ECC interrupt if the interrupt is enabled.

39.4.1.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	D1TCM_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	D1TCM_ECCM_ERRED_DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

39.4.1.36.4 Fields

Field	Function
31-0 D1TCM_ECCM_ERRED_DATA	D1TCM multi-bit ECC error data

39.4.1.37 OCRAM ECC Error Injection Register (OCRAM_ECC_ERROR_INJEC)

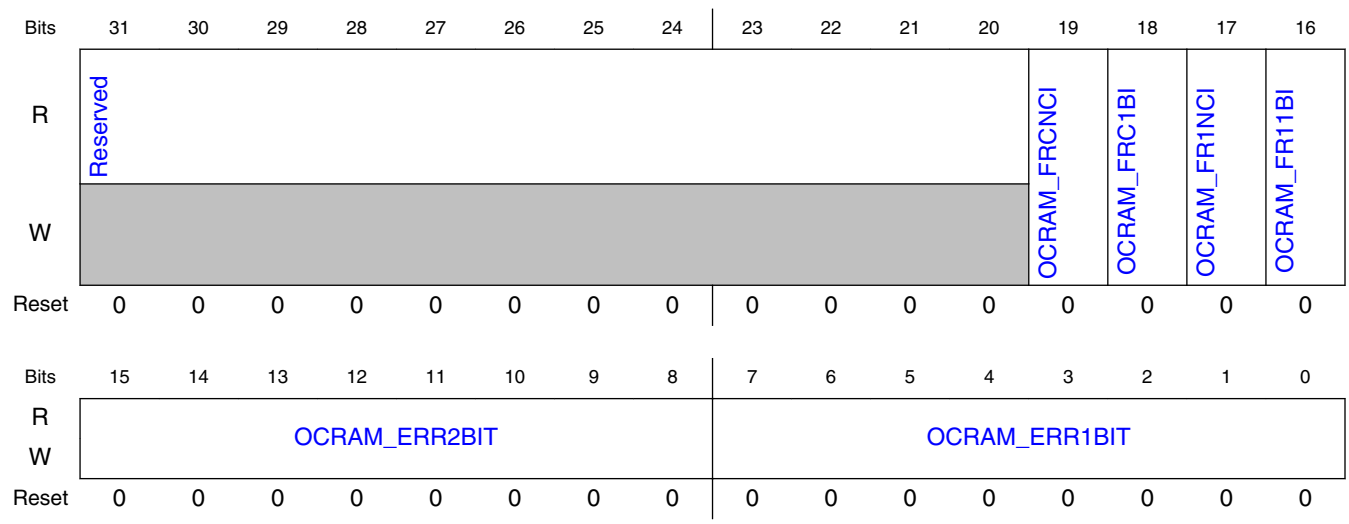
39.4.1.37.1 Offset

Register	Offset
OCRAM_ECC_ERROR_INJEC	8Ch

39.4.1.37.2 Function

This register is used to inject ECC error into OCRAM. Max 2 bits can be inverted to inject Error: if inject single-bit Error, OCRAM_ERR1BIT can be used; if inject multi-bit ECC Error, OCRAM_ERR1BIT and OCRAM_ERR2BIT can be used.

39.4.1.37.3 Diagram



39.4.1.37.4 Fields

Field	Function
31-20 Reserved	Reserved
19 OCRAM_FRCNCI	Force Continuous Non-correctable Data Inversions (multi-bit ECC error) on OCRAM write access When this bit is set, all the followed write access to OCRAM would be injected with multi-bit ECC error on the bits specified by OCRAM_ERR1BIT and OCRAM_ERR2BIT
18 OCRAM_FRC1BI	Force Continuous 1-Bit Data Inversions (single-bit ECC error) on OCRAM write access When this bit is set, all the followed write access to OCRAM would be injected with single-bit ECC error on the bit specified by OCRAM_ERR1BIT
17	Force One Non-correctable Data Inversion(multi-bit ECC error) on OCRAM write access

Table continues on the next page...

Field	Function
OCRAM_FR1NCI	When this bit is set, the first followed write access to OCRAM would be injected with multi-bit ECC error on the bits specified by OCRAM_ERR1BIT and OCRAM_ERR2BIT
16 OCRAM_FR11BI	Force One 1-Bit Data Inversion (single-bit ECC error) on OCRAM write access When this bit is set, the first followed write access to OCRAM would be injected with single-bit ECC error on the bit specified by OCRAM_ERR1BIT
15-8 OCRAM_ERR2BIT	Bit position of the second bit to inject multi-bit ECC Error. • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d63 : inject error on bit[63] of write data • 0d64 : inject error on bit[0] of write ecc • ... • 0d71 : inject error on bit[8] of write ecc
7-0 OCRAM_ERR1BIT	Bit position of the bit to inject ECC Error. Can be used for single-bit and multi-bit ECC Error injection. • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d63 : inject error on bit[63] of write data • 0d64 : inject error on bit[0] of write ecc • ... • 0d71 : inject error on bit[8] of write ecc

39.4.1.38 ITCM ECC Error Injection Register (ITCM_ECC_ERROR_INJEC)

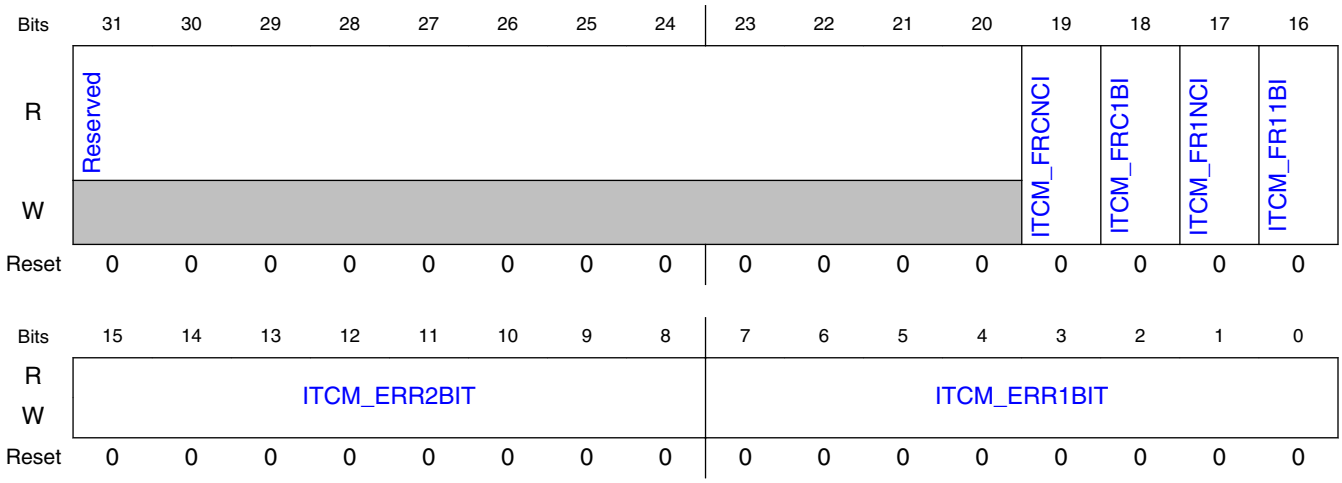
39.4.1.38.1 Offset

Register	Offset
ITCM_ECC_ERROR_INJEC	90h

39.4.1.38.2 Function

This register is used to inject ECC error into ITCM.

39.4.1.38.3 Diagram



39.4.1.38.4 Fields

Field	Function
31-20 Reserved	Reserved
19 ITCM_FRCNCI	Force Continuous Non-correctable Data Inversions (multi-bit ECC error) on ITCM write access When this bit is set, all the followed write access to ITCM would be injected with multi-bit ECC error on the bits specified by ITCM_ERR1BIT and ITCM_ERR2BIT
18 ITCM_FRC1BI	Force Continuous 1-Bit Data Inversions (single-bit ECC error) on ITCM write access When this bit is set, all the followed write access to ITCM would be injected with single-bit ECC error on the bit specified by ITCM_ERR1BIT
17 ITCM_FR1NCI	Force One Non-correctable Data Inversion(multi-bit ECC error) on ITCM write access When this bit is set, the first followed write access to ITCM would be injected with multi-bit ECC error on the bits specified by ITCM_ERR1BIT and ITCM_ERR2BIT
16 ITCM_FR11BI	Force One 1-Bit Data Inversion(single-bit ECC error) on ITCM write access When this bit is set, the first followed write access to ITCM would be injected with single-bit ECC error on the bit specified by ITCM_ERR1BIT
15-8 ITCM_ERR2BIT	Bit position of the second bit to inject multi-bit ECC Error. • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d63 : inject error on bit[63] of write data • 0d64 : inject error on bit[0] of write ecc • ... • 0d71 : inject error on bit[8] of write ecc
7-0 ITCM_ERR1BIT	Bit position of the bit to inject ECC Error. Can be used for single-bit and multi-bit ECC Error injection. • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data

Field	Function
	• ... • 0d63 : inject error on bit[63] of write data • 0d64 : inject error on bit[0] of write ecc • ... • 0d71 : inject error on bit[8] of write ecc

39.4.1.39 D0TCM ECC Error Injection Register (D0TCM_ECC_ERROR_INJEC)

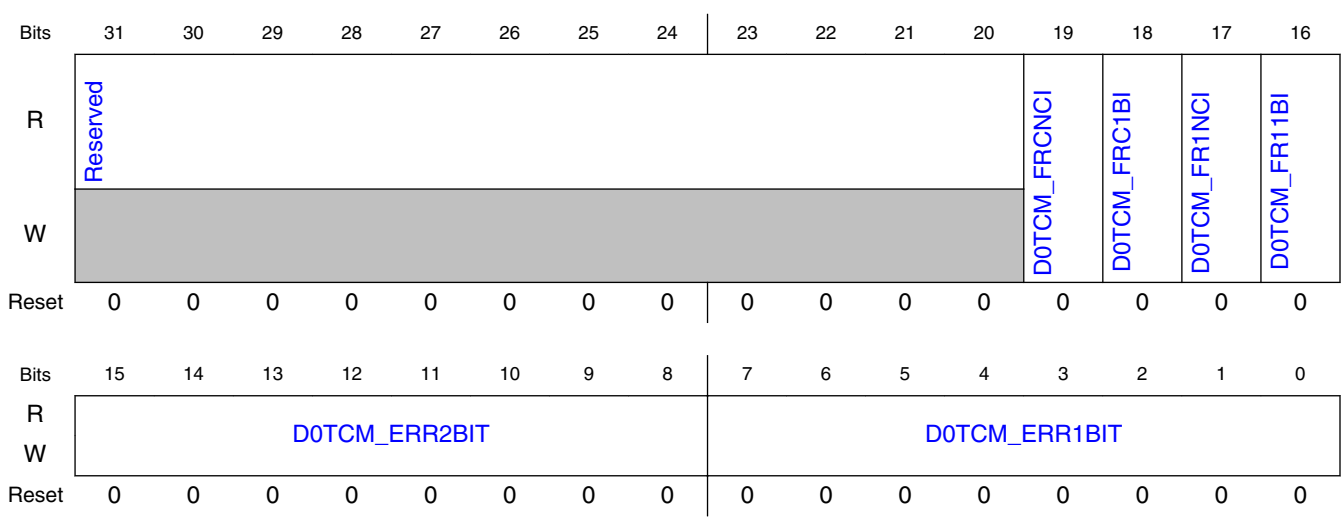
39.4.1.39.1 Offset

Register	Offset
D0TCM_ECC_ERROR_INJEC	94h

39.4.1.39.2 Function

This register is used to inject ECC error into D0TCM.

39.4.1.39.3 Diagram



39.4.1.39.4 Fields

Field	Function
31-20 Reserved	Reserved
19 D0TCM_FRCNCI	Force Continuous Non-correctable Data Inversions (multi-bit ECC error) on D0TCM write access When this bit is set, all the followed write access to D0TCM would be injected with multi-bit ECC error on the bits specified by D0TCM_ERR1BIT and D0TCM_ERR2BIT
18 D0TCM_FRC1BI	Force Continuous 1-Bit Data Inversions (single-bit ECC error) on D0TCM write access When this bit is set, all the followed write access to D0TCM would be injected with single-bit ECC error on the bit specified by D0TCM_ERR1BIT
17 D0TCM_FR1NCI	Force One Non-correctable Data Inversion(multi-bit ECC error) on D0TCM write access When this bit is set, the first followed write access to D0TCM would be injected with multi-bit ECC error on the bits specified by D0TCM_ERR1BIT and D0TCM_ERR2BIT
16 D0TCM_FR11BI	Force One 1-Bit Data Inversion(single-bit ECC error) on D0TCM write access When this bit is set, the first followed write access to D0TCM would be injected with single-bit ECC error on the bit specified by D0TCM_ERR1BIT
15-8 D0TCM_ERR2BIT	Bit position of the second bit to inject multi-bit ECC Error • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d31 : inject error on bit[31] of write data • 0d32 : inject error on bit[0] of write ecc • ... • 0d38 : inject error on bit[7] of write ecc
7-0 D0TCM_ERR1BIT	Bit position of the bit to inject ECC Error. Can be used for single-bit and multi-bit ECC Error injection. • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d31 : inject error on bit[31] of write data • 0d32 : inject error on bit[0] of write ecc • ... • 0d38 : inject error on bit[7] of write ecc

39.4.1.40 D1TCM ECC Error Injection Register (D1TCM_ECC_ERROR_INJEC)

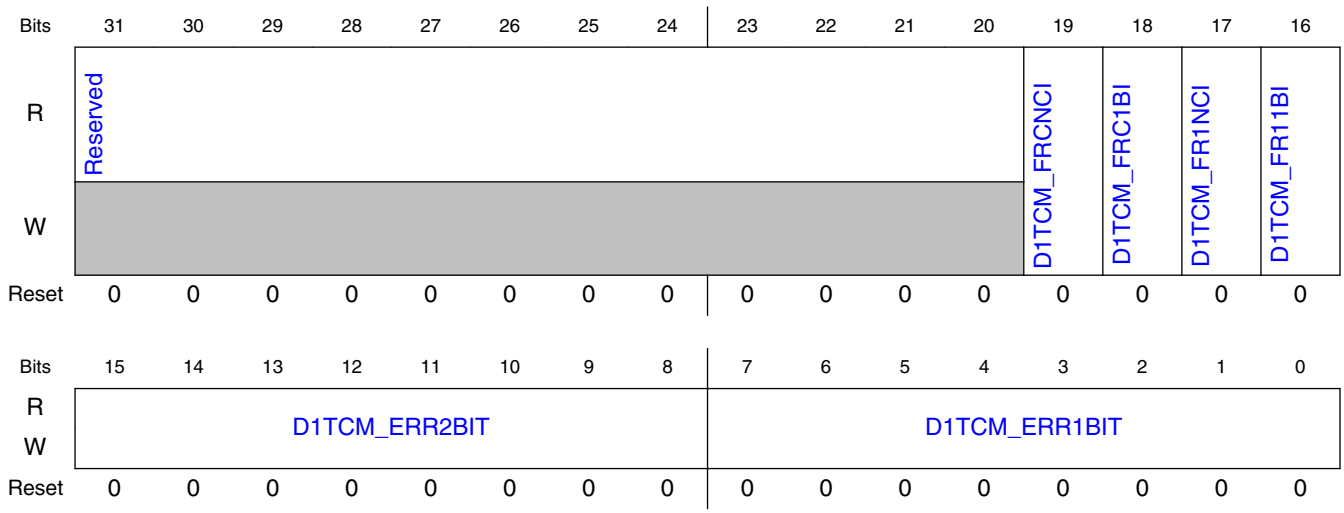
39.4.1.40.1 Offset

Register	Offset
D1TCM_ECC_ERROR_INJEC	98h

39.4.1.40.2 Function

This register is used to inject ECC error into D1TCM.

39.4.1.40.3 Diagram



39.4.1.40.4 Fields

Field	Function
31-20 Reserved	Reserved
19 D1TCM_FRCNCI	Force Continuous Non-correctable Data Inversions (multi-bit ECC error) on D1TCM write access When this bit is set, all the followed write access to D1TCM would be injected with multi-bit ECC error on the bits specified by D1TCM_ERR1BIT and D1TCM_ERR2BIT
18 D1TCM_FRC1BI	Force Continuous 1-Bit Data Inversions (single-bit ECC error) on D1TCM write access When this bit is set, all the followed write access to D1TCM would be injected with single-bit ECC error on the bit specified by D1TCM_ERR1BIT
17 D1TCM_FR1NCI	Force One Non-correctable Data Inversion(multi-bit ECC error) on D1TCM write access When this bit is set, the first followed write access to D1TCM would be injected with multi-bit ECC error on the bits specified by D1TCM_ERR1BIT and D1TCM_ERR2BIT
16 D1TCM_FR11BI	Force One 1-Bit Data Inversion(single-bit ECC error) on D1TCM write access When this bit is set, the first followed write access to D1TCM would be injected with single-bit ECC error on the bit specified by D1TCM_ERR1BIT
15-8 D1TCM_ERR2BIT	Bit position of the second bit to inject multi-bit ECC Error • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d31 : inject error on bit[31] of write data

Table continues on the next page...

Field	Function
	• 0d32 : inject error on bit[0] of write ecc • ... • 0d38 : inject error on bit[7] of write ecc
7-0 D1TCM_ERR1BIT	Bit position of the bit to inject ECC Error. Can be used for single-bit and multi-bit ECC Error injection • 0d00 : inject error on bit[0] of write data • 0d01: inject error on bit[1] of write data • 0d02: inject error on bit[2] of write data • ... • 0d31 : inject error on bit[31] of write data • 0d32 : inject error on bit[0] of write ecc • ... • 0d38 : inject error on bit[7] of write ecc

39.4.1.41 FlexRAM feature Control register (FLEXRAM_CTRL)

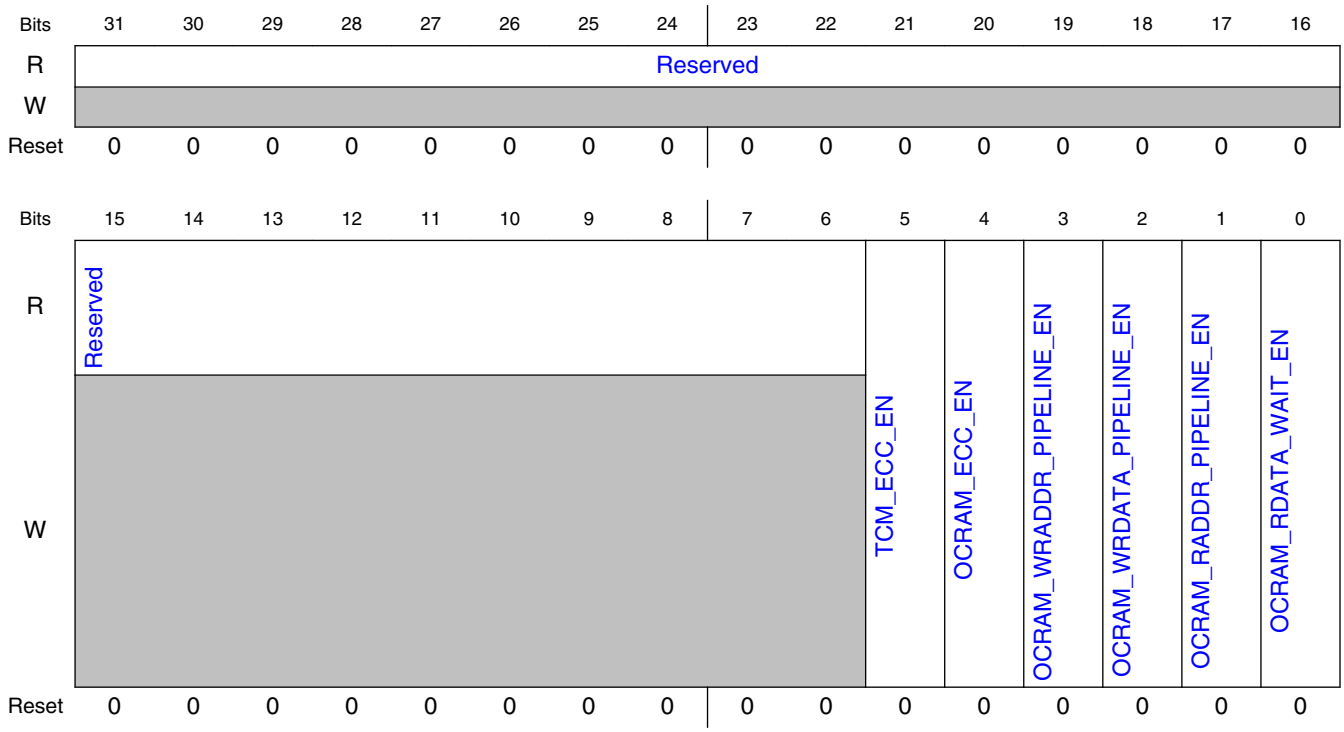
39.4.1.41.1 Offset

Register	Offset
FLEXRAM_CTRL	108h

39.4.1.41.2 Function

Flexram feature ECC and ocram wait pipeline can be configured through this register

39.4.1.41.3 Diagram



39.4.1.41.4 Fields

Field	Function
31-6 Reserved	Reserved
5 TCM_ECC_EN	TCM ECC enable TCM ECC software enable bit, the reset value of this bit is decided by value of register default_tcm_ECC_en.If fuse bit of TCM ECC disable is not fused to 1'b1, then this bit can decide whether TCM ECC is enabled.
4 OCRAM_ECC_EN	OCRAM ECC enable OCRAM ECC software enable bit, the reset value of this bit is decided by value of register default_ocram_ECC_en.If fuse bit of OCRAM ECC disable is not fused to 1'b1, then this bit can decide whether OCRAM ECC is enabled.
3 OCRAM_WRADDR_PIPELINE_EN	Write Address Pipeline Enable Pipeline enable signal for write address path, when this signal is set to 1, the write address will be registered before applied to memory cell
2 OCRAM_WRDATA_PIPELINE_EN	Write Data Pipeline Enable Pipeline enable signal for write data path, when this signal is set to 1, the write data will be registered before applied to memory cell
1	Read Address Pipeline Enable

Table continues on the next page...

Field	Function
OCRAM_RADDR_PIPELINE_EN	Pipeline enable signal for read address path, when this signal is set to 1, the read address will be registered before applied to memory cell
0 OCRAM_RDATA_WAIT_EN	Read Data Wait Enable 1-cycle wait state in read access enable signal, when this signal is set to 1, 1-cycle wait state will be inserted into each read access from OCRAM

39.4.1.42 OCRAM Pipeline Status register (OCRAM_PIPELINE_STATUS)

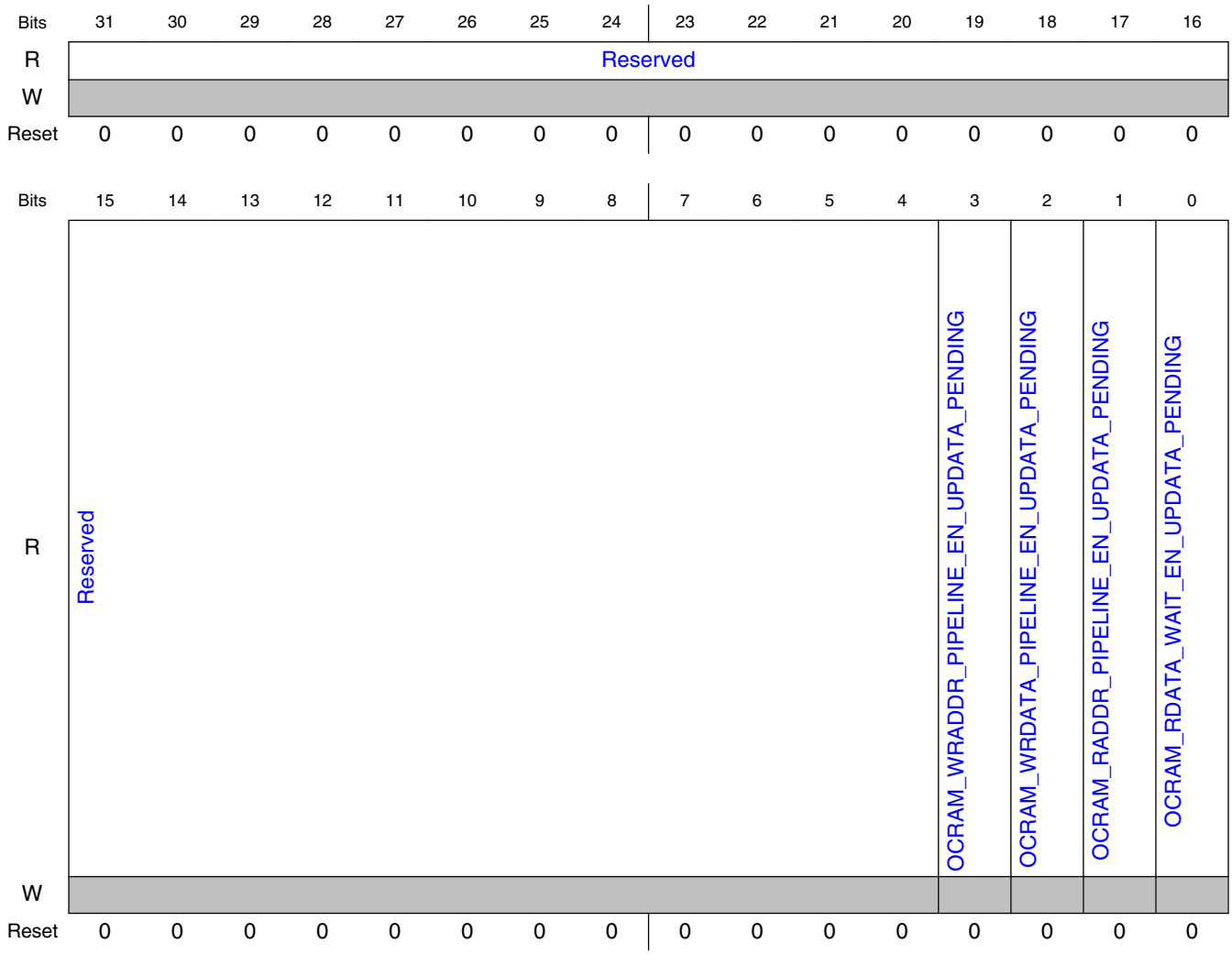
39.4.1.42.1 Offset

Register	Offset
OCRAM_PIPELINE_STATUS	10Ch

39.4.1.42.2 Function

Update pending signal for ocram wait and pipeline enable

39.4.1.42.3 Diagram



39.4.1.42.4 Fields

Field	Function
31-4 Reserved	Reserved
3 OCRAM_WRADDR_PIPELINE_EN_UPDATA_PENDING	Write Address Pipeline Enable Pending Update pending status for WRITE_ADDR_PIPE_EN. When WRITE_ADDR_PIPE_EN bit is changed, this register bit will be set until the new setup becomes valid in the controller.
2	Write Data Pipeline Enable Pending Update pending status for WRITE_DATA_PIPE_EN. When WRITE_DATA_PIPE_EN bit is changed, this register bit will be set until the new setup becomes valid in the controller.

Table continues on the next page...

Field	Function
OCRAM_WRDATA_PIPELINE_EN_UPDATA_PENDING	
1 OCRAM_RADDR_PIPELINE_EN_UPDATA_PENDING	Read Address Pipeline Enable Pending Update pending status for READ_ADDR_PIPE_EN. When READ_ADDR_PIPE_EN bit is changed, this register bit will be set until the new setup becomes valid in the controller.
0 OCRAM_RDATA_WAIT_EN_UPDATA_PENDING	Read Data Wait Enable Pending Update pending status for READ_DATA_WAIT_EN. When READ_DATA_WAIT_EN bit is changed, this register bit will be set until the new setup becomes valid in the controller.

Chapter 40
AHB to IP Bridge (AIPS Lite)

40.1 Chip-specific AIPS Lite information

Table 40-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit

40.2 Overview

The peripheral bridge (AIPS Lite) converts the crossbar switch interface to an interface that can access most of the slave peripherals on this chip. The peripheral bridge occupies 4MB of the address space, which is divided into peripheral slots of 16KB.

NOTE

Not all peripheral slots might be in use. See the memory map chapter for details on slot assignments.

40.3 Functional Description

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

It supports peripheral slots with 8, 16, and 32-bit data width. All accesses to the peripheral slot must be sized less than or equal to the designated peripheral slot size. If an access is attempted that is larger than the targeted port, and error response is generated.

40.3.1 Clocks

The clock roots of AIPS1, AIPS2 and AIPS3 use BUS_CLK_ROOT, while AIPS4 uses BUS_LPSR_CLK_ROOT. Please see the "System Clocks" section of CCM chapter for clock setting, configuration and gating information.

40.4 Memory Map/Register Definition

This module does not contain any user programmable registers.

Chapter 41
Semaphores (SEMA4)

41.1 Chip-specific SEMA4 information

Table 41-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

In this device, core processor 0 (logical processor number=0) is Cortex-M7 processor, and core processor 1 (logical processor number=1) is Cortex-M4 processor.

NOTE

This device only supports two master IDs for SEMA4. CM7 master ID is 0, CM4 master ID is 1. All the other masters should not access SEMA4.

41.2 Overview

The Semaphores (SEMA4) module provides a platform IPS slave device which implements 16 hardware-enforced gates.

41.2.1 Block Diagram

A simplified block diagram of the Semaphores module is shown in Figure 41-1. This is a dual-processor configuration, where cp0 is core processor 0 and cp1 is core processor 1. In the diagram, the register blocks named gate0, gate1,…, gate 15 include the finite state machines (FSM) implementing the semaphore gates plus the interrupt notification logic.

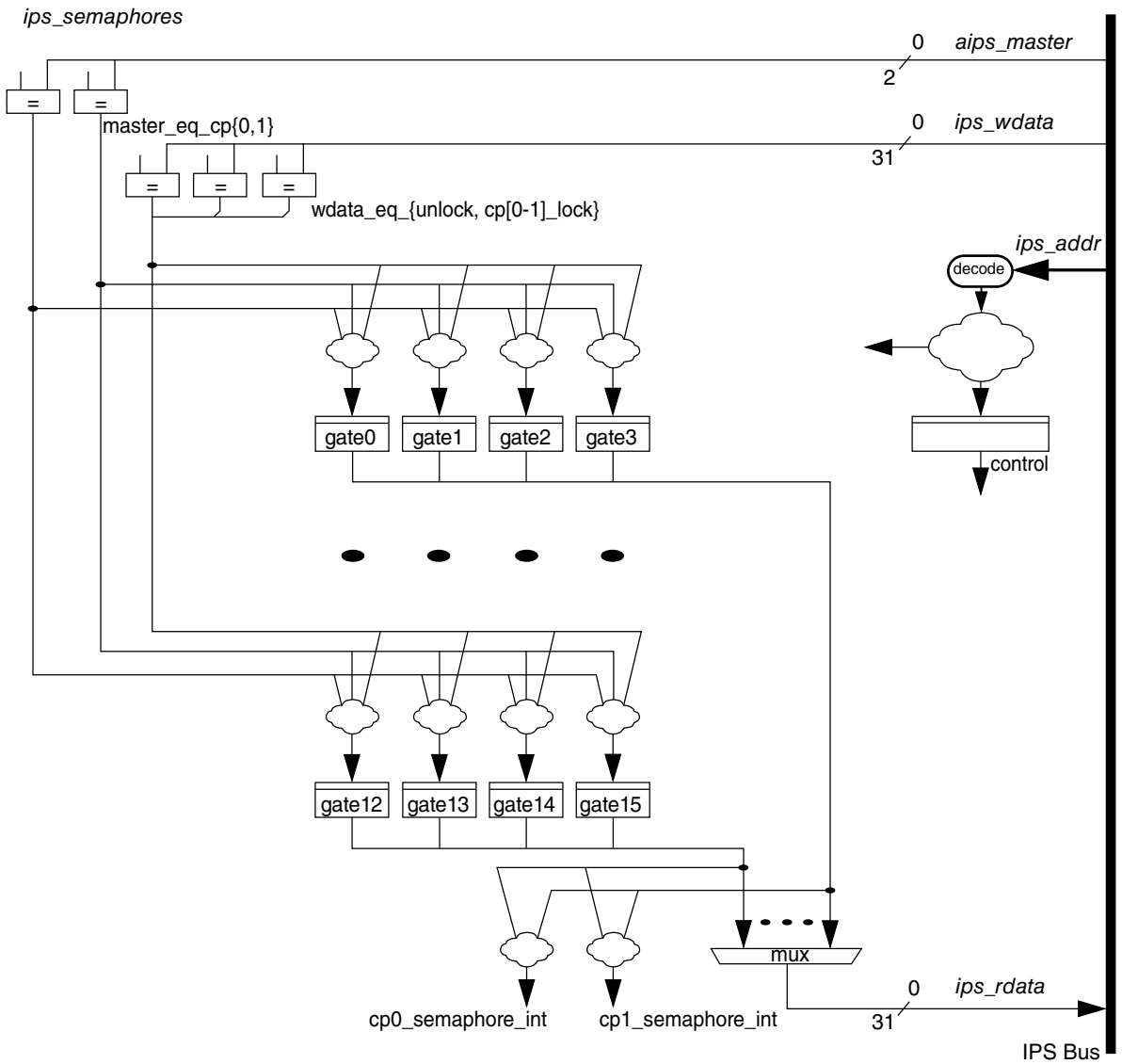


Figure 41-1. Semaphores Block Diagram

41.2.2 Features

The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Support for 16 hardware-enforced gates in a dual-processor configuration

- Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
  - Processors lock gates by writing "processor_number+1" to the appropriate gate and must read back the gate value to verify the lock operation was successful.

    3-state implementation:

    if gate = 0b00, then state = unlocked
    if gate = 0b01, then state = locked by processor 0
    if gate = 0b10, then state = locked by processor 1
  - Uses the bus master number/ID as a reference attribute plus the specified data patterns to validate all write operations
  - Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor
- Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
- Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear_all capability

- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses
  - Two outputs (one per processor) for interrupt notification of failed lock writes

41.3 Functional Description

In this section, the functional operation of the Semaphores module, specifically the state machines of the SEMA4_GATEn and SEMA4_CPnNTF registers are detailed.

41.3.1 Modes of Operation

The Semaphores module does not support any special modes of operation. As a slave peripheral memory-mapped device located on the platform's IPS slave bus, it responds based strictly on the memory addresses of the connected bus. The IPS bus is used to access the Semaphores' programming model.

41.3.2 SEMA4_GATEn Operation

Recall each of the SEMA4_GATEn registers implements a 2-bit, 3-state machine. The state transitions for each gate are shown in the following figure.

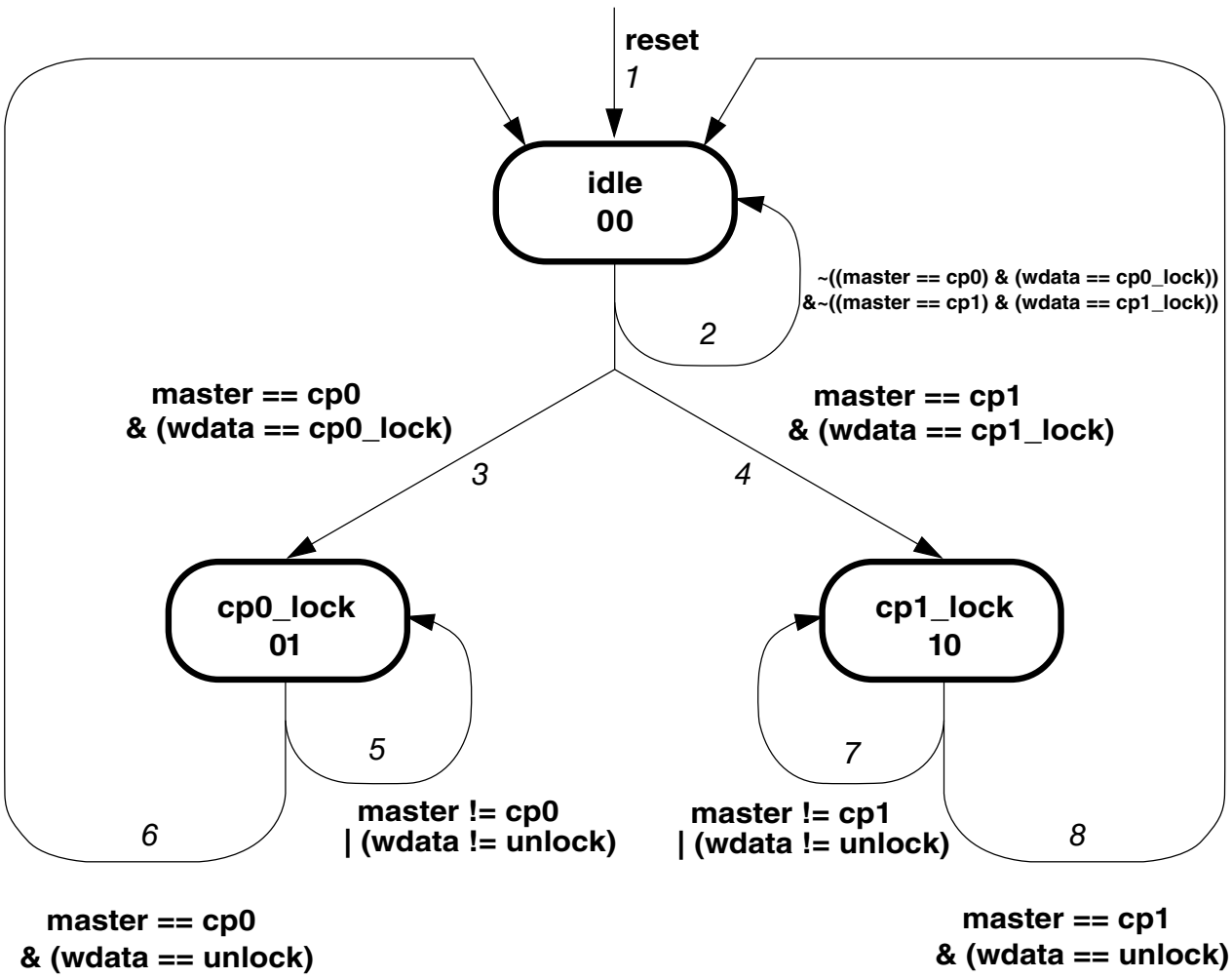


Figure 41-2. SEMA4_GATEn State Machine

The bus master number/ID is used to identify core processor 0 (cp0) or core processor 1 (cp1).

The state transitions for SEMA4_GATEn are defined in the following table.

Table 41-2. SEMA4_GATEn State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, whether a system reset or an individual gate reset, unconditionally forces the gate into the idle state.

Table continues on the next page...

Table 41-2. SEMA4_GATEn State Transitions (continued)

Current State	Next State	Transition	Description
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor occurs, the gate remains in the idle state.
idle	cp0_lock	3	When a write of the "cp0_lock" data value is initiated by processor 0, the gate transitions into the cp0_lock state.
idle	cp1_lock	4	When a write of the "cp1_lock" value is initiated by processor 1, the gate transitions into the cp1_lock state.
cp0_lock	cp0_lock	5	Once in this state, the gate remains here if any attempted write is not from cp0 with the unlock data value.
cp0_lock	idle	6	The gate returns to the idle (unlocked) state once a write from cp0 with the unlock data value occurs.
cp1_lock	cp1_lock	7	Once in this state, the gate remains here if any attempted write is not from cp1 with the unlock data value.
cp1_lock	idle	8	The gate returns to the idle (unlocked) state once a write from cp1 with the unlock data value occurs.

41.3.3 SEMA4_CPnNTF Operation

The failed lock write notification interrupt request is implemented in a 3-bit, 5-state machine which records failed lock attempts and transitions based on gate locking and unlocking. Two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn]. See the following figure.

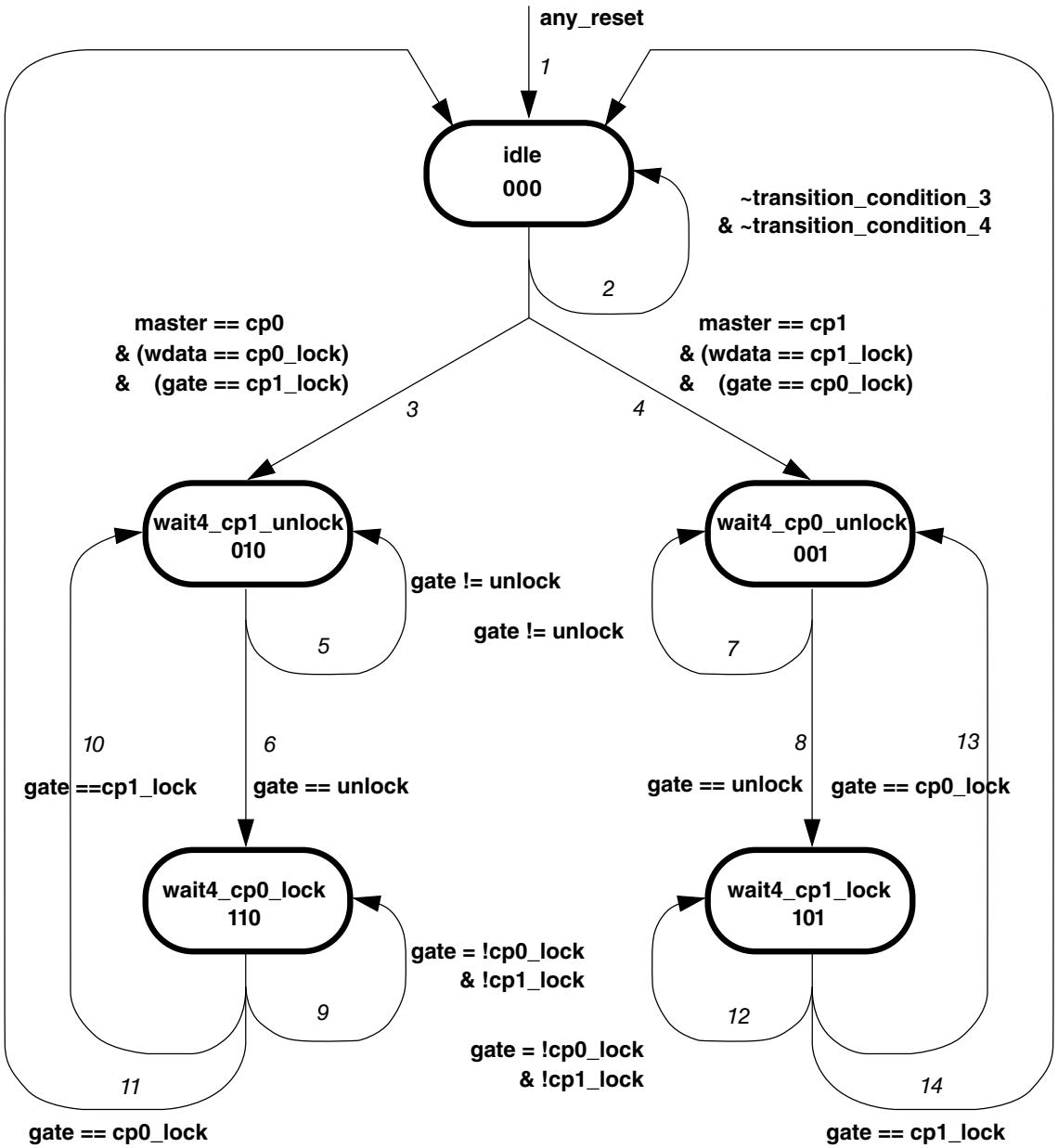


Figure 41-3. IRQ Notification State Machine

The state transitions of the IRQ notification function are defined in the following table. Specific states of this machine are program-visible as the SEMA4_CPnNTF registers. In particular, two states are program-visible:

```
if state = wait4_cp0_lock (0b110) // generate cp0_semaphore_int if properly enabled
then SEMA4_CP0NTF[GNn] = 1;
else SEMA4_CP0NTF[GNn] = 0

if state = wait4_cp1_lock (0b101) // generate cp1_semaphore_int if properly enabled
then SEMA4_CP1NTF[GNn] = 1;
else SEMA4_CP1NTF[GNn] = 0
```

Table 41-3. IRQ Notification State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, including a system reset or an individual notification or secure gate reset, unconditionally forces the machine into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor to an already-locked gate occurs, the machine remains in the idle state.
idle	wait4_cp1_unlock	3	When a write of the "cp0_lock" data value is initiated by processor 0 but the gate is already locked by cp1, the machine transitions into this state, where it waits for cp1 to unlock the gate.
idle	wait4_cp0_unlock	4	When a write of the "cp1_lock" data value is initiated by processor 1but the gate is already locked by cp0, the machine transitions into this state, where it waits for cp0 to unlock the gate
wait4_cp1_unlock	wait4_cp1_unlock	5	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp1_unlock	wait4_cp0_lock	6	From this state, the machine transitions into the next state, waiting for cp0 to lock the gate, once it has been unlocked.
wait4_cp0_unlock	wait4_cp0_unlock	7	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp0_unlock	wait4_cp1_lock	8	From this state, the machine transitions into the next state, waiting for cp1 to lock the gate, once it has been unlocked.
wait4_cp0_lock	wait4_cp0_lock	9	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 0 or the gate is again locked by processor 1.
wait4_cp0_lock	wait4_cp1_unlock	10	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 1. With this transition, the notification interrupt request is negated.
wait4_cp0_lock	idle	11	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	wait4_cp1_lock	12	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 1or the gate is again locked by processor 0.
wait4_cp1_lock	wait4_cp0_unlock	13	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	idle	14	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 1. With this transition, the notification interrupt request is negated.

The Semaphores module generates two interrupt request output signals, one per processor, combining the SEMA4_CPnINE and SEMA4_CPnNTF registers, where the boolean equations are:

```
cp0_semaphore_int
    =    SEMA4_CP0INE[INE0]    &    SEMA4_CP0NTF[GN0]
    |    SEMA4_CP0INE[INE1]    &    SEMA4_CP0NTF[GN1]
    |    SEMA4_CP0INE[INE2]    &    SEMA4_CP0NTF[GN2]
    ...
    |    SEMA4_CP0INE[INE15]    &    SEMA4_CP0NTF[GN15]
cp1_semaphore_int
    =    SEMA4_CP1INE[INE0]    &    SEMA4_CP1NTF[GN0]
    |    SEMA4_CP1INE[INE1]    &    SEMA4_CP1NTF[GN1]
    |    SEMA4_CP1INE[INE2]    &    SEMA4_CP1NTF[GN2]
    ...
    |    SEMA4_CP1INE[INE15]    &    SEMA4_CP1NTF[GN15]
```

41.4 External Signal Description

The Semaphores module does not include any external interfaces.

41.5 Initialization Information

The reset state of the Semaphores module allows it to begin operation without the need for any further initialization. All the internal state machines are cleared by any reset event, allowing the module to immediately begin operation.

41.6 Application Information

In an operational multi-core system, most interactions involving the Semaphores module involves reads and writes to the SEMA4_GATEn registers for implementation of the hardware-enforced software gate functions. Typical code segments for gate functions perform the following operations:

- To lock (close) a gate
  - The processor performs a byte write of "logical_processor_number + 1" to gate[i]
  - The processor reads back gate[i] and checks for a value of "logical_processor_number + 1"

If the compare indicates the expected value, then the gate is locked; proceed with the protected code segment. If the compare does not indicate the expected value, the lock operation failed; repeat the process beginning with byte write to gate[i] in spin-wait loop, or proceed with another execution path and wait for failed lock interrupt notification.

A simple C-language example of a `gateLock` function is shown in the following figure.

```
#define UNLOCK     0
#define CP0_LOCK   1
#define CP1_LOCK   2

void gateLock (n)
int   n;                                          /* gate number to lock */
{
   int i;
   int current_value;
   int locked_value;

   i = processor_number();  /* obtain logical CPU number */

   if (i == 0)
      locked_value = CP0_LOCK;
   else
      locked_value = CP1_LOCK;

   /* read the current value of the gate and wait until the state == UNLOCK */
   do {
        current_value = gate[n];
   } while (current_value != UNLOCK);

   /* the current value of the gate == UNLOCK. attempt to lock the gate for this
         processor. spin-wait in this loop until gate ownership is obtained */
   do {
      gate[n] = locked_value;  /* write gate with processor_number + 1 */
      current_value = gate[n]; /* read gate to verify ownership was obtained */
   } while (current_value != locked_value);
```

Figure 41-4. Sample gateLock Function

- To unlock (open) a gate
    - After completing the protected code segment, the locking processor performs a byte write of zeroes to gate[i], opening (unlocking) the gate

A few comments on the logical CPU number are appropriate. In this example, a reference to `processor_number()` is used to retrieve this hardware configuration value. Typically, the logical processor numbers are defined by a hardwired input vector to the individual cores. The exact method for accessing the logical processor number varies by architecture.

If the optional failed lock IRQ notification mechanisms are used, then accesses to the related registers (SEMA4_CPnINE, SEMA4_ CPnNTF) are required. Note that there is no required negation of the failed lock write notification interrupt, as the request is automatically cleared by the Semaphores module once the gate has been successfully locked by the "failing" processor.

Finally, in the event a system state requires a software-controlled reset of a gate or IRQ notification register(s), accesses to the secure reset control registers (SEMA4_RSTGT, SEMA4_RSTNTF) are required. For these situations, it is recommended that the

appropriate IRQ notification enable (SEMA4_CPnINE) bits should be disabled *before* initiating the secure reset 2-write sequence (write RSTGT and write RSTNTF), to avoid any race conditions involving spurious notification interrupt requests.

41.7 Memory map and register definition

The Semaphores module provides an IPS programming model mapped to an on-platform 16 KB space. The description here specifies a dual-core configuration with 16 semaphore gates. All the register names are prefixed with "SEMA4" as an abbreviation for the full module name.

The programming model is referenced using 8-, 16- and 32-bit accesses. Reads can use any reference size, while writes are generally restricted to the size of the register. Exceptions to the write size restrictions are detailed in the individual register descriptions. Attempted references using inappropriate access sizes, to undefined (reserved) addresses, or with a non-supported access type (for example, a write to a read-only register) generate an IPS error termination.

The 16 KB Semaphores programming model map is shown in the following table.

SEMA4 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40CC_8000	Semaphores Gate n Register (SEMA4_Gate0)	8	R/W	00h	41.7.1/2845
40CC_8001	Semaphores Gate n Register (SEMA4_Gate1)	8	R/W	00h	41.7.1/2845
40CC_8002	Semaphores Gate n Register (SEMA4_Gate2)	8	R/W	00h	41.7.1/2845
40CC_8003	Semaphores Gate n Register (SEMA4_Gate3)	8	R/W	00h	41.7.1/2845
40CC_8004	Semaphores Gate n Register (SEMA4_Gate4)	8	R/W	00h	41.7.1/2845
40CC_8005	Semaphores Gate n Register (SEMA4_Gate5)	8	R/W	00h	41.7.1/2845
40CC_8006	Semaphores Gate n Register (SEMA4_Gate6)	8	R/W	00h	41.7.1/2845
40CC_8007	Semaphores Gate n Register (SEMA4_Gate7)	8	R/W	00h	41.7.1/2845
40CC_8008	Semaphores Gate n Register (SEMA4_Gate8)	8	R/W	00h	41.7.1/2845
40CC_8009	Semaphores Gate n Register (SEMA4_Gate9)	8	R/W	00h	41.7.1/2845
40CC_800A	Semaphores Gate n Register (SEMA4_Gate10)	8	R/W	00h	41.7.1/2845
40CC_800B	Semaphores Gate n Register (SEMA4_Gate11)	8	R/W	00h	41.7.1/2845
40CC_800C	Semaphores Gate n Register (SEMA4_Gate12)	8	R/W	00h	41.7.1/2845
40CC_800D	Semaphores Gate n Register (SEMA4_Gate13)	8	R/W	00h	41.7.1/2845
40CC_800E	Semaphores Gate n Register (SEMA4_Gate14)	8	R/W	00h	41.7.1/2845
40CC_800F	Semaphores Gate n Register (SEMA4_Gate15)	8	R/W	00h	41.7.1/2845

Table continues on the next page...

SEMA4 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40CC_8040	Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE)	16	R/W	0000h	41.7.2/2846
40CC_8048	Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE)	16	R/W	0000h	41.7.2/2846
40CC_8080	Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)	16	R	0000h	41.7.3/2848
40CC_8088	Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)	16	R	0000h	41.7.3/2848
40CC_8100	Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)	16	R/W	0000h	41.7.4/2850
40CC_8104	Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)	16	R/W	0000h	41.7.5/2852

41.7.1 Semaphores Gate n Register (SEMA4_Gate*n*)

Each semaphore gate is implemented in a 2-bit finite state machine (FSM), right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 40CC_8000h base + 0h offset + (1d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate*n* field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine.

Table continues on the next page...

SEMA4_Gate*n* field descriptions (continued)

Field	Description
	Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

41.7.2 Semaphores Processor n IRQ Notification Enable (SEMA4_CP*n*INE)

The application of a hardware semaphore module provides an opportunity for implementation of helpful system-level features. An example is an optional mechanism to generate a processor interrupt after a failed lock attempt. Recall traditional software gate functions execute a spin-wait loop in an effort to obtain and lock the referenced gate. With this module, the processor that fails in the lock attempt could continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function.

The optional notification interrupt function consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnINE registers, unimplemented bits read as zeroes, and writes are ignored.

Address: 40CC_8000h base + 40h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read Write	INE8	INE9	INE10	INE11	INE12	INE13	INE14	INE15
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	INE0	INE1	INE2	INE3	INE4	INE5	INE6	INE7
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*INE field descriptions

Field	Description
15 INE8	Interrupt Request Notification Enable 8. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 8.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
14 INE9	Interrupt Request Notification Enable 9. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 9. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
13 INE10	Interrupt Request Notification Enable 10. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 10. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
12 INE11	Interrupt Request Notification Enable 11. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 11. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
11 INE12	Interrupt Request Notification Enable 12. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 12. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
10 INE13	Interrupt Request Notification Enable 13. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 13. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
9 INE14	Interrupt Request Notification Enable 14. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 14. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
8 INE15	Interrupt Request Notification Enable 15. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 15. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
7 INE0	Interrupt Request Notification Enable 0. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 0. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
6 INE1	Interrupt Request Notification Enable 1. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 1. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
5 INE2	Interrupt Request Notification Enable 2. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 2.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
4 INE3	Interrupt Request Notification Enable 3. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 3. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
3 INE4	Interrupt Request Notification Enable 4. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 4. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
2 INE5	Interrupt Request Notification Enable 5. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 5. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
1 INE6	Interrupt Request Notification Enable 6. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 6. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
0 INE7	Interrupt Request Notification Enable 7. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 7. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.

41.7.3 Semaphores Processor n IRQ Notification (SEMA4_CP*n*NTF)

The Semaphores module optionally allows the processor that fails in the lock attempt to continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function rather than simply execute in a spin-wait loop.

The optional notification interrupt mechanism consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the read-only notification interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnNTF registers, unimplemented bits read as zeroes.

The notification interrupt is generated via a unique finite state machine (FSM), one per hardware gate. This machine operates in the following manner:

1. When an attempted lock fails, the FSM enters a first state where it waits until the gate is unlocked.

2. Once unlocked, the FSM enters a second state where it generates an interrupt request to the “failed lock” processor.
3. When the “failed lock” processor succeeds in locking the gate, the IRQ is automatically cleared and the FSM returns to the idle state. However, if the other processor again locks the gate, the FSM returns to the first state, clears the interrupt request, and then waits for the gate to be unlocked (again).

The notification interrupt request is implemented in a 3-bit, 5-state machine, where two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn].

Address: 40CC_8000h base + 80h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read	GN8	GN9	GN10	GN11	GN12	GN13	GN14	GN15
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	GN0	GN1	GN2	GN3	GN4	GN5	GN6	GN7
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*NTF field descriptions

Field	Description
15 GN8	Gate 8 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 8. For more details, see SEMA4_CPnNTF Operation .
14 GN9	Gate 9 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 9. For more details, see SEMA4_CPnNTF Operation .
13 GN10	Gate 10 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 10. For more details, see SEMA4_CPnNTF Operation .
12 GN11	Gate 11 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 11. For more details, see SEMA4_CPnNTF Operation .
11 GN12	Gate 12 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 12. For more details, see SEMA4_CPnNTF Operation .
10 GN13	Gate 13 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 13. For more details, see SEMA4_CPnNTF Operation .
9 GN14	Gate 14 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 14. For more details, see SEMA4_CPnNTF Operation .
8 GN15	Gate 15 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 15. For more details, see SEMA4_CPnNTF Operation .
7 GN0	Gate 0 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 0. For more details, see SEMA4_CPnNTF Operation .
6 GN1	Gate 1 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 1. For more details, see SEMA4_CPnNTF Operation .
5 GN2	Gate 2 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 2. For more details, see SEMA4_CPnNTF Operation .

Table continues on the next page...

SEMA4_CP*n*NTF field descriptions (continued)

Field	Description
4 GN3	Gate 3 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 3. For more details, see SEMA4_CPnNTF Operation .
3 GN4	Gate 4 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 4. For more details, see SEMA4_CPnNTF Operation .
2 GN5	Gate 5 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 5. For more details, see SEMA4_CPnNTF Operation .
1 GN6	Gate 6 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 6. For more details, see SEMA4_CPnNTF Operation .
0 GN7	Gate 7 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 7. For more details, see SEMA4_CPnNTF Operation .

41.7.4 Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the Semaphores module implements a "secure" reset mechanism which allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTGT memory location. The least significant byte (SEMA4_RSTGT[RSTGDP]) must be 0xe2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTGT location. For this write, the lower byte (SEMA4_RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1d) and the upper byte (SEMA4_RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or that all gates are cleared.
3. Reads of the SEMA4_RSTGT location return information on the 2-bit state machine (SEMA4_RSTGT[RSTGSM]) which implements this function, the bus master performing the reset (SEMA4_RSTGT[RSTGMS]) and the gate number(s) last cleared (SEMA4_RSTGT[RSTGTN]). Reads of the SEMA4_RSTGT register do not affect the secure reset finite state machine in any manner.

Address: 40CC_8000h base + 100h offset = 40CC_8100h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTGTN								RSTGSM_RSTGMS_RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTGT field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates. The corresponding secure IRQ notification state machine(s) are also reset.
RSTGSM_ RSTGMS_ RSTGDP	**NOTE:** This field contains sub-fields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.

Access	Sub-Field	Description
Read-Only	7-6 Reserved	Reserved. Always reads 0.
	5-4 RSTGSM	Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write. 10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value 11 This state encoding is never used and therefore reserved.
	3 Reserved	Reserved. Always reads 0.
	2-0 RSTGMS	Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip-specific information section.
Write-Only	7-0 RSTGDP	Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xe2 while the second write requires RSTGDP = 0x1d.

41.7.5 Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)

As with the case of the secure reset function of the hardware gates, it is recognized that system operation may require a reset function to re-initialize the state of the IRQ notification logic without requiring a system-level reset.

To support this special notification reset requirement, the Semaphores module implements a "secure" reset mechanism which allows an IRQ notification (or all the notifications) to be initialized by following a specific dual-write access pattern. When successful, the specified IRQ notification state machine(s) are reset. Using a technique similar to that required for the servicing of a software watchdog timer, the secure reset mechanism requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the IRQ notification(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTNTF memory location. The least significant byte (SEMA4_RSTNTF[RSTNDP]) must be 0x47; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTNTF location. For this write, the lower byte (SEMA4_RSTNTF[RSTNDP]) is the logical complement of the first data pattern (0xb8) and the upper byte (SEMA4_RSTNTF[RSTNTN]) specifies the notification(s) to be reset. This field can specify a single notification be cleared, or that all notifications are cleared.
3. Reads of the SEMA4_RSTNTF location return information on the 2-bit state machine (SEMA4_RSTNTF[RSTNSM]) which implements this function, the bus master performing the reset (SEMA4_RSTNTF[RSTNMS]) and the notification number(s) last cleared (SEMA4_RSTNTF[RSTNTN]). Reads of the SEMA4_RSTNTF register do not affect the secure reset finite state machine in any manner.

Address: 40CC_8000h base + 104h offset = 40CC_8104h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTNTN								RSTNSM_RSTNMS_RSTNDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTNTF field descriptions

Field	Description
15–8 RSTNTN	Reset Notification Number. This 8-bit field specifies the specific IRQ notification state machine to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTNTN < 64, then reset the single IRQ notification machine defined by RSTNTN, else reset all the notifications.

Table continues on the next page...

SEMA4_RSTNTF field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>RSTNSM_
RSTNMS_
RSTNDP</td><td>
NOTE: This field contains sub-fields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.
<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>7-6
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-4
RSTNSM</td><td>Reset Notification Finite State Machine. Reads of the SEMA4_RSTNTF register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
00 Idle, waiting for the first data pattern write.
01 Waiting for the second data pattern write.
10 The 2-write sequence has completed. Generate the specified notification reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note the RSTNSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value.
11 This state encoding is never used and therefore reserved..</td></tr>
<tr><td>3
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-0
RSTNMS</td><td>Reset Notification Bus Master. This 3-bit read-only field records the logical number of the bus master performing the notification reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip-specific information section.</td></tr>
<tr><td>Write-Only</td><td>7-0
RSTNDP</td><td>Reset Notification Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the notification reset mechanism. For the first write, RSTNDP = 0x47 while the second write requires RSTNDP = 0xb8.</td></tr>
</table>
</td></tr>
</table>

Chapter 42
Resource Domain Controller (RDC)

42.1 Chip-specific RDC information

Table 42-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

In this device, only Cortex-M7 and Cortex-M4 are supported to use SEMA42 (within RDC) or SEMA4. Domain ID is used to distinguish CM7 and CM4, and no master index is used.

NOTE

RDC is not available or accessible on single core versions of RT117x (RT1172 and RT1171).

NOTE

RDC configuration by ROM is not supported.

42.2 Overview

The Resource Domain Controller (RDC) provides robust support for the isolation of destination memory mapped locations such as peripherals and memory to a single cluster, a bus master, or set of clusters and bus masters.

42.2.1 Block Diagram

Many of today's processors have multiple clusters for increased performance and flexibility. In some cases, the clusters serve different functions (for example, user level applications versus real time machine control) and in such cases the software for each cluster may be developed by different providers.

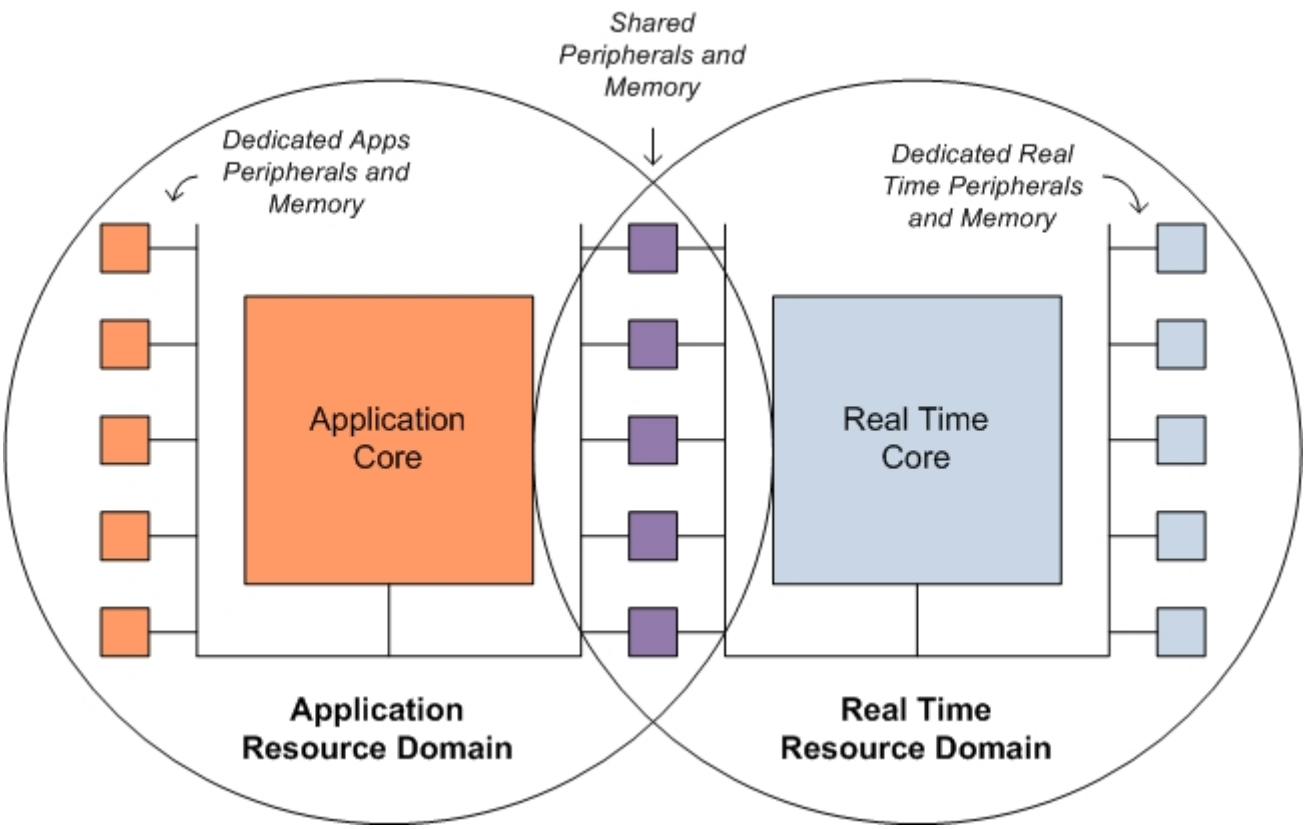


Figure 42-1. Dedicated and Shared Peripherals

For efficiency reasons the code on the clusters may share chip resources such as peripherals and memory. The sharing of chip resources between the somewhat independent processing domains allows for the opportunity of data collisions where

information stored in peripherals or memory by a process on one cluster is overwritten by software running on another cluster. Without careful collaboration between the two operating systems inadvertent malfunction or degradation in performance may result.

The RDC provides a mechanism to allow boot time configuration code to establish resource domains by assigning clusters, bus masters, peripherals and memory regions to domain identifiers. Once configured, bus transactions are monitored to restrict accesses initiated by clusters and bus masters to their respective peripherals and memory.

For shared peripherals, the RDC provides a semaphore-based locking mechanism to provide for temporary exclusivity while the domain software uses the peripheral. Once the software of one domain has finished the task and finished with the peripheral then it may release the semaphore making the peripheral available to the other domain.

42.2.2 Features

Resource domain subsystem has the following features:

- Assignment of clusters, bus masters, peripherals, and memory regions to a resource domain
- Fixed memory resolution of 128 Bytes for small address spaces and 4 KB for large address spaces
- Two resource domain identifiers (DIDs)
- Memory read/write access controls for each resource domain and region
- Optional semaphore-based, hardware-enforced exclusive access of shared peripherals to a resource domain
- Prioritized access permissions for overlapping memory regions
- Automatic restoration of resource domain access permissions to memory regions in the power-down domain

42.3 Functional Description

The RDC is the central location for creation of isolated resource domains and for the enablement of semaphore-based access also known as “safe sharing”. Configuration software assigns one of resource domain identifiers to each cluster and bus master, and allocates each memory region and peripheral to one or more resource domains.

Memory Read or Write access privileges for each resource domain are declared for each memory region. In addition, the software configuration determines which shared peripherals (those peripherals allocated to more than one domain) require safe sharing by setting the semaphore-required configuration for each peripheral.

The RDC configuration information is sent to the fabric ports, memories gaskets, semaphore controller, and peripherals to control access based on domain assignments. The fabric uses the domain identifier associated with each port to include this information along with the bus transaction. When the slave gasket encounters a bus transaction, it makes a comparison of the transaction domain ID to the RDC-provided list of allowed domains. If the transaction domain ID is on the list, then access may be permitted.

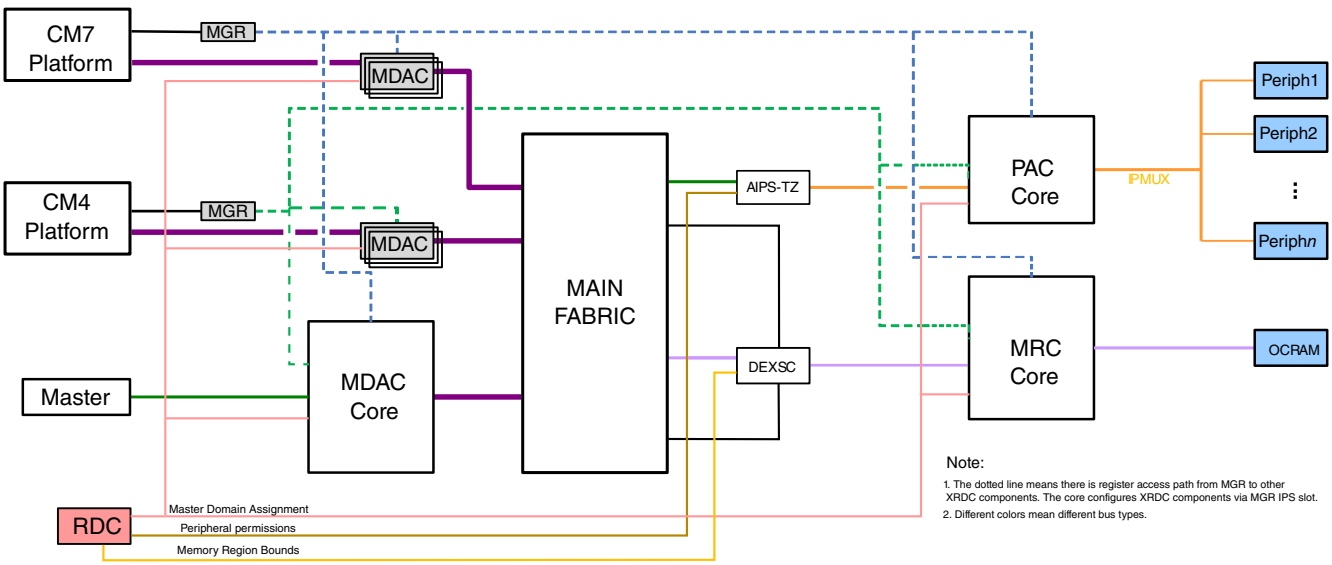


Figure 42-2. Example RDC Connections

For shared peripherals, RDC permits more than one domain access to a single peripheral. RDC also provides three ways to control synchronized use of shared peripherals. These methods include hardware-enforced synchronization, software-based semaphores, or no synchronization. The latter may be suitable for well-tuned multi-cluster operating systems that handle synchronization in the cluster platform, for instance.

For hardware-enforced synchronization, also known as "safe sharing", ownership of the peripheral must be claimed in the semaphore controller before access is allowed to the shared peripheral. Each shared peripheral has a corresponding Peripheral Domain Access Permissions (PDAP) register. When the Semaphore Required (SREQ) bit in a PDAP register is set, a master cannot access this shared peripheral until obtaining a semaphore. During the time that the domain has the semaphore in possession, its bus masters have exclusive access to the peripheral.

When the semaphore is released, then no domain masters have access until the semaphore is obtained again. When the SREQ bit is set, RDC module does not allow masters to obtain semaphores of peripherals to which the domain is not allocated; the master must have designated access in the D-bits of the corresponding PDAP register (for example, D1R bit set for Domain 1 access of the shared peripheral). There is a one-to-one mapping

between the semaphore controller gate and the resource domain controller peripheral. The mapping of PDAP registers and peripherals can be found in the Peripheral Map section of the RDC chapter.

42.3.1 Domain ID

The RDC provides for an isolation of domain resources by use of an identifier called the Domain ID (DID). A cluster and its resources including memory, bus masters, and peripherals are all associated with a single DID. When software or a DMA attempts to access a peripheral or memory, the corresponding bus transaction includes the DID along with the other bus control information such as Read, Write, and privilege mode.

42.3.2 Resource Assignment

The RDC allows assignment of peripherals and memories to one or more domains while each bus master or cluster is placed in one of several domains. The masters are assigned a domain in the Master Domain Assignment (MDA) register. A peripheral is given R/W access permissions to each domain in the PDAP register. Memory regions are bound by address space in start and end registers, i.e. MRSA and MREA. Each memory region is assigned one or more allowed domains and R/W permissions in the Memory Region Control (MRC) register. Memory regions must be enabled before the permissions are active. Otherwise the permissions are not restricted.

The RDC itself should be isolated to ensure that only a trustworthy resource manager can configure the RDC registers. This process may either be present initially during secure boot, or during the runtime in the secure world, for example. If the operating system does not support a runtime trusted execution, then during the secure boot process the RDC configuration can be locked to prevent further modification after the operating systems are running.

NOTE

The CCM supports multi-cluster awareness based on resource domain assignments programmed into the RDC. Refer to the CCM chapter regarding the relationship between cluster resource domains and their respective CCM resources. Failing to follow the proper sequence when updating the resource domain assignments of the cluster can result in clocks being inadvertently gated.

42.3.3 Safe Sharing

For shared peripherals, the RDC can be configured to require a domain to obtain a semaphore lock before access to the peripheral is allowed. This feature helps prevent collisions from processes on separate clusters that may want to use the same peripheral at the same time. The RDC sends a list of eligible domains to the semaphore module for each gate/peripheral. The eligible domains are those that are set in the peripheral domain access permissions (PDAP) registers. There is a one-to-one correspondence between semaphore gates and peripherals so each gate in the semaphore block represents a peripheral. The RDC receives semaphore locks from the hardware semaphore module (SEMA42). A semaphore lock is acquired when a cluster or bus master from a given domain requests a lock for a particular gate. The semaphore module compares the request's domain ID against the list of eligible domain IDs. If the domain ID is on the list and the lock is available then the lock is set and a signal is sent back to the RDC module indicating a lock has been acquired for a particular gate and a domain ID the lock belongs to. The RDC then restricts access to the corresponding peripheral to only transactions originating from the domain that has the lock. Another domain, though on the shared list to access the peripheral, must then wait until the lock is released before acquiring the lock and gaining access to the peripheral. To enable this feature of hardware enforcement for the semaphore locks, the SREQ bit is set in the RDC resource register.

If the SREQ is set, then when a process determines it needs a shared peripheral, it must first lock the resource in the semaphore module. Once the resource is locked, the semaphore module sends a signal to the RDC indicating the domain has access to the resource. The RDC will then set the access permissions to allow that domain access to the peripheral.

For a domain to acquire a lock on a peripheral, the domain must have been assigned to the peripheral in the RDC Peripheral Domain Access Permissions register (PDAP). The semaphore module only allows safe-sharing locks for those domains that are assigned to the peripheral. The semaphore module does not consider the access type (Read or Write) when allowing domains to acquire locks.

The SEMA42 module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Module definition supporting 64 hardware-enforced gates in a multi-processor configuration, where up to 15 processors can be supported; cp*X* is meant to represent cluster processor *X*
    - Gates appear as an *n*-entry byte-size array with read and write accesses ($n = 64$).

- Processors lock gates by writing logic 1 to the appropriate gate and must read back the gate value to verify the lock operation was successful. Also note that after locking, the gate register contains the value of the locking domain (in the field LDOM).
- Once locked, the gate is unlocked by a write of zeroes from the locking processor.
- The number of implemented gates is specified by a hardware configuration define.

- Each hardware gate appears as a 16-state, 4-bit state machine.
  - 16-state implementation

    if gate = 0x0, then state = unlocked
    if gate = 0x1, then state = locked by processor (master_index) 0
    if gate = 0x2, then state = locked by processor (master_index) 1
    …
    if gate = 0xF, then state = locked by processor (master_index) 14
  - Uses the logical bus master number (master_index) as a reference attribute plus the specified data patterns to validate all write operations.
  - Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor.
- Secure reset mechanisms are supported to clear the contents of individual gates, as well as a clear_all capability.

- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses

42.3.4 Resource Domain Control and Security Considerations

Conceptually, the RDC configuration is independent of the processor privilege mode and security domain. It is intended to allow for isolation between cluster processing environments to prevent collisions and increase reliability. Access between resource domains is mutually exclusive and each domain should be in control of its own privilege modes and access rights.

However, it is important to realize multi-cluster processors may have a multiple resource domains but only one overarching security domain. Chip security controls reside in one resource domain. In this configuration, a domain can affect at least one level of access privileges in the other domain. This may be acceptable but clarity and care is needed to ensure expected functionality.

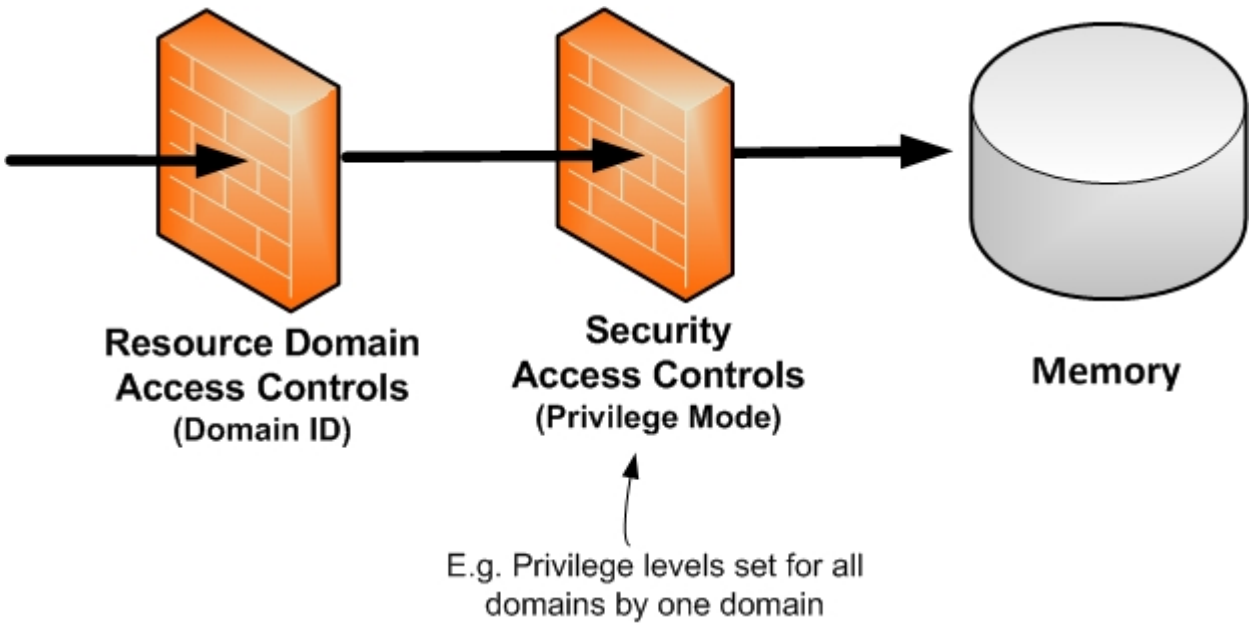


Figure 42-3. Access Control to Memory

Therefore, access to the security controls should be restricted to the most trustworthy operating mode of the cluster and privilege levels should be coordinated to ensure that shared peripherals and memory regions are accessible by both clusters. For instance, if a memory region is designated for secure accesses then all domain masters that share that region must have secure privileges.

42.3.5 Modes of Operation

The RDC provides access controls to the resource domain subsystem. When the device is in a low power mode then some memory regions in the subsystem may be powered off. RDC responds to the impacted memory regions by automatically reconfiguring the memory regions once power returns and blocking access to those memory regions until the reconfiguration process is complete.

42.3.5.1 Low Power Modes

The RDC loads configuration information for memory regions (MRSA, MRSE, MRC) into access control mechanisms (gaskets) at the memory interface. The location of this configuration information may reside inside power domains that lose power during sleep modes for energy savings. To restore configuration information upon return from sleep mode, the RDC receives a global power control signal indicating power is restored. The RDC then automatically reconfigures the memory regions with the configuration information.

During reconfiguration, access is blocked to the previously powered down memories. When the RDC completes reconfiguration it issues an interrupt and allows access to the memory regions. Only the powered down regions are blocked during the reconfiguration. Memory regions in the "always-on" power domain (still powered during sleep mode) remain available according to the programmed access rights. If no memory regions were enabled then the powered down regions are available immediately when power is restored.

The figure below shows the Global Power Control signal which RDC uses to invalidate the configuration upon deassertion and to restore the configuration when re-asserted. The configuration is valid and bus transactions allowed once the memory regions have been restored.

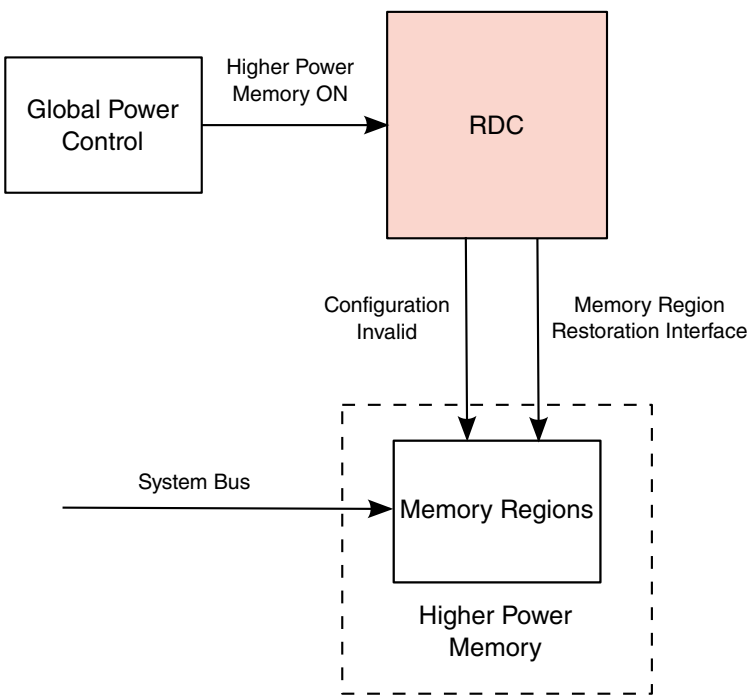


Figure 42-4. Memory Restoration Signaling

42.4 External Signals

RDC has no external signals pinned out.

42.5 Programming Interface

This section provides product specific details describing the mapping of resources - peripherals, bus masters, and memory regions - to corresponding resource domain controls RDC registers.

The RDC and RDC_SEMA42 register maps are combined in this chapter. The base address for the one RDC map and two SEMA42 maps are each separated by 4KB. While there are two SEMA42 submodules and therefore two sets of SEMA42 registers, this chapter describes one. Please refer to the peripheral memory map for the base addresses of the RDC and SEMA42 modules.

42.5.1 Master Assignment Registers

Table 42-2. Master Assignment Mapping

Master	RDC MDA register/Tied Value/Other Source
CM7	Tied to 0
CM7 DMA	Tied to 0
CM4	Tied to 1
CM4 DMA	Tied to 1
CAAM	From CAAM module: JRaDID_MS[PRIM_ICID]
LCDIF2 (LCDIFv2)	From LCDIF2 module
SSARC	From SSARC module: DESC_CTRL1_*n*[CPUD]
ENET 1G TX	RDC_MDA1
ENET 1G RX	RDC_MDA2
ENET	RDC_MDA3
ENET QOS	RDC_MDA4
USDHC1	RDC_MDA5
USDHC2	RDC_MDA6
USB	RDC_MDA7
GC355	RDC_MDA8
PXP	RDC_MDA9

Table continues on the next page...

Table 42-2. Master Assignment Mapping (continued)

Master	RDC MDA register/Tied Value/Other Source
LCDIF1 (eLCDIF)	RDC_MDA10
CSI	RDC_MDA11

42.5.2 Peripheral Mapping

Each peripheral has a corresponding resource domain assignment register in the RDC and semaphore lock register in the RDC_SEMA42 module. The following table shows allocation of the RDC_PDAP and RDC_SEMAPHORE*x*_GATE registers for peripheral resource domain assignment.

Table 42-3. RDC Peripheral Mapping

AIPS	Instance	RDC PDAP Register	RDC_SEMAPHOREx_GATE Register
aips_tz1	Reserved	RDC_PDAP0	RDC_SEMAPHORE1_GATE0
aips_tz1	MECC1	RDC_PDAP1	RDC_SEMAPHORE1_GATE1
aips_tz1	MECC2	RDC_PDAP2	RDC_SEMAPHORE1_GATE2
aips_tz1	FlexSPI1	RDC_PDAP3	RDC_SEMAPHORE1_GATE3
aips_tz1	XECC_FLEXSPI1	RDC_PDAP3	RDC_SEMAPHORE1_GATE3
aips_tz1	FlexSPI2	RDC_PDAP4	RDC_SEMAPHORE1_GATE4
aips_tz1	XECC_FLEXSPI2	RDC_PDAP4	RDC_SEMAPHORE1_GATE4
aips_tz1	SEMC	RDC_PDAP5	RDC_SEMAPHORE1_GATE5
aips_tz1	XECC_SEMC	RDC_PDAP5	RDC_SEMAPHORE1_GATE5
aips_tz1	FLEXRAM	RDC_PDAP6	RDC_SEMAPHORE1_GATE6
aips_tz1	EWM	RDC_PDAP7	RDC_SEMAPHORE1_GATE7
aips_tz1	WDOG1	RDC_PDAP8	RDC_SEMAPHORE1_GATE8
aips_tz1	WDOG2	RDC_PDAP9	RDC_SEMAPHORE1_GATE9
aips_tz1	WDOG3	RDC_PDAP10	RDC_SEMAPHORE1_GATE10
aips_tz1	AOI2	RDC_PDAP11	RDC_SEMAPHORE1_GATE11
aips_tz1	AOI1	RDC_PDAP11	RDC_SEMAPHORE1_GATE11
aips_tz1	XBAR3	RDC_PDAP11	RDC_SEMAPHORE1_GATE11
aips_tz1	XBAR2	RDC_PDAP11	RDC_SEMAPHORE1_GATE11
aips_tz1	XBAR1	RDC_PDAP11	RDC_SEMAPHORE1_GATE11
aips_tz1	ADC_ETC	RDC_PDAP12	RDC_SEMAPHORE1_GATE12
aips_tz2	CAAM (0x4045_0000-0x4045_3FFF)	RDC_PDAP13	RDC_SEMAPHORE1_GATE13
aips_tz1	ADC1	RDC_PDAP14	RDC_SEMAPHORE1_GATE14
aips_tz1	ADC2	RDC_PDAP15	RDC_SEMAPHORE1_GATE15

Table continues on the next page...

Table 42-3. RDC Peripheral Mapping (continued)

AIPS	Instance	RDC PDAP Register	RDC_SEMAPHOREx_GATE Register
aips_tz1	Reserved	RDC_PDAP16	RDC_SEMAPHORE1_GATE16
aips_tz1	DAC	RDC_PDAP17	RDC_SEMAPHORE1_GATE17
aips_tz1	IEE	RDC_PDAP18	RDC_SEMAPHORE1_GATE18
aips_tz1	APC_IEE	RDC_PDAP18	RDC_SEMAPHORE1_GATE18
aips_tz1	DMA_CH_MUX	RDC_PDAP19	RDC_SEMAPHORE1_GATE19
aips_tz1	EDMA	RDC_PDAP19	RDC_SEMAPHORE1_GATE19
aips_tz1	LPUART1	RDC_PDAP20	RDC_SEMAPHORE1_GATE20
aips_tz1	LPUART2	RDC_PDAP21	RDC_SEMAPHORE1_GATE21
aips_tz1	LPUART3	RDC_PDAP22	RDC_SEMAPHORE1_GATE22
aips_tz1	LPUART4	RDC_PDAP23	RDC_SEMAPHORE1_GATE23
aips_tz1	LPUART5	RDC_PDAP24	RDC_SEMAPHORE1_GATE24
aips_tz1	LPUART6	RDC_PDAP25	RDC_SEMAPHORE1_GATE25
aips_tz1	LPUART7	RDC_PDAP26	RDC_SEMAPHORE1_GATE26
aips_tz1	LPUART8	RDC_PDAP27	RDC_SEMAPHORE1_GATE27
aips_tz1	LPUART9	RDC_PDAP28	RDC_SEMAPHORE1_GATE28
aips_tz1	LPUART10	RDC_PDAP29	RDC_SEMAPHORE1_GATE29
aips_tz1	FlexIO1	RDC_PDAP30	RDC_SEMAPHORE1_GATE30
aips_tz1	FlexIO2	RDC_PDAP31	RDC_SEMAPHORE1_GATE31
aips_tz1	CAN1	RDC_PDAP32	RDC_SEMAPHORE1_GATE32
aips_tz1	CAN2	RDC_PDAP33	RDC_SEMAPHORE1_GATE33
aips_tz1	PIT1	RDC_PDAP34	RDC_SEMAPHORE1_GATE34
aips_tz1	KPP	RDC_PDAP35	RDC_SEMAPHORE1_GATE35
aips_tz1	IOMUXC_GPR	RDC_PDAP36	RDC_SEMAPHORE1_GATE36
aips_tz1	IOMUXC	RDC_PDAP37	RDC_SEMAPHORE1_GATE37
aips_tz1	GPT1	RDC_PDAP38	RDC_SEMAPHORE1_GATE38
aips_tz1	GPT2	RDC_PDAP39	RDC_SEMAPHORE1_GATE39
aips_tz1	GPT3	RDC_PDAP40	RDC_SEMAPHORE1_GATE40
aips_tz1	GPT4	RDC_PDAP41	RDC_SEMAPHORE1_GATE41
aips_tz1	GPT5	RDC_PDAP42	RDC_SEMAPHORE1_GATE42
aips_tz1	GPT6	RDC_PDAP43	RDC_SEMAPHORE1_GATE43
aips_tz1	LPI2C1	RDC_PDAP44	RDC_SEMAPHORE1_GATE44
aips_tz1	LPI2C2	RDC_PDAP45	RDC_SEMAPHORE1_GATE45
aips_tz1	LPI2C3	RDC_PDAP46	RDC_SEMAPHORE1_GATE46
aips_tz1	LPI2C4	RDC_PDAP47	RDC_SEMAPHORE1_GATE47
aips_tz1	LPSPI1	RDC_PDAP48	RDC_SEMAPHORE1_GATE48
aips_tz1	LPSPI2	RDC_PDAP49	RDC_SEMAPHORE1_GATE49
aips_tz1	LPSPI3	RDC_PDAP50	RDC_SEMAPHORE1_GATE50
aips_tz1	LPSPI4	RDC_PDAP51	RDC_SEMAPHORE1_GATE51
aips_tz1	GPIO6	RDC_PDAP52	RDC_SEMAPHORE1_GATE52

Table continues on the next page...

Table 42-3. RDC Peripheral Mapping (continued)

AIPS	Instance	RDC PDAP Register	RDC_SEMAPHOREx_GATE Register
aips_tz1	GPIO5	RDC_PDAP52	RDC_SEMAPHORE1_GATE52
aips_tz1	GPIO4	RDC_PDAP52	RDC_SEMAPHORE1_GATE52
aips_tz1	GPIO3	RDC_PDAP52	RDC_SEMAPHORE1_GATE52
aips_tz1	GPIO2	RDC_PDAP52	RDC_SEMAPHORE1_GATE52
aips_tz1	GPIO1	RDC_PDAP52	RDC_SEMAPHORE1_GATE52
aips_tz1	CCM(OBS)	RDC_PDAP53	RDC_SEMAPHORE1_GATE53
aips_tz1	SIM1	RDC_PDAP54	RDC_SEMAPHORE1_GATE54
aips_tz1	SIM2	RDC_PDAP55	RDC_SEMAPHORE1_GATE55
aips_tz1	QTimer1	RDC_PDAP56	RDC_SEMAPHORE1_GATE56
aips_tz1	QTimer2	RDC_PDAP57	RDC_SEMAPHORE1_GATE57
aips_tz1	QTimer3	RDC_PDAP58	RDC_SEMAPHORE1_GATE58
aips_tz1	QTimer4	RDC_PDAP59	RDC_SEMAPHORE1_GATE59
aips_tz1	DEC1	RDC_PDAP60	RDC_SEMAPHORE1_GATE60
aips_tz1	DEC2	RDC_PDAP61	RDC_SEMAPHORE1_GATE61
aips_tz1	DEC3	RDC_PDAP62	RDC_SEMAPHORE1_GATE62
aips_tz1	DEC4	RDC_PDAP63	RDC_SEMAPHORE1_GATE63
aips_tz1	FLEXPWM1	RDC_PDAP64	RDC_SEMAPHORE2_GATE0
aips_tz1	FLEXPWM2	RDC_PDAP65	RDC_SEMAPHORE2_GATE1
aips_tz1	FLEXPWM3	RDC_PDAP66	RDC_SEMAPHORE2_GATE2
aips_tz1	FLEXPWM4	RDC_PDAP67	RDC_SEMAPHORE2_GATE3
aips_tz2	CAAM (0x4046_0000-0x4046_3FFF)	RDC_PDAP68	RDC_SEMAPHORE2_GATE4
aips_tz2	CAAM (0x4047_0000-0x4047_3FFF)	RDC_PDAP69	RDC_SEMAPHORE2_GATE5
aips_tz1	ACMP1	RDC_PDAP70	RDC_SEMAPHORE2_GATE6
aips_tz1	ACMP2	RDC_PDAP71	RDC_SEMAPHORE2_GATE7
aips_tz1	ACMP3	RDC_PDAP72	RDC_SEMAPHORE2_GATE8
aips_tz1	ACMP4	RDC_PDAP73	RDC_SEMAPHORE2_GATE9
aips_tz2	CAAM (0x4048_0000-0x4048_3FFF)	RDC_PDAP74	RDC_SEMAPHORE2_GATE10
aips_tz2	SPDIF	RDC_PDAP75	RDC_SEMAPHORE2_GATE11
aips_tz2	SAI1	RDC_PDAP76	RDC_SEMAPHORE2_GATE12
aips_tz2	SAI2	RDC_PDAP77	RDC_SEMAPHORE2_GATE13
aips_tz2	SAI3	RDC_PDAP78	RDC_SEMAPHORE2_GATE14
aips_tz2	ASRC	RDC_PDAP79	RDC_SEMAPHORE2_GATE15
aips_tz2	USDHC1	RDC_PDAP80	RDC_SEMAPHORE2_GATE16
aips_tz2	USDHC2	RDC_PDAP81	RDC_SEMAPHORE2_GATE17
aips_tz2	ENET_1G	RDC_PDAP82	RDC_SEMAPHORE2_GATE18
aips_tz2	ENET	RDC_PDAP83	RDC_SEMAPHORE2_GATE19
aips_tz2	USB(pl301)	RDC_PDAP84	RDC_SEMAPHORE2_GATE20

Table continues on the next page...

Table 42-3. RDC Peripheral Mapping (continued)

AIPS	Instance	RDC PDAP Register	RDC_SEMAPHOREx_GATE Register
aips_tz2	USBPHY2	RDC_PDAP85	RDC_SEMAPHORE2_GATE21
aips_tz2	USB(OTG2)	RDC_PDAP85	RDC_SEMAPHORE2_GATE21
aips_tz2	USBPHY1	RDC_PDAP86	RDC_SEMAPHORE2_GATE22
aips_tz2	USB(OTG1)	RDC_PDAP86	RDC_SEMAPHORE2_GATE22
aips_tz2	ENET QOS	RDC_PDAP87	RDC_SEMAPHORE2_GATE23
aips_tz2	CAAM (0x4053_0000-0x4053_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x4052_0000-0x4052_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x4051_0000-0x4051_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x4050_0000-0x4050_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x404F_0000-0x404F_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x404E_0000-0x404E_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x404D_0000-0x404D_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x404B_0000-0x404B_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x4049_0000-0x4049_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz2	CAAM (0x4044_0000-0x4044_3FFF)	RDC_PDAP88	RDC_SEMAPHORE2_GATE24
aips_tz3	CSI	RDC_PDAP89	RDC_SEMAPHORE2_GATE25
aips_tz3	LCDIF1 (eLCDIF)	RDC_PDAP90	RDC_SEMAPHORE2_GATE26
aips_tz3	LCDIF2 (LCDIFv2)	RDC_PDAP91	RDC_SEMAPHORE2_GATE27
aips_tz3	MIPI_DSI	RDC_PDAP92	RDC_SEMAPHORE2_GATE28
aips_tz3	MIPI_CSI	RDC_PDAP93	RDC_SEMAPHORE2_GATE29
aips_tz3	PXP	RDC_PDAP94	RDC_SEMAPHORE2_GATE30
aips_tz3	VIDEO_MUX	RDC_PDAP95	RDC_SEMAPHORE2_GATE31
aips_tz4	PGMC	RDC_PDAP96	RDC_SEMAPHORE2_GATE32
aips_tz4	SRC	RDC_PDAP96	RDC_SEMAPHORE2_GATE32
aips_tz4	GPC	RDC_PDAP96	RDC_SEMAPHORE2_GATE32
aips_tz4	IOMUXC_LPSR	RDC_PDAP97	RDC_SEMAPHORE2_GATE33
aips_tz4	IOMUXC_LPSR_GPR	RDC_PDAP98	RDC_SEMAPHORE2_GATE34
aips_tz4	WDOG4	RDC_PDAP99	RDC_SEMAPHORE2_GATE35
aips_tz4	DMA_CH_MUX_LPSR	RDC_PDAP100	RDC_SEMAPHORE2_GATE36
aips_tz4	EDMA_LPSR	RDC_PDAP100	RDC_SEMAPHORE2_GATE36
aips_tz4	Reserved	RDC_PDAP101	RDC_SEMAPHORE2_GATE37
aips_tz4	PDM (MIC)	RDC_PDAP102	RDC_SEMAPHORE2_GATE38

Table continues on the next page...

Table 42-3. RDC Peripheral Mapping (continued)

AIPS	Instance	RDC PDAP Register	RDC_SEMAPHOREx_GATE Register
aips_tz4	LPUART11	RDC_PDAP103	RDC_SEMAPHORE2_GATE39
aips_tz4	LPUART12	RDC_PDAP104	RDC_SEMAPHORE2_GATE40
aips_tz4	LPSPI5	RDC_PDAP105	RDC_SEMAPHORE2_GATE41
aips_tz4	LPSPI6	RDC_PDAP106	RDC_SEMAPHORE2_GATE42
aips_tz4	LPI2C5	RDC_PDAP107	RDC_SEMAPHORE2_GATE43
aips_tz4	LPI2C6	RDC_PDAP108	RDC_SEMAPHORE2_GATE44
aips_tz4	CAN3	RDC_PDAP109	RDC_SEMAPHORE2_GATE45
aips_tz4	SAI4	RDC_PDAP110	RDC_SEMAPHORE2_GATE46
aips_tz4	XRDC2_MGR_D1	RDC_PDAP111	RDC_SEMAPHORE2_GATE47
aips_tz4	GPIO12	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	GPIO11	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	GPIO10	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	GPIO9	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	GPIO8	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	GPIO7	RDC_PDAP112	RDC_SEMAPHORE2_GATE48
aips_tz4	KEY_MANAGER	RDC_PDAP113	RDC_SEMAPHORE2_GATE49
aips_tz4	ANALOG/ANADIG	RDC_PDAP114	RDC_SEMAPHORE2_GATE50
aips_tz4	SNVS_HP	RDC_PDAP115	RDC_SEMAPHORE2_GATE51
aips_tz4	IOMUXC_SNVS	RDC_PDAP116	RDC_SEMAPHORE2_GATE52
aips_tz4	IOMUXC_SNVS_GPR	RDC_PDAP117	RDC_SEMAPHORE2_GATE53
aips_tz4	SNVS_SRAM	RDC_PDAP118	RDC_SEMAPHORE2_GATE54
aips_tz4	GPIO13	RDC_PDAP119	RDC_SEMAPHORE2_GATE55
aips_tz4	ROMCP	RDC_PDAP120	RDC_SEMAPHORE2_GATE56
aips_tz4	DCDC	RDC_PDAP121	RDC_SEMAPHORE2_GATE57
aips_tz4	OCOTP_CTRL	RDC_PDAP122	RDC_SEMAPHORE2_GATE58
aips_tz4	PIT2	RDC_PDAP123	RDC_SEMAPHORE2_GATE59
aips_tz4	SSARC_LP	RDC_PDAP124	RDC_SEMAPHORE2_GATE60
aips_tz4	SSARC_HP	RDC_PDAP124	RDC_SEMAPHORE2_GATE60
aips_tz4	CCM	RDC_PDAP125	RDC_SEMAPHORE2_GATE61
aips_tz2	CAAM (0x404A_0000-0x404A_3FFF)	RDC_PDAP126	RDC_SEMAPHORE2_GATE62
aips_tz2	CAAM (0x404C_0000-0x404C_3FFF)	RDC_PDAP127	RDC_SEMAPHORE2_GATE63

42.5.3 Memory Region Map

The number of memories with domain isolation support varies per device. The number of memory regions for a particular memory and the size of those regions varies per memory gasket. Each region of memory has a set of registers to define the boundaries of the region based on start and end addresses, a control register to set the domain access permissions and enable the region, and a status register to determin if access was denied to a region.

For this device, refer to the table below to determine the memories with domain support, the number of regions for each memory, the region resolution, the identifying numbers for the sets of memory region registers, and the addresses of the RDC registers to access the sets of Memory Region registers.

Table 42-4. Memory Region Mapping

Memory/Port	Regions	Region Resolution	MEM Region register set	Register Addr Range	DEXSC/DSEC Parameters
SEMC	8	4KB	0-7	0x800 – 0x87C	DEXSC_START_ADDR 11'h000
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 20
FlexSPI1	8	4KB	8-15	0x880 – 0x8FC	DEXSC_START_ADDR 11'h080
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 20
FlexSPI2	8	4KB	16-23	0x900 – 0x97C	DEXSC_START_ADDR 11'h100
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 20
OCRAM1	8	128B	24-31	0x980 – 0x9FC	DEXSC_START_ADDR 11'h180
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 25
OCRAM2	8	128B	32-39	0xA00 – 0xA7C	DEXSC_START_ADDR 11'h200
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 25

Table continues on the next page...

Table 42-4. Memory Region Mapping (continued)

Memory/Port	Regions	Region Resolution	MEM Region register set	Register Addr Range	DEXSC/DSEC Parameters
M7 OCRAM	4	128B	40-43	0xA80 – 0xABC	DEXSC_START_ADDR 11'h280
					NUM_OF_SECTIONS 4
					MEM_ADDR_WIDTH 25
M4 OCRAM	4	128B	44-47	0xAC0 – 0xAFC	DEXSC_START_ADDR 11'h2C0
					NUM_OF_SECTIONS 4
					MEM_ADDR_WIDTH 25
CAAM	8	8KB	48-55	0xB00 – 0xB7C	DEXSC_START_ADDR 11'h300
					NUM_OF_SECTIONS 8
					MEM_ADDR_WIDTH 19
GPV	3	1MB	56-58	0xB80 – 0xBAC	DEXSC_START_ADDR 11'h380
					NUM_OF_SECTIONS 3

42.6 RDC Memory Map/Register Definition

RDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8000	Version Information (RDC_VIR)	32	R	03B8_00C2h	42.6.1/2883
40C7_8024	Status (RDC_STAT)	32	R/W	0000_0100h	42.6.2/2883
40C7_8028	Interrupt and Control (RDC_INTCTRL)	32	R/W	0000_0000h	42.6.3/2884
40C7_802C	Interrupt Status (RDC_INTSTAT)	32	R/W	See section	42.6.4/2885
40C7_8200	Master Domain Assignment (RDC_MDA0)	32	R/W	0000_0000h	42.6.5/2885
40C7_8204	Master Domain Assignment (RDC_MDA1)	32	R/W	0000_0000h	42.6.5/2885
40C7_8208	Master Domain Assignment (RDC_MDA2)	32	R/W	0000_0000h	42.6.5/2885
40C7_820C	Master Domain Assignment (RDC_MDA3)	32	R/W	0000_0000h	42.6.5/2885
40C7_8210	Master Domain Assignment (RDC_MDA4)	32	R/W	0000_0000h	42.6.5/2885
40C7_8214	Master Domain Assignment (RDC_MDA5)	32	R/W	0000_0000h	42.6.5/2885

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8218	Master Domain Assignment (RDC_MDA6)	32	R/W	0000_0000h	42.6.5/2885
40C7_821C	Master Domain Assignment (RDC_MDA7)	32	R/W	0000_0000h	42.6.5/2885
40C7_8220	Master Domain Assignment (RDC_MDA8)	32	R/W	0000_0000h	42.6.5/2885
40C7_8224	Master Domain Assignment (RDC_MDA9)	32	R/W	0000_0000h	42.6.5/2885
40C7_8228	Master Domain Assignment (RDC_MDA10)	32	R/W	0000_0000h	42.6.5/2885
40C7_822C	Master Domain Assignment (RDC_MDA11)	32	R/W	0000_0000h	42.6.5/2885
40C7_8400	Peripheral Domain Access Permissions (RDC_PDAP0)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8404	Peripheral Domain Access Permissions (RDC_PDAP1)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8408	Peripheral Domain Access Permissions (RDC_PDAP2)	32	R/W	0000_000Fh	42.6.6/2886
40C7_840C	Peripheral Domain Access Permissions (RDC_PDAP3)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8410	Peripheral Domain Access Permissions (RDC_PDAP4)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8414	Peripheral Domain Access Permissions (RDC_PDAP5)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8418	Peripheral Domain Access Permissions (RDC_PDAP6)	32	R/W	0000_000Fh	42.6.6/2886
40C7_841C	Peripheral Domain Access Permissions (RDC_PDAP7)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8420	Peripheral Domain Access Permissions (RDC_PDAP8)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8424	Peripheral Domain Access Permissions (RDC_PDAP9)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8428	Peripheral Domain Access Permissions (RDC_PDAP10)	32	R/W	0000_000Fh	42.6.6/2886
40C7_842C	Peripheral Domain Access Permissions (RDC_PDAP11)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8430	Peripheral Domain Access Permissions (RDC_PDAP12)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8434	Peripheral Domain Access Permissions (RDC_PDAP13)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8438	Peripheral Domain Access Permissions (RDC_PDAP14)	32	R/W	0000_000Fh	42.6.6/2886
40C7_843C	Peripheral Domain Access Permissions (RDC_PDAP15)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8440	Peripheral Domain Access Permissions (RDC_PDAP16)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8444	Peripheral Domain Access Permissions (RDC_PDAP17)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8448	Peripheral Domain Access Permissions (RDC_PDAP18)	32	R/W	0000_000Fh	42.6.6/2886
40C7_844C	Peripheral Domain Access Permissions (RDC_PDAP19)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8450	Peripheral Domain Access Permissions (RDC_PDAP20)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8454	Peripheral Domain Access Permissions (RDC_PDAP21)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8458	Peripheral Domain Access Permissions (RDC_PDAP22)	32	R/W	0000_000Fh	42.6.6/2886
40C7_845C	Peripheral Domain Access Permissions (RDC_PDAP23)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8460	Peripheral Domain Access Permissions (RDC_PDAP24)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8464	Peripheral Domain Access Permissions (RDC_PDAP25)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8468	Peripheral Domain Access Permissions (RDC_PDAP26)	32	R/W	0000_000Fh	42.6.6/2886
40C7_846C	Peripheral Domain Access Permissions (RDC_PDAP27)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8470	Peripheral Domain Access Permissions (RDC_PDAP28)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8474	Peripheral Domain Access Permissions (RDC_PDAP29)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8478	Peripheral Domain Access Permissions (RDC_PDAP30)	32	R/W	0000_000Fh	42.6.6/2886
40C7_847C	Peripheral Domain Access Permissions (RDC_PDAP31)	32	R/W	0000_000Fh	42.6.6/2886

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8480	Peripheral Domain Access Permissions (RDC_PDAP32)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8484	Peripheral Domain Access Permissions (RDC_PDAP33)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8488	Peripheral Domain Access Permissions (RDC_PDAP34)	32	R/W	0000_000Fh	42.6.6/2886
40C7_848C	Peripheral Domain Access Permissions (RDC_PDAP35)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8490	Peripheral Domain Access Permissions (RDC_PDAP36)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8494	Peripheral Domain Access Permissions (RDC_PDAP37)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8498	Peripheral Domain Access Permissions (RDC_PDAP38)	32	R/W	0000_000Fh	42.6.6/2886
40C7_849C	Peripheral Domain Access Permissions (RDC_PDAP39)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84A0	Peripheral Domain Access Permissions (RDC_PDAP40)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84A4	Peripheral Domain Access Permissions (RDC_PDAP41)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84A8	Peripheral Domain Access Permissions (RDC_PDAP42)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84AC	Peripheral Domain Access Permissions (RDC_PDAP43)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84B0	Peripheral Domain Access Permissions (RDC_PDAP44)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84B4	Peripheral Domain Access Permissions (RDC_PDAP45)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84B8	Peripheral Domain Access Permissions (RDC_PDAP46)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84BC	Peripheral Domain Access Permissions (RDC_PDAP47)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84C0	Peripheral Domain Access Permissions (RDC_PDAP48)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84C4	Peripheral Domain Access Permissions (RDC_PDAP49)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84C8	Peripheral Domain Access Permissions (RDC_PDAP50)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84CC	Peripheral Domain Access Permissions (RDC_PDAP51)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84D0	Peripheral Domain Access Permissions (RDC_PDAP52)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84D4	Peripheral Domain Access Permissions (RDC_PDAP53)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84D8	Peripheral Domain Access Permissions (RDC_PDAP54)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84DC	Peripheral Domain Access Permissions (RDC_PDAP55)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84E0	Peripheral Domain Access Permissions (RDC_PDAP56)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84E4	Peripheral Domain Access Permissions (RDC_PDAP57)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84E8	Peripheral Domain Access Permissions (RDC_PDAP58)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84EC	Peripheral Domain Access Permissions (RDC_PDAP59)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84F0	Peripheral Domain Access Permissions (RDC_PDAP60)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84F4	Peripheral Domain Access Permissions (RDC_PDAP61)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84F8	Peripheral Domain Access Permissions (RDC_PDAP62)	32	R/W	0000_000Fh	42.6.6/2886
40C7_84FC	Peripheral Domain Access Permissions (RDC_PDAP63)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8500	Peripheral Domain Access Permissions (RDC_PDAP64)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8504	Peripheral Domain Access Permissions (RDC_PDAP65)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8508	Peripheral Domain Access Permissions (RDC_PDAP66)	32	R/W	0000_000Fh	42.6.6/2886
40C7_850C	Peripheral Domain Access Permissions (RDC_PDAP67)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8510	Peripheral Domain Access Permissions (RDC_PDAP68)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8514	Peripheral Domain Access Permissions (RDC_PDAP69)	32	R/W	0000_000Fh	42.6.6/2886

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8518	Peripheral Domain Access Permissions (RDC_PDAP70)	32	R/W	0000_000Fh	42.6.6/2886
40C7_851C	Peripheral Domain Access Permissions (RDC_PDAP71)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8520	Peripheral Domain Access Permissions (RDC_PDAP72)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8524	Peripheral Domain Access Permissions (RDC_PDAP73)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8528	Peripheral Domain Access Permissions (RDC_PDAP74)	32	R/W	0000_000Fh	42.6.6/2886
40C7_852C	Peripheral Domain Access Permissions (RDC_PDAP75)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8530	Peripheral Domain Access Permissions (RDC_PDAP76)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8534	Peripheral Domain Access Permissions (RDC_PDAP77)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8538	Peripheral Domain Access Permissions (RDC_PDAP78)	32	R/W	0000_000Fh	42.6.6/2886
40C7_853C	Peripheral Domain Access Permissions (RDC_PDAP79)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8540	Peripheral Domain Access Permissions (RDC_PDAP80)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8544	Peripheral Domain Access Permissions (RDC_PDAP81)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8548	Peripheral Domain Access Permissions (RDC_PDAP82)	32	R/W	0000_000Fh	42.6.6/2886
40C7_854C	Peripheral Domain Access Permissions (RDC_PDAP83)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8550	Peripheral Domain Access Permissions (RDC_PDAP84)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8554	Peripheral Domain Access Permissions (RDC_PDAP85)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8558	Peripheral Domain Access Permissions (RDC_PDAP86)	32	R/W	0000_000Fh	42.6.6/2886
40C7_855C	Peripheral Domain Access Permissions (RDC_PDAP87)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8560	Peripheral Domain Access Permissions (RDC_PDAP88)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8564	Peripheral Domain Access Permissions (RDC_PDAP89)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8568	Peripheral Domain Access Permissions (RDC_PDAP90)	32	R/W	0000_000Fh	42.6.6/2886
40C7_856C	Peripheral Domain Access Permissions (RDC_PDAP91)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8570	Peripheral Domain Access Permissions (RDC_PDAP92)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8574	Peripheral Domain Access Permissions (RDC_PDAP93)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8578	Peripheral Domain Access Permissions (RDC_PDAP94)	32	R/W	0000_000Fh	42.6.6/2886
40C7_857C	Peripheral Domain Access Permissions (RDC_PDAP95)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8580	Peripheral Domain Access Permissions (RDC_PDAP96)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8584	Peripheral Domain Access Permissions (RDC_PDAP97)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8588	Peripheral Domain Access Permissions (RDC_PDAP98)	32	R/W	0000_000Fh	42.6.6/2886
40C7_858C	Peripheral Domain Access Permissions (RDC_PDAP99)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8590	Peripheral Domain Access Permissions (RDC_PDAP100)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8594	Peripheral Domain Access Permissions (RDC_PDAP101)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8598	Peripheral Domain Access Permissions (RDC_PDAP102)	32	R/W	0000_000Fh	42.6.6/2886
40C7_859C	Peripheral Domain Access Permissions (RDC_PDAP103)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85A0	Peripheral Domain Access Permissions (RDC_PDAP104)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85A4	Peripheral Domain Access Permissions (RDC_PDAP105)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85A8	Peripheral Domain Access Permissions (RDC_PDAP106)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85AC	Peripheral Domain Access Permissions (RDC_PDAP107)	32	R/W	0000_000Fh	42.6.6/2886

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_85B0	Peripheral Domain Access Permissions (RDC_PDAP108)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85B4	Peripheral Domain Access Permissions (RDC_PDAP109)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85B8	Peripheral Domain Access Permissions (RDC_PDAP110)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85BC	Peripheral Domain Access Permissions (RDC_PDAP111)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85C0	Peripheral Domain Access Permissions (RDC_PDAP112)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85C4	Peripheral Domain Access Permissions (RDC_PDAP113)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85C8	Peripheral Domain Access Permissions (RDC_PDAP114)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85CC	Peripheral Domain Access Permissions (RDC_PDAP115)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85D0	Peripheral Domain Access Permissions (RDC_PDAP116)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85D4	Peripheral Domain Access Permissions (RDC_PDAP117)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85D8	Peripheral Domain Access Permissions (RDC_PDAP118)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85DC	Peripheral Domain Access Permissions (RDC_PDAP119)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85E0	Peripheral Domain Access Permissions (RDC_PDAP120)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85E4	Peripheral Domain Access Permissions (RDC_PDAP121)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85E8	Peripheral Domain Access Permissions (RDC_PDAP122)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85EC	Peripheral Domain Access Permissions (RDC_PDAP123)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85F0	Peripheral Domain Access Permissions (RDC_PDAP124)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85F4	Peripheral Domain Access Permissions (RDC_PDAP125)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85F8	Peripheral Domain Access Permissions (RDC_PDAP126)	32	R/W	0000_000Fh	42.6.6/2886
40C7_85FC	Peripheral Domain Access Permissions (RDC_PDAP127)	32	R/W	0000_000Fh	42.6.6/2886
40C7_8800	Memory Region Start Address (RDC_MRSA0)	32	R/W	Undefined	42.6.7/2887
40C7_8804	Memory Region End Address (RDC_MREA0)	32	R/W	Undefined	42.6.8/2888
40C7_8808	Memory Region Control (RDC_MRC0)	32	R/W	0000_00FFh	42.6.9/2888
40C7_880C	Memory Region Violation Status (RDC_MRVS0)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8810	Memory Region Start Address (RDC_MRSA1)	32	R/W	Undefined	42.6.7/2887
40C7_8814	Memory Region End Address (RDC_MREA1)	32	R/W	Undefined	42.6.8/2888
40C7_8818	Memory Region Control (RDC_MRC1)	32	R/W	0000_00FFh	42.6.9/2888
40C7_881C	Memory Region Violation Status (RDC_MRVS1)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8820	Memory Region Start Address (RDC_MRSA2)	32	R/W	Undefined	42.6.7/2887
40C7_8824	Memory Region End Address (RDC_MREA2)	32	R/W	Undefined	42.6.8/2888
40C7_8828	Memory Region Control (RDC_MRC2)	32	R/W	0000_00FFh	42.6.9/2888
40C7_882C	Memory Region Violation Status (RDC_MRVS2)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8830	Memory Region Start Address (RDC_MRSA3)	32	R/W	Undefined	42.6.7/2887
40C7_8834	Memory Region End Address (RDC_MREA3)	32	R/W	Undefined	42.6.8/2888
40C7_8838	Memory Region Control (RDC_MRC3)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_883C	Memory Region Violation Status (RDC_MRVS3)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8840	Memory Region Start Address (RDC_MRSA4)	32	R/W	Undefined	42.6.7/2887
40C7_8844	Memory Region End Address (RDC_MREA4)	32	R/W	Undefined	42.6.8/2888
40C7_8848	Memory Region Control (RDC_MRC4)	32	R/W	0000_00FFh	42.6.9/2888
40C7_884C	Memory Region Violation Status (RDC_MRVS4)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8850	Memory Region Start Address (RDC_MRSA5)	32	R/W	Undefined	42.6.7/2887
40C7_8854	Memory Region End Address (RDC_MREA5)	32	R/W	Undefined	42.6.8/2888
40C7_8858	Memory Region Control (RDC_MRC5)	32	R/W	0000_00FFh	42.6.9/2888
40C7_885C	Memory Region Violation Status (RDC_MRVS5)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8860	Memory Region Start Address (RDC_MRSA6)	32	R/W	Undefined	42.6.7/2887
40C7_8864	Memory Region End Address (RDC_MREA6)	32	R/W	Undefined	42.6.8/2888
40C7_8868	Memory Region Control (RDC_MRC6)	32	R/W	0000_00FFh	42.6.9/2888
40C7_886C	Memory Region Violation Status (RDC_MRVS6)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8870	Memory Region Start Address (RDC_MRSA7)	32	R/W	Undefined	42.6.7/2887
40C7_8874	Memory Region End Address (RDC_MREA7)	32	R/W	Undefined	42.6.8/2888
40C7_8878	Memory Region Control (RDC_MRC7)	32	R/W	0000_00FFh	42.6.9/2888
40C7_887C	Memory Region Violation Status (RDC_MRVS7)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8880	Memory Region Start Address (RDC_MRSA8)	32	R/W	Undefined	42.6.7/2887
40C7_8884	Memory Region End Address (RDC_MREA8)	32	R/W	Undefined	42.6.8/2888
40C7_8888	Memory Region Control (RDC_MRC8)	32	R/W	0000_00FFh	42.6.9/2888
40C7_888C	Memory Region Violation Status (RDC_MRVS8)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8890	Memory Region Start Address (RDC_MRSA9)	32	R/W	Undefined	42.6.7/2887
40C7_8894	Memory Region End Address (RDC_MREA9)	32	R/W	Undefined	42.6.8/2888
40C7_8898	Memory Region Control (RDC_MRC9)	32	R/W	0000_00FFh	42.6.9/2888
40C7_889C	Memory Region Violation Status (RDC_MRVS9)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88A0	Memory Region Start Address (RDC_MRSA10)	32	R/W	Undefined	42.6.7/2887
40C7_88A4	Memory Region End Address (RDC_MREA10)	32	R/W	Undefined	42.6.8/2888
40C7_88A8	Memory Region Control (RDC_MRC10)	32	R/W	0000_00FFh	42.6.9/2888
40C7_88AC	Memory Region Violation Status (RDC_MRVS10)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88B0	Memory Region Start Address (RDC_MRSA11)	32	R/W	Undefined	42.6.7/2887
40C7_88B4	Memory Region End Address (RDC_MREA11)	32	R/W	Undefined	42.6.8/2888
40C7_88B8	Memory Region Control (RDC_MRC11)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_88BC	Memory Region Violation Status (RDC_MRVS11)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88C0	Memory Region Start Address (RDC_MRSA12)	32	R/W	Undefined	42.6.7/2887
40C7_88C4	Memory Region End Address (RDC_MREA12)	32	R/W	Undefined	42.6.8/2888
40C7_88C8	Memory Region Control (RDC_MRC12)	32	R/W	0000_00FFh	42.6.9/2888
40C7_88CC	Memory Region Violation Status (RDC_MRVS12)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88D0	Memory Region Start Address (RDC_MRSA13)	32	R/W	Undefined	42.6.7/2887
40C7_88D4	Memory Region End Address (RDC_MREA13)	32	R/W	Undefined	42.6.8/2888
40C7_88D8	Memory Region Control (RDC_MRC13)	32	R/W	0000_00FFh	42.6.9/2888
40C7_88DC	Memory Region Violation Status (RDC_MRVS13)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88E0	Memory Region Start Address (RDC_MRSA14)	32	R/W	Undefined	42.6.7/2887
40C7_88E4	Memory Region End Address (RDC_MREA14)	32	R/W	Undefined	42.6.8/2888
40C7_88E8	Memory Region Control (RDC_MRC14)	32	R/W	0000_00FFh	42.6.9/2888
40C7_88EC	Memory Region Violation Status (RDC_MRVS14)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_88F0	Memory Region Start Address (RDC_MRSA15)	32	R/W	Undefined	42.6.7/2887
40C7_88F4	Memory Region End Address (RDC_MREA15)	32	R/W	Undefined	42.6.8/2888
40C7_88F8	Memory Region Control (RDC_MRC15)	32	R/W	0000_00FFh	42.6.9/2888
40C7_88FC	Memory Region Violation Status (RDC_MRVS15)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8900	Memory Region Start Address (RDC_MRSA16)	32	R/W	Undefined	42.6.7/2887
40C7_8904	Memory Region End Address (RDC_MREA16)	32	R/W	Undefined	42.6.8/2888
40C7_8908	Memory Region Control (RDC_MRC16)	32	R/W	0000_00FFh	42.6.9/2888
40C7_890C	Memory Region Violation Status (RDC_MRVS16)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8910	Memory Region Start Address (RDC_MRSA17)	32	R/W	Undefined	42.6.7/2887
40C7_8914	Memory Region End Address (RDC_MREA17)	32	R/W	Undefined	42.6.8/2888
40C7_8918	Memory Region Control (RDC_MRC17)	32	R/W	0000_00FFh	42.6.9/2888
40C7_891C	Memory Region Violation Status (RDC_MRVS17)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8920	Memory Region Start Address (RDC_MRSA18)	32	R/W	Undefined	42.6.7/2887
40C7_8924	Memory Region End Address (RDC_MREA18)	32	R/W	Undefined	42.6.8/2888
40C7_8928	Memory Region Control (RDC_MRC18)	32	R/W	0000_00FFh	42.6.9/2888
40C7_892C	Memory Region Violation Status (RDC_MRVS18)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8930	Memory Region Start Address (RDC_MRSA19)	32	R/W	Undefined	42.6.7/2887
40C7_8934	Memory Region End Address (RDC_MREA19)	32	R/W	Undefined	42.6.8/2888
40C7_8938	Memory Region Control (RDC_MRC19)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_893C	Memory Region Violation Status (RDC_MRVS19)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8940	Memory Region Start Address (RDC_MRSA20)	32	R/W	Undefined	42.6.7/2887
40C7_8944	Memory Region End Address (RDC_MREA20)	32	R/W	Undefined	42.6.8/2888
40C7_8948	Memory Region Control (RDC_MRC20)	32	R/W	0000_00FFh	42.6.9/2888
40C7_894C	Memory Region Violation Status (RDC_MRVS20)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8950	Memory Region Start Address (RDC_MRSA21)	32	R/W	Undefined	42.6.7/2887
40C7_8954	Memory Region End Address (RDC_MREA21)	32	R/W	Undefined	42.6.8/2888
40C7_8958	Memory Region Control (RDC_MRC21)	32	R/W	0000_00FFh	42.6.9/2888
40C7_895C	Memory Region Violation Status (RDC_MRVS21)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8960	Memory Region Start Address (RDC_MRSA22)	32	R/W	Undefined	42.6.7/2887
40C7_8964	Memory Region End Address (RDC_MREA22)	32	R/W	Undefined	42.6.8/2888
40C7_8968	Memory Region Control (RDC_MRC22)	32	R/W	0000_00FFh	42.6.9/2888
40C7_896C	Memory Region Violation Status (RDC_MRVS22)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8970	Memory Region Start Address (RDC_MRSA23)	32	R/W	Undefined	42.6.7/2887
40C7_8974	Memory Region End Address (RDC_MREA23)	32	R/W	Undefined	42.6.8/2888
40C7_8978	Memory Region Control (RDC_MRC23)	32	R/W	0000_00FFh	42.6.9/2888
40C7_897C	Memory Region Violation Status (RDC_MRVS23)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8980	Memory Region Start Address (RDC_MRSA24)	32	R/W	Undefined	42.6.7/2887
40C7_8984	Memory Region End Address (RDC_MREA24)	32	R/W	Undefined	42.6.8/2888
40C7_8988	Memory Region Control (RDC_MRC24)	32	R/W	0000_00FFh	42.6.9/2888
40C7_898C	Memory Region Violation Status (RDC_MRVS24)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8990	Memory Region Start Address (RDC_MRSA25)	32	R/W	Undefined	42.6.7/2887
40C7_8994	Memory Region End Address (RDC_MREA25)	32	R/W	Undefined	42.6.8/2888
40C7_8998	Memory Region Control (RDC_MRC25)	32	R/W	0000_00FFh	42.6.9/2888
40C7_899C	Memory Region Violation Status (RDC_MRVS25)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89A0	Memory Region Start Address (RDC_MRSA26)	32	R/W	Undefined	42.6.7/2887
40C7_89A4	Memory Region End Address (RDC_MREA26)	32	R/W	Undefined	42.6.8/2888
40C7_89A8	Memory Region Control (RDC_MRC26)	32	R/W	0000_00FFh	42.6.9/2888
40C7_89AC	Memory Region Violation Status (RDC_MRVS26)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89B0	Memory Region Start Address (RDC_MRSA27)	32	R/W	Undefined	42.6.7/2887
40C7_89B4	Memory Region End Address (RDC_MREA27)	32	R/W	Undefined	42.6.8/2888
40C7_89B8	Memory Region Control (RDC_MRC27)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_89BC	Memory Region Violation Status (RDC_MRVS27)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89C0	Memory Region Start Address (RDC_MRSA28)	32	R/W	Undefined	42.6.7/2887
40C7_89C4	Memory Region End Address (RDC_MREA28)	32	R/W	Undefined	42.6.8/2888
40C7_89C8	Memory Region Control (RDC_MRC28)	32	R/W	0000_00FFh	42.6.9/2888
40C7_89CC	Memory Region Violation Status (RDC_MRVS28)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89D0	Memory Region Start Address (RDC_MRSA29)	32	R/W	Undefined	42.6.7/2887
40C7_89D4	Memory Region End Address (RDC_MREA29)	32	R/W	Undefined	42.6.8/2888
40C7_89D8	Memory Region Control (RDC_MRC29)	32	R/W	0000_00FFh	42.6.9/2888
40C7_89DC	Memory Region Violation Status (RDC_MRVS29)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89E0	Memory Region Start Address (RDC_MRSA30)	32	R/W	Undefined	42.6.7/2887
40C7_89E4	Memory Region End Address (RDC_MREA30)	32	R/W	Undefined	42.6.8/2888
40C7_89E8	Memory Region Control (RDC_MRC30)	32	R/W	0000_00FFh	42.6.9/2888
40C7_89EC	Memory Region Violation Status (RDC_MRVS30)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_89F0	Memory Region Start Address (RDC_MRSA31)	32	R/W	Undefined	42.6.7/2887
40C7_89F4	Memory Region End Address (RDC_MREA31)	32	R/W	Undefined	42.6.8/2888
40C7_89F8	Memory Region Control (RDC_MRC31)	32	R/W	0000_00FFh	42.6.9/2888
40C7_89FC	Memory Region Violation Status (RDC_MRVS31)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A00	Memory Region Start Address (RDC_MRSA32)	32	R/W	Undefined	42.6.7/2887
40C7_8A04	Memory Region End Address (RDC_MREA32)	32	R/W	Undefined	42.6.8/2888
40C7_8A08	Memory Region Control (RDC_MRC32)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A0C	Memory Region Violation Status (RDC_MRVS32)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A10	Memory Region Start Address (RDC_MRSA33)	32	R/W	Undefined	42.6.7/2887
40C7_8A14	Memory Region End Address (RDC_MREA33)	32	R/W	Undefined	42.6.8/2888
40C7_8A18	Memory Region Control (RDC_MRC33)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A1C	Memory Region Violation Status (RDC_MRVS33)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A20	Memory Region Start Address (RDC_MRSA34)	32	R/W	Undefined	42.6.7/2887
40C7_8A24	Memory Region End Address (RDC_MREA34)	32	R/W	Undefined	42.6.8/2888
40C7_8A28	Memory Region Control (RDC_MRC34)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A2C	Memory Region Violation Status (RDC_MRVS34)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A30	Memory Region Start Address (RDC_MRSA35)	32	R/W	Undefined	42.6.7/2887
40C7_8A34	Memory Region End Address (RDC_MREA35)	32	R/W	Undefined	42.6.8/2888
40C7_8A38	Memory Region Control (RDC_MRC35)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8A3C	Memory Region Violation Status (RDC_MRVS35)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A40	Memory Region Start Address (RDC_MRSA36)	32	R/W	Undefined	42.6.7/2887
40C7_8A44	Memory Region End Address (RDC_MREA36)	32	R/W	Undefined	42.6.8/2888
40C7_8A48	Memory Region Control (RDC_MRC36)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A4C	Memory Region Violation Status (RDC_MRVS36)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A50	Memory Region Start Address (RDC_MRSA37)	32	R/W	Undefined	42.6.7/2887
40C7_8A54	Memory Region End Address (RDC_MREA37)	32	R/W	Undefined	42.6.8/2888
40C7_8A58	Memory Region Control (RDC_MRC37)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A5C	Memory Region Violation Status (RDC_MRVS37)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A60	Memory Region Start Address (RDC_MRSA38)	32	R/W	Undefined	42.6.7/2887
40C7_8A64	Memory Region End Address (RDC_MREA38)	32	R/W	Undefined	42.6.8/2888
40C7_8A68	Memory Region Control (RDC_MRC38)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A6C	Memory Region Violation Status (RDC_MRVS38)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A70	Memory Region Start Address (RDC_MRSA39)	32	R/W	Undefined	42.6.7/2887
40C7_8A74	Memory Region End Address (RDC_MREA39)	32	R/W	Undefined	42.6.8/2888
40C7_8A78	Memory Region Control (RDC_MRC39)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A7C	Memory Region Violation Status (RDC_MRVS39)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A80	Memory Region Start Address (RDC_MRSA40)	32	R/W	Undefined	42.6.7/2887
40C7_8A84	Memory Region End Address (RDC_MREA40)	32	R/W	Undefined	42.6.8/2888
40C7_8A88	Memory Region Control (RDC_MRC40)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A8C	Memory Region Violation Status (RDC_MRVS40)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8A90	Memory Region Start Address (RDC_MRSA41)	32	R/W	Undefined	42.6.7/2887
40C7_8A94	Memory Region End Address (RDC_MREA41)	32	R/W	Undefined	42.6.8/2888
40C7_8A98	Memory Region Control (RDC_MRC41)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8A9C	Memory Region Violation Status (RDC_MRVS41)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AA0	Memory Region Start Address (RDC_MRSA42)	32	R/W	Undefined	42.6.7/2887
40C7_8AA4	Memory Region End Address (RDC_MREA42)	32	R/W	Undefined	42.6.8/2888
40C7_8AA8	Memory Region Control (RDC_MRC42)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8AAC	Memory Region Violation Status (RDC_MRVS42)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AB0	Memory Region Start Address (RDC_MRSA43)	32	R/W	Undefined	42.6.7/2887
40C7_8AB4	Memory Region End Address (RDC_MREA43)	32	R/W	Undefined	42.6.8/2888
40C7_8AB8	Memory Region Control (RDC_MRC43)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8ABC	Memory Region Violation Status (RDC_MRVS43)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AC0	Memory Region Start Address (RDC_MRSA44)	32	R/W	Undefined	42.6.7/2887
40C7_8AC4	Memory Region End Address (RDC_MREA44)	32	R/W	Undefined	42.6.8/2888
40C7_8AC8	Memory Region Control (RDC_MRC44)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8ACC	Memory Region Violation Status (RDC_MRVS44)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AD0	Memory Region Start Address (RDC_MRSA45)	32	R/W	Undefined	42.6.7/2887
40C7_8AD4	Memory Region End Address (RDC_MREA45)	32	R/W	Undefined	42.6.8/2888
40C7_8AD8	Memory Region Control (RDC_MRC45)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8ADC	Memory Region Violation Status (RDC_MRVS45)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AE0	Memory Region Start Address (RDC_MRSA46)	32	R/W	Undefined	42.6.7/2887
40C7_8AE4	Memory Region End Address (RDC_MREA46)	32	R/W	Undefined	42.6.8/2888
40C7_8AE8	Memory Region Control (RDC_MRC46)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8AEC	Memory Region Violation Status (RDC_MRVS46)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8AF0	Memory Region Start Address (RDC_MRSA47)	32	R/W	Undefined	42.6.7/2887
40C7_8AF4	Memory Region End Address (RDC_MREA47)	32	R/W	Undefined	42.6.8/2888
40C7_8AF8	Memory Region Control (RDC_MRC47)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8AFC	Memory Region Violation Status (RDC_MRVS47)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B00	Memory Region Start Address (RDC_MRSA48)	32	R/W	Undefined	42.6.7/2887
40C7_8B04	Memory Region End Address (RDC_MREA48)	32	R/W	Undefined	42.6.8/2888
40C7_8B08	Memory Region Control (RDC_MRC48)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B0C	Memory Region Violation Status (RDC_MRVS48)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B10	Memory Region Start Address (RDC_MRSA49)	32	R/W	Undefined	42.6.7/2887
40C7_8B14	Memory Region End Address (RDC_MREA49)	32	R/W	Undefined	42.6.8/2888
40C7_8B18	Memory Region Control (RDC_MRC49)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B1C	Memory Region Violation Status (RDC_MRVS49)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B20	Memory Region Start Address (RDC_MRSA50)	32	R/W	Undefined	42.6.7/2887
40C7_8B24	Memory Region End Address (RDC_MREA50)	32	R/W	Undefined	42.6.8/2888
40C7_8B28	Memory Region Control (RDC_MRC50)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B2C	Memory Region Violation Status (RDC_MRVS50)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B30	Memory Region Start Address (RDC_MRSA51)	32	R/W	Undefined	42.6.7/2887
40C7_8B34	Memory Region End Address (RDC_MREA51)	32	R/W	Undefined	42.6.8/2888
40C7_8B38	Memory Region Control (RDC_MRC51)	32	R/W	0000_00FFh	42.6.9/2888

Table continues on the next page...

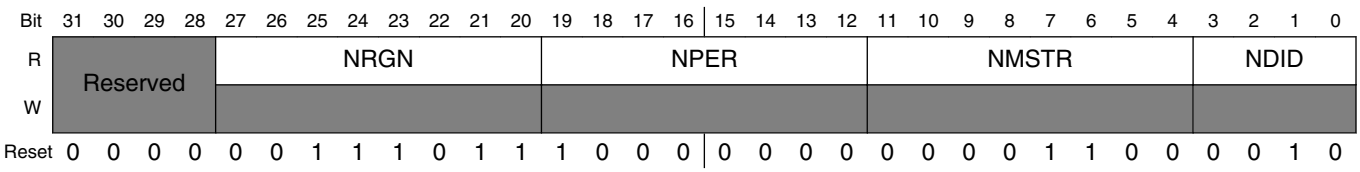
RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C7_8B3C	Memory Region Violation Status (RDC_MRVS51)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B40	Memory Region Start Address (RDC_MRSA52)	32	R/W	Undefined	42.6.7/2887
40C7_8B44	Memory Region End Address (RDC_MREA52)	32	R/W	Undefined	42.6.8/2888
40C7_8B48	Memory Region Control (RDC_MRC52)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B4C	Memory Region Violation Status (RDC_MRVS52)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B50	Memory Region Start Address (RDC_MRSA53)	32	R/W	Undefined	42.6.7/2887
40C7_8B54	Memory Region End Address (RDC_MREA53)	32	R/W	Undefined	42.6.8/2888
40C7_8B58	Memory Region Control (RDC_MRC53)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B5C	Memory Region Violation Status (RDC_MRVS53)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B60	Memory Region Start Address (RDC_MRSA54)	32	R/W	Undefined	42.6.7/2887
40C7_8B64	Memory Region End Address (RDC_MREA54)	32	R/W	Undefined	42.6.8/2888
40C7_8B68	Memory Region Control (RDC_MRC54)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B6C	Memory Region Violation Status (RDC_MRVS54)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B70	Memory Region Start Address (RDC_MRSA55)	32	R/W	Undefined	42.6.7/2887
40C7_8B74	Memory Region End Address (RDC_MREA55)	32	R/W	Undefined	42.6.8/2888
40C7_8B78	Memory Region Control (RDC_MRC55)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B7C	Memory Region Violation Status (RDC_MRVS55)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B80	Memory Region Start Address (RDC_MRSA56)	32	R/W	Undefined	42.6.7/2887
40C7_8B84	Memory Region End Address (RDC_MREA56)	32	R/W	Undefined	42.6.8/2888
40C7_8B88	Memory Region Control (RDC_MRC56)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B8C	Memory Region Violation Status (RDC_MRVS56)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8B90	Memory Region Start Address (RDC_MRSA57)	32	R/W	Undefined	42.6.7/2887
40C7_8B94	Memory Region End Address (RDC_MREA57)	32	R/W	Undefined	42.6.8/2888
40C7_8B98	Memory Region Control (RDC_MRC57)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8B9C	Memory Region Violation Status (RDC_MRVS57)	32	R/W	0000_0000h	42.6.10/ 2889
40C7_8BA0	Memory Region Start Address (RDC_MRSA58)	32	R/W	Undefined	42.6.7/2887
40C7_8BA4	Memory Region End Address (RDC_MREA58)	32	R/W	Undefined	42.6.8/2888
40C7_8BA8	Memory Region Control (RDC_MRC58)	32	R/W	0000_00FFh	42.6.9/2888
40C7_8BAC	Memory Region Violation Status (RDC_MRVS58)	32	R/W	0000_0000h	42.6.10/ 2889

42.6.1 Version Information (RDC_VIR)

The VIR provides version information including the number of domains, number of master slots, number of peripheral slots, and number of memory regions.

Address: 40C7_8000h base + 0h offset = 40C7_8000h

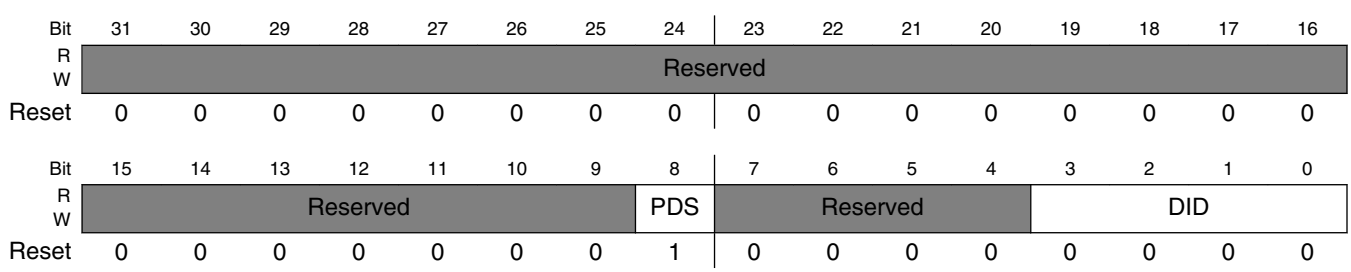


RDC_VIR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–20 NRGN	Number of Memory Regions Indicates the number of memory regions in this instance of the RDC.
19–12 NPER	Number of Peripherals Indicates the number of peripherals that can be isolated or safe-shared.
11–4 NMSTR	Number of Masters Indicates the number of masters supported by this instance of RDC.
NDID	Number of Domains Indicates the number of domain IDs supported by this instance of the RDC. For example, value '0010' means the actual number of domains is 2.

42.6.2 Status (RDC_STAT)

Address: 40C7_8000h base + 24h offset = 40C7_8024h

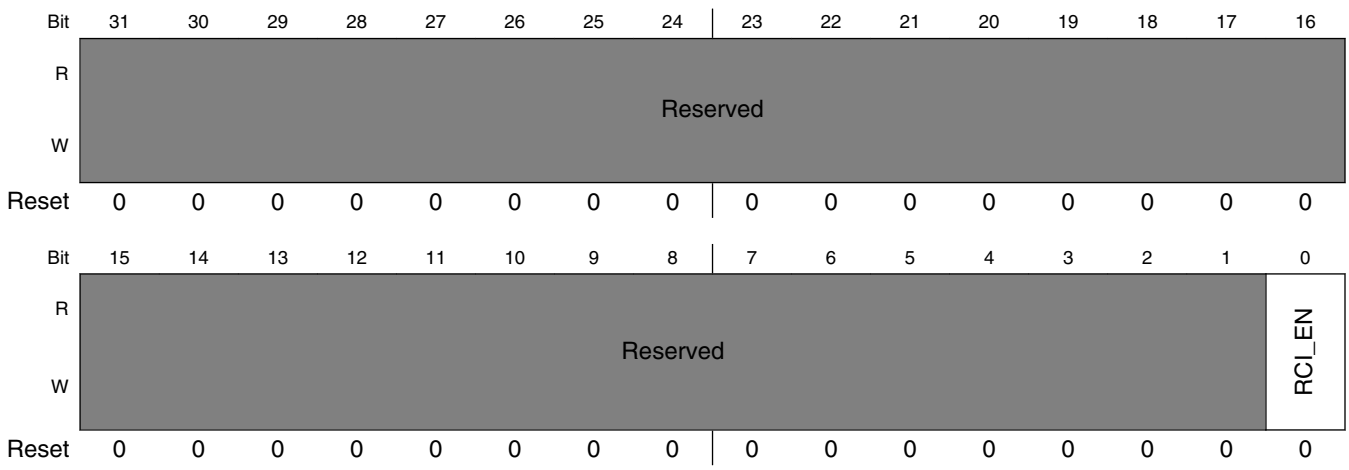


RDC_STAT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 PDS	Power Domain Status Indicates if the "Power Down" memory regions are powered and available. Power Down memory regions are only those memory regions susceptible to power outage for power savings are unavailable if this is zero. "Always-On" memory regions remain available. Always On memory regions are those regions that are not powered down unless the entire SoC is powered down. This signal remains low until all access controls have been restored to the domain. 0 Power Down Domain is OFF 1 Power Down Domain is ON
7–4 Reserved	This field is reserved.
DID	Domain ID The Domain ID of the core or bus master that is reading this. The value is different for requests from different domains.

42.6.3 Interrupt and Control (RDC_INTCTRL)

Address: 40C7_8000h base + 28h offset = 40C7_8028h



RDC_INTCTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 RCI_EN	Restoration Complete Interrupt Interrupt generated when the RDC has completed restoring state to a recently re-powered memory regions.

Table continues on the next page...

RDC_INTCTRL field descriptions (continued)

Field	Description
	0 Interrupt Disabled 1 Interrupt Enabled

42.6.4 Interrupt Status (RDC_INTSTAT)

Indication of Interrupt Pending for State Restoration

Address: 40C7_8000h base + 2Ch offset = 40C7_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															INT w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_INTSTAT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 INT	Interrupt Status Indicates state of interrupt signal for state restoration. This is that status of the interrupt enabled in RDC_INTCTRL. Write one to interrupt status to clear it. 0 No Interrupt Pending 1 Interrupt Pending

42.6.5 Master Domain Assignment (RDC_MDA*n*)

Address: 40C7_8000h base + 200h offset + (4d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DID	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_MDA*n* field descriptions

Field	Description
31 LCK	Assignment Lock 0 Not Locked 1 Locked
30–2 Reserved	This field is reserved.
DID	Domain ID Indicates the domain to which the Master is assigned 00 Master assigned to Processing Domain 0 01 Master assigned to Processing Domain 1 10 Reserved 11 Reserved

42.6.6 Peripheral Domain Access Permissions (RDC_PDAP*n*)

Address: 40C7_8000h base + 400h offset + (4d × i), where i=0d to 127d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	SREQ	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved				D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

RDC_PDAP*n* field descriptions

Field	Description
31 LCK	Peripheral Permissions Lock When set prevents further modification of the Peripheral Domain Access Permissions (sticky bit until reset) 0 Not Locked 1 Locked
30 SREQ	Semaphore Required When set the hardware semaphore state enforces the semaphore lock. If a domain has access permissions and a semaphore has locked a shared peripheral then only the domain holding the semaphore signal can access this peripheral.

Table continues on the next page...

RDC_PDAP*n* field descriptions (continued)

Field	Description
	0 Semaphores have no effect 1 Semaphores are enforced
29–8 Reserved	This field is reserved.
7–4 Reserved	This field is reserved.
3 D1R	Domain 1 Read Access 0 No Read Access 1 Read Access Allowed
2 D1W	Domain 1 Write Access 0 No Write Access 1 Write Access Allowed
1 D0R	Domain 0 Read Access 0 No Read Access 1 Read Access Allowed
0 D0W	Domain 0 Write Access 0 No Write Access 1 Write Access Allowed

42.6.7 Memory Region Start Address (RDC_MRSA*n*)

Address: 40C7_8000h base + 800h offset + (16d × i), where i=0d to 58d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

RDC_MRSA*n* field descriptions

Field	Description
31–7 SADR	Start address for memory region Lower bound (inclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

42.6.8 Memory Region End Address (RDC_MREA*n*)

Address: 40C7_8000h base + 804h offset + (16d × i), where i=0d to 58d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

RDC_MREA*n* field descriptions

Field	Description
31–7 EADR	Upper bound for memory region Upper bound (exclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

42.6.9 Memory Region Control (RDC_MRC*n*)

Address: 40C7_8000h base + 808h offset + (16d × i), where i=0d to 58d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	ENA	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved				D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RDC_MRC*n* field descriptions

Field	Description
31 LCK	Region Lock Locks all region fields from further modification except ENA, which can be set but not reset after LCK is set. LCK is a sticky bit. 0 No Lock. All fields in this register may be modified. 1 Locked. No fields in this register may be modified except ENA, which may be set but not cleared.
30 ENA	Region Enable Activates the memory region. If the region is not activated then the permissions and address boundaries have not affect and the region will be fully accessible.

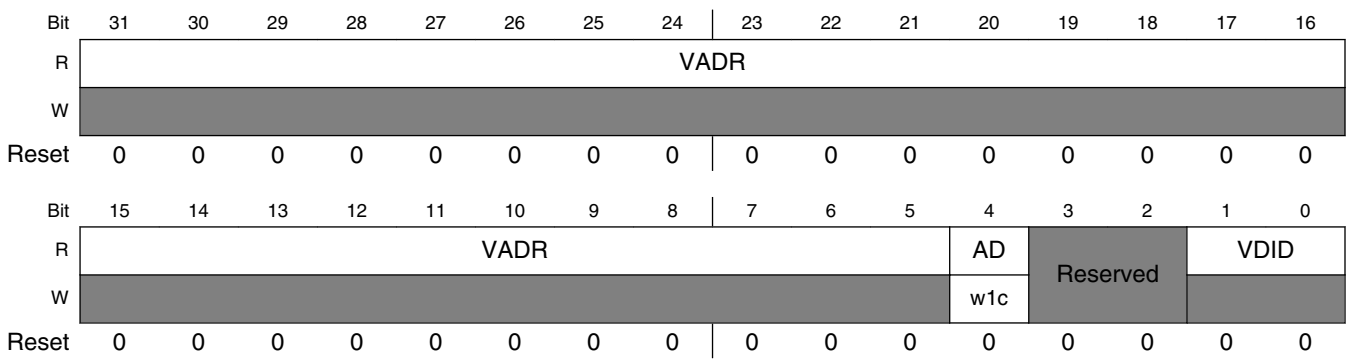
Table continues on the next page...

RDC_MRC*n* field descriptions (continued)

Field	Description
	0 Memory region is not defined or restricted. 1 Memory boundaries, domain permissions and controls are in effect.
29–8 Reserved	This field is reserved.
7–4 Reserved	This field is reserved.
3 D1R	Domain 1 Read Access to Region 0 Processing Domain 1 does not have Read access to the memory region 1 Processing Domain 1 has Read access to the memory region
2 D1W	Domain 1 Write Access to Region 0 Processing Domain 1 does not have Write access to the memory region 1 Processing Domain 1 has Write access to the memory region
1 D0R	Domain 0 Read Access to Region 0 Processing Domain 0 does not have Read access to the memory region 1 Processing Domain 0 has Read access to the memory region
0 D0W	Domain 0 Write Access to Region 0 Processing Domain 0 does not have Write access to the memory region 1 Processing Domain 0 has Write access to the memory region

42.6.10 Memory Region Violation Status (RDC_MRVS*n*)

Address: 40C7_8000h base + 80Ch offset + (16d × i), where i=0d to 58d



RDC_MRVS*n* field descriptions

Field	Description
31–5 VADR	Violating Address The address of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. Clearing of contents occurs only when the status is read by the memory region's associated domain ID (s).

Table continues on the next page...

RDC_MRVS*n* field descriptions (continued)

Field	Description
4 AD	Access Denied Access to a memory region denied. This bit is cleared when this bit is written by one of the allowed domains.
3–2 Reserved	This field is reserved.
VDID	Violating Domain ID The domain ID of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. 00 Processing Domain 0 01 Processing Domain 1 10 Reserved 11 Reserved

42.7 RDC SEMA42 Memory Map/Register Definition

Only Supervisor Mode accesses are allowed on these registers. User accesses generate an error termination.

RDC_SEMAPHORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C4_4000	Gate Register (RDC_SEMAPHORE1_GATE0)	8	R/W	00h	42.7.1/2894
40C4_4001	Gate Register (RDC_SEMAPHORE1_GATE1)	8	R/W	00h	42.7.1/2894
40C4_4002	Gate Register (RDC_SEMAPHORE1_GATE2)	8	R/W	00h	42.7.1/2894
40C4_4003	Gate Register (RDC_SEMAPHORE1_GATE3)	8	R/W	00h	42.7.1/2894
40C4_4004	Gate Register (RDC_SEMAPHORE1_GATE4)	8	R/W	00h	42.7.1/2894
40C4_4005	Gate Register (RDC_SEMAPHORE1_GATE5)	8	R/W	00h	42.7.1/2894
40C4_4006	Gate Register (RDC_SEMAPHORE1_GATE6)	8	R/W	00h	42.7.1/2894
40C4_4007	Gate Register (RDC_SEMAPHORE1_GATE7)	8	R/W	00h	42.7.1/2894
40C4_4008	Gate Register (RDC_SEMAPHORE1_GATE8)	8	R/W	00h	42.7.1/2894
40C4_4009	Gate Register (RDC_SEMAPHORE1_GATE9)	8	R/W	00h	42.7.1/2894
40C4_400A	Gate Register (RDC_SEMAPHORE1_GATE10)	8	R/W	00h	42.7.1/2894
40C4_400B	Gate Register (RDC_SEMAPHORE1_GATE11)	8	R/W	00h	42.7.1/2894
40C4_400C	Gate Register (RDC_SEMAPHORE1_GATE12)	8	R/W	00h	42.7.1/2894
40C4_400D	Gate Register (RDC_SEMAPHORE1_GATE13)	8	R/W	00h	42.7.1/2894
40C4_400E	Gate Register (RDC_SEMAPHORE1_GATE14)	8	R/W	00h	42.7.1/2894

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C4_400F	Gate Register (RDC_SEMAPHORE1_GATE15)	8	R/W	00h	42.7.1/2894
40C4_4010	Gate Register (RDC_SEMAPHORE1_GATE16)	8	R/W	00h	42.7.1/2894
40C4_4011	Gate Register (RDC_SEMAPHORE1_GATE17)	8	R/W	00h	42.7.1/2894
40C4_4012	Gate Register (RDC_SEMAPHORE1_GATE18)	8	R/W	00h	42.7.1/2894
40C4_4013	Gate Register (RDC_SEMAPHORE1_GATE19)	8	R/W	00h	42.7.1/2894
40C4_4014	Gate Register (RDC_SEMAPHORE1_GATE20)	8	R/W	00h	42.7.1/2894
40C4_4015	Gate Register (RDC_SEMAPHORE1_GATE21)	8	R/W	00h	42.7.1/2894
40C4_4016	Gate Register (RDC_SEMAPHORE1_GATE22)	8	R/W	00h	42.7.1/2894
40C4_4017	Gate Register (RDC_SEMAPHORE1_GATE23)	8	R/W	00h	42.7.1/2894
40C4_4018	Gate Register (RDC_SEMAPHORE1_GATE24)	8	R/W	00h	42.7.1/2894
40C4_4019	Gate Register (RDC_SEMAPHORE1_GATE25)	8	R/W	00h	42.7.1/2894
40C4_401A	Gate Register (RDC_SEMAPHORE1_GATE26)	8	R/W	00h	42.7.1/2894
40C4_401B	Gate Register (RDC_SEMAPHORE1_GATE27)	8	R/W	00h	42.7.1/2894
40C4_401C	Gate Register (RDC_SEMAPHORE1_GATE28)	8	R/W	00h	42.7.1/2894
40C4_401D	Gate Register (RDC_SEMAPHORE1_GATE29)	8	R/W	00h	42.7.1/2894
40C4_401E	Gate Register (RDC_SEMAPHORE1_GATE30)	8	R/W	00h	42.7.1/2894
40C4_401F	Gate Register (RDC_SEMAPHORE1_GATE31)	8	R/W	00h	42.7.1/2894
40C4_4020	Gate Register (RDC_SEMAPHORE1_GATE32)	8	R/W	00h	42.7.1/2894
40C4_4021	Gate Register (RDC_SEMAPHORE1_GATE33)	8	R/W	00h	42.7.1/2894
40C4_4022	Gate Register (RDC_SEMAPHORE1_GATE34)	8	R/W	00h	42.7.1/2894
40C4_4023	Gate Register (RDC_SEMAPHORE1_GATE35)	8	R/W	00h	42.7.1/2894
40C4_4024	Gate Register (RDC_SEMAPHORE1_GATE36)	8	R/W	00h	42.7.1/2894
40C4_4025	Gate Register (RDC_SEMAPHORE1_GATE37)	8	R/W	00h	42.7.1/2894
40C4_4026	Gate Register (RDC_SEMAPHORE1_GATE38)	8	R/W	00h	42.7.1/2894
40C4_4027	Gate Register (RDC_SEMAPHORE1_GATE39)	8	R/W	00h	42.7.1/2894
40C4_4028	Gate Register (RDC_SEMAPHORE1_GATE40)	8	R/W	00h	42.7.1/2894
40C4_4029	Gate Register (RDC_SEMAPHORE1_GATE41)	8	R/W	00h	42.7.1/2894
40C4_402A	Gate Register (RDC_SEMAPHORE1_GATE42)	8	R/W	00h	42.7.1/2894
40C4_402B	Gate Register (RDC_SEMAPHORE1_GATE43)	8	R/W	00h	42.7.1/2894
40C4_402C	Gate Register (RDC_SEMAPHORE1_GATE44)	8	R/W	00h	42.7.1/2894
40C4_402D	Gate Register (RDC_SEMAPHORE1_GATE45)	8	R/W	00h	42.7.1/2894
40C4_402E	Gate Register (RDC_SEMAPHORE1_GATE46)	8	R/W	00h	42.7.1/2894
40C4_402F	Gate Register (RDC_SEMAPHORE1_GATE47)	8	R/W	00h	42.7.1/2894
40C4_4030	Gate Register (RDC_SEMAPHORE1_GATE48)	8	R/W	00h	42.7.1/2894
40C4_4031	Gate Register (RDC_SEMAPHORE1_GATE49)	8	R/W	00h	42.7.1/2894
40C4_4032	Gate Register (RDC_SEMAPHORE1_GATE50)	8	R/W	00h	42.7.1/2894
40C4_4033	Gate Register (RDC_SEMAPHORE1_GATE51)	8	R/W	00h	42.7.1/2894
40C4_4034	Gate Register (RDC_SEMAPHORE1_GATE52)	8	R/W	00h	42.7.1/2894

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40C4_4035	Gate Register (RDC_SEMAPHORE1_GATE53)	8	R/W	00h	42.7.1/2894
40C4_4036	Gate Register (RDC_SEMAPHORE1_GATE54)	8	R/W	00h	42.7.1/2894
40C4_4037	Gate Register (RDC_SEMAPHORE1_GATE55)	8	R/W	00h	42.7.1/2894
40C4_4038	Gate Register (RDC_SEMAPHORE1_GATE56)	8	R/W	00h	42.7.1/2894
40C4_4039	Gate Register (RDC_SEMAPHORE1_GATE57)	8	R/W	00h	42.7.1/2894
40C4_403A	Gate Register (RDC_SEMAPHORE1_GATE58)	8	R/W	00h	42.7.1/2894
40C4_403B	Gate Register (RDC_SEMAPHORE1_GATE59)	8	R/W	00h	42.7.1/2894
40C4_403C	Gate Register (RDC_SEMAPHORE1_GATE60)	8	R/W	00h	42.7.1/2894
40C4_403D	Gate Register (RDC_SEMAPHORE1_GATE61)	8	R/W	00h	42.7.1/2894
40C4_403E	Gate Register (RDC_SEMAPHORE1_GATE62)	8	R/W	00h	42.7.1/2894
40C4_403F	Gate Register (RDC_SEMAPHORE1_GATE63)	8	R/W	00h	42.7.1/2894
40C4_4042	Reset Gate Write (RDC_SEMAPHORE1_RSTGT_W)	16	R/W	0000h	42.7.2/2895
40C4_4042	Reset Gate Read (RDC_SEMAPHORE1_RSTGT_R)	16	R/W	0000h	42.7.3/2896
40CC_C000	Gate Register (RDC_SEMAPHORE2_GATE0)	8	R/W	00h	42.7.1/2894
40CC_C001	Gate Register (RDC_SEMAPHORE2_GATE1)	8	R/W	00h	42.7.1/2894
40CC_C002	Gate Register (RDC_SEMAPHORE2_GATE2)	8	R/W	00h	42.7.1/2894
40CC_C003	Gate Register (RDC_SEMAPHORE2_GATE3)	8	R/W	00h	42.7.1/2894
40CC_C004	Gate Register (RDC_SEMAPHORE2_GATE4)	8	R/W	00h	42.7.1/2894
40CC_C005	Gate Register (RDC_SEMAPHORE2_GATE5)	8	R/W	00h	42.7.1/2894
40CC_C006	Gate Register (RDC_SEMAPHORE2_GATE6)	8	R/W	00h	42.7.1/2894
40CC_C007	Gate Register (RDC_SEMAPHORE2_GATE7)	8	R/W	00h	42.7.1/2894
40CC_C008	Gate Register (RDC_SEMAPHORE2_GATE8)	8	R/W	00h	42.7.1/2894
40CC_C009	Gate Register (RDC_SEMAPHORE2_GATE9)	8	R/W	00h	42.7.1/2894
40CC_C00A	Gate Register (RDC_SEMAPHORE2_GATE10)	8	R/W	00h	42.7.1/2894
40CC_C00B	Gate Register (RDC_SEMAPHORE2_GATE11)	8	R/W	00h	42.7.1/2894
40CC_C00C	Gate Register (RDC_SEMAPHORE2_GATE12)	8	R/W	00h	42.7.1/2894
40CC_C00D	Gate Register (RDC_SEMAPHORE2_GATE13)	8	R/W	00h	42.7.1/2894
40CC_C00E	Gate Register (RDC_SEMAPHORE2_GATE14)	8	R/W	00h	42.7.1/2894
40CC_C00F	Gate Register (RDC_SEMAPHORE2_GATE15)	8	R/W	00h	42.7.1/2894
40CC_C010	Gate Register (RDC_SEMAPHORE2_GATE16)	8	R/W	00h	42.7.1/2894
40CC_C011	Gate Register (RDC_SEMAPHORE2_GATE17)	8	R/W	00h	42.7.1/2894
40CC_C012	Gate Register (RDC_SEMAPHORE2_GATE18)	8	R/W	00h	42.7.1/2894
40CC_C013	Gate Register (RDC_SEMAPHORE2_GATE19)	8	R/W	00h	42.7.1/2894
40CC_C014	Gate Register (RDC_SEMAPHORE2_GATE20)	8	R/W	00h	42.7.1/2894
40CC_C015	Gate Register (RDC_SEMAPHORE2_GATE21)	8	R/W	00h	42.7.1/2894
40CC_C016	Gate Register (RDC_SEMAPHORE2_GATE22)	8	R/W	00h	42.7.1/2894
40CC_C017	Gate Register (RDC_SEMAPHORE2_GATE23)	8	R/W	00h	42.7.1/2894
40CC_C018	Gate Register (RDC_SEMAPHORE2_GATE24)	8	R/W	00h	42.7.1/2894

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40CC_C019	Gate Register (RDC_SEMAPHORE2_GATE25)	8	R/W	00h	42.7.1/2894
40CC_C01A	Gate Register (RDC_SEMAPHORE2_GATE26)	8	R/W	00h	42.7.1/2894
40CC_C01B	Gate Register (RDC_SEMAPHORE2_GATE27)	8	R/W	00h	42.7.1/2894
40CC_C01C	Gate Register (RDC_SEMAPHORE2_GATE28)	8	R/W	00h	42.7.1/2894
40CC_C01D	Gate Register (RDC_SEMAPHORE2_GATE29)	8	R/W	00h	42.7.1/2894
40CC_C01E	Gate Register (RDC_SEMAPHORE2_GATE30)	8	R/W	00h	42.7.1/2894
40CC_C01F	Gate Register (RDC_SEMAPHORE2_GATE31)	8	R/W	00h	42.7.1/2894
40CC_C020	Gate Register (RDC_SEMAPHORE2_GATE32)	8	R/W	00h	42.7.1/2894
40CC_C021	Gate Register (RDC_SEMAPHORE2_GATE33)	8	R/W	00h	42.7.1/2894
40CC_C022	Gate Register (RDC_SEMAPHORE2_GATE34)	8	R/W	00h	42.7.1/2894
40CC_C023	Gate Register (RDC_SEMAPHORE2_GATE35)	8	R/W	00h	42.7.1/2894
40CC_C024	Gate Register (RDC_SEMAPHORE2_GATE36)	8	R/W	00h	42.7.1/2894
40CC_C025	Gate Register (RDC_SEMAPHORE2_GATE37)	8	R/W	00h	42.7.1/2894
40CC_C026	Gate Register (RDC_SEMAPHORE2_GATE38)	8	R/W	00h	42.7.1/2894
40CC_C027	Gate Register (RDC_SEMAPHORE2_GATE39)	8	R/W	00h	42.7.1/2894
40CC_C028	Gate Register (RDC_SEMAPHORE2_GATE40)	8	R/W	00h	42.7.1/2894
40CC_C029	Gate Register (RDC_SEMAPHORE2_GATE41)	8	R/W	00h	42.7.1/2894
40CC_C02A	Gate Register (RDC_SEMAPHORE2_GATE42)	8	R/W	00h	42.7.1/2894
40CC_C02B	Gate Register (RDC_SEMAPHORE2_GATE43)	8	R/W	00h	42.7.1/2894
40CC_C02C	Gate Register (RDC_SEMAPHORE2_GATE44)	8	R/W	00h	42.7.1/2894
40CC_C02D	Gate Register (RDC_SEMAPHORE2_GATE45)	8	R/W	00h	42.7.1/2894
40CC_C02E	Gate Register (RDC_SEMAPHORE2_GATE46)	8	R/W	00h	42.7.1/2894
40CC_C02F	Gate Register (RDC_SEMAPHORE2_GATE47)	8	R/W	00h	42.7.1/2894
40CC_C030	Gate Register (RDC_SEMAPHORE2_GATE48)	8	R/W	00h	42.7.1/2894
40CC_C031	Gate Register (RDC_SEMAPHORE2_GATE49)	8	R/W	00h	42.7.1/2894
40CC_C032	Gate Register (RDC_SEMAPHORE2_GATE50)	8	R/W	00h	42.7.1/2894
40CC_C033	Gate Register (RDC_SEMAPHORE2_GATE51)	8	R/W	00h	42.7.1/2894
40CC_C034	Gate Register (RDC_SEMAPHORE2_GATE52)	8	R/W	00h	42.7.1/2894
40CC_C035	Gate Register (RDC_SEMAPHORE2_GATE53)	8	R/W	00h	42.7.1/2894
40CC_C036	Gate Register (RDC_SEMAPHORE2_GATE54)	8	R/W	00h	42.7.1/2894
40CC_C037	Gate Register (RDC_SEMAPHORE2_GATE55)	8	R/W	00h	42.7.1/2894
40CC_C038	Gate Register (RDC_SEMAPHORE2_GATE56)	8	R/W	00h	42.7.1/2894
40CC_C039	Gate Register (RDC_SEMAPHORE2_GATE57)	8	R/W	00h	42.7.1/2894
40CC_C03A	Gate Register (RDC_SEMAPHORE2_GATE58)	8	R/W	00h	42.7.1/2894
40CC_C03B	Gate Register (RDC_SEMAPHORE2_GATE59)	8	R/W	00h	42.7.1/2894
40CC_C03C	Gate Register (RDC_SEMAPHORE2_GATE60)	8	R/W	00h	42.7.1/2894
40CC_C03D	Gate Register (RDC_SEMAPHORE2_GATE61)	8	R/W	00h	42.7.1/2894
40CC_C03E	Gate Register (RDC_SEMAPHORE2_GATE62)	8	R/W	00h	42.7.1/2894

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40CC_C03F	Gate Register (RDC_SEMAPHORE2_GATE63)	8	R/W	00h	42.7.1/2894
40CC_C042	Reset Gate Write (RDC_SEMAPHORE2_RSTGT_W)	16	R/W	0000h	42.7.2/2895
40CC_C042	Reset Gate Read (RDC_SEMAPHORE2_RSTGT_R)	16	R/W	0000h	42.7.3/2896

42.7.1 Gate Register (RDC_SEMAPHORE*x*_GATE*n*)

Each semaphore gate is implemented in a 4-bit finite state machine, right-justified in a byte data structure. The hardware uses the logical bus master number (master_index) in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. Attempted writes with a data value that is neither the unlock value nor the appropriate lock value (master_index + 1) are simply treated as "no operation" and do not affect any gate state. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes.

Address: Base address + 0h offset + (1d × i), where i=0d to 63d

Bit	7	6	5	4	3	2	1	0
Read	0		LDOM		GTFSM			
Write								
Reset	0	0	0	0	0	0	0	0

RDC_SEMAPHORE*x*_GATE*n* field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LDOM	Read-only bits. They indicate which domain had currently locked the gate. 00 The gate is locked by domain 0. (True if the field GTFSM does not equal to 0000.) 01 The gate has been locked by domain 1. 10 Reserved 11 Reserved
GTFSM	Gate Finite State Machine. The state of the gate reflects the last processor that locked it, which can be useful during system debug. The hardware gate is maintained in a 16-state implementation, defined as:

Table continues on the next page...

RDC_SEMAPHORE*x*_GATE*n* field descriptions (continued)

Field	Description
	0000 The gate is unlocked (free). 0001 The gate has been locked by processor with master_index = 0. 0010 The gate has been locked by processor with master_index = 1. 0011 The gate has been locked by processor with master_index = 2. 0100 The gate has been locked by processor with master_index = 3. 0101 The gate has been locked by processor with master_index = 4. 0110 The gate has been locked by processor with master_index = 5. 0111 The gate has been locked by processor with master_index = 6. 1000 The gate has been locked by processor with master_index = 7. 1001 The gate has been locked by processor with master_index = 8. 1010 The gate has been locked by processor with master_index = 9. 1011 The gate has been locked by processor with master_index = 10. 1100 The gate has been locked by processor with master_index = 11. 1101 The gate has been locked by processor with master_index = 12. 1110 The gate has been locked by processor with master_index = 13. 1111 The gate has been locked by processor with master_index = 14.

42.7.2 Reset Gate Write (RDC_SEMAPHORE*x*_RSTGT_W)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte (RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.

3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 42h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0							
Write									RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_W field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
RSTGDP	Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D.

42.7.3 Reset Gate Read (RDC_SEMAPHOREx_RSTGT_R)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte (RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data

pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.

3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 42h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0		RSTGSM		RSTGMS			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_R field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. This field contains the hexidecimal value of the gate number. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RSTGSM	Reset Gate Finite State Machine. Reads of the RDC_SEMA42RSTGT register return the encoded state machine value. Note the RSTGSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write. 10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. The "01" state persists for only one clock cycle. Software will never be able to observe this state. 11 This state encoding is never used and therefore reserved.
RSTGMS	Reset Gate Bus Master. This 4-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register must be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device, and the logical processor number is SoC-specific. See the chip-specific information section before the Semaphores (SEMA4) chapter for more details.

Chapter 43
Extended Resource Domain Controller 2 (XRDC2)

43.1 Chip-specific XRDC2 information

Table 43-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

The XRDC2 is used in RT1170 to support both single-core and multi-core use cases with max flexible resource management.

XRDC2 is used as second-level control (RDC is used as first-level control). Protect the access to resources within one CPU based on access type:

- Secure/non-secure access
- Privileged/user access

Each CPU has its own XRDC2, which has up to 16 domains.

Below is the diagram to show resource management concept in i.MX RT1170.

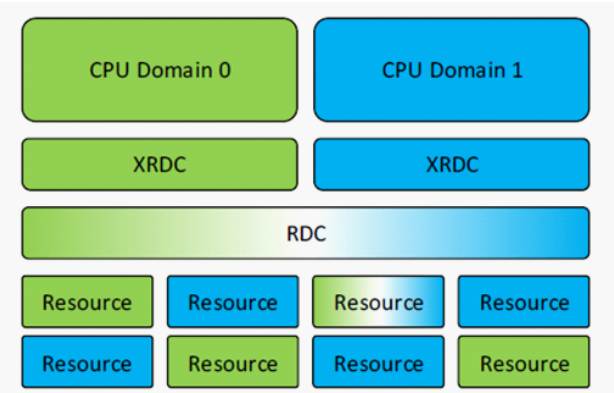


Figure 43-1. Resource Management Concept in i.MX RT1170

The following tables show the i.MX RT1170 XRDC2 configuration details.

Table 43-2. i.MX RT1170 XRDC2 MDAC Details

MDAC	j	i in MGR for M7	i in MGR for M4	MATCH
M7 AHB MDAC	0~15	0	N/A	MATCH[5:0]= XID[5:0]
M4 AHBC MDAC	0~15	N/A	0	MATCH[13:0]= {PID[7:0],XID[5:0]}
M7 AXI MDAC	0~15	1	N/A	MATCH[27:0]= {AWUSER[15:8],AWXID[5:0],ARUSER[15:8],ARXID[5:0]}
M4 AHBS MDAC	0~15	N/A	1	MATCH[13:0]= {PID[7:0],XID[5:0]}
CAAM MDAC	0~15	2	2	MATCH[19:0]= {AWICID[3:0],AWXID[5:0],ARICID[3:0],ARXID[5:0]}
CSI MDAC	0~7	3	3	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
M7 EDMA MDAC	0~15	4	N/A	MATCH[14:0]= {CHANNEL_ID[4:0],HMASTER[3:0],XID[5:0]}
M4 EDMA MDAC	0~15	N/A	4	MATCH[14:0]= {CHANNEL_ID[4:0],HMASTER[3:0],XID[5:0]}
ENET MDAC	0~7	5	5	MATCH[5:0]= XID[5:0]
ENET_1G RX MDAC	0~7	6	6	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
ENET_1G TX MDAC	0~7	7	7	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
ENET_QOS MDAC	0~7	8	8	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
GPU2D MDAC	0~7	9	9	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
eLCDIF MDAC	0~7	10	10	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
LCDIFv2 MDAC	0~7	11	11	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
PXP MDAC	0~7	12	12	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}

Table continues on the next page...

Table 43-2. i.MX RT1170 XRDC2 MDAC Details (continued)

MDAC	j	i in MGR for M7	i in MGR for M4	MATCH
Reserved	-	13	13	-
SSARC MDAC	N/A	14	14	No match input
USB MDAC	0~7	15	15	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
USDHC1 MDAC	0~7	16	16	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}
USDHC2 MDAC	0~7	17	17	MATCH[11:0]= {AWXID[5:0],ARXID[5:0]}

ID information:

- XID: AHB bus version of the Extended ID format
- AWXID/ARXID: AXI bus version of the Extended ID format
- AWUSER/ ARUSER: AXI bus user bits
- AWICID/ARICID: AXI Write/Read Isolation Context Identifiers(ICID) from CAAM
- PID: Process id from M4 MCM. See the Arm Cortex M4 Platform chapter for more information
- CHANNEL_ID: channel id from EDMA, indicates DMA channel index
- HMASTER: Master id from EDMA, indicates the master id of master that programmed the DMA channel that is active

Table 43-3. Extended ID

Extended ID	Field Name	Description
[7:6]	RDC domain ID[1:0]	Transmit CID from input to output transparently
[5]	Instruction Fetch Flag	Indicates if the access is instruction fetch. In AHB bus version, it is equal to ~HPROT[0], In AXI bus version, it is equal to ARPORT[2]
[4]	Reserved	-
[3]	Access Region Match	Indicate if ADDR is in defined 8 regions. See APC IOMUXC GPR register descriptions for more information
[2:0]	Access Region ID[2:0]	Indicates ADDR is in which region. See APC chapter for more information

Table 43-4. i.MX RT1170 XRDC2 MRC Details

MRC	j	i in MGR for M7	i in MGR for M4
CAAM MRC	0~15	0	0
FLEXSPI1 MRC	0~15	1	1
FLEXSPI2 MRC	0~15	2	2
M4 Local memory MRC	0~15	3	3
M7 OCRAM MRC	0~15	4	4
MECC1 MRC	0~15	5	5
MECC2 MRC	0~15	6	6
SEMC MRC	0~15	7	7

Table 43-5. i.MX RT1170 XRDC2 PAC Details

PAC	j	i in MGR for M7	i in MGR for M4
AIPS1 (aips_tz1)	0~129	0	0
AIPS2 (aips_tz2)	0~129	1	1
AIPS3 (aips_tz3)	0~129	2	2
AIPS4 (aips_tz4)	0~129	3	3
GC355 (GPU2D)	0	4	4
CDOG (CDOG)	0	5	5

Table 43-6. Peripheral Mapping

AIPS Slot (j)	AIPS	Instance
0-4	aips_tz1	Reserved
5	aips_tz1	MECC1
6	aips_tz1	MECC2
7	aips_tz1	XECC_FLEXSPI1
8	aips_tz1	XECC_FLEXSPI2
9	aips_tz1	XECC_SEMC
10	aips_tz1	CM7 (FLEXRAM)
11	aips_tz1	EWM
12	aips_tz1	WDOG1
13	aips_tz1	WDOG2
14	aips_tz1	WDOG3
15	aips_tz1	XBAR1
16	aips_tz1	XBAR2
17	aips_tz1	XBAR3
18	aips_tz1	ADC_ETC
19	aips_tz1	Reserved
20	aips_tz1	LPADC1
21	aips_tz1	LPADC2
22-24	aips_tz1	Reserved
25	aips_tz1	DAC
26	aips_tz1	APC_IEE
27	aips_tz1	IEE
28	aips_tz1	EDMA
29	aips_tz1	DMAMUX0
30	aips_tz1	Reserved
31	aips_tz1	LPUART1
32	aips_tz1	LPUART2
33	aips_tz1	LPUART3
34	aips_tz1	LPUART4

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
35	aips_tz1	LPUART5
36	aips_tz1	LPUART6
37	aips_tz1	LPUART7
38	aips_tz1	LPUART8
39	aips_tz1	LPUART9
40	aips_tz1	LPUART10
41	aips_tz1	Reserved
42	aips_tz1	Reserved
43	aips_tz1	FlexIO1
44	aips_tz1	FlexIO2
45	aips_tz1	Reserved
46	aips_tz1	AOI1
47	aips_tz1	AOI2
48	aips_tz1	Reserved
49	aips_tz1	CAN1
50	aips_tz1	CAN2
51	aips_tz1	FlexSPI1
52	aips_tz1	FlexSPI2
53	aips_tz1	SEMC
54	aips_tz1	PIT1
55	aips_tz1	Reserved
56	aips_tz1	KPP
57	aips_tz1	IOMUXC_GPR
58	aips_tz1	IOMUXC
59	aips_tz1	GPT1
60	aips_tz1	GPT2
61	aips_tz1	GPT3
62	aips_tz1	GPT4
63	aips_tz1	GPT5
64	aips_tz1	GPT6
65	aips_tz1	LPI2C1
66	aips_tz1	LPI2C2
67	aips_tz1	LPI2C3
68	aips_tz1	LPI2C4
69	aips_tz1	LPSPI1
70	aips_tz1	LPSPI2
71	aips_tz1	LPSPI3
72	aips_tz1	LPSPI4
73	aips_tz1	Reserved

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
74	aips_tz1	Reserved
75	aips_tz1	GPIO1
76	aips_tz1	GPIO2
77	aips_tz1	GPIO3
78	aips_tz1	GPIO4
79	aips_tz1	GPIO5
80	aips_tz1	GPIO6
81	aips_tz1	Reserved
82	aips_tz1	Reserved
83	aips_tz1	Reserved
84	aips_tz1	Reserved
85	aips_tz1	EVMSIM1
86	aips_tz1	EVMSIM2
87	aips_tz1	QTimer1
88	aips_tz1	QTimer2
89	aips_tz1	QTimer3
90	aips_tz1	QTimer4
91	aips_tz1	Reserved
92	aips_tz1	Reserved
93	aips_tz1	QDC1
94	aips_tz1	QDC2
95	aips_tz1	QDC3
96	aips_tz1	QDC4
97	aips_tz1	Reserved
98	aips_tz1	Reserved
99	aips_tz1	FLEXPWM1
100	aips_tz1	FLEXPWM2
101	aips_tz1	FLEXPWM3
102	aips_tz1	FLEXPWM4
103	aips_tz1	Reserved
104	aips_tz1	Reserved
105	aips_tz1	ACMP1
106	aips_tz1	ACMP2
107	aips_tz1	ACMP3
108	aips_tz1	ACMP4
109-129	aips_tz1	Reserved
0	aips_tz2	SPDIF
1	aips_tz2	SAI1
2	aips_tz2	SAI2

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
3	aips_tz2	SAI3
4	aips_tz2	Reserved
5	aips_tz2	ASRC
6	aips_tz2	USDHC1
7	aips_tz2	USDHC2
8	aips_tz2	ENET_1G
9	aips_tz2	ENET
10	aips_tz2	USB(pl301)
11	aips_tz2	USB(otg2)
12	aips_tz2	USB(otg)
13	aips_tz2	USBPHY1
14	aips_tz2	USBPHY2
15	aips_tz2	ENET_QOS
16	aips_tz2	CAAM
17	aips_tz2	Reserved
18	aips_tz2	Reserved
19	aips_tz2	Reserved
20	aips_tz2	CAAM
21	aips_tz2	Reserved
22	aips_tz2	Reserved
23	aips_tz2	Reserved
24	aips_tz2	CAAM
25	aips_tz2	Reserved
26	aips_tz2	Reserved
27	aips_tz2	Reserved
28	aips_tz2	CAAM
29	aips_tz2	Reserved
30	aips_tz2	Reserved
31	aips_tz2	Reserved
32	aips_tz2	CAAM
33	aips_tz2	Reserved
34	aips_tz2	Reserved
35	aips_tz2	Reserved
36	aips_tz2	CAAM
37	aips_tz2	Reserved
38	aips_tz2	Reserved
39	aips_tz2	Reserved
40	aips_tz2	CAAM
41	aips_tz2	Reserved

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
42	aips_tz2	Reserved
43	aips_tz2	Reserved
44	aips_tz2	CAAM
45	aips_tz2	Reserved
46	aips_tz2	Reserved
47	aips_tz2	Reserved
48	aips_tz2	CAAM
49	aips_tz2	Reserved
50	aips_tz2	Reserved
51	aips_tz2	Reserved
52	aips_tz2	CAAM
53	aips_tz2	Reserved
54	aips_tz2	Reserved
55	aips_tz2	Reserved
56	aips_tz2	CAAM
57	aips_tz2	Reserved
58	aips_tz2	Reserved
59	aips_tz2	Reserved
60	aips_tz2	CAAM
61	aips_tz2	Reserved
62	aips_tz2	Reserved
63	aips_tz2	Reserved
64	aips_tz2	CAAM
65	aips_tz2	Reserved
66	aips_tz2	Reserved
67	aips_tz2	Reserved
68	aips_tz2	CAAM
69	aips_tz2	Reserved
70	aips_tz2	Reserved
71	aips_tz2	Reserved
72	aips_tz2	CAAM
73	aips_tz2	Reserved
74	aips_tz2	Reserved
75	aips_tz2	Reserved
76	aips_tz2	CAAM
77-129	aips_tz2	Reserved
0	aips_tz3	CSI
1	aips_tz3	eLCDIF
2	aips_tz3	LCDIFv2

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
3	aips_tz3	MIPI_DSI
4	aips_tz3	MIPI_CSI
5	aips_tz3	PXP
6	aips_tz3	VIDEO_MUX
7-129	aips_tz3	Reserved
0	aips_tz4	GPC
1	aips_tz4	SRC
2	aips_tz4	IOMUXC_LPSR
3	aips_tz4	IOMUXC_LPSR_GPR
4	aips_tz4	WDOG4
5	aips_tz4	EDMA_LPSR
6	aips_tz4	DMAMUX1 (LPSR)
7	aips_tz4	Reserved
8	aips_tz4	PDM
9	aips_tz4	LPUART11
10	aips_tz4	LPUART12
11	aips_tz4	LPSPI5
12	aips_tz4	LPSPI6
13	aips_tz4	LPI2C5
14	aips_tz4	LPI2C6
15	aips_tz4	CAN3
16	aips_tz4	SAI4
17	aips_tz4	SEMA1
18	aips_tz4	MU(A)
19	aips_tz4	MU(B)
20	aips_tz4	Reserved
21	aips_tz4	Reserved
22	aips_tz4	Reserved
23	aips_tz4	GPIO7
24	aips_tz4	GPIO8
25	aips_tz4	GPIO9
26	aips_tz4	GPIO10
27	aips_tz4	GPIO11
28	aips_tz4	GPIO12
29	aips_tz4	Reserved
30	aips_tz4	RDC
31	aips_tz4	Reserved
32	aips_tz4	KEYMGR
33	aips_tz4	ANALOG/ANADIG

Table continues on the next page...

Table 43-6. Peripheral Mapping (continued)

AIPS Slot (j)	AIPS	Instance
34	aips_tz4	PGMC
35	aips_tz4	Reserved
36	aips_tz4	SNVS
37	aips_tz4	IOMUXC_SNVS
38	aips_tz4	IOMUXC_SNVS_GPR
39	aips_tz4	SNVS_SRAM
40	aips_tz4	GPIO13
41	aips_tz4	ROMCP
42	aips_tz4	DCDC
43	aips_tz4	OCOTP
44	aips_tz4	PIT2
45	aips_tz4	SSARC_HP
46	aips_tz4	SSARC_LP
47	aips_tz4	Reserved
48	aips_tz4	CCM
49	aips_tz4	CCM
50	aips_tz4	SEMA4
51	aips_tz4	RDC_SEMAPHORE2
52	aips_tz4	XRDC2 MGR M4
53	aips_tz4	XRDC2 MGR M4
54	aips_tz4	XRDC2 MGR M4
55	aips_tz4	XRDC2 MGR M4
56	aips_tz4	XRDC2 MGR M7
57	aips_tz4	XRDC2 MGR M7
58	aips_tz4	XRDC2 MGR M7
59	aips_tz4	XRDC2 MGR M7
60-127	aips_tz4	Reserved
128	aips_tz4	Reserved for global slot
129	aips_tz4	Reserved for global slot

Table 43-7. i.MX RT1170 XRDC2 MSC Details

MSC	i in MGR for M7	i in MGR for M4
GPV0 MSC	0	0
GPV1 MSC	1	1
GPV4 MSC	2	2
ROM MSC	3	3

43.2 Overview

The Extended Resource Domain Controller 2 (XRDC2) provides an integrated, scalable architectural framework for access control, system memory protection and peripheral isolation. It allows software to assign chip resources including processor cores, non-core bus masters, memory regions and slave peripherals to processing domains to support enforcement of robust operational environments.

The XRDC2 is a second generation design that has the same core functions and shares many conventions with the original XRDC. However, the XRDC2 programming model is not compatible with original XRDC. For the sake of brevity, this document uses the generic XRDC2 to refer to the second generation design.

43.2.1 Block diagram

The XRDC2 implementation is distributed across multiple submodules instantiated throughout the device. The XRDC2 submodules include:

- Manager (MGR) coordinates all programming model reads and writes
- Master Domain Assignment Controller (MDAC) is responsible for the generation of the DID, nonsecure and privileged attributes for every system bus transaction in the device based on pre-programmed Master Domain Assignment (MDA) registers
- Memory Region Controller (MRC) implements the access controls for slave memories based on the pre-programmed Memory Region Descriptor (MRGD) registers
- Peripheral Access Controller (PAC) implements the access controls for slave peripherals based on the pre-programmed Peripheral Domain Access Control (PDAC) registers
- Memory Slot Controller (MSC) implements the access controls for a memory or peripheral based on the pre-programmed Memory Slot Access Control (MSAC) registers

See Figure 43-2 for a simplified XRDC2 block diagram focusing on topology and connections.

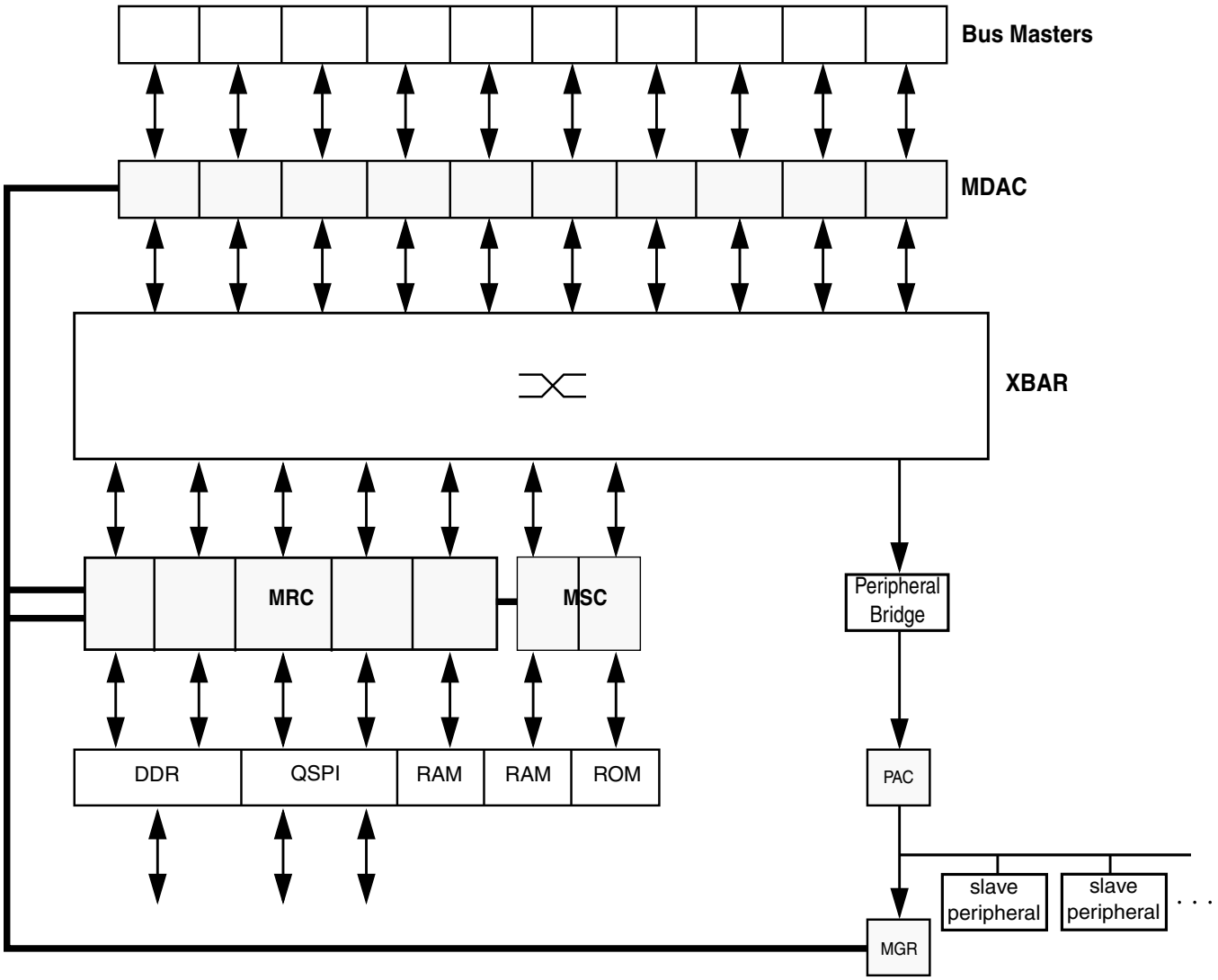


Figure 43-2. Simplified XRDC2 Block Diagram

43.2.2 Features

The key features of the XRDC2 include:

- Assignment of chip resources to processing “domains”
  - Each processing domain is assigned a unique domain identifier (DID)
  - DID is a new attribute associated with every system bus transaction
  - DID used in conjunction with user/privileged, secure/nonsecure attributes for access control
- Access rights to slave memory and peripheral targets defined by region or memory slot descriptors
- Supports sharing of a memory regions and peripherals with exclusive access locks to dynamically determine access rights

- Built upon a 4-level hierarchical access control model
  - SecurePriv(ileged) > SecureUser > NonsecurePriv(ileged) > NonsecureUser
  - Encoded into a 3-bit per-domain access control policy (ACP) used throughout the XRDC2
- Global and individual descriptor locking of configuration registers
- Programming model and hardware implementation is distributed across multiple submodules
  - Supports a broad, highly-configurable architecture definition
- Very configurable hardware design and implementation
  - Scalable across a wide spectrum of SoC devices

43.3 Functional description

The XRDC2 memory protection scheme has two components. First each bus mastering resource has an associated Master Domain Assignment Controller (MDAC) for assigning a domain identifier (DID) attribute to each bus transaction. Second each memory and peripheral has an associated access control submodule that enforces access control policies based on the DID and privilege level of the transaction. The XRDC2 access control submodules are the Memory Region Controller (MRC), Peripheral Access Controller (PAC) and Memory Slot Controller (MSC). All accesses throughout the device are monitored concurrently to determine the validity of each and every access. If a reference from a given domain has sufficient access rights, it is allowed to continue, else the access is denied and a bus error is generated.

The access control scheme defined by the XRDC2 supports a 4-level model, combining the traditional privileged (also known as supervisor) and user modes with an additional signal defining the secure, nonsecure attributes of each memory reference. The result is a 4-level hierarchical access control mechanism, where:

```
SecurePriv(ileged) > SecureUser > NonsecurePriv(ileged) > NonsecureUser
```

with different access control policies based on read and write references. Combined with the DID, the user/privileged and secure/nonsecure attributes are the basis for implementation of the XRDC2's access control mechanisms.

Access to shared memory regions and slave peripherals can be dynamically controlled with hardware exclusive access locks. This capability allows the access control policy for a given resource to be dynamically revised based on lock ownership.

43.3.1 Modes of Operation

The XRDC2 module does not support any special modes of operation. The XRDC2 responds based only on the memory addresses of the connected system buses. Domain assignment and access control functions are evaluated on a transfer-by-transfer basis using the addresses and attributes of each transfer. The implementation of the XRDC2 supports AMBA-AHB, AMBA-AXI, AMBA-APB and IPS bus protocols.

43.3.2 Manager (MGR)

The MGR submodule is responsible for coordinating XRDC2's programming model reads and writes.

The complete programming model is large, requiring a 64 Kbyte slave peripheral address space slot, although most devices define a sparsely populated subset. The MGR submodule implements the global registers. Other registers are distributed across the various submodule instances (MSC, PAC, MDAC, MRC). The MGR submodule provides the required hardware management for all programming model accesses, generating and distributing the appropriate address decodes and write data to the MSC, PAC, MDAC and MRC submodule instances, collecting and combining all the register read data buses and responses.

43.3.3 Master Domain Assignment Controller (MDAC)

The MDAC submodule is responsible for the generation of the DID, nonsecure and privileged attributes for every system bus transaction in the device. The MDAC compares its *match* input to the MATCH field for each enabled MDA descriptor for each bus transaction. If the MDAC *match* input is equal to the MATCH field for all bit positions which are a zero in the MASK field, the MDA generates a hit. The highest priority MDA with a hit takes precedence and its DID, PA, SA field values determine the DID, secure and privilege attributes for the bus transaction. See Master Domain Assignment (MDAC 0_MDA0_W0 - MDAC31_MDA31_W0) for details.

The MDAC is a configurable submodule, both in terms of hardware configurability options and software programmability. Internally, the MDACi_MDAj_Wk registers are implemented in this submodule. Each instance also supports a number of hardware parameters that are device-specific. These include parameters defining the number of

implemented domains, *match* input size, and the number of MDA descriptors. The MDAC can also optionally support multiple bus master ports with a single set of MDA descriptors.

43.3.4 DxACP Evaluation

Fundamental to the XRDC2's operation is the actual access violation check performed by the memory region (MRC), memory slot controller (MSC) and peripheral access (PAC) controllers.

Previous descriptions have detailed the domain assignment mechanisms and the tracking of the domain identifier as an address attribute through the system bus switching fabric(s). As transactions reach the slave memory and peripheral controllers, the address is used to select the appropriate memory region descriptor or access control descriptor. Once the appropriate descriptor has been selected, the next function is the actual access evaluation.

For this step, the DID of the current transaction selects the appropriate 3-bit DxACP (domain "x" access control policy) from the source register. The DxACP field defines the access rights for the domain based on the 4-level model combined with the transaction's access attributes (read = 0/write = 1, secure = 0/nonsecure = 1, user = 0/privileged = 1).

The 4-level model is defined by the concatenation of two address attributes: secure/nonsecure and user/privileged attributes. These two attribute signals (nonsecure, privileged) and the resulting access levels are defined as:

```
if {nonsecure, privileged} == 0b01, then level = SecurePriv(ileged)
if {nonsecure, privileged} == 0b00, then level = SecureUser
if {nonsecure, privileged} == 0b11, then level = NonsecurePriv(ileged)
if {nonsecure, privileged} == 0b10, then level = NonsecureUser
```

Recall the resulting hierarchical access control model specifies:

```
SecurePriv(ileged) > SecureUser > NonsecurePriv(ileged) > NonsecureUser
```

The XRDC2's evaluation of the access control policy is defined by the following Table 43-8 table.

Table 43-8. Domain x Access Control Policy Definition

	Allowable Accesses			
Policy	Secure Priv	Secure User	Nonsecure Priv	Nonsecure User
111	r, w	r, w	r, w	r, w
110	r, w	r, w	r, w	none
101	r, w	r, w	r	r
100	r, w	r, w	r	none
011	r, w	r, w	none	none
010	r, w	none	none	none
001	r	r	none	none
000	none	none	none	none

Unimplemented domains default to an all-zero DxACP field providing no access rights. After the DxACP evaluation is complete, the transaction is allowed to continue if it has sufficient access rights, else the access is denied and a bus error is generated. The XRDC2 also supports the inclusion of an exclusive access lock to dynamically alter the write permissions of a memory region or peripheral access. See Exclusive Access Lock for details.

43.3.5 Exclusive Access Lock

The exclusive access lock is a hardware mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access to a peripheral or memory region at any given time. Once an exclusive access lock is enabled, it is enabled until the next reset and the lock must be acquired by a domain to enable its access permissions. Only one domain can own the lock at a time and only the lock owner's access permissions are enabled, read and write permissions for all other domains are revoked.

Owning an exclusive access lock only enables permissions granted by the DxACP field, it does not grant permissions. Only domains with some read or write permissions (non-zero DxACP value) can acquire or release a lock. A lock can be held indefinitely. Typically a lock is released after required accesses are completed to allow other domains to acquire the lock as needed. In normal operation, only the domain that owns the lock can release it. There is a safeguarded means for a domain which does not own a lock to force a lock release.

The exclusive access lock is enabled, acquired and released by writing to the 2-bit Exclusive Access Lock (EAL) descriptor field. Neither the global configuration lock nor the descriptor lock apply to this field. The EAL field can be written to independently of

the rest of the descriptor fields. Non-zero writes (01,10,11) to the EAL field do not affect other fields of the register. Likewise, a write of 00 does not affect the exclusive access lock and other fields of the register are updated. This convention eliminates the need to perform a read, modify, write operation to control the exclusive access lock. For devices without the exclusive access lock feature, the EAL field is a read only, reserved space and register writes are not affected by non-zero writes to the EAL field.

The EAL values are shown in the following table.

Table 43-9. Exclusive Access Lock (EAL)

EAL	Lock Enabled	Lock State	Access Permissions	Allowed Transitions
00	No	---	Per DxACP evaluation.	01 disable lock 10 enable lock
01	No	---	Per DxACP evaluation.	None
10	Yes	Available	Read and write permissions revoked for all domains.	11 acquire lock
11	Yes	Not Available	Per DxACP evaluation for the domain that owns the lock, read and write permissions revoked for all other domains.	10 release lock

If EAL=11, the owner of the lock is indicated by the 4-bit, read only EALO field. The EALO field is used in the lock acquisition procedure to determine when the lock acquisition is successful. Domains with a DxACP field value of 000 can not acquire or release the lock.

The EAL field has a reset value of 00 which disables the exclusive access lock. The lock is disabled until the next reset by writing 0x01000000 to the W1 register (PAC/MSC) or W6 register (MRC) which sets EAL=01. Alternatively, the exclusive access lock is enabled until the next reset and set to the available state by writing 0x02000000 to the W1 register (PAC/MSC) or W6 register (MRC) which sets EAL=10.

Once the exclusive access lock is enabled, domains must acquire the lock to enable their access permissions. Only one domain can own the lock at a time. The procedure for acquiring the lock is as follows:

1. Set EAL = 11 by writing 0x03000000 to the W1 register (PAC/MSC) or W6 register (MRC)
2. Read EALO
3. If EALO = DID then the lock is acquired, else delay and repeat steps 1-3.

The DID value can be determined by reading the Status Register (SR).

The exclusive access lock is released by writing 0x02000000 to the W1 register (PAC/MSC) or W6 register (MRC) which sets EAL=10. Only the domain that owns the lock (EALO=DID) may release the lock with the normal lock release write.

A lock can be forced to the available state (EAL=10) by a domain that does not own the lock through the forced lock release procedure. The procedure to force an exclusive access lock release is as follows:

1. Write 0x02000046 to W1 register (PAC/MSC) or W6 register (MRC)
2. Write 0x02000052 to W1 register (PAC/MSC) or W6 register (MRC)

Only the EAL field and the least significant byte of the forced release writes are significant and detected by hardware. The two writes must be consecutive, any intervening write to the register resets the sequence. No other register fields are affected by the forced release write. Once the lock is released, it can be acquired by any domain. As a convention, normal lock release writes should use 0x02000000 to clearly distinguish normal release writes from other configuration register writes and forced release writes.

43.3.6 Memory Region Controller (MRC)

The Memory Region Controller (MRC) provides domain-based, hardware access control for all system bus references targeted at non-peripheral memory spaces. Using pre-programmed Memory Region Descriptors (MRGDs) which define memory spaces and their associated access rights per domain identifier, the MRC concurrently monitors system bus transactions and evaluates the appropriateness of each transfer. Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any defined memory region or have insufficient rights are terminated with a bus error.

The XRDC2 architectural framework supports up to 32 MRC instances, where each MRC instance can support up to 32 region descriptors and monitors one AHB or AXI slave port.

When MCR[GVLDC]=1, the MRC submodules evaluate memory access addresses versus the enabled MRGDs to determine which regions have a hit. To determine if the current reference hits in a given region, two magnitude comparators are used in conjunction with the region's start and end addresses. The equation for the hit determination is as follows:

```
region_hitj = (addr[35:12] >= srtaddrj) & (addr[35:12] <= endaddrj) & vldj
```

where addr[35:0] is the current system reference address, srtaddr and endaddr are the start and end address fields, and vld is the valid bit, all from the memory region descriptor j.

There are no hardware checks to verify that endaddr > srtaddr, software must ensure proper values are loaded into these fields of the region descriptor. Regions can be disabled via system power mode controls, see the appropriate section of the reference manual for details. The use of region addresses aligned to 4 kB boundaries matches the AXI bus specification which prohibits bus transactions from crossing a 4 kB boundary.

If a transfer hits one or more regions, it is evaluated with the highest priority MRGD with a hit. MRGDs have a fixed priority order starting with index j=0 as the lowest priority, index j=1 next highest, etc. Using the highest priority MRGD and the transfer attributes {DID, write, nonsecure, privileged} the transfer is evaluated as described in DxACP Evaluation. If the access has sufficient rights based on the DxACP evaluation and the exclusive access lock, it is allowed to continue, otherwise the access is denied and a bus error is generated. If a transfer does not hit any region, the access is denied and a bus error is generated.

43.3.7 Memory Slot Controller (MSC)

The Memory Slot Controller (MSC) performs access control for a peripheral or memory space with a fixed address range. For accesses within that address range the MSC uses the DID associated with the access to select the appropriate DxACP field in the MSAC descriptor. The selected DxACP field is then evaluated against the attributes associated with the memory transaction to determine the validity of the access. See DxACP Evaluation for more details. If the access has sufficient rights based on the DxACP evaluation and the exclusive access lock, it is allowed to continue, else the access is denied and a bus error is generated.

43.3.8 Peripheral Access Controller (PAC)

The Peripheral Access Controller (PAC) submodule performs access control for a set of peripherals connected to a peripheral bus bridge or integrated into a peripheral subsystem. The PAC selects the PDAC descriptor based on the peripheral slot accessed and then uses the DID associated with the access to select the appropriate DxACP field. The selected DxACP field is then evaluated against the attributes associated with the memory transaction to determine the validity of the access. See DxACP Evaluation for more details. If the access has sufficient rights based on the DxACP evaluation and the exclusive access lock, it is allowed to continue, else the access is denied and a bus error is generated. Unimplemented peripheral slots allow all accesses to continue without any protection controls.

43.4 External Signals

The XRDC2 module does not directly support any external interfaces.

The internal interfaces include a standard 32-bit slave bus for all programming model accesses, connections to the address phase signals associated with AHB and/or AXI system buses and connections to the slave peripheral buses as shown in the Figure 43-2.

43.5 Initialization

Out of reset, MCR[GVLDM] and MCR[GVLDC] are cleared and XRDC2 disabled allowing secure privileged startup code to configure the entire programming model. The initialization process typically is performed in 3 steps:

1. Program all of the domain assignment (MDACi_MDAj_Wk), memory region (MRCi_MRGDj_Wk), peripheral domain access control (PACi_PDACj_Wk) and memory slot access control (MSCi_Wk) descriptors. There are individual valid bits included in all these registers which typically would be asserted. Additionally, there are also lock mechanisms available if register settings need to be configured and marked as read-only.
2. Once the XRDC2 programming model is loaded, the MCR[GVLDM] bit is set.
3. After a delay to account for the maximum latency through the system bus fabric the MCR[GVLDC] bit is set and the XRDC2 is fully operational.

43.6 Application information

As described in the Overview, the XRDC2 architecture is intended to provide a broad, highly-capable framework for access control, system memory protection and peripheral isolation. The resulting microarchitecture implementation is distributed throughout the SoC, highly configurable via hardware design parameters and software programmability. The considerable number of hardware design parameters allows the XRDC2 architecture to be a very scalable solution, able to service a wide spectrum of SoC devices.

43.6.1 Master Domain Assignments

The typical use case related to master domain assignments is to include one (or more) processor core(s) in a single domain, possibly coupled with other bus master devices like DMA, etc. The definition of a domain may be static, based simply on the combination of a processor and other optional bus masters. Alternatively, the processor's operation may be configured to dynamically select between a very small number of domains. For example, "critical" tasks, whether safety-critical, performance-critical, etc. can be grouped together in one domain and all other tasks into a second domain. The various submodule descriptors all have lock features that allow the register resources to be converted into read-only resources to protect the written configuration.

43.6.2 Memory Region Descriptor Management

There are two important concepts to consider in managing the memory region descriptors.

First, recall that the association between each MRCi instance and the slave memory regions it monitors is SoC-dependent. The device-specific configuration specifies the number of implemented memory region descriptors in a given instance and the specific port numbers associated with the slave memories being monitored.

Second, as detailed in the section, Memory Region Controller (MRC), recall that, once the XRDC2 is enabled, *every memory reference must hit in one or more regions*, else a bus error is generated. MRGD0(lowest priority) can be programmed to cover the entire address space to provide a default set of access permissions.

43.7 Memory Map and register definition

The XRDC2 module supports a large number of architecturally-defined, program-visible registers distributed across a 64 Kbyte peripheral address slot. The actual registers present in any given device are application (and SoC) specific.

The programming model registers can be accessed via 8-, 16- or 32-bit reads and 32-bit write references. Attempted accesses using unsupported write data sizes, writes to read-only resources, or to reserved spaces are terminated with an error. The XRDC2 programming model is partitioned into 5 groups of registers:

- Global Registers
- Memory Slot Access Control (MSAC) Registers
- Master Domain Assignment (MDA) Registers
- Peripheral Domain Access Control (PDAC) Registers
- Memory Region Descriptor (MRGD) Registers

The MSAC, PDAC, MDA and MRGD registers are organized as 3-dimensional data structures with the following format {submodule instance}_{descriptor instance}_{word index} as shown in the following table, Table 43-10. There is only one MSAC descriptor instance for each MSC. The PDAC, MSAC, and MDA descriptors are 2 words in length. The MRGD descriptor is 8 words in length.

Table 43-10. XRDC2 Register Address Offsets

Submodule	Descriptor Registers	Address Offset
MSC	MSCi_MSAC_Wk	0x1000 + 8*i + 4*k
MDAC	MDACi_MDAj_Wk	0x2000 + 0x100*i + 8*j + 4*k
PAC	PACi_PDACj_Wk	0x4000 + 0x800*i + 8*j + 4*k
MRC	MRCi_MRGDj_Wk	0x8000 + 0x400*i + 32*j + 4*k

The architectural definition of the complete XRDC2 programming model is presented in the XRDC2 memory map table below. Each implementation is SoC-specific and highly dependent on the device requirements.

43.7.1 XRDC2 register descriptions

43.7.1.1 XRDC2 memory map

XRDC2_D0 base address: 40CE_0000h

XRDC2_D1 base address: 40CD_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Module Control Register (MCR)	32	RW	0000_0000
4	Status Register (SR)	32	RO	0000_0010
1000 - 13F8	Memory Slot Access Control (MSC0_MSAC_W0 - MSC127_MSAC_W0)	32	RW	0000_0000
1004 - 13FC	Memory Slot Access Control (MSC0_MSAC_W1 - MSC127_MSAC_W1)	32	RW	0000_0000
2000 - 3FF8	Master Domain Assignment (MDAC0_MDA0_W0 - MDAC31_MDA31_W0)	32	RW	0000_0000
2004 - 3FFC	Master Domain Assignment (MDAC0_MDA0_W1 - MDAC31_MDA31_W1)	32	RW	0000_0000
4000 - 7FF8	Peripheral Domain Access Control (PAC0_PDAC0_W0 - PAC7_PDAC255_W0)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
4004 - 7FFC	Peripheral Domain Access Control (PAC0_PDAC0_W1 - PAC7_PDAC255_W1)	32	RW	0000_0000
8000 - FFE0	Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0)	32	RW	0000_0000
8004 - FFE4	Memory Region Descriptor (MRC0_MRGD0_W1 - MRC31_MRGD31_W1)	32	RW	0000_0000
8008 - FFE8	Memory Region Descriptor (MRC0_MRGD0_W2 - MRC31_MRGD31_W2)	32	RW	0000_0FFF
800C - FFEC	Memory Region Descriptor (MRC0_MRGD0_W3 - MRC31_MRGD31_W3)	32	RW	0000_0000
8014 - FFF4	Memory Region Descriptor (MRC0_MRGD0_W5 - MRC31_MRGD31_W5)	32	RW	0000_0000
8018 - FFF8	Memory Region Descriptor (MRC0_MRGD0_W6 - MRC31_MRGD31_W6)	32	RW	0000_0000

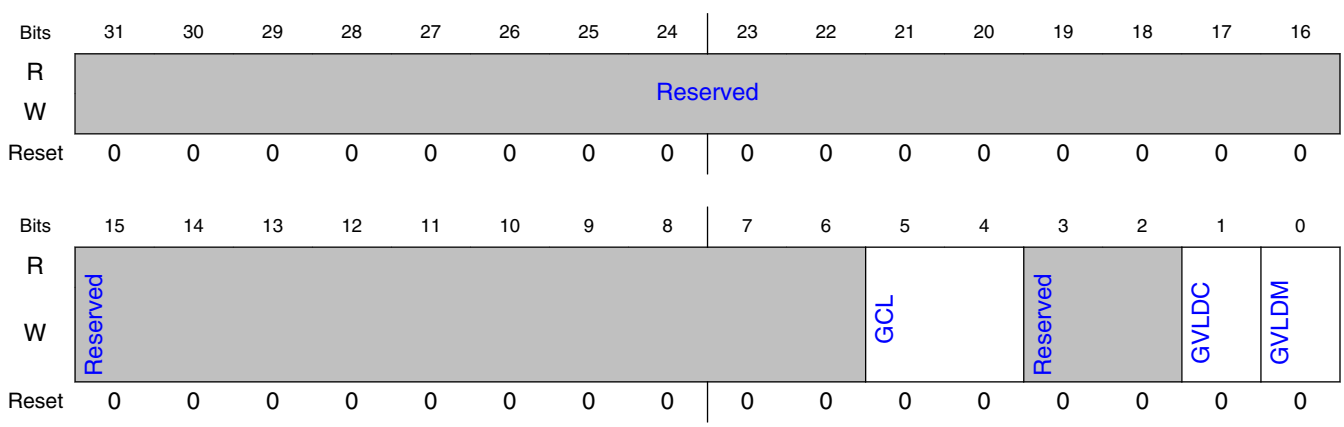
43.7.1.2 Module Control Register (MCR)

The MCR provides the global enables for the XRDC2.

43.7.1.2.1 Offset

Register	Offset
MCR	0h

43.7.1.2.2 Diagram



43.7.1.2.3 Fields

Field	Description
31-6 —	Reserved
5-4 GCL	Global Configuration Lock This 2-bit field provides a mechanism to limit write access to all registers. Once set to a non-zero value, this field can't be modified until the next reset. Exclusive access locks are not affected by the the global configuration lock. 00 - Lock disabled, registers can be written by any domain. 01 - Lock disabled until the next reset, registers can be written by any domain. 10 - Lock enabled, only the global configuration lock owner (SR[GCLO]) can write to registers. 11 - Lock enabled, all registers are read only until the next reset.
3-2 —	Reserved
1 GVLDC	Global Valid Access Control 0 - Access controls are disabled, XRDC2 allows all transactions. 1 - Access controls are enabled.
0 GVLDM	Global Valid MDAC 0 - DID set to 0 for all transactions. 1 - MDACs are enabled.

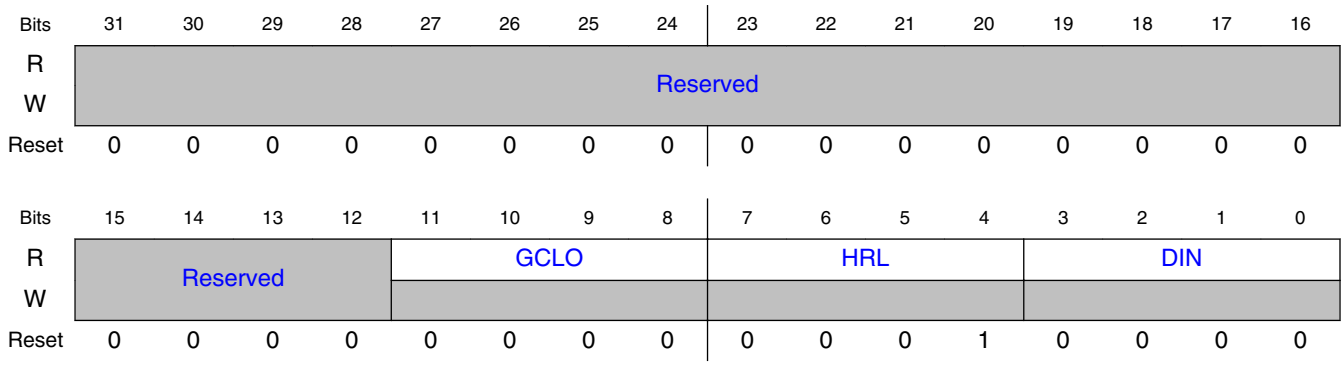
43.7.1.3 Status Register (SR)

The SR provides read-only status about the XRDC2.

43.7.1.3.1 Offset

Register	Offset
SR	4h

43.7.1.3.2 Diagram



43.7.1.3.3 Fields

Field	Description
31-12 —	Reserved
11-8 GCLO	Global Configuration Lock Owner This 4-bit read only field is set to the DID of the domain that set MCR[GCL]=10. When MCR[GCL]=10 this field indicates which domain owns the global configuration lock and is the only domain that can write to registers. GCLO values are defined as follows: • 0000: global configuration lock owner is domain 0 • 0001: global configuration lock owner is domian id 1 • ... • 1111: global configuration lock owenr is domain id 15
7-4 HRL	Hardware Revision Level This 4-bit read-only field specifies the XRDC2's hardware and definition revision level. It can be read by software to determine the functional definition of the module. It is hardwired to 1 as it represents the first XRDC2 implementation. HRL values are defined as follows: • 0000: version 0 • 0001: version 1 • .... • 1111: version 15
3-0 DIN	Domain Identifier Number This field provides the domain number [0-15] of the requesting bus master. DIN values are defined as follows: • 0000: domain id number is 0 • 0001: domain id number is 1 • ... • 1111: domain id number is 15

43.7.1.4 Memory Slot Access Control (MSC0_MSAC_W0 - MSC127_MSAC_W0)

The MSCi_MSAC_Wk registers hold the MSAC descriptors used by the MSC submodules to enforce access control policies for memories and peripherals. Each MSAC descriptor is formed by two 32-bit registers (MSCi_MSAC_W0/W1). Each MSC is assigned to one peripheral or memory and has one MSAC descriptor.

This device includes an exclusive access lock in the MSAC descriptor. The exclusive access lock provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. The exclusive access lock is controlled by writing to the EAL field. The EAL field is written independently of the other W1 register fields. Writes with a non-zero EAL value do not update other fields of the W1 register. Writes with a 00 EAL value update other W1 register fields but do not affect the write access lock. See Exclusive Access Lock for details.

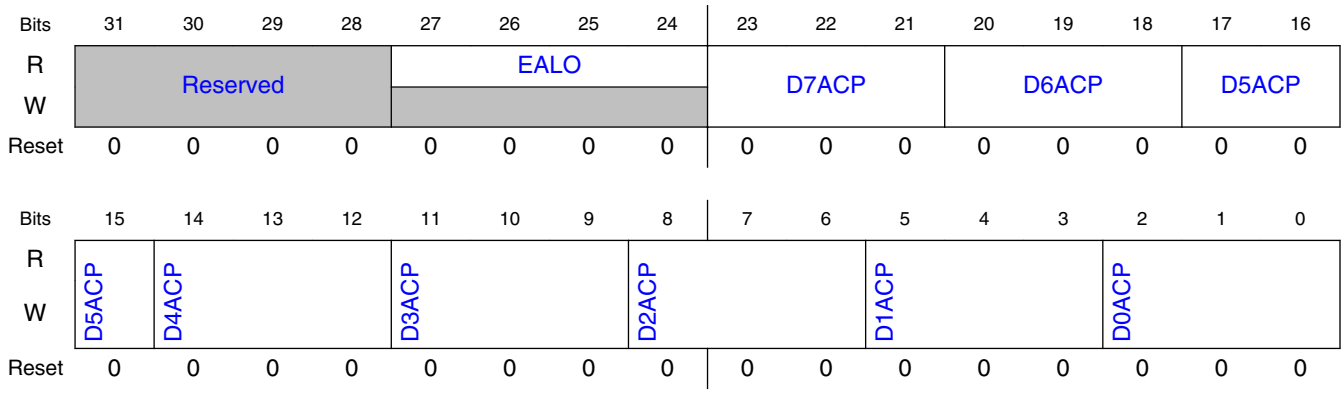
A MSC submodule is associated with one peripheral or memory with a fixed address range. For accesses within that address range the MSC uses the DID associated with the access to select the appropriate DxACP field. The selected DxACP field is then evaluated against the attributes associated with the memory transaction to determine the validity of the access. See DxACP Evaluation for more details. If the access has sufficient rights, it is allowed to continue, else the access is denied and a bus error is generated.

43.7.1.4.1 Offset

For i = 0 to 127:

Register	Offset
MSCi_MSAC_W0	1000h + (i × 8h)

43.7.1.4.2 Diagram



43.7.1.4.3 Fields

Field	Description
31-28 —	Reserved
27-24 EALO	Exclusive Access Lock Owner When EAL=11, this 4-bit read-only field indicates which domain owns the exclusive access lock. See Exclusive Access Lock for details on usage of the exclusive access lock.
23-21 D7ACP	Domain “x” access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D6ACP	Domain “x” access control policy Refer to Table 43-8
17-15 D5ACP	Domain “x” access control policy Refer to Table 43-8
14-12 D4ACP	Domain “x” access control policy Refer to Table 43-8
11-9 D3ACP	Domain “x” access control policy Refer to Table 43-8
8-6 D2ACP	Domain “x” access control policy Refer to Table 43-8
5-3 D1ACP	Domain “x” access control policy Refer to Table 43-8
2-0 D0ACP	Domain “x” access control policy Refer to Table 43-8

43.7.1.5 Memory Slot Access Control (MSC0_MSAC_W1 - MSC127_MSAC_W1)

Refer to Memory Slot Access Control (MSC0_MSAC_W0 - MSC127_MSAC_W0) for description.

43.7.1.5.1 Offset

For i = 0 to 127:

Register	Offset
MSCi_MSAC_W1	1004h + (i × 8h)

43.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VLD	DL2		Reserved			EAL		D15ACP			D14ACP			D13ACP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D13ACP	D12ACP			D11ACP			D10ACP			D9ACP			D8ACP		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.5.3 Fields

Field	Description
31 VLD	Valid This field indicates the memory slot access control definition is valid. It is further qualified by MCR[GVLDC] = 1. If either MCR[GVLDC] or the VLD flag is cleared, all accesses to the memory or peripheral are allowed.. 0 - The MSAC assignment is invalid. 1 - The MSAC assignment is valid.
30-29 DL2	Descriptor Lock

Table continues on the next page...

Field	Description
	This 2-bit field provides a mechanism to limit writes to the MSAC descriptor to protect its contents. Once set to a non-zero value, this field can't be modified until the next reset. This lock is in addition to the global configuration lock, both locks must grant permissions to write to the MSAC descriptor. Neither lock applies to the EAL field. 00 - Lock disabled, descriptor registers can be written. 01 - Lock disabled until the next reset, descriptor registers can be written. 10 - Lock enabled, only domain "x" can only update the DxACP field; no other fields can be written. 11 - Lock enabled, descriptor registers are read-only until the next reset.
28-26 —	Reserved
25-24 EAL	Exclusive Access Lock This 2-bit field provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. This field is written independently of the rest of the W1 register. Writes with a value of 00 do not affect the exclusive access lock and are used to update other fields of the W1 register. Writes with a non-zero value do not update other fields of the W1 register. Neither the global configuration lock nor the descriptor lock apply to this field. See Exclusive Access Lock for details on usage of the exclusive access lock. 00 - Lock disabled. 01 - Lock disabled until next reset. 10 - Lock enabled, lock state = available. 11 - Lock enabled, lock state = not available.
23-21 D15ACP	Domain "x" access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D14ACP	Domain "x" access control policy Refer to Table 43-8
17-15 D13ACP	Domain "x" access control policy Refer to Table 43-8
14-12 D12ACP	Domain "x" access control policy Refer to Table 43-8
11-9 D11ACP	Domain "x" access control policy Refer to Table 43-8
8-6 D10ACP	Domain "x" access control policy Refer to Table 43-8
5-3 D9ACP	Domain "x" access control policy Refer to Table 43-8
2-0 D8ACP	Domain "x" access control policy Refer to Table 43-8

43.7.1.6 Master Domain Assignment (MDAC0_MDA0_W0 - MDAC31_MDA31_W0)

The MDACi_MDAj_Wk registers hold the MDA descriptors used by the MDAC submodules to assign the DID, secure and privilege attributes to bus transactions. Each MDA descriptor is formed by two 32-bit registers (MDACi_MDAj_W0/W1). Each bus master port is assigned to a MDAC submodule that can have up to 32 MDAs. The number of implemented MDAs varies per MDAC instance. See chip-specific XRDC information for details. An MDAC instance may support multiple master ports. The MDA descriptor includes the MATCH, MASK, VLD, DL, SA, PA, and DID fields as shown below in the register diagram.

When MCR[GVLDM]=1, setting the VLD bit enables comparison of the MDAC *match* input to the MATCH field for each bus transaction. If the MDAC *match* input is equal to the MATCH field for all bit positions which are a zero in the MASK field, the MDA generates a hit. The highest priority MDA with a hit takes precedence and its DID, SA, PA field values determine the DID, secure and privilege attributes for the bus transaction. MDAs have a fixed priority order starting with index j=0 as the lowest priority, index j=1 next highest, etc. If MCR[GVLDM]=0 or there are no MDA hits then the DID attributes are set to zero and the secure and privilege attributes from the master are assigned to the transaction.

The *match* input, MATCH and corresponding MASK fields vary in width from 0 to 16 bits by MDAC instance. Likewise the signals used for the *match* input vary by master. See chip-specific XRDC information for more details. A MDAC may have only one MDA descriptor with a MATCH width of 0 bits. In this case the programmed DID, SA and PA values are associated with every transaction and any access to the MDAC_MDA_W0 register will result in a bus error. The MASK field for MDA index j=0 can be set to all ones to create a default set of DID, SA, and PA values to be applied if there are no other MDA hits.

43.7.1.6.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MDACi_MDAj_W0	2000h + (i × 100h) + (j × 8h)

43.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.6.3 Fields

Field	Description
31-16 MATCH	Match The MATCH field specifies the reference value for the comparison with the MDAC *match* input. The MATCH field width varies by MDAC instance from 0 to 16 bits. Unimplemented bits are read as 0. A size of 0 bits generates a hit on all comparisons.
15-0 MASK	Mask The MASK field specifies which bits are valid for the MATCH comparison. Only bit positions in which the MASK value is zero are compared. The MASK field width is the same as the MATCH field which varies by MDAC instance from 0 to 16 bits. A MASK value of all ones generates a hit on all comparisons.

43.7.1.7 Master Domain Assignment (MDAC0_MDA0_W1 - MDAC31_MDA31_W1)

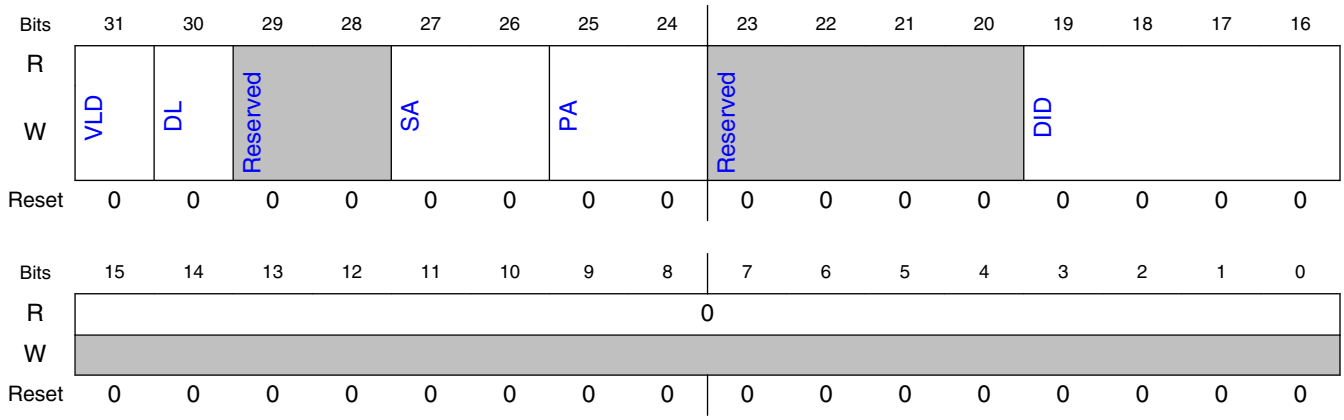
Refer to Master Domain Assignment (MDAC0_MDA0_W0 - MDAC31_MDA31_W0) for description.

43.7.1.7.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MDACi_MDAj_W1	2004h + (i × 100h) + (j × 8h)

43.7.1.7.2 Diagram



43.7.1.7.3 Fields

Field	Description
31 VLD	Valid This field indicates if the MDA descriptor is valid. When MCR[GVLDM]=1, setting the VLD bit enables comparison of the MDAC *match* input to the MATCH field for each bus transaction. A write to the W0 register clears the VLD bit so a coherent register state can be supported. 0 - The MDA is invalid. 1 - The MDA is valid.
30 DL	Descriptor Lock The DL bit provides a means to make the MDA descriptor registers read only. Once set, this bit remains asserted until the next reset. 0 - Lock disabled, registers can be written. 1 - Lock enabled, registers are read-only until the next reset.
29-28 —	Reserved
27-26 SA	Secure attribute This 2-bit field defines the secure/nonsecure attribute output on a hit. 00 - Use the bus master's secure/nonsecure attribute directly. 01 - Use the bus master's secure/nonsecure attribute directly. 10 - Force the bus attribute for this master to secure. 11 - Force the bus attribute for this master to nonsecure.
25-24 PA	Privileged attribute This 2-bit field defines the privileged/user attribute on a hit. 00 - Use the bus master's privileged/user attribute directly. 01 - Use the bus master's privileged/user attribute directly. 10 - Force the bus attribute for this master to user. 11 - Force the bus attribute for this master to privileged.

Table continues on the next page...

Field	Description
23-20 —	Reserved
19-16 DID	Domain Identifier This field defines the Domain Identifier (DID) attribute on a hit.
15-0 —	Reserved

43.7.1.8 Peripheral Domain Access Control (PAC0_PDAC0_W0 - PAC7_PDAC255_W0)

The PACi_PDACj_Wk registers hold the PDAC descriptors used by the PAC submodules to enforce access control policies for peripherals. Each PDAC descriptor is formed by two 32-bit registers (PACi_PDACj_W0/W1). Each peripheral bridge or peripheral subsystem is assigned to a PAC submodule that can have up to 256 PDAC descriptors, one per peripheral slot. In some PAC hardware configurations, certain slots may be mirror slots that use the PDAC descriptor of other slots. For example, slots 8 to 15 could be mirror slots that use the PDAC descriptors from slots 0 to 7, with slot 8 using PDAC0, slot 9 using PDAC1, etc. See chip-specific XRDC information for the PAC configurations used.

This device includes an exclusive access lock in the PDAC descriptor. The exclusive access lock provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. The exclusive access lock is controlled by writing to the EAL field. The EAL field is written independently of the other W1 register fields. Writes with a non-zero EAL value do not update other fields of the W1 register. Writes with a 00 EAL value update other W1 register fields but do not affect the exclusive access lock. See Exclusive Access Lock for details on usage of the exclusive access lock.

The PAC submodule uses the peripheral access address to select the appropriate PDAC descriptor, and then uses the DID associated with the access to select the appropriate DxACP field. The selected DxACP field is then evaluated against the attributes associated with the memory transaction to determine the validity of the access. See DxACP Evaluation for more details. If the access has sufficient rights, it is allowed to continue, else the access is denied and a bus error is generated.

43.7.1.8.1 Offset

For i = 0 to 7; j = 0 to 255:

Register	Offset
PACi_PDACj_W0	4000h + (i × 800h) + (j × 8h)

43.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				EALO				D7ACP			D6ACP			D5ACP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D5ACP	D4ACP			D3ACP			D2ACP			D1ACP			D0ACP		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.8.3 Fields

Field	Description
31-28 —	Reserved
27-24 EALO	Exclusive Access Lock Owner When EAL=11, this 4-bit read only field indicates which domain owns the exclusive access lock. See Exclusive Access Lock for details on usage of the exclusive access lock.
23-21 D7ACP	Domain “x” access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D6ACP	Domain “x” access control policy Refer to Table 43-8
17-15 D5ACP	Domain “x” access control policy Refer to Table 43-8
14-12 D4ACP	Domain “x” access control policy Refer to Table 43-8
11-9	Domain “x” access control policy

Table continues on the next page...

Field	Description
D3ACP	Refer to Table 43-8
8-6 D2ACP	Domain “x” access control policy Refer to Table 43-8
5-3 D1ACP	Domain “x” access control policy Refer to Table 43-8
2-0 D0ACP	Domain “x” access control policy Refer to Table 43-8

43.7.1.9 Peripheral Domain Access Control (PAC0_PDAC0_W1 - PAC7_PDAC255_W1)

Refer to Peripheral Domain Access Control (PAC0_PDAC0_W0 - PAC7_PDAC255_W0) for description.

43.7.1.9.1 Offset

For i = 0 to 7; j = 0 to 255:

Register	Offset
PACi_PDACj_W1	4004h + (i × 800h) + (j × 8h)

43.7.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VLD	DL2		Reserved			EAL		D15ACP			D14ACP			D13ACP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D13ACP	D12ACP			D11ACP			D10ACP			D9ACP			D8ACP		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.9.3 Fields

Field	Description
31 VLD	Valid This field indicates the peripheral domain access control definition is valid. It is further qualified by MCR[GVLDC] = 1. If either MCR[GVLDC] or this VLD flag is cleared, all accesses to the peripheral are allowed. 0 - The PDAC assignment is invalid. 1 - The PDAC assignment is valid.
30-29 DL2	Descriptor Lock This 2-bit field provides a mechanism to limit writes to the PDAC descriptor to protect its contents. Once set to a non-zero value, this field can't be modified until the next reset. This lock is in addition to the global configuration lock, both locks must grant permissions to write to the PDAC descriptor. Neither lock applies to the EAL field. 00 - Lock disabled, descriptor registers can be written.. 01 - Lock disabled until the next reset, descriptor registers can be written.. 10 - Lock enabled, only domain "x" can only update the DxACP field; no other fields can be written.. 11 - Lock enabled, descriptor registers are read-only until the next reset.
28-26 —	Reserved
25-24 EAL	Exclusive Access Lock This 2-bit field provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. This field is written independently of the rest of the W1 register. Writes with a value of 00 do not affect the exclusive access lock and are used to update other fields of the W1 register. Writes with a non-zero value do not update other fields of the W1 register. Neither the global configuration lock nor the descriptor lock apply to this field. See Exclusive Access Lock for details on usage of the exclusive access lock. 00 - Lock disabled. 01 - Lock disabled until next reset. 10 - Lock enabled, lock state = available. 11 - Lock enabled, lock state = not available.
23-21 D15ACP	Domain "x" access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D14ACP	Domain "x" access control policy Refer to Table 43-8
17-15 D13ACP	Domain "x" access control policy Refer to Table 43-8
14-12 D12ACP	Domain "x" access control policy Refer to Table 43-8
11-9	Domain "x" access control policy

Table continues on the next page...

Field	Description
D11ACP	Refer to Table 43-8
8-6 D10ACP	Domain “x” access control policy Refer to Table 43-8
5-3 D9ACP	Domain “x” access control policy Refer to Table 43-8
2-0 D8ACP	Domain “x” access control policy Refer to Table 43-8

43.7.1.10 Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0)

The MRCi_MRGDj_Wk registers hold the Memory Region Descriptors (MRGD) used by the MRC submodules to enforce access control policies for memories. Each MRGD is formed by a set of seven 32-bit registers (MRCi_MRGDj_W0-W6) which specify the start address, end address and access permissions for a region of memory. Each slave memory port is assigned to an MRC submodule that can have up to 32 memory regions and associated MRGDs. The number of implemented MRGDs varies per MRC instance, please see the appropriate section of the reference manual for details.

This device includes an exclusive access lock in the MRGD.The exclusive access lock provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. The exclusive access lock is controlled by writing to the EAL field. The EAL field is written independently of the other W6 register fields. Writes with a non-zero EAL value do not update other fields of the W6 register. Writes with a 00 EAL value update other W6 register fields but do not affect the exclusive access lock. See Exclusive Access Lock for details on usage of the exclusive access lock.

When MCR[GVLDC]=1, the MRC submodules evaluate memory access addresses versus the enabled MRGDs to determine which regions have a hit. The MRC uses the domain identifier associated with the access to select the appropriate DxACP field from the highest priority MRGD that has a hit. MRGDs have a fixed priority order starting with index j=0 as the lowest priority, index j=1 next highest, etc. The selected DxACP field is evaluated against the attributes associated with the memory transaction to determine the validity of the access. See Memory Region Controller (MRC) for details. If the access has sufficient rights, it is allowed to continue, else the access is denied and a

bus error is generated. If an access has no region hits the access is denied and a bus error is generated. MRGD0 can be programmed to cover the entire memory space to provide default permissions if no other MRGDs hit.

43.7.1.10.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W0	8000h + (i × 400h) + (j × 20h)

43.7.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRTADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRTADDR				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.10.3 Fields

Field	Description
31-12 SRTADDR	Start Address The SRTADDR field (W0) defines [19:0] of 24 most significant bits of the 36-bit start byte address of the memory region. Addresses matching the start address are included in the region.
11-0 —	Reserved

43.7.1.11 Memory Region Descriptor (MRC0_MRGD0_W1 - MRC31_MRGD31_W1)

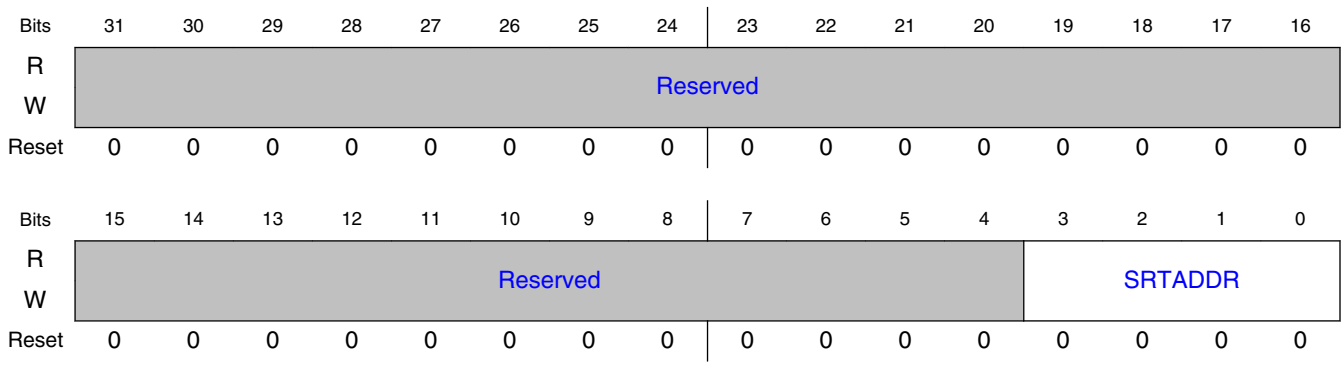
Refer to Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0) for description.

43.7.1.11.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W1	8004h + (i × 400h) + (j × 20h)

43.7.1.11.2 Diagram



43.7.1.11.3 Fields

Field	Description
31-4 —	Reserved
3-0 SRTADDR	Start Address The SRTADDR field (W1) defines [23:20] of 24 most significant bits of the 36-bit start byte address of the memory region. Addresses matching the start address are included in the region.

43.7.1.12 Memory Region Descriptor (MRC0_MRGD0_W2 - MRC31_MRGD31_W2)

Refer to Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0) for description.

43.7.1.12.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W2	8008h + (i × 400h) + (j × 20h)

43.7.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ENDADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENDADDR				Reserved											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

43.7.1.12.3 Fields

Field	Description
31-12 ENDADDR	End Address The ENDADDR field (W2) defines [19:0] of 24 most significant bits of the 36-bit end byte address of the memory region. Addresses matching the end address are included in the region.
11-0 —	Reserved

43.7.1.13 Memory Region Descriptor (MRC0_MRGD0_W3 - MRC31_MRGD31_W3)

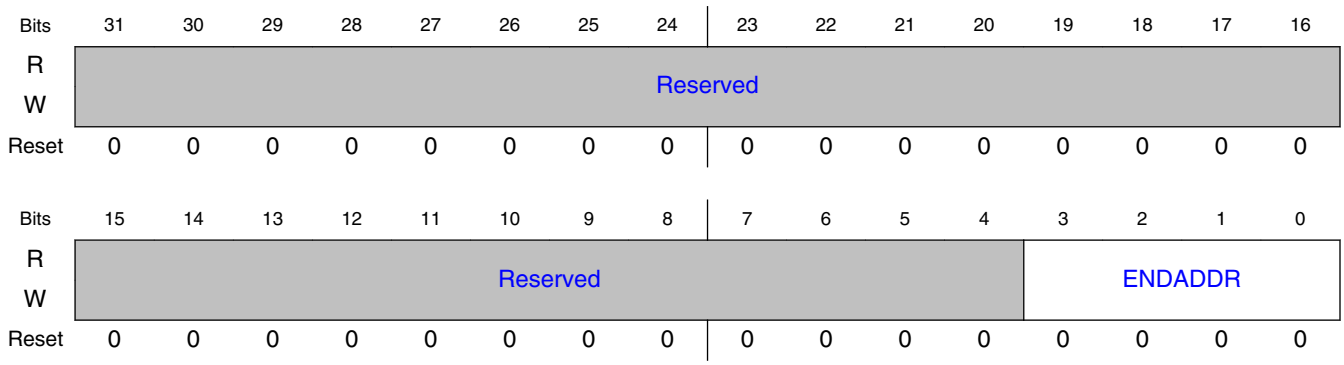
Refer to Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0) for description.

43.7.1.13.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W3	800Ch + (i × 400h) + (j × 20h)

43.7.1.13.2 Diagram



43.7.1.13.3 Fields

Field	Description
31-4 —	Reserved
3-0 ENDADDR	End Address The ENDADDR field (W3) defines [23:20] of 24 most significant bits of the 36-bit end byte address of the memory region. Addresses matching the end address are included in the region.

43.7.1.14 Memory Region Descriptor (MRC0_MRGD0_W5 - MRC31_MRGD31_W5)

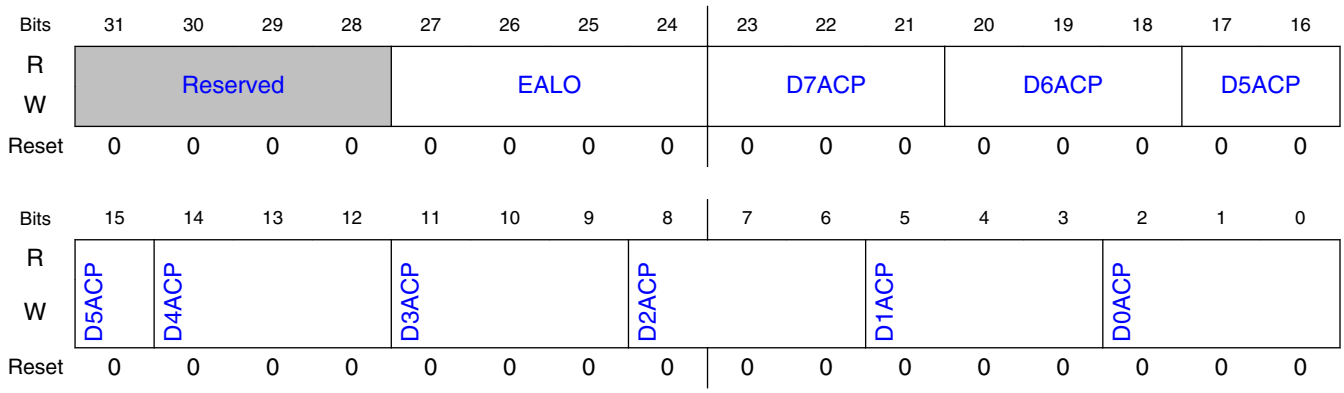
Refer to Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0) for description.

43.7.1.14.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W5	8014h + (i × 400h) + (j × 20h)

43.7.1.14.2 Diagram



43.7.1.14.3 Fields

Field	Description
31-28 —	Reserved
27-24 EALO	Exclusive Access Lock Owner When EAL=11, this 4-bit read only field indicates which domain owns the exclusive access lock. See Exclusive Access Lock for details.
23-21 D7ACP	Domain “x” access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D6ACP	Domain “x” access control policy Refer to Table 43-8
17-15 D5ACP	Domain “x” access control policy Refer to Table 43-8
14-12 D4ACP	Domain “x” access control policy Refer to Table 43-8
11-9 D3ACP	Domain “x” access control policy Refer to Table 43-8
8-6 D2ACP	Domain “x” access control policy Refer to Table 43-8
5-3 D1ACP	Domain “x” access control policy Refer to Table 43-8
2-0 D0ACP	Domain “x” access control policy Refer to Table 43-8

43.7.1.15 Memory Region Descriptor (MRC0_MRGD0_W6 - MRC31_MRGD31_W6)

Refer to Memory Region Descriptor (MRC0_MRGD0_W0 - MRC31_MRGD31_W0) for description.

43.7.1.15.1 Offset

For i = 0 to 31; j = 0 to 31:

Register	Offset
MRCi_MRGDj_W6	8018h + (i × 400h) + (j × 20h)

43.7.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VLD	DL2		Reserved			EAL		D15ACP			D14ACP			D13ACP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	D13ACP	D12ACP			D11ACP			D10ACP			D9ACP			D8ACP		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.7.1.15.3 Fields

Field	Description
31 VLD	Valid This field indicates the MRGD is valid. When MCR[GVLDC]=1 and VLD=1, bus transaction addresses are compared to SRTADDR and ENDADDR fields to determine if a hit is generated for the MRGD. 0 - The MRGD is invalid. 1 - The MRGD is valid.
30-29 DL2	Descriptor Lock This 2-bit field provides a mechanism to limit writes to the MRGD to protect its contents. Once set to a non-zero value, this field can't be modified until the next reset. This lock is in addition to the global

Table continues on the next page...

Field	Description
	configuration lock, both locks must grant permissions to write to the MRGD. Neither lock applies to the EAL field. 00 - Lock disabled, descriptor registers can be written. 01 - Lock disabled until the next reset, descriptor registers can be written. 10 - Lock enabled, only domain “x” can only update the DxACP field; no other fields can be written. 11 - Lock enabled, descriptor registers are read-only until the next reset.
28-26 —	Reserved
25-24 EAL	Exclusive Access Lock Exclusive Access Lock. This 2-bit field provides a mechanism to dynamically modify the DxACP permission evaluation such that only one domain has access permissions at a time. This field is written independently of the rest of the W1 register. Writes with a value of 00 do not affect the exclusive access lock and are used to update other fields of the W1 register. Writes with a non-zero value do not update other fields of the W1 register. Neither the global configuration lock nor the descriptor lock apply to this field. See Exclusive Access Lock for details on usage of the exclusive access lock. 00 - Lock disabled. 01 - Lock disabled until next reset. 10 - Lock enabled, lock state = available. 11 - Lock enabled, lock state = not available.
23-21 D15ACP	Domain “x” access control policy This 3-bit field defines the access control rights for the selected domain. The encodings specify read and write access capabilities based on the 4 operating states. These fields are only implemented for supported DIDs; unsupported DIDs are not implemented and default to a zero value providing no access rights. Refer to Table 43-8
20-18 D14ACP	Domain “x” access control policy Refer to Table 43-8
17-15 D13ACP	Domain “x” access control policy Refer to Table 43-8
14-12 D12ACP	Domain “x” access control policy Refer to Table 43-8
11-9 D11ACP	Domain “x” access control policy Refer to Table 43-8
8-6 D10ACP	Domain “x” access control policy Refer to Table 43-8
5-3 D9ACP	Domain “x” access control policy Refer to Table 43-8
2-0 D8ACP	Domain “x” access control policy Refer to Table 43-8

Chapter 44
Display And Camera Overview

44.1 Overview

The following modules are a part of the Display and Camera system:

- CMOS Sensor Interface (CSI): Support for 8/16/24-bit data port for YCbCr, YUV, or RGB data input
- MIPI Serial Interface (MIPI CSI2): Camera interface that supports 80 Mbps to 1.5 Gbps data rate in high speed mode, and 10Mbps in low power mode
- Enhanced LCD Interface (eLCDIF): 8/16/18/24-bit RGB LCD interface
- LCD Interface v2 (LCDIFv2): Advanced LCD interface with multi-layer blending
- MIPI Display Serial Interface (MIPI DSI): Display interface that support MIPI DSI-compliant interfaces
- Video Multiplexer (VIDEO_MUX): Mux controls between supported display and camera interfaces
    - Digital Content Integrity Checker (DCIC): Verifies that display data is not corrupted (supports up to 24-bit pixel data)
- PXP pixel pipeline: 2D graphics and pixel/image processing
- Graphics Processing Unit (GPU2D): 2D Graphics engine that supports raster and vector graphics encompassing the majority of the embedded graphics use-case scenarios

The following figure shows the high-level integration scheme of the chip display and camera system.

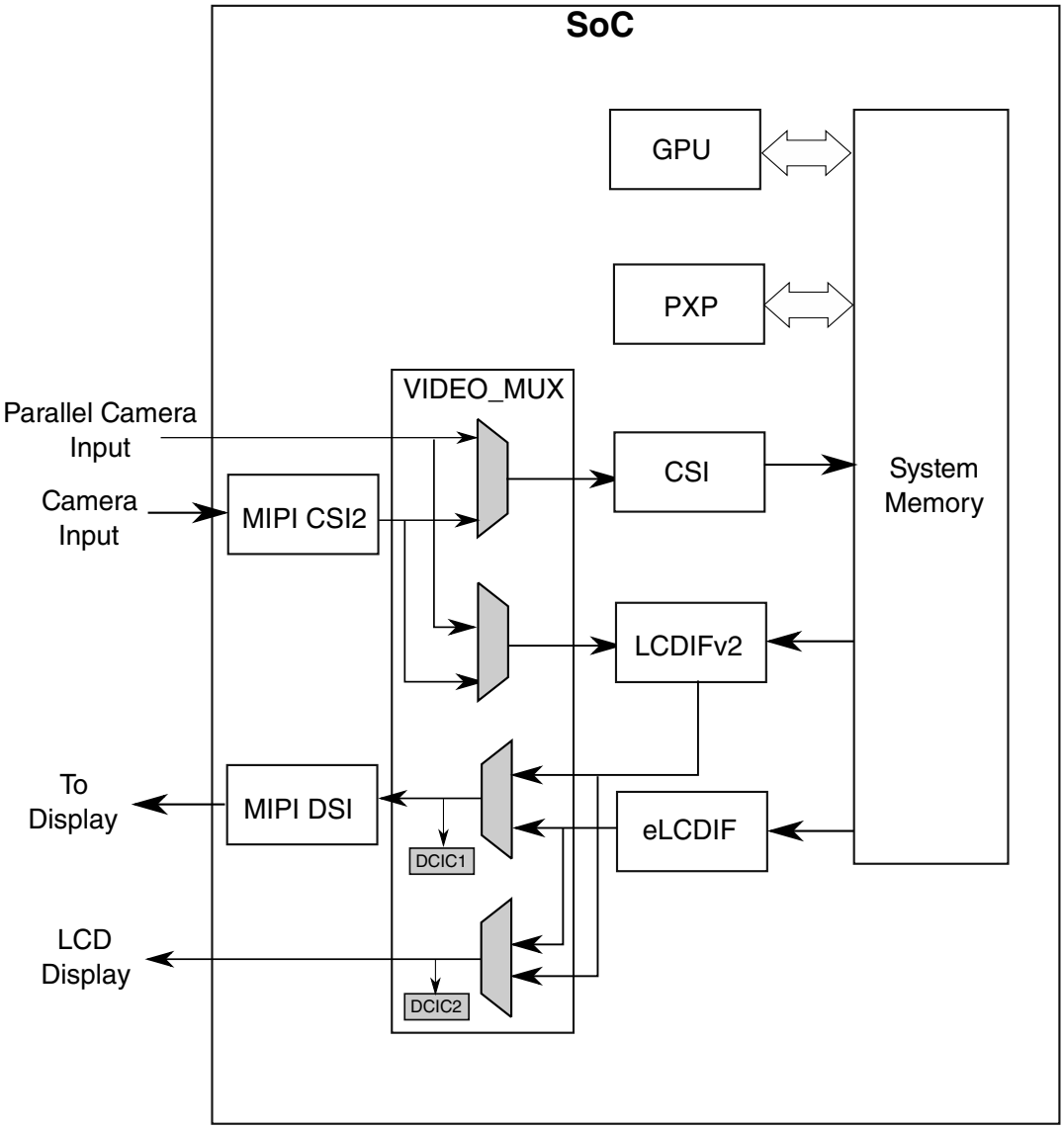


Figure 44-1. High-level Integration block diagram

NOTE

The camera input to the CSI module can be chosen from either Parallel Camera or MIPI CSI2. However, the display output can be fed to MIPI DSI and/or eLCDIF.

The LCDIFv2 acts as the primary LCD controller, while the eLCDIF acts as the secondary/backup controller.

The GPU2D is a vector engine and PXP acts as the blending accelerator.

44.1.1 CMOS Sensor Interface (CSI)

The CSI enables the chip to connect directly to external CMOS image sensors. CMOS image sensors are separated into two classes, dumb and smart. Dumb sensors are those that support only traditional sensor timing (Vertical SYNC, Horizontal SYNC), and output RGB, YUV, YCbCr, RAW formats, as well as Bayer and statistics data. The smart sensors support BT.656 video decoder formats and perform additional processing of the image (for example, image compression, image pre-filtering, and various data output formats).

44.1.2 MIPI Camera Serial Interface 2 (MIPI_CSI2)

The MIPI CSI2 is a camera interface that supports two data lanes, and one clock input between all data lanes. It provides support for virtual channels.

44.1.3 Enhanced LCD Interface (eLCDIF)

The eLCDIF is a general purpose display controller that is used to drive a wide range of display devices. These displays can vary in size and capability. There are other popular displays that support moving pictures and require the RGB interface mode (called DOTCLK interface in this document) for high-speed data transfers.

The block has several major features:

- Bus master interface to source frame buffer data for display refresh and a DMA interface to manage input data transfers from the LCD requiring minimal CPU overhead.
- 8/16/18/24 bit LCD data bus support available depending on I/O mux options.
- Programmable timing and parameters for DOTCLK LCD interfaces to support a wide variety of displays.

44.1.4 LCD Interface v2 (LCDIFv2)

The LCDIFv2 controller blends graphics from memory in real time, and output the data to a display interface.

It includes the following features:

- CRC check for configurable region on the final display output after alpha blending

- Eight layers of alpha blending
- Write-back channel to save the final output into memory

44.1.5 MIPI Display Serial Interface (MIPI_DSI)

MIPI DSI is a display interface that provides an interface to connect to MIPI DSI-compliant peripherals. It provides support for command and video modes, and virtual channels.

The MIPI DSI controller is associated with a MIPI D-PHY which complies with the MIPI D-PHY 1.1 standard. It supports two data lanes and one clock lane.

44.1.6 Video Multiplexer (VIDEO_MUX)

Video Multiplexer is a mux that lets the user choose between several camera and display interfaces. It also includes a Digital Content Integrity Checker (DCIC) to check for data corruption.

44.1.7 Pixel Processing Pipeline (PXP)

The pixel processing pipeline is used to perform image processing on image/video buffers before sending to an LCD display.

The main features of PXP include:

- Multiple input/output format support, including YUV/RGB/Grayscale
- Supports both RGB/YUV scaling
- Supports overlay with Alpha blending
- Supports Rotation of 0, 90, 180, and 270 degrees in conjunction with vertical and horizontal flip options

44.1.8 Graphics Processing Unit (GPU2D)

The 2D GPU comprises of several components like Tessellation engine, Vector Graphics Engine, Imaging Engine that perform various processing. It supports OpenVG 1.1 API Standard for VGA and VGLite Graphics API.

Chapter 45
CMOS Sensor Interface (CSI)

45.1 Chip-specific CSI information

Table 45-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

45.2 Overview

This chapter presents the CMOS Sensor Interface (CSI) architecture, operation principles, and programming model. The CSI enables the chip to connect directly to external CMOS image sensors.

CMOS image sensors are separated into two classes, dumb and smart. Dumb sensors are those that support only traditional sensor timing (Vertical SYNC and Horizontal SYNC) and output only RGB, YUV, and Bayer and statistics data, while smart sensors support ITU-R BT.656 video decoder formats and perform additional processing of the image (for example, image compression, image pre-filtering, and various data output formats). Statistics only work for Bayer data is 8-bit per pixel format.

45.2.1 Block Diagram

The block diagram of the CSI module is shown below:

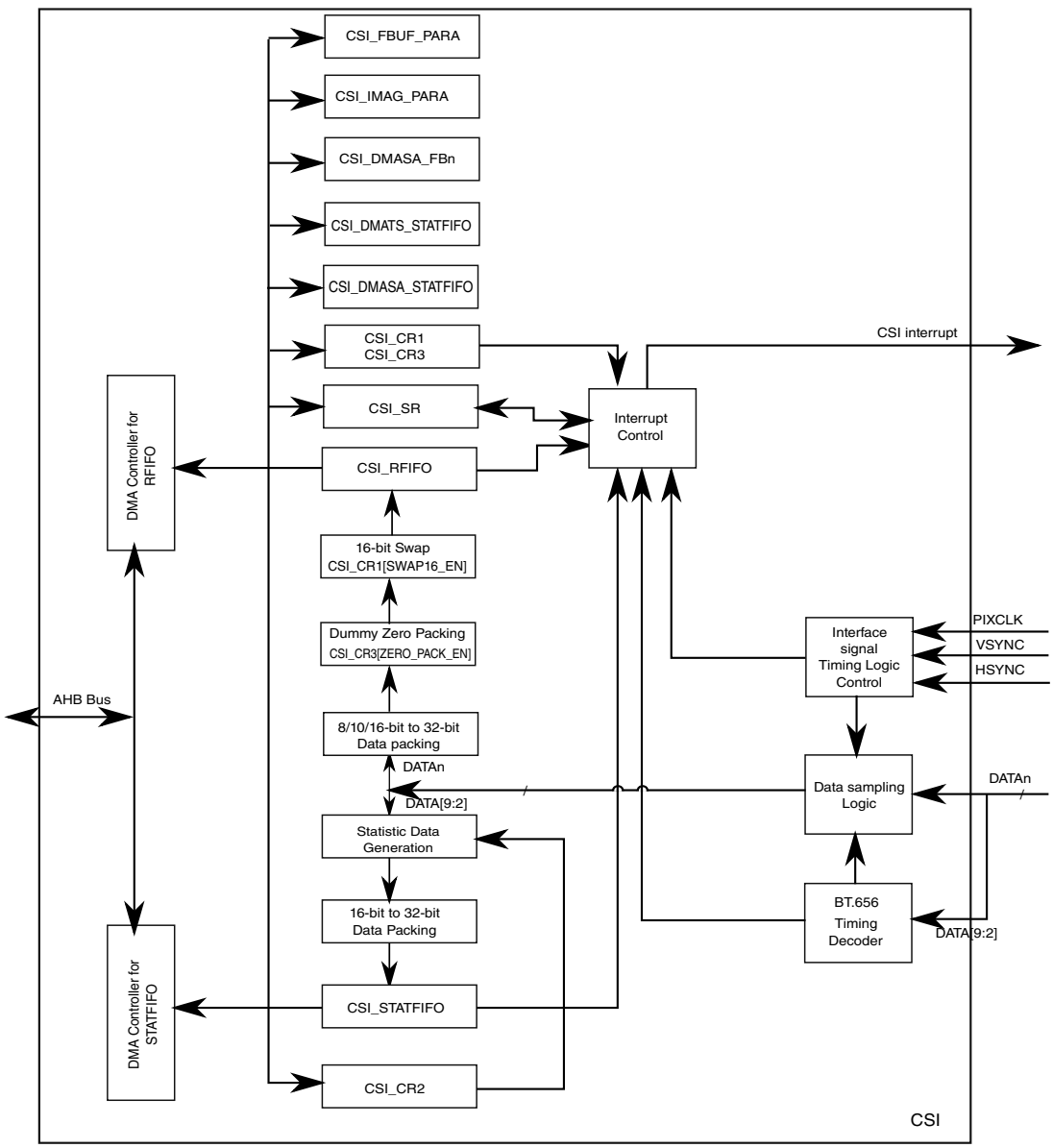


Figure 45-1. Block Diagram

45.2.2 Features

The features of the CSI include:

- Configurable interface logic to support most commonly available CMOS sensors.

- Support for ITU-R BT.656 video interface as well as traditional sensor interface.
- 8-bit / 16-bit / 24-bit data port for YCbCr, YUV, or RGB data input.
- 8-bit / 10-bit / 16-bit data port for Bayer data input.
- Full control of 8-bit/pixel, 10-bit/pixel or 16-bit / pixel data format to 64-bit receive FIFO packing.
- 256 x 64 FIFO to store received image pixel data.
- Statistic FIFO size of 64x32 bits
- Receive FIFO overrun protection mechanism.
- Embedded DMA controllers to transfer data from receive FIFO or statistic FIFO through AHB bus.
- Support 2D (data is chosen as x/y coordinate) DMA transfer from the receive FIFO to the frame buffers
- Support double buffering two frames in the external memory.
- Single interrupt source to interrupt controller from maskable interrupt sources: Start of Frame, End of Frame, Change of Field, FIFO full, FIFO overrun, DMA transfer done, BT.656 error and AHB bus response error.
- Configurable master clock frequency output to sensor.
- Support decoloring and binarization to accelerate image processing and recognition, and barcode/QR code decoding applications. It reduces memory space required and bus loading.

45.3 Functional Description

45.3.1 Principles of Operation

The information found here describes the modes of operation of the sensor interface.

The CSI is designed to support generic sensor interface timing as well as ITU-R BT.656 video interface timing. Traditional CMOS sensors typically use VSYNC (SOF), HSYNC (BLANK), and PIXCLK signals to output Bayer, RGB or YUV data. Smart CMOS sensors, that come with on-chip imaging processing, usually support video mode transfer. They use an embedded timing codec to replace the VSYNC and HSYNC signal. The timing codec is defined by the ITU-R BT.656 standard.

The CSI can support connection with the sensor as follows.

- To connect with one 8-bit sensor, the sensor data interface should connect to CSI_DATA[9:2].
- To connect with one 10-bit sensor, the sensor data interface should connect to CSI_DATA[9:0].

- To connect with one 12-bit sensor, the sensor data interface should connect to CSI_DATA[11:0].
- To connect with one 16-bit sensor, the sensor data interface should connect to CSI_DATA[15:0].
- To connect with one 24-bit sensor, connect the sensor data interface to CSI_DATA[23:0].

Table 45-2. CSI input data format

Data line	TVdecoder YCbCr	RGB888	RGB888/ YUV4444	RGB666	RGB565	YCbCr422	YCbCr422	Generic 10 bit	ITU-R BT. 656
DATA[23]	Y[7]	R[7]		R[5]					
DATA[22]	Y[6]	R[6]		R[4]					
DATA[21]	Y[5]	R[5]		R[3]					
DATA[20]	Y[4]	R[4]		R[2]					
DATA[19]	Y[3]	R[3]		R[1]					
DATA[18]	Y[2]	R[2]		R[0]					
DATA[17]	Y[1]	R[1]		Y[5]					
DATA[16]	Y[0]	R[0]		R[4]					
DATA[15]	Cb[7]	G[7]		G[5]	R[4]	Y[7]			
DATA[14]	Cb[6]	G[6]		G[4]	R[3]	Y[6]			
DATA[13]	Cb[5]	G[5]		G[3]	R[2]	Y[5]			
DATA[12]	Cb[4]	G[4]		G[2]	R[1]	Y[4]			
DATA[11]	Cb[3]	G[3]		G[1]	R[0]	Y[3]			
DATA[10]	Cb[2]	G[2]		G[0]	G[5]	Y[2]			
DATA[9]	Cb[1]	G[1]	R/G/B[7]	G[5]	G[4]	Y[1]	Y/C[7]	Ge[9]	C/Y[7]
DATA[8]	Cb[0]	G[0]	R/G/B[6]	G[4]	G[3]	Y[0]	Y/C[6]	Ge[8]	C/Y[6]
DATA[7]	Cr[7]	B[7]	R/G/B[5]	B[5]	G[2]	C[7]	Y/C[5]	Ge[7]	C/Y[5]
DATA[6]	Cr[6]	B[6]	R/G/B[4]	B[4]	G[1]	C[6]	Y/C[4]	Ge[6]	C/Y[4]
DATA[5]	Cr[5]	B[5]	R/G/B[3]	B[3]	G[0]	C[5]	Y/C[3]	Ge[5]	C/Y[3]
DATA[4]	Cr[4]	B[4]	R/G/B[2]	B[2]	B[4]	C[4]	Y/C[2]	Ge[4]	C/Y[2]
DATA[3]	Cr[3]	B[3]	R/G/B[1]	B[1]	B[3]	C[3]	Y/C[1]	Ge[3]	C/Y[1]
DATA[2]	Cr[2]	B[2]	R/G/B[0]	B[0]	B[2]	C[2]	Y/C[0]	Ge[2]	C/Y[0]
DATA[1]	Cr[1]	B[1]		B[5]	B[1]	C[1]		Ge[1]	
DATA[0]	Cr[0]	B[0]		B[4]	B[0]	C[0]		Ge[0]	

45.3.1.1 Data Transfer with the Embedded DMA Controllers

The CSI has two embedded DMA controllers, one for the receive FIFO and the other for the statistic FIFO. It supports 2D DMA transfer from the receive FIFO to the frame buffers in the external memory and linear DMA transfer from the statistic FIFO.

NOTE

One word = 4 bytes

Double words = 8 bytes

To transfer data from the RxFIFO to the external memory, the user should set the start address in the frame buffer where the transferred data is stored, the parameters of the frame buffers, and the parameters of the image coming from the sensor. The user can have two frame buffers in the external memory. Each one will store a frame of image coming from the sensor. The embedded DMA controller will first write the frame buffer1 and then frame buffer2. These two frame buffers will be written by turns. The start address should be aligned in double words and set in the CSI_DMASA-FB1 and CSI_DMASA-FB2 registers. In the CSI_FBUF_PARA register, the user should set the stride of the frame buffer to show how many double words to skip before starting to write the next row of the image. In the CSI_IMAG_PARA register, the user should set the width and height of the image coming from the sensor. The CSI_CR3[RxFF_LEVEL] and CSI_CR3[DMA_REQ_EN_RFF] bits also need to be set before the data transfer starts. When the number of the data in the RxFIFO reaches the trigger level, a DMA request will be sent to the embedded DMA controller and the data will be read out from the RxFIFO and written through AHB bus into the external frame buffers. The burst type of transfer can be INCR4, INCR8 and INCR16 by setting CSI_CR2[DMA_BURST_TYPE_RFF] bits. INCR4/INCR8/INCR16 are AHB protocols. After all data in an image frame are transferred, the CSI_SR[DMA_TSF_DONE_FB1] or CSI_SR[DMA_TSF_DONE_FB2] bit will be set and the interrupt can be triggered if the corresponding enable bit is set in CSI_CR1 register. The CSI_CR3[DMA_REFLASH_RFF] bit can be used to activate or restart the embedded DMA controller.

The RxFIFO has the overrun protection mechanism in case the RxFIFO is overrun during data transfer. If the RxFIFO is full and more data needs to be received during the data transfer, the RxFIFO will be overwritten continuously and all 128 words of data in the RxFIFO before overrun occurred will be discarded; the corresponding 128 words memory space in the frame buffer will keep the previous values. In other words, the DMA will skip the 128 word space fragment as if data is written.

To transfer data from the statistic FIFO to the external memory, the user should set the start address of the external memory where the transferred data is stored and the total transfer sizes. The start address and the transfer sizes are all aligned in double words and should be set in the CSIDMASA-STATFIFO and CSIDMATS-STATFIFO registers. The CSI_CR3[STATFF_LEVEL] and CSI_CR3[DMA_REQ_EN_SFF] bits should also be set before the data transfer starts. When the number of the data in the CSI_STATFIFO register reaches the trigger level, a dma request will be sent to the embedded DMA controller and the data will be read out from the STATFIFO and written through AHB

bus into the external memory. The burst type of transfer can be INCR4, INCR8 and INCR16 by setting CSI_CR2[DMA_BURST_TYPE_SFF] bit. After all expected data (defined by the total transfer sizes) are transferred, the CSI_SR[DMA_TSF_DONE_SFF] bit will be set and an interrupt can be triggered if the CSI_CR1[SFF_DMA_DONE_INTEN] is enabled. The CSI_CR3[DMA_REFLASH_SFF] bit can be used to activate or re-start the embedded DMA controller.

45.3.1.2 Gated Clock Mode

VSYNC, HSYNC, and PIXCLK signals are used in gated clock mode.

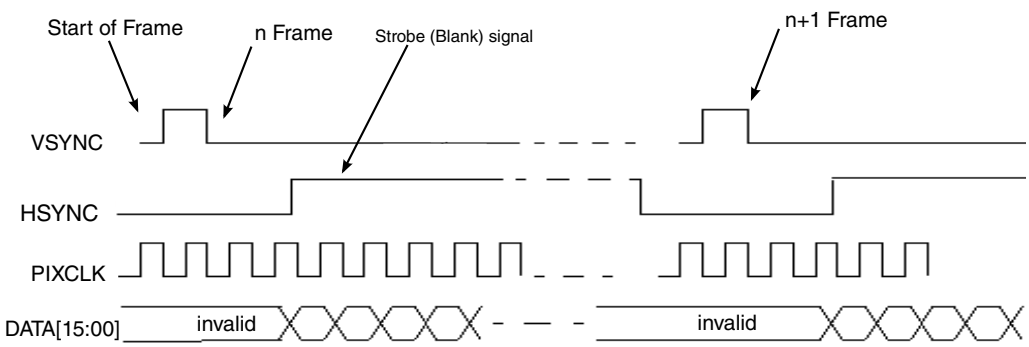


Figure 45-2. Gated Clock Mode Timing Diagram

NOTE

The CSI can be programmed to support rising/falling-edge triggered VSYNC through CSI_CR1[SOF_POL], active-high/low HSYNC using CSI_CR1[HSYNC_POL], and rising/falling-edge triggered PIXCLK through CSI_CR1[REDGE]. The configuration chosen by the user is referred to as an active edge.

A frame starts with an active edge on VSYNC, then HSYNC asserts and holds for the entire line. The Pixel clock is valid as long as HSYNC is asserted. Data is latched at the active edge of the valid pixel clocks. HSYNC deasserts at the end of line. Pixel clocks then become invalid and CSI stops receiving data from the stream. For the next line the HSYNC timing repeats. For the next frame the VSYNC timing repeats.

45.3.1.3 Non-Gated Clock Mode

In non-gated clock mode, only the VSYNC and PIXCLK signals are used; the HSYNC signal is ignored.

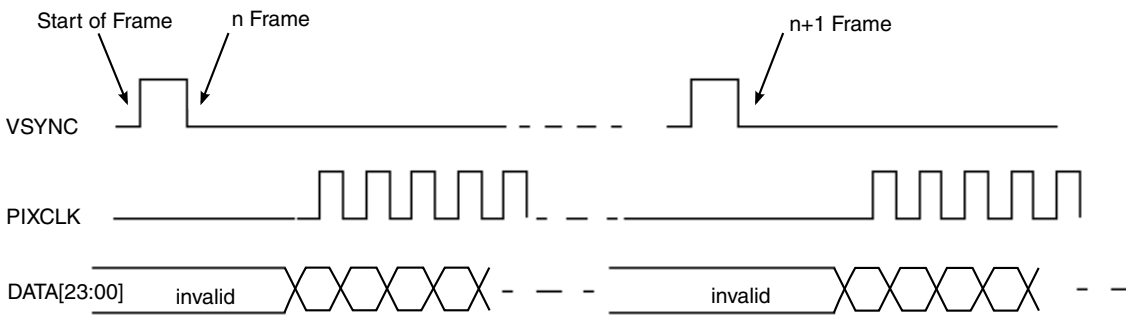


Figure 45-3. Non-Gated Clock Mode Timing Diagram

The overall timing of non-gated mode is the same as the gated-clock mode, except for the HSYNC signal. HSYNC signal is ignored by the CSI. All incoming pixel clocks are valid and cause data to be latched into RxFIFO. The PIXCLK signal is inactive (states low) until valid data is ready to be transmitted over the bus.

Figure 45-3 shows the timing of a typical sensor. Other sensors may have the slightly different timing from that shown. The CSI can be programed to support rising/falling-edge triggered VSYNC, active-high/low HSYNC, and rising/falling-edge triggered PIXCLK.

Below is the TV Decoder input mode timing diagram.

45.3.1.4 ITU-R BT.656 Interlace Mode

In interlace mode, an even field and an odd field are interlaced to create a frame. An even field consists of even-numbered lines (0,2,4,6, etc.), and an odd field consist of odd-numbered lines (1,3,5,7,etc.). Some vertical and horizontal blanking data is also added to the fields.

In BT.656 interlace mode, only the PIXCLK and CSI_DATA[9:2] signals are used. The start of frame and blank signals are replaced by a timing code which is embedded in the data stream. Each active line starts with an Start of Active Video (SAV) code and ends with an End of Active Video (EAV) code. In some cases, digital blanking is inserted in between EAV and SAV code. The CSI decodes and filters out the timing-coding from the data stream, recovering VSYNC and HSYNC signals for internal use, such as statistical

block control. Data is forwarded to the data receive and packing block in a sequential manner without reordering-that is, field 1 followed by field 2. The fields must be reordered in software to get back the original image.

Change of Field (COF) interrupt is triggered upon every field change. The interrupt service routine reads the status register to check for the current field.

According to the BT.656 specification, the image must be in 625/50 PAL or 525/60 NTSC format. Data must be in YCbCr422 format, each pixel contains 2 bytes, either Y + Cr or Y + Cb. These requirements are set for TV systems. The CSI module supports PAL and NTSC format only.

The following figure describes the frame structure in PAL system, showing vertical and horizontal blanking.

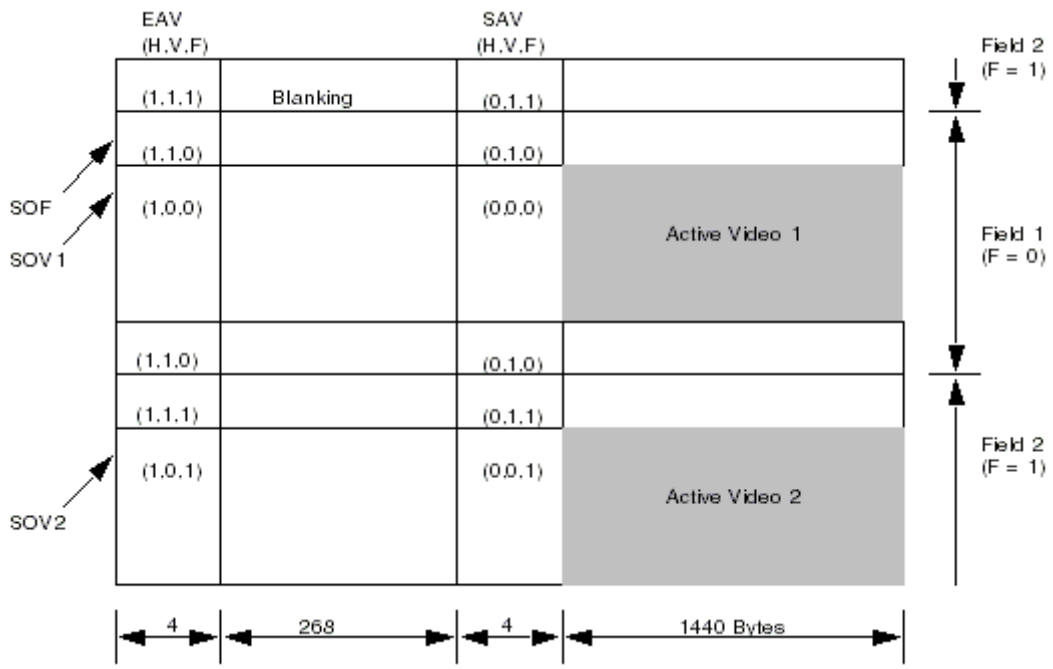


Figure 45-4. BT.656 Interlace Mode (PAL)

The following figure describes the general timing for a single line, showing SAV and EAV.

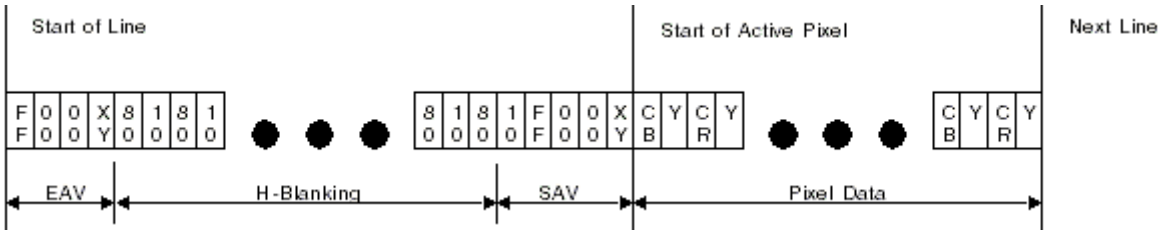


Figure 45-5. BT.656 General Line Timing

The coding tables recommended by the BT.656 specification are shown below. It is used in the BT.656 mode to decode the video stream. An interrupt is generated for SOF, which is decoded from the embedded timing codec.

Table 45-3. Coding for SAV and EAV

Data Bit Number	1st Byte 0xFF	2nd Byte 0x00	3rd Byte 0x00	4th Byte 0xXY
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

Table 45-4. Codes with Protection bits for Error Detection/Correction

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Table 45-5. Representations by F-Bit

F-Bit	Representations
0	ODD FIELD (FIELD 1)
1	EVEN FIELD (FIELD 2)

45.3.1.5 ITU-R BT.656 Progressive Mode

For a CMOS camera system of VGA or CIF resolution, strict adherence to the interlace requirements stated in the BT.656 standard is not required.

The image is considered to have only 1 active field which is scanned in a progressive manner. This active field is regarded as field 1 and the F-bit in the timing codec is ignored by the decoder. Most sensors support BT.656 timing in this mode (progressive) by default.

The following figure shows the typical flow of progressive mode.

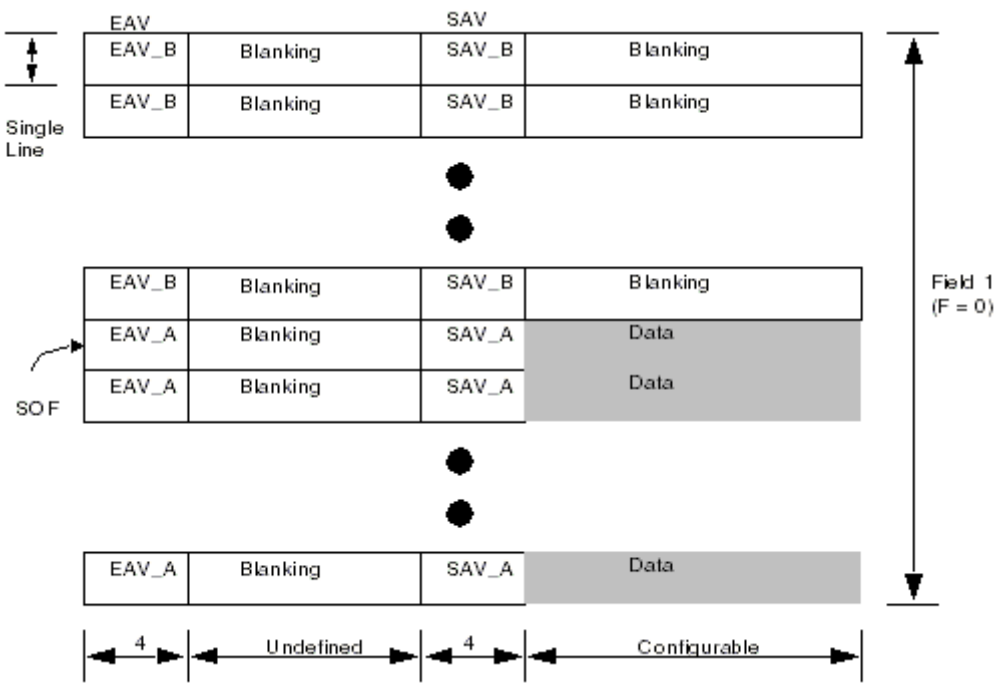


Figure 45-6. BT.656 Progressive Mode (General Case)

An interrupt is generated for SOF but not for COF. In the general case, when SOF information is retrieved from the embedded coding, it is known as internal VSYNC mode. In other cases, when the VSYNC signal is provided by the sensor, it is known as external VSYNC mode. The CSI can be operated in internal or external VSYNC mode.

45.3.1.6 Error Correction for ITU-R BT.656 Coding

According to the algorithm for BT.656 coding, protection bits in the SAV and EAV are encoded in the way that allows a 1-bit error to be corrected, or a 2-bit error to be detected by the decoder. This feature is supported by the interlace mode BT.656 decoder in CSI.

For the 1-bit error case, users can select the error to be corrected automatically, or simply shown as a status flag instead. For the 2-bit error case, because the decoder is unable to make a correction, the error would be shown as a status flag only.

An interrupt can be generated upon the detection of an error. This signal can be enabled or disabled without affecting the operation of the status bit.

45.3.1.7 Deinterlacer

Deinterlacing is the process of converting interlaced video, such as ITU-R BT.656 input or TV Decoder (VADC) input, into a non-interlaced form.

The CSI uses the weaving method to do deinterlacing. Weaving is done by adding consecutive fields together. CSI uses top-field detection function. No matter input is NTSC or PAL mode, the combined frame will always put the top-field first and then bottom-field.

45.3.2 Decoloring and Binarization

CSI involves the decoloring and binarization function and cooperates with SW to enormously accelerate the decoloring and binarization scanner speed and reduce the memory size.

45.3.2.1 Decoloring and Binarization Introduction

A barcode, including 1 dimensional (1-D) and 2 dimensional (2-D), can be read by an imaging device such as a camera and appropriately interpreted by process module.

Before detector and decoder, a decoloring and binarization scanner needs to convert the image to binary format. Previous chips usually use CPU to do such work (as shown in the figure below):

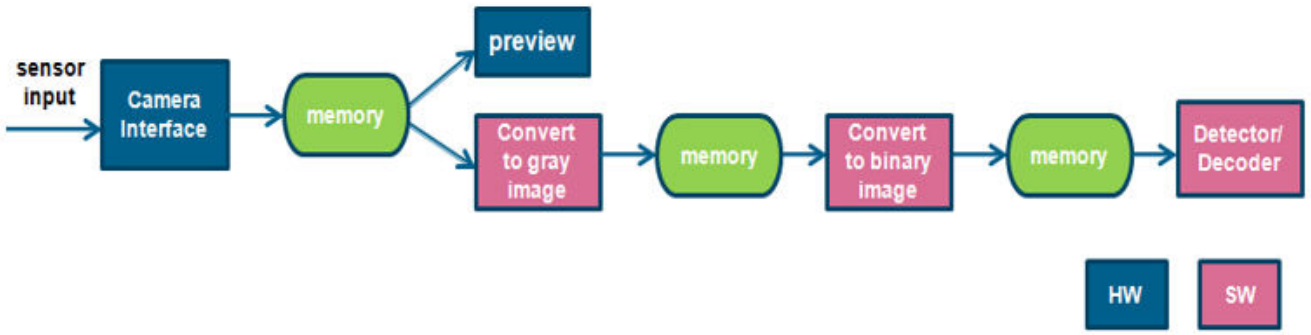


Figure 45-7. Previous Flow of Decoloring and Binarization Scanner

CSI is directly connected to external camera and synchronizes and processes the input image before storing to memory. After CSI, displayer will fetch the data from memory and display for customer review, meanwhile, CPU will fetch the data and convert the YUV data to binary data through two times memory read and write.

Using above architecture, previous barcode scanner flow not only consumes big memory size but also decrease the barcode scanner speed. Due to above disadvantage, upgraded CSI combines normal image preview and Decoloring and binarization together (as shown in the figure below).

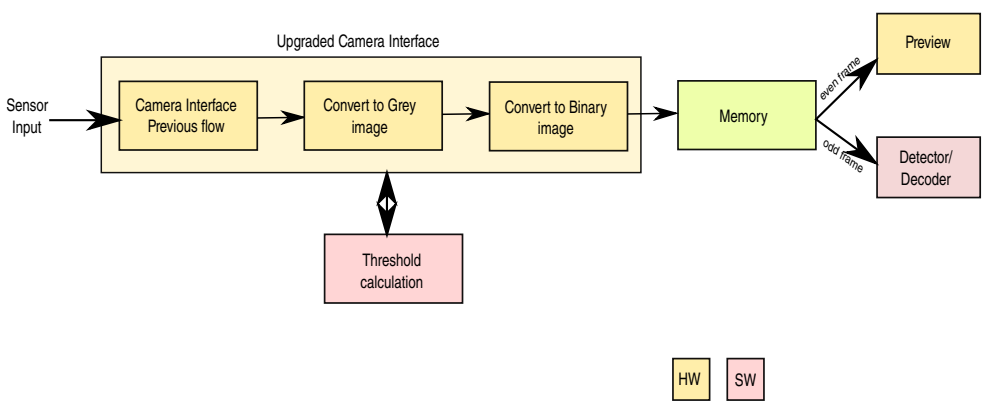


Figure 45-8. Updated Flow of Decoloring and Binarization Scanner

In a real-time camera sensor input, there is no much difference between two continuous frames, thus we can use the threshold calculated by SW based on the histogram of previous frame to do the binarization in current frame.

Upgraded flow combines the Decoloring and Binarization scanner binarization and preview to one flow. Even frame stores image to memory for preview while statistics the gray level histogram and sends to SW; odd frame uses the SW computed threshold based on previous gray level histogram and coverts the image to binary format before storing to memory for the scanner detector and decoder.

Using HW to take some binary work instead of SW can enormously accelerates the scanner speed and meanwhile reduces the memory size.

45.3.2.2 Upgraded CSI module

The upgraded architecture can help to achieve preview and binarization. The preview can be achieved using even frame and binarization using an odd frame. It takes maximum advantage of the normal SYNC, FIFO and DMA modules. By inserting the Binarization and Histogram modules between the sensor input and memory, the conversion of a YUV image to Binary is completed in a single pipeline.

The detailed diagram is shown below:

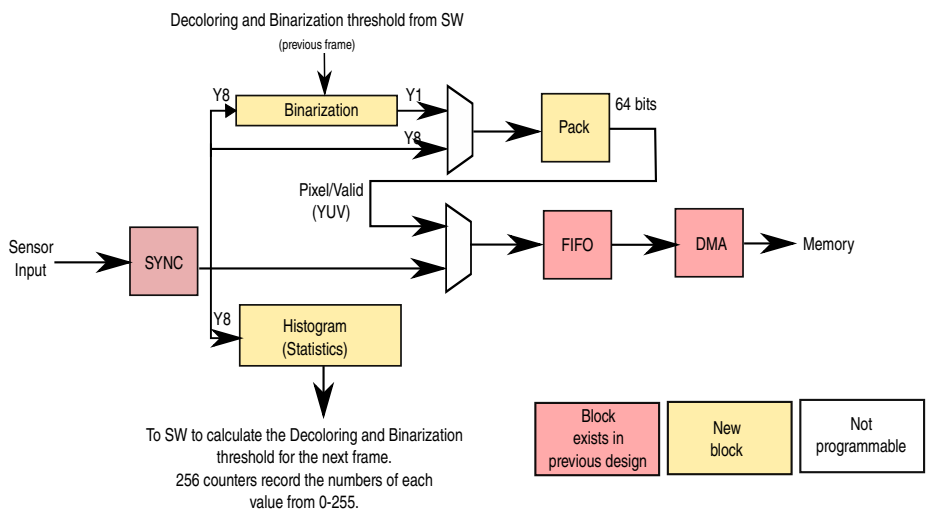


Figure 45-9. Upgraded CSI block diagram

The Binarization module utilizes a threshold based on grey level histogram to transform the image into black and white pixels. Y1 and Y8 data will be muxed together in the Pack module and packed to 32/64/128 bits per cycle as a system requires. Then it is muxed with normal image input. The output gets stored to the FIFO, read out by the DMA engine, and then stored in the external memory.

This upgraded method greatly reduces memory utilization.

The Even Frame data path is shown below:

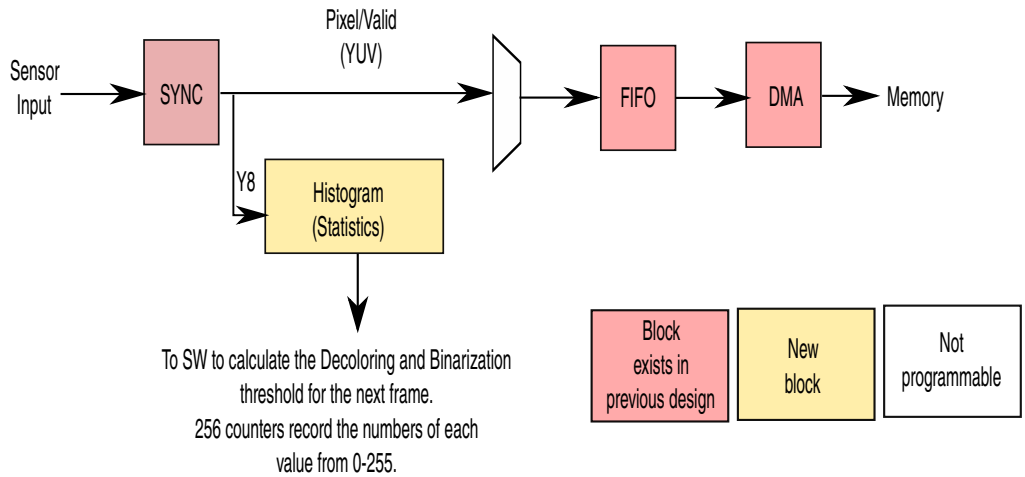


Figure 45-10. Even Frame Path

The Odd Frame data path is shown below:

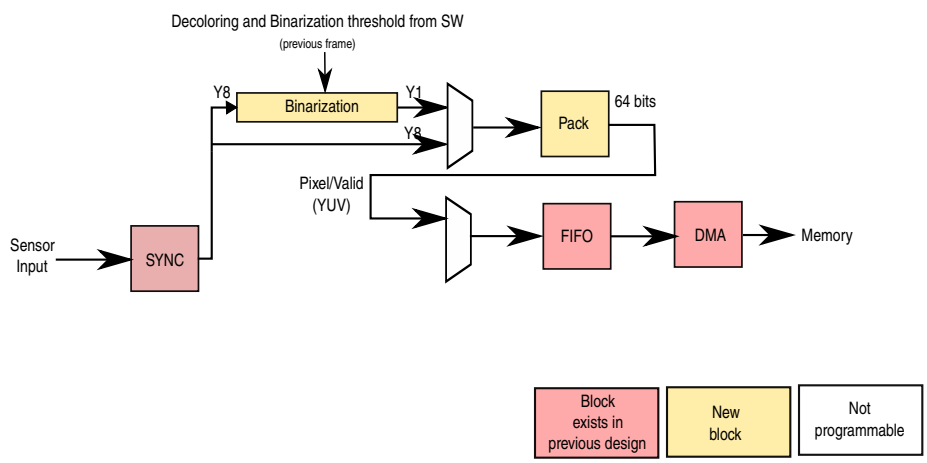


Figure 45-11. Odd Frame Path

Through above pipeline, an image, from YUV422 CMOS sensor, with 600KB size, will be transformed to 600K/16=37.5KB size. Only 1/16 memory needed using upgraded CSI IP compared to previous one. Besides, deleting two times read/write memory will

significantly accelerate the Decoloring and Binarization scanner speed. The color data is always saved to FB2, and gray scaled data always saved to FB1 in case of grey scale mode.

45.3.2.2.1 Decoloring and Binarization Histogram

Decoloring and Binarization uses a threshold to define the boundary between black and white pixels. The threshold is based on the image gray level histogram. Decoloring and Binarization histogram calculation consists of three steps as shown below:

- Extracting Y8 component from various data formats
- Counting each value (0~255) pixel numbers and storing to 256 counter registers
- Generating calculated done interrupt

When SW receives the calculated done interrupt, it reads out the 256 counter registers and chooses the suitable threshold by a certain algorithm based on the histogram and re-sends it to CSI module.

45.3.2.2.2 Binarization

Binarization transforms the Y8 data into 1-bit value (0 or 1) by comparing the value with the threshold configured by SW, and then sends output to the packing module.

Alternatively, the Y8 data can bypass the binarization module and stored directly to memory for backup function.

45.3.2.2.3 Limitation

Decoloring and Binarization scanner histogram starts at the start of a frame and stops as soon as the end of a frame is reached in order to leave maximum time period for SW involvement. It computes the threshold using this data. Due to camera video blanking limitation, it only works well in systems that satisfies hard real-time constraints, such as bare metal or real-time Operating Systems (OS).

45.3.3 Data Packing Style

Careful attention to endianess is needed given the different port sizes at different stages of the image capture path.

To enable flexible packing of image data before storage in the FIFOs, the CSI module can swap data fields by use of the CSI_CR1[PACK_DIR] and CSI_CR1[SWAP16_EN]).

The CSI module accepts 8-bit, 10-bit or 16-bit data from the sensor by configuring CSI_CR1[PIXEL_BIT] and CSI_CR3[SENSOR_16BITS]. The input data is packed according to the setting of CSI_CR1[PACK_DIR] bit. The packed data is stored in the RX FIFO according to the setting of the CSI_CR1[SWAP16_EN].

For 10-bit per pixel data format, each pixel is expanded to 16 bits by appending 6 zeros bits to the most significant bit.

When gray scale mode (CSI_CR20[QRCODE_EN]==1'b1) and Binary mode (CSI_CR20[BINARY_EN] == 1'b1) are enabled, data from sensor input will be converted from YUV to binary Y and packed to memory.

When gray scale mode (CSI_CR20[QRCODE_EN]==1'b1) and Y8 mode (CSI_CR20[BINARY_EN] == 1'b0) are enabled, data from sensor input will be converted from YUV to Y8 and packed to memory.

In binary mode, number of bits per pixel is 1. In Y8 mode, the number of bits per pixel is 8.

45.3.3.1 RX FIFO Path

45.3.3.1.1 Bayer Data

Bayer data is a type of raw data from the image sensor. This byte-wide data must be converted to the RGB space or YUV space by software. The data path for Bayer data is from the CSI to memory. If the system is in little endian, then the CSI_CR1[PACK_DIR] bit should be set to 0. 8-bit data format from a sensor is packed to 64 bits as P7.P6.P5.P4.P3.P2.P1.P0, where P0 is the pixel coming in time slot 0 (first data) and P3 is the pixel coming in time slot 3 . When the data is addressed as bytes by software, P0 is transferred first, P1 is transferred next, and so on. 10-bit data format is packed to 64 bits as 000000.P3.000000.P2.000000.P1.000000.P0, where P0 is the 10-bit data coming in time slot 0 (first pixel) and P3 is the 10-bit data coming in time slot 3 (fourth pixel). 16-bit data is packed to 64 bits as P3.P2.P1.P0, where P0-P3 are 16-bit data.

45.3.3.1.2 RGB565 Data

RGB565 data is processed data from the image sensor, which can be put directly into the display buffer. The data is 16 bits wide. The data path is from CSI to memory. It could be directed to the display controller from the memory. On the sensor side, data must be transmitted as P0 first, followed by P1, and so on. For each pixel, whether the MSB or LSB is sent first depends on the endianness of the sensor. Data is 16 bits wide with the MSB labeled RG, and the LSB labeled GB. P0 is represented as RG0 and GB0.

CSI receives data in one of the following sequence:

- RG0, GB0, RG1, GB1, while RG0 comes out at time slot 0 (first data), and GB1 comes out at time slot 3 (last data)
- GB0, RG0, GB1, RG1

Using the first sequence as an example, and assuming the system is running in little endian, the data is presented as:

- 8-bit data from sensor: RG0, GB0, RG1, GB1, …
- 64-bit data before storage in the CSI RX FIFO (PACK_DIR bit = 1): RG0GB0RG1GB1RG2GB2RG3GB3
- 64-bitdata in CSI RX FIFO (SWAP16_EN bit enabled): RG3GB3RG2GB2RG1GB1RG0GB0
- 64-bit transfer to system memory: RG3GB3RG2GB2RG1GB1RG0GB0
- 16-bit read by display controller: RG0GB0, RG1GB1

45.3.3.1.3 RGB888 Data

This is another kind of processed data from image sensor, which can be used for further image processing directly. Each of the data consist of 8-bit Red, 8-bit Green, and 8-bit Blue data. An example of timing scheme is shown in the following figure.

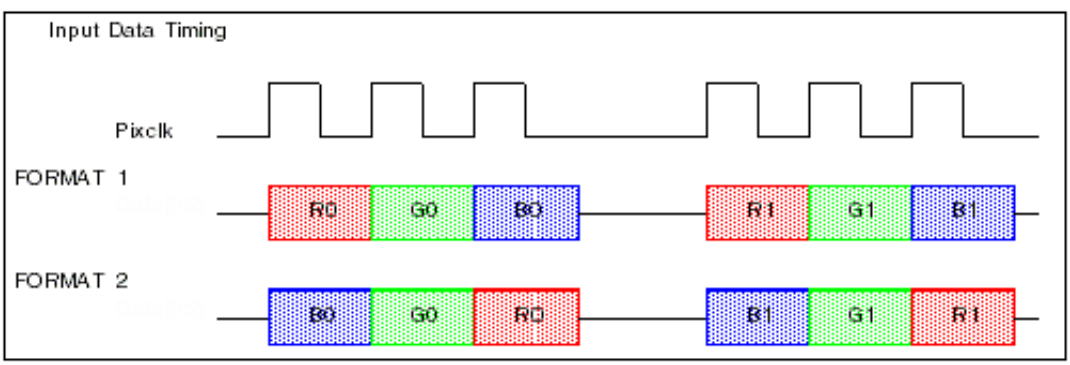


Figure 45-12. Sample Timing Diagram for RGB888 8 bits/cycle Data

An optional scheme to pack a dummy byte is provided. For every group of 3 bytes data, a dummy zero is packed to form a 32-bit word as shown in the following figure. The dummy zero can be packed at the LSB position or MSB position. Using RGB888A_FORMAT_SEL in CSI_CSICR18[18] to determine to put the dummy bytes packed at LSB or MSB position.

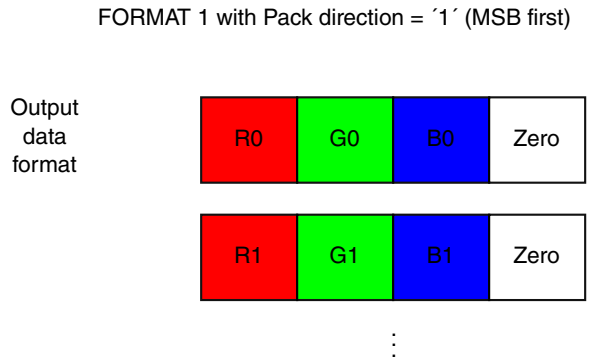


Figure 45-13. Dummy Byte Packing Scheme

45.3.3.1.4 Binary Data

This is one of processed data formats from the image sensor. For Y1 (Binary) mode (CSI_CR20[BINARY_EN] = 1), when grey scale mode is enabled, the data will be packed as {Y63,Y62,.....Y0} if CSI_CR20[BIG_EN]=1. Alternatively, if CSI_CR20[BIG_END]=0, the data will be packed as{Y0,Y1,... , Y63}. In this case, one pixel is 1 bit.

For Y8 mode (CSI_CR20[BINARY_EN]=0), when grey scale mode is enabled, the data will be packed as {Y7,Y6,.....Y0} if CSI_CR20[BIG_END] =1. Alternatively, if CSI_CR20[BIG_END]=0, the data will be packed as {Y0,Y1,... , Y7}. In this case, one pixel is 8 bits.

NOTE

The data is packed as 8 bits per byte.

45.3.3.2 STAT FIFO Path

Statistics only works for Bayer data in 8-bit per pixel format. It generates 16-bit statistical output from the 8-bit Bayer input (CSI_DATA[13:6]). The outputs are Sum of Green (G), Sum of Red (R), Sum of Blue (B), and Auto Focus (F). Each output is 16-bits wide.

The settings of CSI_CR1[PACK_DIR] and CSI_CR1[SWAP16_EN] bits have no effect on the input path. The CSI_CR1[PACK_DIR] only controls how the 16-bit stat output is packed into the 32-bit STAT FIFO.

When the CSI_CR1[PACK_DIR] = 1, the stat data is packed as:

First 32-bit: RG

Second 32-bit: BF

…

When the CSI_CR1[PACK_DIR] = 0, the stat data is packed as:

First 32-bit: GR

Second 32-bit: FB

...

The diagram shows how 16-bit data stat output is packed into 32-bit STATFIFO register:

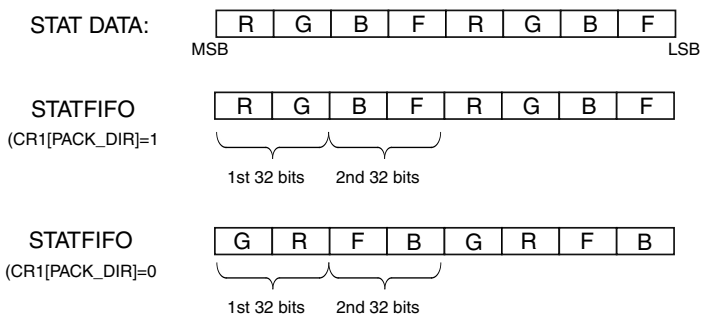


Figure 45-14. Data Packing

45.3.4 Clocks

The following table describes the clock sources for CSI. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 45-6. Clocks

Clock	Description
hclk	Module clock
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock
ipg_clk_s_raw	Peripheral raw data clock

45.3.5 Reset

CSI has one hardware reset. When set, it will reset all the logic and registers to their default values.

To re-enable CSI without using the hardware reset, follow the steps below:

1. Disable CSI by setting CSI_CR18[CSI_ENABLE]=0
2. Reflash the DMA controller for RFIFO by setting CSI_CR3[DMA_REFLASH_RFF]=1

3. Enable CSI by setting CSI_CR18[CSI_ENABLE]=1

45.3.6 Interrupt Generation

The information found here describes CSI events that generate interrupts.

45.3.6.1 Start Of Frame Interrupt (SOF_INT)

The source of an SOF interrupt is dependent on the mode of operation.

In traditional mode, VSYNC signal is taken from sensor and SOF_INT is generated at the rising or falling edge (programmable) of VSYNC.

In BT.656 interlace mode, the SOF interrupt information is retrieved from the embedded coding and SOF_INT is generated.

In BT.656 progressive mode, there are two sources of an SOF interrupt:

- In *internal* VSYNC mode, SOF is retrieved from the embedded coding.
- In *external* VSYNC mode, VSYNC is taken from the sensor and SOF is generated at the rising edge of VSYNC.

45.3.6.2 End Of Frame Interrupt (EOF_INT)

An EOF interrupt is generated when the frame ends and the complete frame data in RXFIFO is read.

The EOF event triggering works with the RX count register (CSIRXCNT). Software sets the RX count register to the frame size (in words). The CSI RX logic then counts the number of pixel data being received and compares it with the RX count. If the preset value is reached, an EOF interrupt is generated and the data in the RXFIFO are read. If a SOF event is detected before this happens, the EOF interrupt is not generated.

45.3.6.3 Change Of Field Interrupt (COF_INT)

The Change of Field interrupt is only valid in BT.656 Interlace mode. The COF interrupt is generated when the field toggles, either from field 1 to field 2, or field 2 to field 1.

Software should first check CSI_SR[COF_INT] bit before checking that F1_INT or F2_INT is turned on.

In PAL systems, the field changes at the beginning of the frame and coincides with SOF. For the first field, a COF interrupt is not generated, only an SOF. The COF interrupt is generated for the second field.

45.3.6.4 BT.656 Error Interrupt (ECC_INT)

The BT.656 Error Interrupt is only valid for BT.656 Interlace mode. An ECC interrupt is generated when an error is found on the SAV or EAV codes in the incoming stream. When this happens, the CSI_SR[ECC_INT] status bit is set.

45.3.6.5 RxFIFO Full Interrupt (RxFF_INT)

A RxFIFO full interrupt is generated when the number of data in RXFIFO reaches the water mark defined by RxFF_LEVEL in CSICR3.

45.3.6.6 Statistic FIFO Full Interrupt (STATFF_INT)

A StatFIFO full interrupt is generated when the number of data in STATFIFO reaches the water mark defined by STATFF_LEVEL in CSICR3.

45.3.6.7 RxFIFO Overrun Interrupt (RFF_OR_INT)

A RxFIFO Overrun interrupt is generated when the RxFIFO has 128 words data and more data is being written in.

45.3.6.8 Statistic FIFO Overrun Interrupt (SFF_OR_INT)

A StatFIFO Overrun interrupt is generated when the STATFIFO has 64 words data and more data is being written in.

45.3.6.9 Frame Buffer1 DMA Transfer Done Interrupt (DMA_TSF_DONE_FB1)

A DMA transfer done interrupt of frame buffer1 is generated when one frame of data are transferred from RxFIFO to the frame buffer1 in the external memory.

45.3.6.10 Frame Buffer2 DMA Transfer Done Interrupt (DMA_TSF_DONE_FB2)

A DMA transfer done interrupt of frame buffer2 is generated when one frame of data are transferred from RxFIFO to the frame buffer2 in the external memory.

45.3.6.11 Statistic FIFO DMA Transfer Done Interrupt (DMA_TSF_DONE_SFF)

A StatFIFO DMA transfer done interrupt is generated when all the data are transferred from StatFIFO to the external memory. The transfer size is defined in the STATFIFO DMA Transfer Size Register.

45.3.6.12 AHB Bus Response Error Interrupt (HRESP_ERR_INT)

An AHB Bus response error interrupt is generated when a bus error is detected.

This error can occur if an address is wrong. For more details, refer to AHB protocol.

45.3.6.13 Histogram Calculation Done Interrupt (HISTOGRAM_CALC_DONE)

A histogram calculation interrupt is generated when all the data are calculated by the histogram module. When this interrupt happens, driver can read the final statistic value for further process.

45.3.6.14 DMA Field 0 Transfer Done Interrupt (DMA_FIELD0_DONE)

A DMA transfer done interrupt of field 0 is generated when one field of data are transferred from RxFIFO to the frame buffer in external memory. This signal should work on interlaced mode.

45.3.6.15 DMA Field 1 Transfer Done Interrupt (DMA_FIELD1_DONE)

A DMA transfer done interrupt of field 1 is generated when one field of data are transferred from RxFIFO to the frame buffer in external memory. This signal should work on interlaced mode.

45.3.6.16 Base Address Change Error Interrupt (BASEADDR_CHANGE_ERROR)

A Base Address change error is generated when the base address changed while the last frame data transfer are not finished.

45.4 External Signals

The table below describes the external signals for the CSI. The external signals are tied between the CSI module and an external CMOS sensor.

Table 45-7. CSI External Signals

Signal	Description	Direction
DATAn	Data Sensor Signal, part of 16-bit Sensor Data Bus (Bayer, YUV, YCrCb, RGB)	I
FIELD	CSI Field Signal	I
HSYNC	Horizontal Sync (Blank Signal)	I
MCLK	CMOS Sensor Master Clock Note: MCLK is provided by the CCM module directly, not from the CSI module itself	O
PIXCLK	Pixel Clock	I
VSYNC	Vertical Sync (Start Of Frame)	I

NOTE

DATA[1:0] cannot be used for 8-bit bus.

45.5 Initialization

To initialize CSI, follow the steps below:

1. Enable the external sensor
2. Configure the CSI registers
3. Enable CSI by setting CSI_CR18[CSI_ENABLE]=1

The steps 1 and 2 can be interchanged.

45.6 CSI Memory Map/Register Definition

All the 32-bit registers of the CSI module are summarized in the Memory Map below:

CSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_0000	CSI Control Register 1 (CSI_CR1)	32	R/W	4000_0800h	45.6.1/2983
4080_0004	CSI Control Register 2 (CSI_CR2)	32	R/W	0000_0000h	45.6.2/2987
4080_0008	CSI Control Register 3 (CSI_CR3)	32	R/W	0000_0000h	45.6.3/2989
4080_000C	CSI Statistic FIFO Register (CSI_STATFIFO)	32	R	0000_0000h	45.6.4/2991
4080_0010	CSI RX FIFO Register (CSI_RFIFO)	32	R	0000_0000h	45.6.5/2992
4080_0014	CSI RX Count Register (CSI_RXCNT)	32	R/W	0000_9600h	45.6.6/2992
4080_0018	CSI Status Register (CSI_SR)	32	R/W	8000_4000h	45.6.7/2993
4080_0020	CSI DMA Start Address Register - for STATFIFO (CSI_DMASA_STATFIFO)	32	R/W	0000_0000h	45.6.8/2996
4080_0024	CSI DMA Transfer Size Register - for STATFIFO (CSI_DMATS_STATFIFO)	32	R/W	0000_0000h	45.6.9/2996
4080_0028	CSI DMA Start Address Register - for Frame Buffer1 (CSI_DMASA_FB1)	32	R/W	0000_0000h	45.6.10/ 2997
4080_002C	CSI DMA Transfer Size Register - for Frame Buffer2 (CSI_DMASA_FB2)	32	R/W	0000_0000h	45.6.11/ 2998
4080_0030	CSI Frame Buffer Parameter Register (CSI_FBUF_PARA)	32	R/W	0000_0000h	45.6.12/ 2998
4080_0034	CSI Image Parameter Register (CSI_IMAG_PARA)	32	R/W	0000_0000h	45.6.13/ 2999
4080_0048	CSI Control Register 18 (CSI_CR18)	32	R/W	0002_D000h	45.6.14/ 3000
4080_004C	CSI Control Register 19 (CSI_CR19)	32	R/W	0000_0000h	45.6.15/ 3003
4080_0050	CSI Control Register 20 (CSI_CR20)	32	R/W	0000_0000h	45.6.16/ 3004
4080_0054	CSI Control Register (CSI_CR21)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0058	CSI Control Register (CSI_CR22)	32	R/W	0000_0000h	45.6.17/ 3005
4080_005C	CSI Control Register (CSI_CR23)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0060	CSI Control Register (CSI_CR24)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0064	CSI Control Register (CSI_CR25)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0068	CSI Control Register (CSI_CR26)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_006C	CSI Control Register (CSI_CR27)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0070	CSI Control Register (CSI_CR28)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0074	CSI Control Register (CSI_CR29)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0078	CSI Control Register (CSI_CR30)	32	R/W	0000_0000h	45.6.17/ 3005
4080_007C	CSI Control Register (CSI_CR31)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0080	CSI Control Register (CSI_CR32)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0084	CSI Control Register (CSI_CR33)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0088	CSI Control Register (CSI_CR34)	32	R/W	0000_0000h	45.6.17/ 3005
4080_008C	CSI Control Register (CSI_CR35)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0090	CSI Control Register (CSI_CR36)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0094	CSI Control Register (CSI_CR37)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0098	CSI Control Register (CSI_CR38)	32	R/W	0000_0000h	45.6.17/ 3005
4080_009C	CSI Control Register (CSI_CR39)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00A0	CSI Control Register (CSI_CR40)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00A4	CSI Control Register (CSI_CR41)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00A8	CSI Control Register (CSI_CR42)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00AC	CSI Control Register (CSI_CR43)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00B0	CSI Control Register (CSI_CR44)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00B4	CSI Control Register (CSI_CR45)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00B8	CSI Control Register (CSI_CR46)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00BC	CSI Control Register (CSI_CR47)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00C0	CSI Control Register (CSI_CR48)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_00C4	CSI Control Register (CSI_CR49)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00C8	CSI Control Register (CSI_CR50)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00CC	CSI Control Register (CSI_CR51)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00D0	CSI Control Register (CSI_CR52)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00D4	CSI Control Register (CSI_CR53)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00D8	CSI Control Register (CSI_CR54)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00DC	CSI Control Register (CSI_CR55)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00E0	CSI Control Register (CSI_CR56)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00E4	CSI Control Register (CSI_CR57)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00E8	CSI Control Register (CSI_CR58)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00EC	CSI Control Register (CSI_CR59)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00F0	CSI Control Register (CSI_CR60)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00F4	CSI Control Register (CSI_CR61)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00F8	CSI Control Register (CSI_CR62)	32	R/W	0000_0000h	45.6.17/ 3005
4080_00FC	CSI Control Register (CSI_CR63)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0100	CSI Control Register (CSI_CR64)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0104	CSI Control Register (CSI_CR65)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0108	CSI Control Register (CSI_CR66)	32	R/W	0000_0000h	45.6.17/ 3005
4080_010C	CSI Control Register (CSI_CR67)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0110	CSI Control Register (CSI_CR68)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0114	CSI Control Register (CSI_CR69)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0118	CSI Control Register (CSI_CR70)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_011C	CSI Control Register (CSI_CR71)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0120	CSI Control Register (CSI_CR72)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0124	CSI Control Register (CSI_CR73)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0128	CSI Control Register (CSI_CR74)	32	R/W	0000_0000h	45.6.17/ 3005
4080_012C	CSI Control Register (CSI_CR75)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0130	CSI Control Register (CSI_CR76)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0134	CSI Control Register (CSI_CR77)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0138	CSI Control Register (CSI_CR78)	32	R/W	0000_0000h	45.6.17/ 3005
4080_013C	CSI Control Register (CSI_CR79)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0140	CSI Control Register (CSI_CR80)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0144	CSI Control Register (CSI_CR81)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0148	CSI Control Register (CSI_CR82)	32	R/W	0000_0000h	45.6.17/ 3005
4080_014C	CSI Control Register (CSI_CR83)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0150	CSI Control Register (CSI_CR84)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0154	CSI Control Register (CSI_CR85)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0158	CSI Control Register (CSI_CR86)	32	R/W	0000_0000h	45.6.17/ 3005
4080_015C	CSI Control Register (CSI_CR87)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0160	CSI Control Register (CSI_CR88)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0164	CSI Control Register (CSI_CR89)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0168	CSI Control Register (CSI_CR90)	32	R/W	0000_0000h	45.6.17/ 3005
4080_016C	CSI Control Register (CSI_CR91)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0170	CSI Control Register (CSI_CR92)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_0174	CSI Control Register (CSI_CR93)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0178	CSI Control Register (CSI_CR94)	32	R/W	0000_0000h	45.6.17/ 3005
4080_017C	CSI Control Register (CSI_CR95)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0180	CSI Control Register (CSI_CR96)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0184	CSI Control Register (CSI_CR97)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0188	CSI Control Register (CSI_CR98)	32	R/W	0000_0000h	45.6.17/ 3005
4080_018C	CSI Control Register (CSI_CR99)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0190	CSI Control Register (CSI_CR100)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0194	CSI Control Register (CSI_CR101)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0198	CSI Control Register (CSI_CR102)	32	R/W	0000_0000h	45.6.17/ 3005
4080_019C	CSI Control Register (CSI_CR103)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01A0	CSI Control Register (CSI_CR104)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01A4	CSI Control Register (CSI_CR105)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01A8	CSI Control Register (CSI_CR106)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01AC	CSI Control Register (CSI_CR107)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01B0	CSI Control Register (CSI_CR108)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01B4	CSI Control Register (CSI_CR109)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01B8	CSI Control Register (CSI_CR110)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01BC	CSI Control Register (CSI_CR111)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01C0	CSI Control Register (CSI_CR112)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01C4	CSI Control Register (CSI_CR113)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01C8	CSI Control Register (CSI_CR114)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_01CC	CSI Control Register (CSI_CR115)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01D0	CSI Control Register (CSI_CR116)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01D4	CSI Control Register (CSI_CR117)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01D8	CSI Control Register (CSI_CR118)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01DC	CSI Control Register (CSI_CR119)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01E0	CSI Control Register (CSI_CR120)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01E4	CSI Control Register (CSI_CR121)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01E8	CSI Control Register (CSI_CR122)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01EC	CSI Control Register (CSI_CR123)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01F0	CSI Control Register (CSI_CR124)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01F4	CSI Control Register (CSI_CR125)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01F8	CSI Control Register (CSI_CR126)	32	R/W	0000_0000h	45.6.17/ 3005
4080_01FC	CSI Control Register (CSI_CR127)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0200	CSI Control Register (CSI_CR128)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0204	CSI Control Register (CSI_CR129)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0208	CSI Control Register (CSI_CR130)	32	R/W	0000_0000h	45.6.17/ 3005
4080_020C	CSI Control Register (CSI_CR131)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0210	CSI Control Register (CSI_CR132)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0214	CSI Control Register (CSI_CR133)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0218	CSI Control Register (CSI_CR134)	32	R/W	0000_0000h	45.6.17/ 3005
4080_021C	CSI Control Register (CSI_CR135)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0220	CSI Control Register (CSI_CR136)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_0224	CSI Control Register (CSI_CR137)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0228	CSI Control Register (CSI_CR138)	32	R/W	0000_0000h	45.6.17/ 3005
4080_022C	CSI Control Register (CSI_CR139)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0230	CSI Control Register (CSI_CR140)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0234	CSI Control Register (CSI_CR141)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0238	CSI Control Register (CSI_CR142)	32	R/W	0000_0000h	45.6.17/ 3005
4080_023C	CSI Control Register (CSI_CR143)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0240	CSI Control Register (CSI_CR144)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0244	CSI Control Register (CSI_CR145)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0248	CSI Control Register (CSI_CR146)	32	R/W	0000_0000h	45.6.17/ 3005
4080_024C	CSI Control Register (CSI_CR147)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0250	CSI Control Register (CSI_CR148)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0254	CSI Control Register (CSI_CR149)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0258	CSI Control Register (CSI_CR150)	32	R/W	0000_0000h	45.6.17/ 3005
4080_025C	CSI Control Register (CSI_CR151)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0260	CSI Control Register (CSI_CR152)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0264	CSI Control Register (CSI_CR153)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0268	CSI Control Register (CSI_CR154)	32	R/W	0000_0000h	45.6.17/ 3005
4080_026C	CSI Control Register (CSI_CR155)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0270	CSI Control Register (CSI_CR156)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0274	CSI Control Register (CSI_CR157)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0278	CSI Control Register (CSI_CR158)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_027C	CSI Control Register (CSI_CR159)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0280	CSI Control Register (CSI_CR160)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0284	CSI Control Register (CSI_CR161)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0288	CSI Control Register (CSI_CR162)	32	R/W	0000_0000h	45.6.17/ 3005
4080_028C	CSI Control Register (CSI_CR163)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0290	CSI Control Register (CSI_CR164)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0294	CSI Control Register (CSI_CR165)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0298	CSI Control Register (CSI_CR166)	32	R/W	0000_0000h	45.6.17/ 3005
4080_029C	CSI Control Register (CSI_CR167)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02A0	CSI Control Register (CSI_CR168)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02A4	CSI Control Register (CSI_CR169)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02A8	CSI Control Register (CSI_CR170)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02AC	CSI Control Register (CSI_CR171)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02B0	CSI Control Register (CSI_CR172)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02B4	CSI Control Register (CSI_CR173)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02B8	CSI Control Register (CSI_CR174)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02BC	CSI Control Register (CSI_CR175)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02C0	CSI Control Register (CSI_CR176)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02C4	CSI Control Register (CSI_CR177)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02C8	CSI Control Register (CSI_CR178)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02CC	CSI Control Register (CSI_CR179)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02D0	CSI Control Register (CSI_CR180)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_02D4	CSI Control Register (CSI_CR181)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02D8	CSI Control Register (CSI_CR182)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02DC	CSI Control Register (CSI_CR183)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02E0	CSI Control Register (CSI_CR184)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02E4	CSI Control Register (CSI_CR185)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02E8	CSI Control Register (CSI_CR186)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02EC	CSI Control Register (CSI_CR187)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02F0	CSI Control Register (CSI_CR188)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02F4	CSI Control Register (CSI_CR189)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02F8	CSI Control Register (CSI_CR190)	32	R/W	0000_0000h	45.6.17/ 3005
4080_02FC	CSI Control Register (CSI_CR191)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0300	CSI Control Register (CSI_CR192)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0304	CSI Control Register (CSI_CR193)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0308	CSI Control Register (CSI_CR194)	32	R/W	0000_0000h	45.6.17/ 3005
4080_030C	CSI Control Register (CSI_CR195)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0310	CSI Control Register (CSI_CR196)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0314	CSI Control Register (CSI_CR197)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0318	CSI Control Register (CSI_CR198)	32	R/W	0000_0000h	45.6.17/ 3005
4080_031C	CSI Control Register (CSI_CR199)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0320	CSI Control Register (CSI_CR200)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0324	CSI Control Register (CSI_CR201)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0328	CSI Control Register (CSI_CR202)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_032C	CSI Control Register (CSI_CR203)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0330	CSI Control Register (CSI_CR204)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0334	CSI Control Register (CSI_CR205)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0338	CSI Control Register (CSI_CR206)	32	R/W	0000_0000h	45.6.17/ 3005
4080_033C	CSI Control Register (CSI_CR207)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0340	CSI Control Register (CSI_CR208)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0344	CSI Control Register (CSI_CR209)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0348	CSI Control Register (CSI_CR210)	32	R/W	0000_0000h	45.6.17/ 3005
4080_034C	CSI Control Register (CSI_CR211)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0350	CSI Control Register (CSI_CR212)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0354	CSI Control Register (CSI_CR213)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0358	CSI Control Register (CSI_CR214)	32	R/W	0000_0000h	45.6.17/ 3005
4080_035C	CSI Control Register (CSI_CR215)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0360	CSI Control Register (CSI_CR216)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0364	CSI Control Register (CSI_CR217)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0368	CSI Control Register (CSI_CR218)	32	R/W	0000_0000h	45.6.17/ 3005
4080_036C	CSI Control Register (CSI_CR219)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0370	CSI Control Register (CSI_CR220)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0374	CSI Control Register (CSI_CR221)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0378	CSI Control Register (CSI_CR222)	32	R/W	0000_0000h	45.6.17/ 3005
4080_037C	CSI Control Register (CSI_CR223)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0380	CSI Control Register (CSI_CR224)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_0384	CSI Control Register (CSI_CR225)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0388	CSI Control Register (CSI_CR226)	32	R/W	0000_0000h	45.6.17/ 3005
4080_038C	CSI Control Register (CSI_CR227)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0390	CSI Control Register (CSI_CR228)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0394	CSI Control Register (CSI_CR229)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0398	CSI Control Register (CSI_CR230)	32	R/W	0000_0000h	45.6.17/ 3005
4080_039C	CSI Control Register (CSI_CR231)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03A0	CSI Control Register (CSI_CR232)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03A4	CSI Control Register (CSI_CR233)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03A8	CSI Control Register (CSI_CR234)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03AC	CSI Control Register (CSI_CR235)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03B0	CSI Control Register (CSI_CR236)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03B4	CSI Control Register (CSI_CR237)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03B8	CSI Control Register (CSI_CR238)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03BC	CSI Control Register (CSI_CR239)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03C0	CSI Control Register (CSI_CR240)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03C4	CSI Control Register (CSI_CR241)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03C8	CSI Control Register (CSI_CR242)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03CC	CSI Control Register (CSI_CR243)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03D0	CSI Control Register (CSI_CR244)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03D4	CSI Control Register (CSI_CR245)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03D8	CSI Control Register (CSI_CR246)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_03DC	CSI Control Register (CSI_CR247)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03E0	CSI Control Register (CSI_CR248)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03E4	CSI Control Register (CSI_CR249)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03E8	CSI Control Register (CSI_CR250)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03EC	CSI Control Register (CSI_CR251)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03F0	CSI Control Register (CSI_CR252)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03F4	CSI Control Register (CSI_CR253)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03F8	CSI Control Register (CSI_CR254)	32	R/W	0000_0000h	45.6.17/ 3005
4080_03FC	CSI Control Register (CSI_CR255)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0400	CSI Control Register (CSI_CR256)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0404	CSI Control Register (CSI_CR257)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0408	CSI Control Register (CSI_CR258)	32	R/W	0000_0000h	45.6.17/ 3005
4080_040C	CSI Control Register (CSI_CR259)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0410	CSI Control Register (CSI_CR260)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0414	CSI Control Register (CSI_CR261)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0418	CSI Control Register (CSI_CR262)	32	R/W	0000_0000h	45.6.17/ 3005
4080_041C	CSI Control Register (CSI_CR263)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0420	CSI Control Register (CSI_CR264)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0424	CSI Control Register (CSI_CR265)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0428	CSI Control Register (CSI_CR266)	32	R/W	0000_0000h	45.6.17/ 3005
4080_042C	CSI Control Register (CSI_CR267)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0430	CSI Control Register (CSI_CR268)	32	R/W	0000_0000h	45.6.17/ 3005

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_0434	CSI Control Register (CSI_CR269)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0438	CSI Control Register (CSI_CR270)	32	R/W	0000_0000h	45.6.17/ 3005
4080_043C	CSI Control Register (CSI_CR271)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0440	CSI Control Register (CSI_CR272)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0444	CSI Control Register (CSI_CR273)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0448	CSI Control Register (CSI_CR274)	32	R/W	0000_0000h	45.6.17/ 3005
4080_044C	CSI Control Register (CSI_CR275)	32	R/W	0000_0000h	45.6.17/ 3005
4080_0450	CSI Control Register (CSI_CR276)	32	R/W	0000_0000h	45.6.17/ 3005

45.6.1 CSI Control Register 1 (CSI_CR1)

This register controls the sensor interface timing and interrupt generation. The interrupt enable bits in this register control the interrupt signals and the status bits. That means status bits will only function when the corresponding interrupt bits are enabled.

Address: 4080_0000h base + 0h offset = 4080_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SWAP16_EN	EXT_VSYNC	EOF_INT_EN	Reserved	VIDEO_MODE	COF_INT_EN	SF_OR_INTEN	RF_OR_INTEN	Reserved	SFF_DMA_DONE_INTEN	STATFF_INTEN	FB2_DMA_DONE_INTEN	FB1_DMA_DONE_INTEN	RXFF_INTEN	SOF_POL	SOF_INTEN
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			HISTOGRAM_CALC_DONE_IE	HSYNC_POL	CCIR_EN	Reserved	FCC	PACK_DIR	CLR_STATFIFO	CLR_RXFIFO	GCLK_MODE	INV_DATA	INV_PCLK	REDGE	PIXEL_BIT
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

CSI_CR1 field descriptions

Field	Description
31 SWAP16_EN	SWAP 16-Bit Enable. This bit enables the swapping of 16-bit data. Data is packed from 8-bit or 10-bit to 32-bit first (according to the setting of PACK_DIR) and then swapped as 16-bit words before being put into the RX FIFO. The action of the bit only affects the RX FIFO and has no affect on the STAT FIFO. **NOTE:** Example of swapping enabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x 33441122 **NOTE:** Example of swapping disabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x11223344

Table continues on the next page...

CSI_CR1 field descriptions (continued)

Field	Description
	0 Disable swapping 1 Enable swapping
30 EXT_VSYNC	External VSYNC Enable. This bit controls the operational VSYNC mode. **NOTE:** This only works when the CSI is in BT.656 progressive mode. 0 Internal VSYNC mode 1 External VSYNC mode
29 EOF_INT_EN	End-of-Frame Interrupt Enable. This bit enables and disables the EOF interrupt. 0 EOF interrupt is disabled. 1 EOF interrupt is generated when RX count value is reached.
28 Reserved	This field is reserved.
27 VIDEO_MODE	Video mode select. This bit controls the video mode in BT.656 mode and TV decoder input. 0 Progressive mode is selected 1 Interlace mode is selected
26 COF_INT_EN	Change Of Image Field (COF) Interrupt Enable. This bit enables the COF interrupt. This bit works only in BT.656 interlace mode which is when CCIR_EN = 1 and CCIR_MODE = 1. 0 COF interrupt is disabled 1 COF interrupt is enabled
25 SF_OR_INTEN	STAT FIFO Overrun Interrupt Enable. This bit enables the STATFIFO overrun interrupt. 0 STATFIFO overrun interrupt is disabled 1 STATFIFO overrun interrupt is enabled
24 RF_OR_INTEN	RxFIFO Overrun Interrupt Enable. This bit enables the RX FIFO overrun interrupt. 0 RxFIFO overrun interrupt is disabled 1 RxFIFO overrun interrupt is enabled
23 Reserved	This field is reserved. This bit should read 0.
22 SFF_DMA_ DONE_INTEN	STATFIFO DMA Transfer Done Interrupt Enable. This bit enables the interrupt of STATFIFO DMA transfer done. 0 STATFIFO DMA Transfer Done interrupt disable 1 STATFIFO DMA Transfer Done interrupt enable
21 STATFF_INTEN	STATFIFO Full Interrupt Enable. This bit enables the STAT FIFO interrupt. 0 STATFIFO full interrupt disable 1 STATFIFO full interrupt enable
20 FB2_DMA_ DONE_INTEN	Frame Buffer2 DMA Transfer Done Interrupt Enable. This bit enables the interrupt of Frame Buffer2 DMA transfer done. 0 Frame Buffer2 DMA Transfer Done interrupt disable 1 Frame Buffer2 DMA Transfer Done interrupt enable

Table continues on the next page...

CSI_CR1 field descriptions (continued)

Field	Description
19 FB1_DMA_ DONE_INTEN	Frame Buffer1 DMA Transfer Done Interrupt Enable. This bit enables the interrupt of Frame Buffer1 DMA transfer done. 0 Frame Buffer1 DMA Transfer Done interrupt disable 1 Frame Buffer1 DMA Transfer Done interrupt enable
18 RXFF_INTEN	RxFIFO Full Interrupt Enable. This bit enables the RxFIFO full interrupt. 0 RxFIFO full interrupt disable 1 RxFIFO full interrupt enable
17 SOF_POL	SOF Interrupt Polarity. This bit controls the condition that generates an SOF interrupt. 0 SOF interrupt is generated on SOF falling edge 1 SOF interrupt is generated on SOF rising edge
16 SOF_INTEN	Start Of Frame (SOF) Interrupt Enable. This bit enables the SOF interrupt. 0 SOF interrupt disable 1 SOF interrupt enable
15–13 Reserved	This field is reserved.
12 HISTOGRAM_ CALC_DONE_IE	Histogram Interrupt Enable 0 Histogram done interrupt disable 1 Histogram done interrupt enable
11 HSYNC_POL	HSYNC Polarity Select. This bit controls the polarity of HSYNC. This bit only works in gated-clock-that is, GCLK_MODE = 1 and CCIR_EN = 0. 0 HSYNC is active low 1 HSYNC is active high
10 CCIR_EN	BT.656 Interface Enable. This bit selects the type of interface used. 0 Traditional interface is selected. 1 BT.656 interface is selected.
9 Reserved	This field is reserved.
8 FCC	FIFO Clear Control. This bit determines how the RXFIFO and STATFIFO are cleared. When Synchronous FIFO clear is selected the RXFIFO and STATFIFO are cleared, and STAT block is reset, on every SOF. FIFOs and STAT block restarts immediately after reset. For information on the operation when Asynchronous FIFO clear is selected, refer to the descriptions for the CLR_RXFIFO and CLR_STATFIFO bits. 0 Asynchronous FIFO clear is selected. 1 Synchronous FIFO clear is selected.
7 PACK_DIR	Data Packing Direction. This bit Controls how 8-bit/10-bit image data is packed into 32-bit RX FIFO, and how 16-bit statistical data is packed into 32-bit STAT FIFO. 0 Pack from LSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x44332211 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xBBBBAAAA in STAT FIFO. 1 Pack from MSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x11223344 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xAAAABBBB in STAT FIFO.

Table continues on the next page...

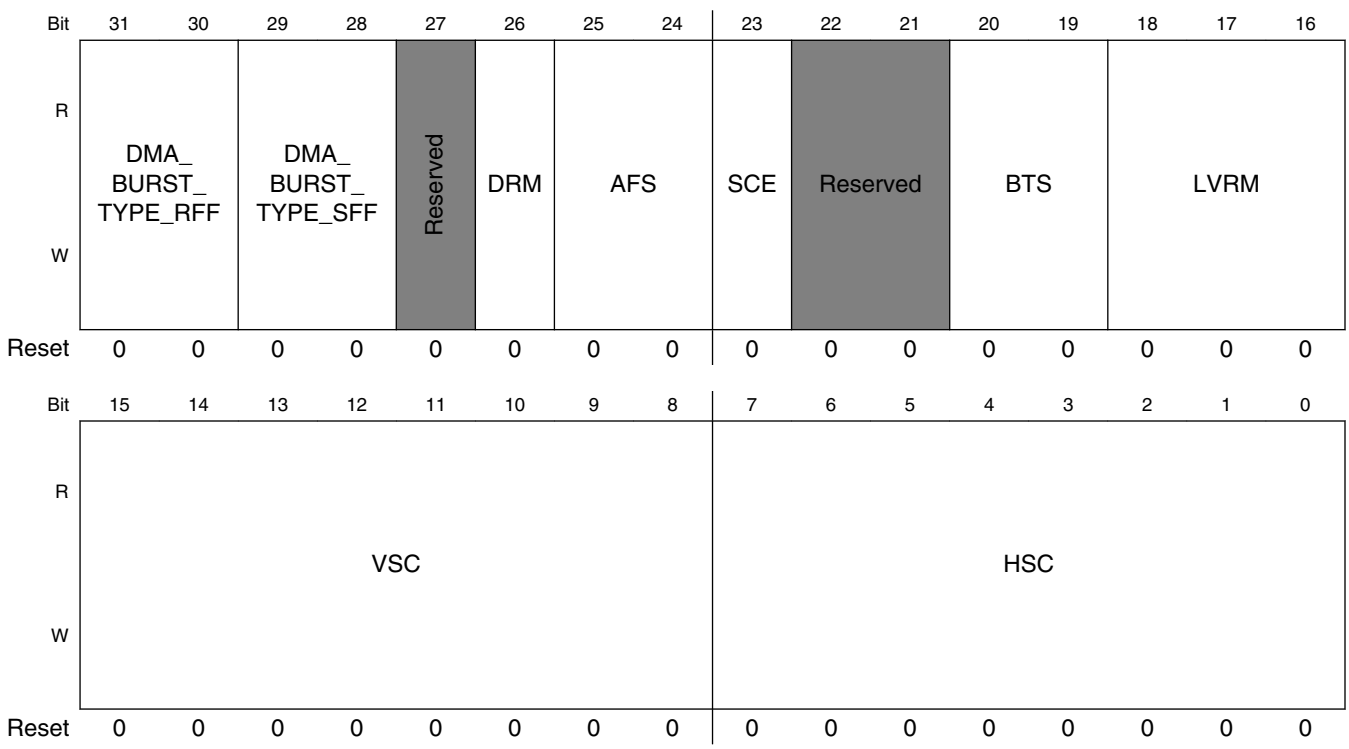
CSI_CR1 field descriptions (continued)

Field	Description
6 CLR_STATFIFO	Asynchronous STATFIFO Clear. This bit clears the STATFIFO and Reset STAT block. This bit works only in async FIFO clear mode-that is, FCC = 0. Otherwise this bit is ignored. Writing 1 will clear STATFIFO and reset STAT block immediately, STATFIFO and STAT block then wait and restart after the arrival of next SOF. The bit is restored to 0 automatically after finish. Normally reads 0.
5 CLR_RXFIFO	Asynchronous RXFIFO Clear. This bit clears the RXFIFO. This bit works only in async FIFO clear mode-that is, FCC = 0. Otherwise this bit is ignored. Writing 1 clears the RXFIFO immediately, RXFIFO restarts immediately after that. The bit is restored to 0 automatically after finish. Normally reads 0.
4 GCLK_MODE	Gated Clock Mode Enable. Controls if CSI is working in gated or non-gated mode. This bit works only in traditional mode-that is, CCIR_EN = 0. Otherwise this bit is ignored. 0 Non-gated clock mode. All incoming pixel clocks are valid. HSYNC is ignored. 1 Gated clock mode. Pixel clock signal is valid only when HSYNC is active.
3 INV_DATA	Invert Data Input. This bit enables or disables internal inverters on the data lines. 0 CSI_D[7:0] data lines are directly applied to internal circuitry 1 CSI_D[7:0] data lines are inverted before applied to internal circuitry
2 INV_PCLK	Invert Pixel Clock Input. This bit determines if the Pixel Clock (CSI_PIXCLK) is inverted before it is applied to the CSI module. 0 CSI_PIXCLK is directly applied to internal circuitry 1 CSI_PIXCLK is inverted before applied to internal circuitry
1 REDGE	Valid Pixel Clock Edge Select. Selects which edge of the CSI_PIXCLK is used to latch the pixel data. 0 Pixel data is latched at the falling edge of CSI_PIXCLK 1 Pixel data is latched at the rising edge of CSI_PIXCLK
0 PIXEL_BIT	Pixel Bit. This bit indicates the bayer data width for each pixel. This bit should be configured before activating or re-starting the embedded DMA controller. 0 8-bit data for each pixel 1 10-bit data for each pixel

45.6.2 CSI Control Register 2 (CSI_CR2)

This register provides the statistic block with data about which live view (sensor input) resolution is being used, and the starting sensor pixel of the Bayer pattern. It also contains the horizontal and vertical count used to determine the number of pixels to skip between the 64 x 64 blocks of statistics when generating statistics on live view image that are greater than 512 x 384.

Address: 4080_0000h base + 4h offset = 4080_0004h



CSI_CR2 field descriptions

Field	Description
31–30 DMA_BURST_ TYPE_RFF	Burst Type of DMA Transfer from RxFIFO. Selects the burst type of DMA transfer from RxFIFO. X0 INCR8 01 INCR4 11 INCR16
29–28 DMA_BURST_ TYPE_SFF	Burst Type of DMA Transfer from STATFIFO. Selects the burst type of DMA transfer from STATFIFO. X0 INCR8 01 INCR4 11 INCR16

Table continues on the next page...

CSI_CR2 field descriptions (continued)

Field	Description
27 Reserved	This field is reserved. This bit should read 0
26 DRM	Double Resolution Mode. Controls size of statistics grid. 0 Stats grid of 8 x 6 1 Stats grid of 8 x 12
25–24 AFS	Auto Focus Spread. Selects which green pixels are used for auto-focus. 00 Abs Diff on consecutive green pixels 01 Abs Diff on every third green pixels 1x Abs Diff on every four green pixels
23 SCE	Skip Count Enable. Skip count is the number of pixels to skip between the 64x64 blocks of statistics. 0 Skip count disable 1 Skip count enable
22–21 Reserved	This field is reserved. These bits should read 0.
20–19 BTS	Bayer Tile Start. Controls the Bayer pattern starting point. 00 GR 01 RG 10 BG 11 GB
18–16 LVRM	Live View Resolution Mode. Selects the grid size used for live view resolution. 0 512 x 384 1 448 x 336 2 384 x 288 3 384 x 256 4 320 x 240 5 288 x 216 6 400 x 300
15–8 VSC	Vertical Skip Count. Contains the number of rows to skip. SCE must be 1, otherwise VSC is ignored. 0-255 Number of rows to skip minus 1
HSC	Horizontal Skip Count. Contains the number of pixels to skip. SCE must be 1, otherwise HSC is ignored. 0-255 Number of pixels to skip minus 1

45.6.3 CSI Control Register 3 (CSI_CR3)

This read/write register acts as an extension of the functionality of the CSI Control register 1, adding additional control and features.

Address: 4080_0000h base + 8h offset = 4080_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FRMCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FRMCNT_RST	DMA_REFLASH_RFF	DMA_REFLASH_SFF	DMA_REQ_EN_RFF	DMA_REQ_EN_SFF	STATFF_LEVEL			HRESP_ERR_EN	RxFF_LEVEL			SENSOR_16BITS	ZERO_PACK_EN	ECC_INT_EN	ECC_AUTO_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CR3 field descriptions

Field	Description
31–16 FRMCNT	Frame Counter. This is a 16-bit Frame Counter (Wraps around automatically after reaching the maximum)
15 FRMCNT_RST	Frame Count Reset. Resets the Frame Counter. (Cleared automatically after reset is done) 0 Do not reset 1 Reset frame counter immediately
14 DMA_ REFLASH_RFF	Reflash DMA Controller for RxFIFO. This bit reflash the embedded DMA controller for RxFIFO. It should be reflashed before the embedded DMA controller starts to work. (Cleared automatically after reflashing is done) 0 No reflashing 1 Reflash the embedded DMA controller
13 DMA_ REFLASH_SFF	Reflash DMA Controller for STATFIFO. This bit reflash the embedded DMA controller for STATFIFO. It should be reflashed before the embedded DMA controller starts to work. (Cleared automatically after reflashing is done) 0 No reflashing 1 Reflash the embedded DMA controller

Table continues on the next page...

CSI_CR3 field descriptions (continued)

Field	Description
12 DMA_REQ_EN_RFF	DMA Request Enable for RxFIFO. This bit enables the dma request from RxFIFO to the embedded DMA controller. 0 Disable the dma request 1 Enable the dma request
11 DMA_REQ_EN_SFF	DMA Request Enable for STATFIFO. This bit enables the dma request from STATFIFO to the embedded DMA controller. 0 Disable the dma request 1 Enable the dma request
10–8 STATFF_LEVEL	STATFIFO Full Level. When the number of data in STATFIFO reach this level, STATFIFO full interrupt is generated, or STATFIFO DMA request is sent. 000 4 Double words 001 8 Double words 010 12 Double words 011 16 Double words 100 24 Double words 101 32 Double words 110 48 Double words 111 64 Double words
7 HRESP_ERR_EN	Hresponse Error Enable. This bit enables the hresponse (AHB protocol standard) error interrupt. 0 Disable hresponse error interrupt 1 Enable hresponse error interrupt
6–4 RxFF_LEVEL	**RxFIFO Full Level**. When the number of data in RxFIFO reaches this level, a RxFIFO full interrupt is generated, or an RXFIFO DMA request is sent. If the number of data is smaller, it will send request to bus with small burst size. 000 4 Double words 001 8 Double words 010 16 Double words 011 24 Double words 100 32 Double words 101 48 Double words 110 64 Double words 111 96 Double words
3 SENSOR_16BITS	16-bit Sensor Mode. This bit indicates one 16-bit sensor connected to the 16-bit data ports. This bit should be set if there is one 16-bit sensor connected. This bit should be configured before re-flashing or restarting the embedded DMA controller. 0 Only one 8-bit sensor is connected. 1 One 16-bit sensor is connected.
2 ZERO_PACK_EN	Dummy Zero Packing Enable. This bit causes a dummy zero to be packed with every 3 incoming bytes, forming a 32-bit word. The dummy zero is always packed to the LSB position. This packing function is only available in 8-bit/pixel mode. 0 Zero packing disabled 1 Zero packing enabled

Table continues on the next page...

CSI_CR3 field descriptions (continued)

Field	Description
1 ECC_INT_EN	Error Detection Interrupt Enable. This bit enables and disables the error detection interrupt. This feature only works in BT.656 interlace mode. 0 No interrupt is generated when error is detected. Only the status bit ECC_INT is set. 1 Interrupt is generated when error is detected.
0 ECC_AUTO_EN	Automatic Error Correction Enable. This bit enables and disables the automatic error correction. If an error occurs and error correction is disabled only the ECC_INT status bit is set. This feature only works in BT. 656 interlace mode. 0 Auto Error correction is disabled. 1 Auto Error correction is enabled.

45.6.4 CSI Statistic FIFO Register (CSI_STATFIFO)

The StatFIFO is a read-only register containing statistic data from the sensor. Writing to this register has no effect.

Address: 4080_0000h base + Ch offset = 4080_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STAT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

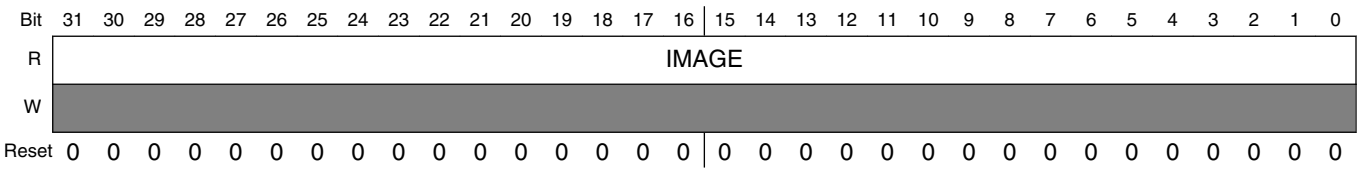
CSI_STATFIFO field descriptions

Field	Description
STAT	Static data from sensor

45.6.5 CSI RX FIFO Register (CSI_RFIFO)

This read-only register contains received image data. Writing to this register has no effect.

Address: 4080_0000h base + 10h offset = 4080_0010h



CSI_RFIFO field descriptions

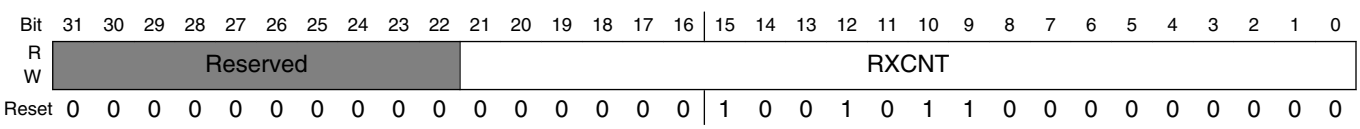
Field	Description
IMAGE	Received image data

45.6.6 CSI RX Count Register (CSI_RXCNT)

This register works for End of Frame (EOF) interrupt generation while width/height is used for DMA done. It should be set to the number of words to receive that would generate an EOF interrupt.

There is an internal counter that counts the number of words read from the RX FIFO. Whenever the RX FIFO is being read, by either the CPU or the embedded DMA controller, the counter value is updated and compared with this register. If the values match, then an EOF interrupt is triggered.

Address: 4080_0000h base + 14h offset = 4080_0014h



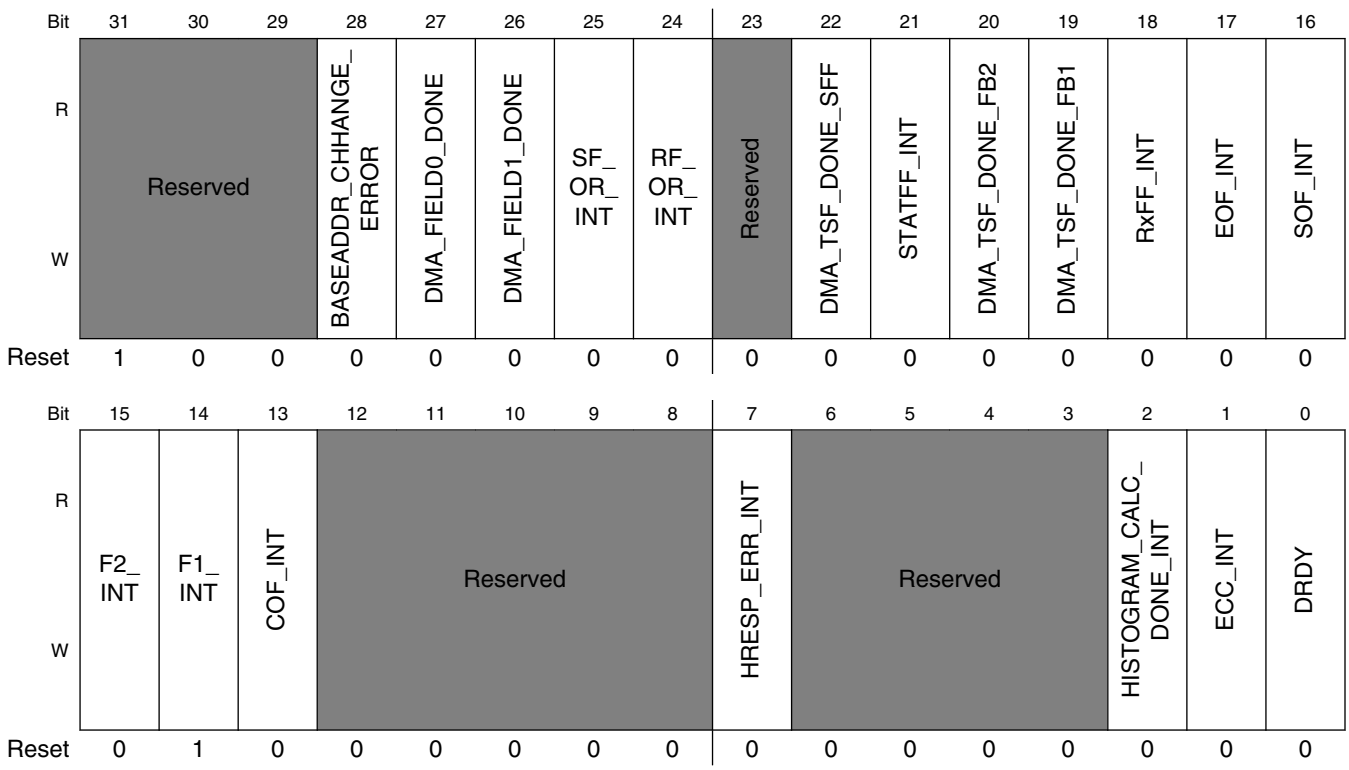
CSI_RXCNT field descriptions

Field	Description
31–22 Reserved	This field is reserved. These bits should read 0.
RXCNT	RxFIFO Count. This 22-bit counter for RXFIFO is updated each time the RXFIFO is read by CPU or DMA.This counter should be set to the expected number of words to receive that would generate an EOF interrupt.

45.6.7 CSI Status Register (CSI_SR)

This read/write register shows sensor interface status, and which kind of interrupt is being generated. The corresponding interrupt bits must be set for the status bit to function. Status bits will function normally even if the corresponding interrupt enable bits are not enabled.

Address: 4080_0000h base + 18h offset = 4080_0018h



CSI_SR field descriptions

Field	Description
31–29 Reserved	This field is reserved.
28 BASEADDR_CHHANGE_ERROR	When using base address switching enable, this bit will be 1 when switching occur before DMA complete. This bit will be clear by writing 1. When this interrupt happens, follow the steps listed below. 1. Unassert the CSI enable, CSIx_CSICR18 bit31, 2. Reflash the DMA, assert the CSIX_CSICR3 bit 14, 3. Assert the CSI enable, CSIx_CSICR18 bit31.
27 DMA_FIELD0_DONE	When DMA field 0 is complete, this bit will be set to 1(clear by writing 1).

Table continues on the next page...

CSI_SR field descriptions (continued)

Field	Description
26 DMA_FIELD1_ DONE	When DMA field 1 is complete, this bit will be set to 1(clear by writing 1).
25 SF_OR_INT	STATFIFO Overrun Interrupt Status. Indicates the overflow status of the STATFIFO register. (Cleared by writing 1) 0 STATFIFO has not overflowed. 1 STATFIFO has overflowed.
24 RF_OR_INT	RxFIFO Overrun Interrupt Status. Indicates the overflow status of the RxFIFO register. (Cleared by writing 1) 0 RXFIFO has not overflowed. 1 RXFIFO has overflowed.
23 Reserved	This field is reserved. This bit should read 0.
22 DMA_TSF_ DONE_SFF	DMA Transfer Done from StatFIFO. Indicates that the dma transfer from StatFIFO is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by writting 1 or reflashing the StatFIFO dma controller in CSICR3.(Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
21 STATFF_INT	STATFIFO Full Interrupt Status. Indicates the number of data in the STATFIFO reaches the trigger level. (this bit is cleared automatically by reading the STATFIFO) 0 STATFIFO is not full. 1 STATFIFO is full.
20 DMA_TSF_ DONE_FB2	DMA Transfer Done in Frame Buffer2. Indicates that the DMA transfer from RxFIFO to Frame Buffer2 is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by by writting 1 or reflashing the RxFIFO dma controller in CSICR3. (Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
19 DMA_TSF_ DONE_FB1	DMA Transfer Done in Frame Buffer1. Indicates that the DMA transfer from RxFIFO to Frame Buffer1 is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by by writting 1 or reflashing the RxFIFO dma controller in CSICR3. (Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
18 RxFF_INT	RXFIFO Full Interrupt Status. Indicates the number of data in the RxFIFO reaches the trigger level. (this bit is cleared automatically by reading the RxFIFO) 0 RxFIFO is not full. 1 RxFIFO is full.
17 EOF_INT	End of Frame (EOF) Interrupt Status. Indicates when EOF is detected. (Cleared by writing 1) 0 EOF is not detected. 1 EOF is detected.
16 SOF_INT	Start of Frame Interrupt Status. Indicates when SOF is detected. (Cleared by writing 1)

Table continues on the next page...

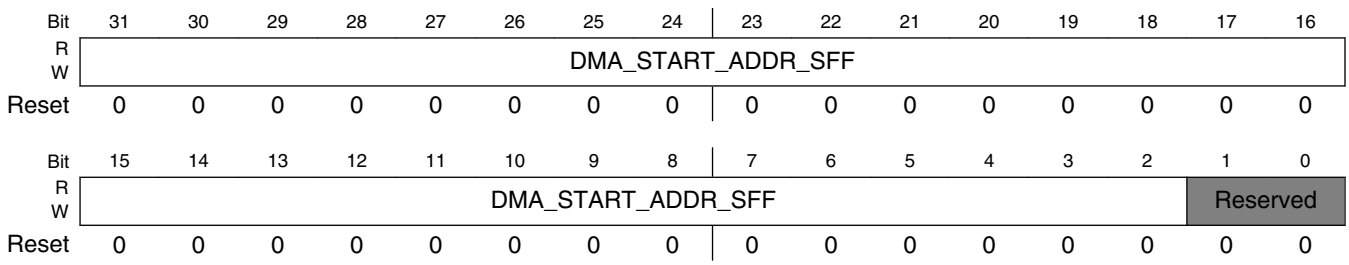
CSI_SR field descriptions (continued)

Field	Description
	0 SOF is not detected. 1 SOF is detected.
15 F2_INT	BT.656 Field 2 Interrupt Status. Indicates the presence of field 2 of video in BT.656 mode. (Cleared automatically when current field does not match) **NOTE:** Only works in BT.656 Interlace mode. 0 Field 2 of video is not detected 1 Field 2 of video is about to start
14 F1_INT	BT.656 Field 1 Interrupt Status. Indicates the presence of field 1 of video in BT.656 mode. (Cleared automatically when current field does not match) **NOTE:** Only works in BT.656 Interlace mode. 0 Field 1 of video is not detected. 1 Field 1 of video is about to start.
13 COF_INT	Change Of Field Interrupt Status. Indicates that a change of the video field has been detected. Only works in BT.656 Interlace mode. Software should read this bit first and then dispatch the new field from F1_INT and F2_INT. (Cleared by writing 1) 0 Video field has no change. 1 Change of video field is detected.
12–8 Reserved	This field is reserved. These bits should read 0.
7 HRESP_ERR_INT	Hresponse Error Interrupt Status. Indicates that a hresponse (AHB protocol standard) error has been detected. (Cleared by writing 1) 0 No hresponse error. 1 Hresponse error is detected.
6–3 Reserved	This field is reserved. These bits should read 0.
2 HISTOGRAM_CALC_DONE_INT	0 Histogram calculation is not finished 1 Histogram calculation is done and driver can access the PIXEL_COUNTERS(CSI_CSICR21~CSI_CSICR276) to get the gray level
1 ECC_INT	BT.656 Error Interrupt. This bit indicates an error has occurred. This only works in BT.656 Interlace mode. (Cleared by writing 1) 0 No error detected 1 Error is detected in BT.656 coding
0 DRDY	RXFIFO Data Ready. Indicates the presence of data that is ready for transfer in the RxFIFO. (Cleared automatically by reading FIFO) 0 No data (word) is ready 1 At least 1 datum (word) is ready in RXFIFO.

45.6.8 CSI DMA Start Address Register - for STATFIFO (CSI_DMASA_STATFIFO)

This register provides the start address for the embedded DMA controller of STATFIFO. The embedded DMA controller will read data from STATFIFO and write it to the external memory from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 20h offset = 4080_0020h



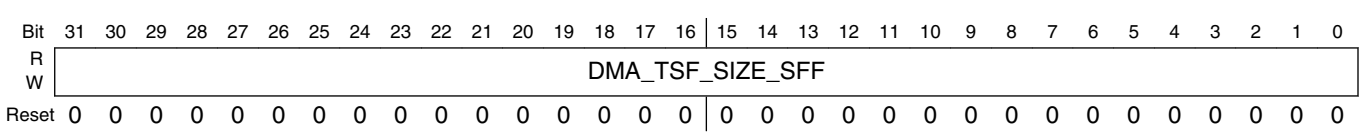
CSI_DMASA_STATFIFO field descriptions

Field	Description
31–2 DMA_START_ADDR_SFF	DMA Start Address for STATFIFO. Indicates the start address to write data. The embedded DMA controller will read data from STATFIFO and write it from this address through AHB bus. **NOTE:** The address should be double words aligned.
Reserved	This field is reserved. These bits should read 0.

45.6.9 CSI DMA Transfer Size Register - for STATFIFO (CSI_DMATS_STATFIFO)

This register provides the total transfer size for the embedded DMA controller of STATFIFO. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 24h offset = 4080_0024h



CSI_DMATS_STATFIFO field descriptions

Field	Description
DMA_TSF_SIZE_SFF	DMA Transfer Size for STATFIFO. Indicates how many words to be transfered by the embedded DMA controller. **NOTE:** The size should be double words aligned.

45.6.10 CSI DMA Start Address Register - for Frame Buffer1 (CSI_DMASA_FB1)

This register provides the start address in the frame buffer1 for the embedded DMA controller of RxFIFO. The embedded DMA controller will read data from RxFIFO and write it to the frame buffer1 from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 28h offset = 4080_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_START_ADDR_FB1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_START_ADDR_FB1														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_DMASA_FB1 field descriptions

Field	Description
31–2 DMA_START_ADDR_FB1	DMA Start Address in Frame Buffer1. Indicates the start address to write data. The embedded DMA controller will read data from RxFIFO and write it from this address through AHB bus. **NOTE:** The address should be double words aligned.
Reserved	This field is reserved. These bits should read 0.

45.6.11 CSI DMA Transfer Size Register - for Frame Buffer2 (CSI_DMASA_FB2)

This register provides the start address in the frame buffer2 for the embedded DMA controller of RxFIFO. The embedded DMA controller will read data from RxFIFO and write it to the frame buffer2 from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 2Ch offset = 4080_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_START_ADDR_FB2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_START_ADDR_FB2														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_DMASA_FB2 field descriptions

Field	Description
31–2 DMA_START_ADDR_FB2	DMA Start Address in Frame Buffer2. Indicates the start address to write data. The embedded DMA controller will read data from RxFIFO and write it from this address through AHB bus. **NOTE:** The address should be double words aligned.
Reserved	This field is reserved. These bits should read 0.

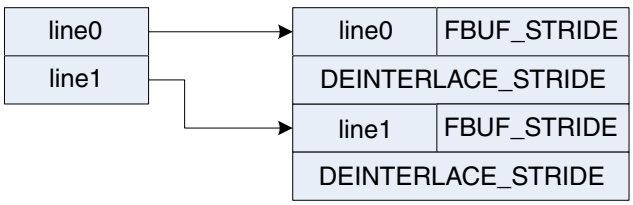
45.6.12 CSI Frame Buffer Parameter Register (CSI_FBUF_PARA)

This register provides the stride of the frame buffer to show how many words to skip before starting to write the next row of the image. The width of the frame buffer minus the width of the image is the stride. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 30h offset = 4080_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEINTERLACE_STRIDE																FBUF_STRIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_FBUF_PARA field descriptions

Field	Description
31–16 DEINTERLACE_ STRIDE	DEINTERLACE_STRIDE is only used in the deinterlace mode. If line stride feature is supported in deinterlace mode, FBUF_STRIDE and DEINTERLACE_STRIDE need to be configured at the same time. DEINTERLACE_STRIDE is configured the same as line width. 
FBUF_STRIDE	Frame Buffer Parameter. Indicates the stride of the frame buffer. The width of the frame buffer(in double words) minus the width of the image(in double words) is the stride. The stride should be double words aligned. The embedded DMA controller will skip the stride before starting to write the next row of the image. **NOTE:** Double words is equivalent to 8 bytes.

45.6.13 CSI Image Parameter Register (CSI_IMAG_PARA)

This register provides the width and the height of the image from the sensor. The width and height should be aligned in pixel. The width of the image multiplied by the height is the total pixel size that will be transferred in a frame by the embedded DMA controller. This register should be configured before activating or restarting the embedded DMA controller.

Address: 4080_0000h base + 34h offset = 4080_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMAGE_WIDTH																IMAGE_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

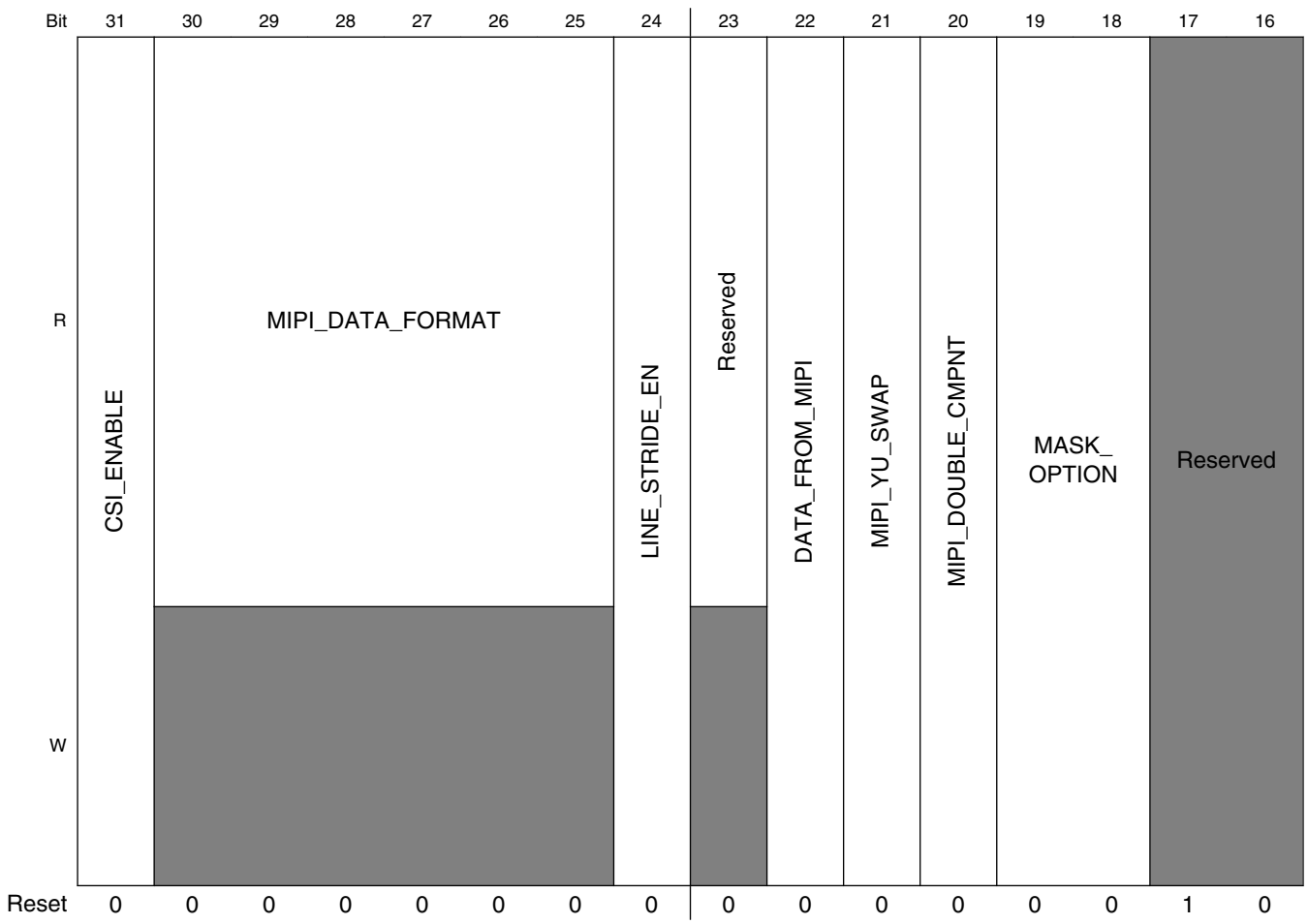
CSI_IMAG_PARA field descriptions

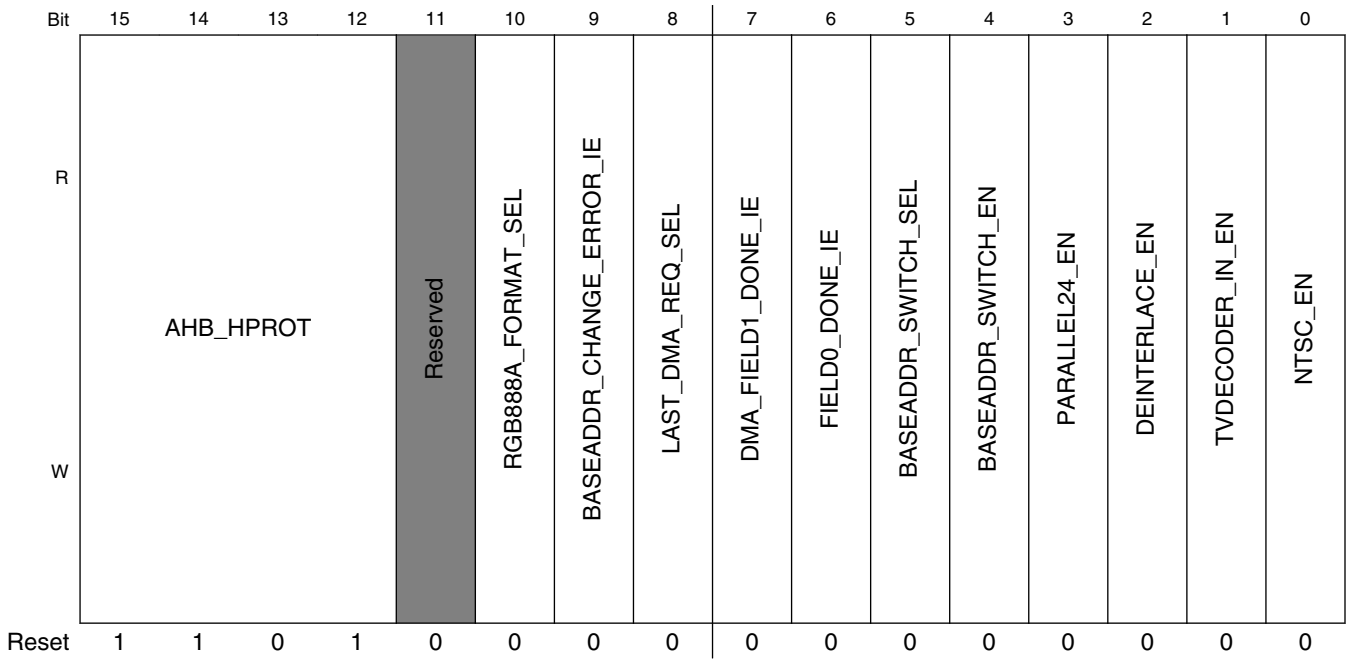
Field	Description
31–16 IMAGE_WIDTH	This field indicates the number of active pixel cycles per line. For 1 cycle/pixel, it equals to IMAGE_WIDTH, and for 2 cycles/pixel it equals to (IMAGE_WIDTH *2). For 24bpp pixel transferred on 8-bit data bus (3 cycle/pixel), it equals to image width. If the input data from the sensor is 8-bit/pixel format, the IMAGE_WIDTH should be a multiple of 8 pixels. If the input data from the sensor is 10-bit/pixel or 16-bit/pixel format, the IMAGE_WIDTH should be a multiple of 4 pixels.
IMAGE_HEIGHT	Image Height. Indicates how many pixels in a column of the image from the sensor. **NOTE:** Image height must be even when deinterlace enabled.

45.6.14 CSI Control Register 18 (CSI_CR18)

This read/write register acts as an extension of the functionality of the CSI Control register 18

Address: 4080_0000h base + 48h offset = 4080_0048h





CSI_CR18 field descriptions

Field	Description
31 CSI_ENABLE	CSI global enable signal. Only when this bit is 1, CSI can start to receive the data and store to memory.
30–25 MIPI_DATA_ FORMAT	Image Data Format Generic Short Packet: 0x08 ~ 0x0F Embedded 8-bit non Image: 0x12 YUV420 (8-bit) 0x18 YUV420 (10-bit) 0x19 YUV420 (8-bit legacy) 0x1A YUV420 (8-bit CSPS) 0x1C YUV420 (10-bit CSPS) 0x1D YUV422 (8-bit) 0x1E YUV422 (10-bit) 0x1F RGB444 0x20 (Not support) RGB555 0x21 (Not support) RGB565 0x22 RGB666 0x23 RGB888 0x24 RAW6 0x28 RAW7 0x29 RAW8 0x2A RAW10 0x2B RAW12 0x2C RAW14 0x2D User defined 1 0x30 User defined 2 0x31

Table continues on the next page...

CSI_CR18 field descriptions (continued)

Field	Description
	User defined 3 0x32 User defined 4 0x33 User defined 5 0x34 User defined 6 0x35 User defined 7 0x36 User defined 8 0x37
24 LINE_STRIDE_EN	When the line width are not the multiple of the burst length, assert this bit.
23 Reserved	This field is reserved.
22 DATA_FROM_MIPI	0 Data from parallel sensor 1 Data from MIPI
21 MIPI_YU_SWAP	It only works in MIPI CSI YUV422 double component mode.
20 MIPI_DOUBLE_CMPNT	Double component per clock cycle in YUV422 formats. 0 Single component per clock cycle (half pixel per clock cycle) 1 Double component per clock cycle (a pixel per clock cycle)
19–18 MASK_OPTION	These bits used to choose the method to mask the CSI input. 00 Writing to memory (OCRAM or external DDR) from first completely frame, when using this option, the CSI_ENABLE should be 1. 01 Writing to memory when CSI_ENABLE is 1. 02 Writing to memory from second completely frame, when using this option, the CSI_ENABLE should be 1. 03 Writing to memory when data comes in, not matter the CSI_ENABLE is 1 or 0.
17–16 Reserved	This field is reserved.
15–12 AHB_HPROT	Hprot value in AHB bus protocol.
11 Reserved	This field is reserved.
10 RGB888A_FORMAT_SEL	Output is 32-bit format. 0 {8'h0, data[23:0]} 1 {data[23:0], 8'h0}
9 BASEADDR_CHANGE_ERROR_IE	Base address change error interrupt enable signal. 0 Interrupt disabled 1 Interrupt enabled
8 LAST_DMA_REQ_SEL	Choosing the last DMA request condition. If fifo_full_level > hburst_length, please select hburst_length. Otherwise, select fifo_full_length.

Table continues on the next page...

CSI_CR18 field descriptions (continued)

Field	Description
	0 fifo_full_level 1 hburst_length
7 DMA_FIELD1_DONE_IE	When in interlace mode, field 1 done interrupt enable. 0 Interrupt disabled 1 Interrupt enabled
6 FIELD0_DONE_IE	In interlace mode, field 0 means interrupt enabled. 0 Interrupt disabled 1 Interrupt enabled
5 BASEADDR_SWITCH_SEL	CSI 2 base addresses switching method. When using this bit, BASEADDR_SWITCH_EN is 1. 0 Switching base address at the edge of the vsync 1 Switching base address at the edge of the first data of each frame
4 BASEADDR_SWITCH_EN	When this bit is enabled, CSI DMA will switch the base address according to BASEADDR_SWITCH_SEL rather than automatically by DMA completed.
3 PARALLEL24_EN	Enable bit for Parallel RGB888/YUV444 24bit input 0 Input is disabled 1 Input is enabled
2 DEINTERLACE_EN	This bit is used to select the output method When input is TVDECODER or standard BT.656 video. 0 Deinterlace disabled 1 Deinterlace enabled
1 TVDECODER_IN_EN	When input is from TV decoder, this bit is enabled.
0 NTSC_EN	This bit is used to select NTSC/PAL mode When input is TVDECODER or standard BT.656 video. 0 PAL 1 NTSC

45.6.15 CSI Control Register 19 (CSI_CR19)

This read/write register acts as an extension of the functionality of the CSI Control register 19

Address: 4080_0000h base + 4Ch offset = 4080_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								DMA_RFIFO_HIGHEST_FIFO_LEVEL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

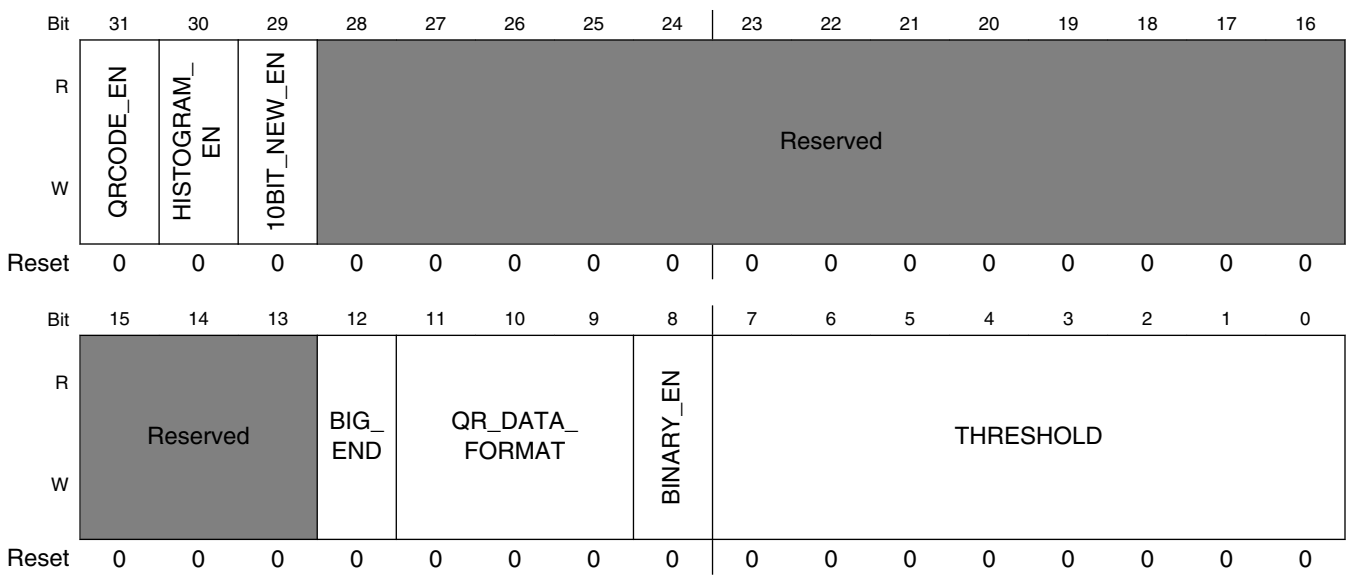
CSI_CR19 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
DMA_RFIFO_ HIGHEST_FIFO_ LEVEL	This byte stores the highest FIFO level achieved by CSI FIFO timely and will be clear by writing 8'ff to it.

45.6.16 CSI Control Register 20 (CSI_CR20)

This read/write register acts as an extension of the functionality of the CSI Control Register 20.

Address: 4080_0000h base + 50h offset = 4080_0050h



CSI_CR20 field descriptions

Field	Description
31 QRCODE_EN	Gray scale mode enable 0 Normal mode 1 Gray scale mode
30 HISTOGRAM_ EN	Histogram enable 0 Histogram disable 1 Histogram enable
29 10BIT_NEW_EN	0 When input 8bits data, it will use the data[9:2] 1 If input is 10bits data, it will use the data[7:0] (optional)

Table continues on the next page...

CSI_CR20 field descriptions (continued)

Field	Description
28–13 Reserved	This field is reserved.
12 BIG_END	0 The newest (most recent) data will be assigned the lowest position when store to memory. 1 The newest (most recent) data will be assigned the highest position when store to memory.
11–9 QR_DATA_ FORMAT	000 YU YV one cycle per 1 pixel input 001 UY VY one cycle per1 pixel input 010 Y U Y V two cycles per 1 pixel input 011 U Y V Y two cycles per 1 pixel input 100 YUV one cycle per 1 pixel input 101 Y U V three cycles per 1 pixel input
8 BINARY_EN	0 Output is Y8 format(8 bits each pixel) 1 Output is Y1 format(1 bit each pixel)
THRESHOLD	THRESHOLD used for binary function. When data value > THRESHOLD, output will be 1 Else will be 0.

45.6.17 CSI Control Register (CSI_CR*n*)

These read/write registers acts as an extension of the functionality of the CSI Control Registers. They store pixel counter values from 0 - 255.

Address: 4080_0000h base + 54h offset + (4d × i), where i=0d to 255d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PIXEL_COUNTERS																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CR*n* field descriptions

Field	Description
31–24 Reserved	This field is reserved.
PIXEL_ COUNTERS	Number of pixels (Y component of the input pixel) equals: 0 (CSICR21) 1 (CSICR22) . . 255 (CSICR276)

Chapter 46
MIPI CSI2 (MIPI_CSI2)

46.1 Chip-specific MIPI CSI-2 information

Table 46-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

46.2 Overview

The Mobile Industry Processor Interface (MIPI) Camera Serial Interface 2 (CSI2) Rx block with the PHY and host controller handles the sensor or image input and process for all the input imaging devices.

46.2.1 Block diagram

The block diagram for the MIPI CSI-2 Rx block is given below:

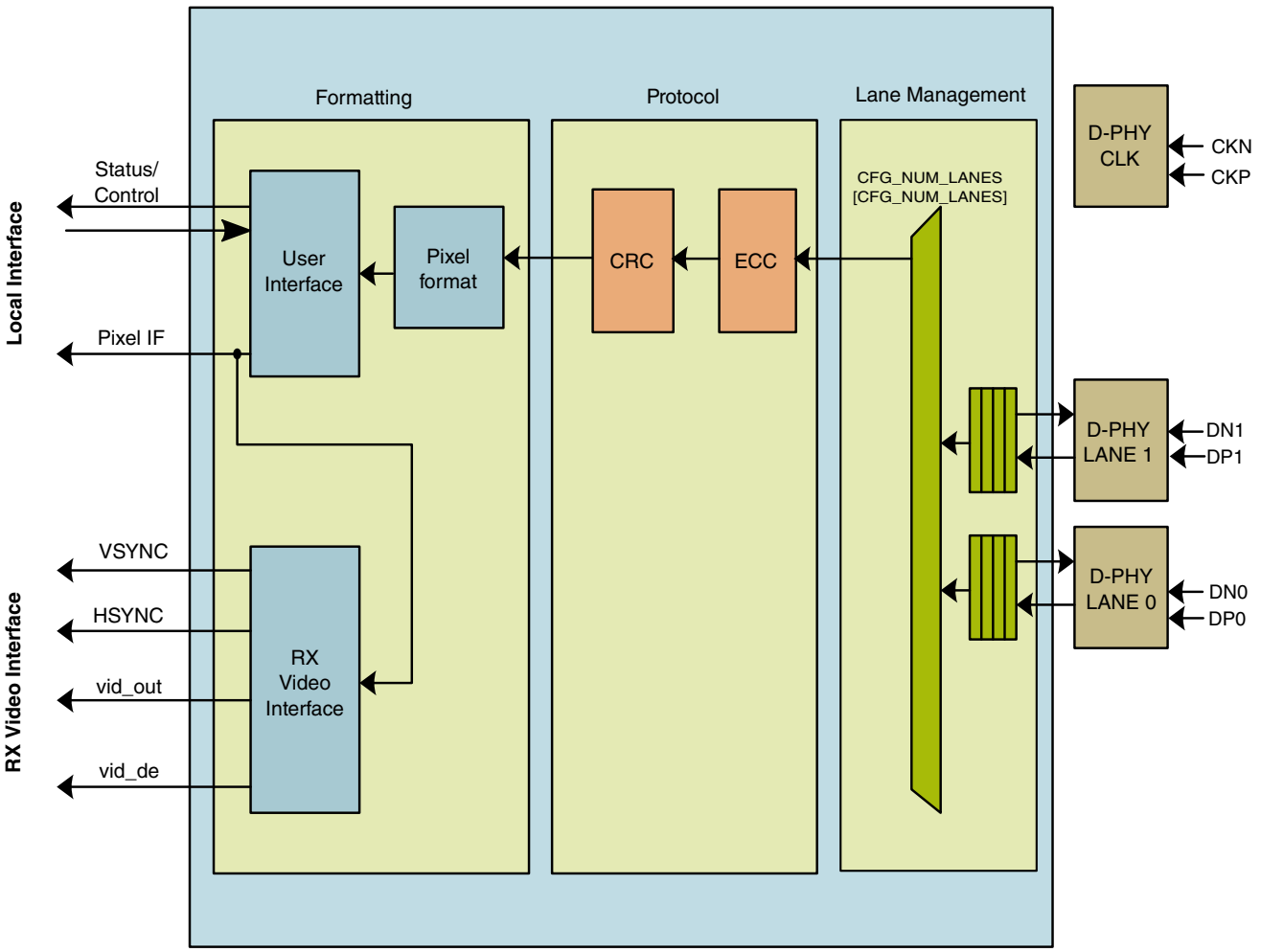


Figure 46-1. MIPI CSI-2 Rx Block Diagram

PHY Layer: It documents the characteristics of the transmission medium, electrical parameters for signaling and for the D-PHY physical layer option, the timing relationship between clock and data lanes. The mechanism for signaling Start Of Transmission (SoT) and End of Transmission (EoT) is specified as well as other out of band information that can be conveyed between transmitting and receiving PHYs. Bit-level and byte-level synchronizations are included as a part of the PHY.

Protocol Layer: It is composed of several layers, each with distinct responsibilities. The CSI2 protocol enables multiple data streams using a single interface on the host processor. The Protocol layer specifies how multiple data streams may be tagged and interleaved so each data stream can be properly reconstructed.

- Byte to Pixel unpacking: The CSI2 specification supports image applications with varying pixel formats from 6bpp to 24bpp. In the receiver, this layer unpacks bytes from the Low level Protocol layer into pixels . Eight bits per pixel data is transferred unchanged by this layer.

- Low Level Protocol (LLP): This includes the means of establishing bit-level and byte-level synchronization for serial data transferred between SoT and EoT events and for passing data to the next layer. The minimum granularity of LLP is one byte. It also includes assignment of bit-value interpretation within the byte, i.e, the Endian assignment.
- Lane Management: The CSI2 controller is lane-scalable for increased performance.On the receiving side, the interface collects bytes from the lanes and merges them together into a recombined data stream that restores the original stream sequence.

Data within the Protocol layer is organized as packets. The header is stripped off at the LLP layer and interpreted by corresponding logic in the receiver.

The CSI-2 Rx Controller Core also includes an Rx Video Interface. This interface watches the local interface of the controller and generates a simplified video output interface. Image packet types which are supported by the Pixel Format block are assembled into a Video output bus which carries the pixel data which is received by the controller. Valid video output data is indicated using the data enable output(s). Horizontal and vertical sync outputs are also provided.

The Local Interface is an easy to use pixel based interface that supports all data types. The Local interface runs at the User Interface clock rate for all implementations. The CSI-2 Rx Controller Core takes care of all packet formatting details and transmission over the MIPI bus.

46.2.2 Features

The MIPI-CSI2 Controller has the following key features:

- Implements the PHY layer, and protocol layer. The protocol layer includes Byte to Pixel unpacking Layer, Low Level protocol, and Lane management
- Supports unidirectional Master operation
- Scalable data lane support, 1 to 2 Data Lanes
- Supports high speed mode (80Mbps - 1.5Gbps) per lane
- Supports 10Mbps data rate in low power mode
- Includes high speed deserializers
- Loopback testability support
- Support for D-PHY Ultra Low Power State (ULPS)
- Error collection support (Rx Only)
- Flexible pixel-based user interface
    - Supports user generated packets
    - Supports single interface

- Supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs
- Support for Rx Video Interface
- Support for APB Control and Status Register (CSR) interface with IRQ

46.3 Functional description

46.3.1 CSI-2 Rx Controller Core Local Interface

The CSI-2 Rx Controller Core Local Interface consists of a Receive Interface.

Packets are received from the MIPI Interface via the Receive Local Interface. The Receive Interface is designed to easily interface to a FIFO on the user side that holds the pixels that are received until the user application can process them.

46.3.1.1 Rx Local Interface Description

The Receive Local Interface is the mechanism with which the user receives packets from the MIPI Interface.

The Receive Local Interface first announces that a packet is being or has been received by asserting the pkt_hdr_valid signal with the other packet information output ports containing details about the packet. Next, the sop_out signal will assert indicating that data is about to arrive at the Receive Local Interface. Dav_out will assert with every cycle of pixel data on the data_out port. The last cycle of receive data is signaled by the assertion of eop_out.

With a single pixel interface, data_out contains a single pixel every clock that dav_out (data valid) is asserted. The single pixel interface must run at a frequency that is equal to number of MIPI lanes times the byte lane frequency.

Table 46-2. Receive Local Interface

Signal	Type	Description
data_out[n-1:0]	Output	Pixel Data output.
dav_out	Output	Data Valid Out. When asserted high, data_out, sop_out, and eop_out outputs are valid.

Table continues on the next page...

Table 46-2. Receive Local Interface (continued)

sop_out	Output	Start of packet signal. This active high signal indicates the first cycle of data on data out. Valid when dav_out asserts high
eop_out[3:0]	Output	Last pixel indicator. Asserts to indicate the last pixel in a transmission. When in single pixel mode, eop_out[0] is used. When in double pixel mode two bits are used, [1:0]. 2'b01 indicates the last pixel is on the lower pixel path. 2'b10 indicates the last pixel is on the upper pixel path. When in quad pixel mode, all four bits are utilized. A value of 4'b0001 indicates the last pixel is on the lowest pixel path portion of data_out. 4'b0010 indicates the last pixel is on the second pixel path of data_out. 4'b0100 indicates the last pixel is on the third pixel path of data_out. 4'b1000 indicates the last pixel is on the fourth and highest pixel path of data_out. Valid when dav_out asserts high.

46.3.1.2 Receive Interface Example Transaction

High Speed receive packets appear immediately on the Receive local interface as they are received.

46.3.1.2.1 Short Packet Receive

The short packets do not contain any data. All the information in a short packet is contained in the 32 bit header. Pkt_hdr_valid asserts indicating that the packet header information contained on pkt_hdr_data_type, pkt_hdr_wc, and pkt_hdr_vc is valid. Short packets are defined as having a pkt_hdr_data_type value that is less than 0x10. With a short packet the CSI-2 Rx Controller Core does not assert sop_out, eop_out, nor dav_out.

The figure below shows the reception of a CSI-2 Short Packet type.

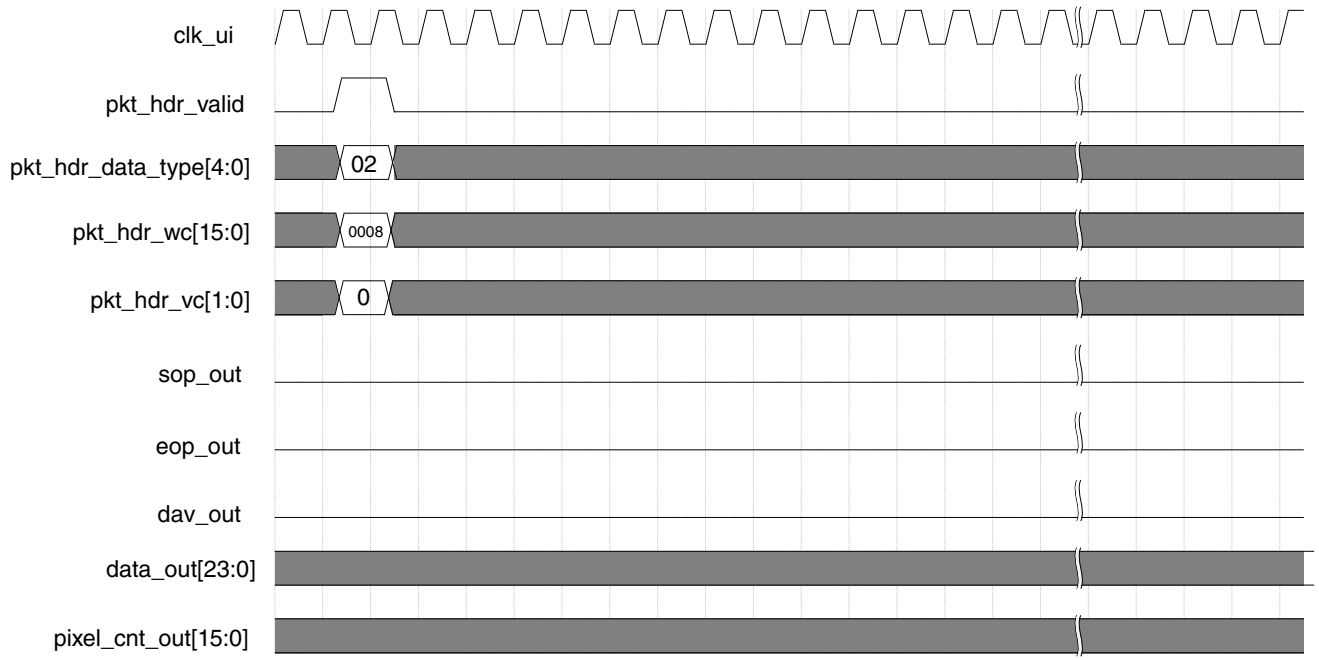


Figure 46-2. Short Packet receive with a data type of Line Start, Line Number of 8 , and Virtual Channel = 0 single pixel configuration

In this example, the data type ix 0x30, a user defined byte data type. Initially, pkt_hdr_valid asserts along with the packet header information. The packet header ports, pkt_hdr_data_type, pkt_hdr_wc, and pkt_hdr_vc indicate that the packet is a user defined data type with a payload of 4 bytes. For unsupported types, the Rx Controller outputs the entire packet minus the two byte CRC value at the end of the packet. The first four bytes are the packet header followed immediately by the 4 bytes of payload data.

The figure below shows a waveform diagram of a received packet on the local interface for an unsupported data type.

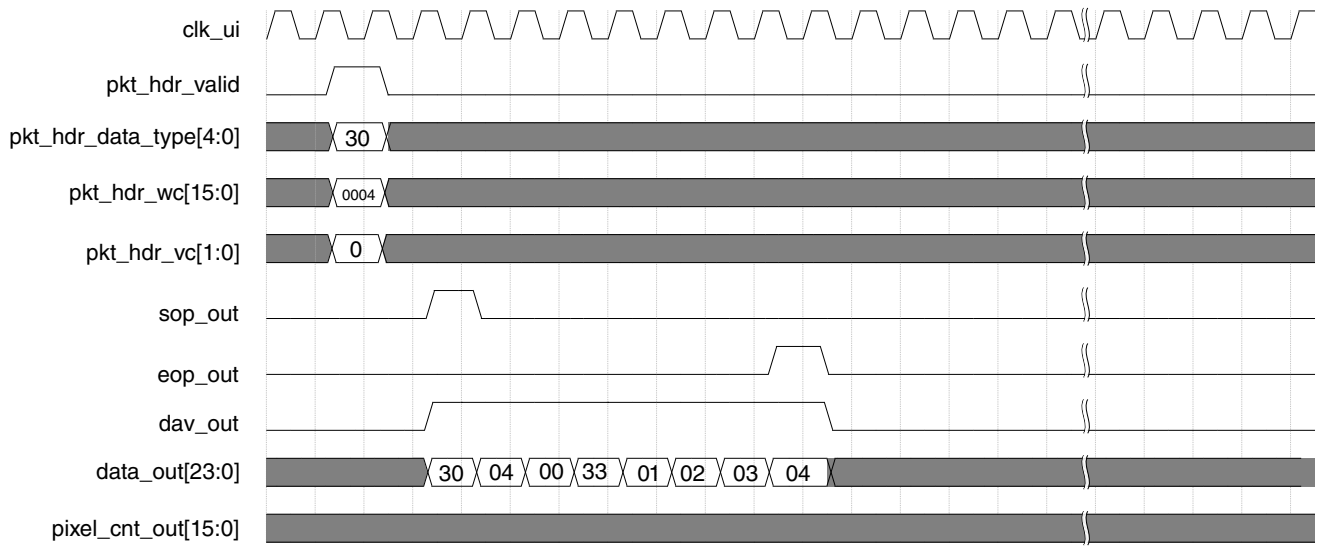


Figure 46-3. Received packet of an unsupported data type, 0x30, user defined data, single pixel configuration

46.3.2 CSI-2 Controller Core Configurations

The CSI-2 Rx Controller Core User Interface supports either a single, double, or quad pixel wide data path. The double and quad wide pixel modes can help reduce the frequency the user interface must run at while the single pixel wide mode can be easier to interface to for some applications.

This chip supports the following:

- Single Pixel Configuration

46.3.2.1 Single Pixel Configuration

The CSI-2 Rx Controller Core User Interface supports a configuration where a single pixel of received data is processed every clock period.

The CSI-2 Rx Controller Core contains minimal pixel buffering (16 pixels or less) so the frequency of that the user interface runs at must be at least high enough to receive the data from the controller at a rate that is equal to or greater than the highest transmission data rate on the MIPI interface for all lanes combined.

46.3.3 CSI-2 Rx Controller Core Data Types Formatting

The CSI-2 Rx Controller Core supports the various CSI-2 Data Types via the data_out port. The following table below describes how the controller presents the various received data types on the data_out output port.

The CSI2 Controller supports the Single Pixel Configuration.

Format	Data Type	Description (single Pixel Configuration)
RGB444	0x20	Blue data is on data_out[3:0], Green on data_out[7:4], and Red on data_out[11:8]
RGB555	0x21	Blue data is on data_out[4:0], Green on data_out[9:5], and Red on data_out[14:10]
RGB565	0x22	Blue data is on data_out[4:0], Green on data_out[10:5], and Red on data_out[15:11
RGB666	0x23	Blue data is on data_out[5:0], Green on data_out[11:6], and Red on data_out[17:12]
RGB888	0x24	Blue data is on data_out[7:0], Green on data_out[15:8], and Red on data_out[23:16]
Legacy YUV420 (8 bit)	0x1A	With Legacy YUV420 8 bit and the Rx Controller configured for single pixel mode the video data is received in UYY…/VYY sequences in odd/even lines. The U component is received with YY in a UYY group on odd lines while the V component is received with the same YY pairs in VYY groups on the even lines. When receiving an odd line data_out[7:0] contains the U component, data_out[15:8] Y1 and data_out[23:16] Y2. On even lines data_out[7:0] contains the V component, data_out[15:8] Y1 and data_out[23:16] Y2.
YUV420 (8bit)	0x18, 0x1C	With YUV420 8 bit and YUV420 8 bit chroma shifted is received as YYYY (odd lines) UYVY (even lines) sequence. With the Rx controller configured for single pixel mode and receiving even numbered lines, data_out[7:0] contains the U1 component, bits [15:8] Y1. On the next data valid cycle bits [7:0] V1, bits [15:8] Y2. When receiving odd lines, data_out[7:0] contains the Y1 component, bits [15:8] Y2. On the next data valid cycle bits [7:0] Y3, bits [15:8] Y4.
YUV420 (10 bit)	0x19, 0x1D	With YUV420 10 bit and YUV420 10 bit chroma shifted is received as YYYY (odd numbered lines) and UYVY (even numbered lines) sequence with each component represented by 10 bits.

Table continues on the next page...

Format	Data Type	Description (single Pixel Configuration)
		With the Rx controller configured for single pixel mode and receiving even numbered lines, data_out[9:0] contains the U1 component, bits [19:10] Y1. On the next data valid cycle bits [9:0] V1, bits [19:10] Y2. When receiving odd lines, data_out[9:0] contains the Y1 component, bits [19:10] Y2. On the next data valid cycle bits [9:0] Y3, bits [19:10] Y4.
YUV422 (8 bit)	0x1E	The data_out[7:0] contains the U1 component, bits [15:8] contains Y1 component. On the next data valid cycle, bits [7:0] contains V1 component, and bits [15:8] contains Y2 component.

46.3.4 Rx Video Interface

The CSI-2 Rx Video Interface is connected internally to the local interface outputs of the CSI-2 Rx Controller Core. This interface provides both the packet header information and the pixel data (after processing by the Pixel Format block). CSI-2 Rx Controller Cores may be built with any number of Pixel Formats supported. The CSI-2 Rx Video Interface only recognizes video formats for which the CSI-2 Rx Controller Core has been configured to support.

The general timing of the video interface is determined by the timing of the packets received by the CSI-2 Rx controller which are used to determine when to generate a Vertical Sync and when to output a line of video data.

The CSI-2 Rx Video Interface may receive several different types of packet data types as shown in the table below.

Packet Type	Data ID	Function
Frame Start Code (FS)	0x00	Indicates start of frame
Frame End Code (FE)	0x01	Indicates end of frame
Line Start Code (LS)	0x02	Indicates start of line
Line End Code (LS)	0x02	Indicates end of line
Image Data Codes (IMG)	various	Image data packets carry the sample data. The CSI-2 Rx Video Interface will respond to those

		Data types which are enabled in the CSI-2 Rx Controller Core. See CSI-2 Rx Controller Core Data Types Formatting

46.3.4.1 Video Interface Output Raster Timing

The output timing of the raster is formed by the packets sent to the CSI-2 Rx Controller Core and is determined by the relative timing of these packets. This timing is controlled by the transmitter and is not determined by the CSI-2 Rx Controller Core. As shown in the figure below, the basic sequence of packets sent by the transmitter (often a sensor or camera) consists of a Frame Start packet (short packet), followed by some number of Image Data packets (long packets) with one Image data packet per video line and the followed by a Frame End packet (short packet). The Frame End packet is followed by pause which determines the Blanking Lines interval. The length of the Image Data packets is determined by the horizontal resolution of the sensor and each line's samples are carried by a single Image Data packet.

NOTE

Note that each Image Data packet may optionally be preceded by a Line Start packet and followed by an optional Line End packet. The Line Start and Line End packets are ignored by the CSI-2 Rx Video Interface.

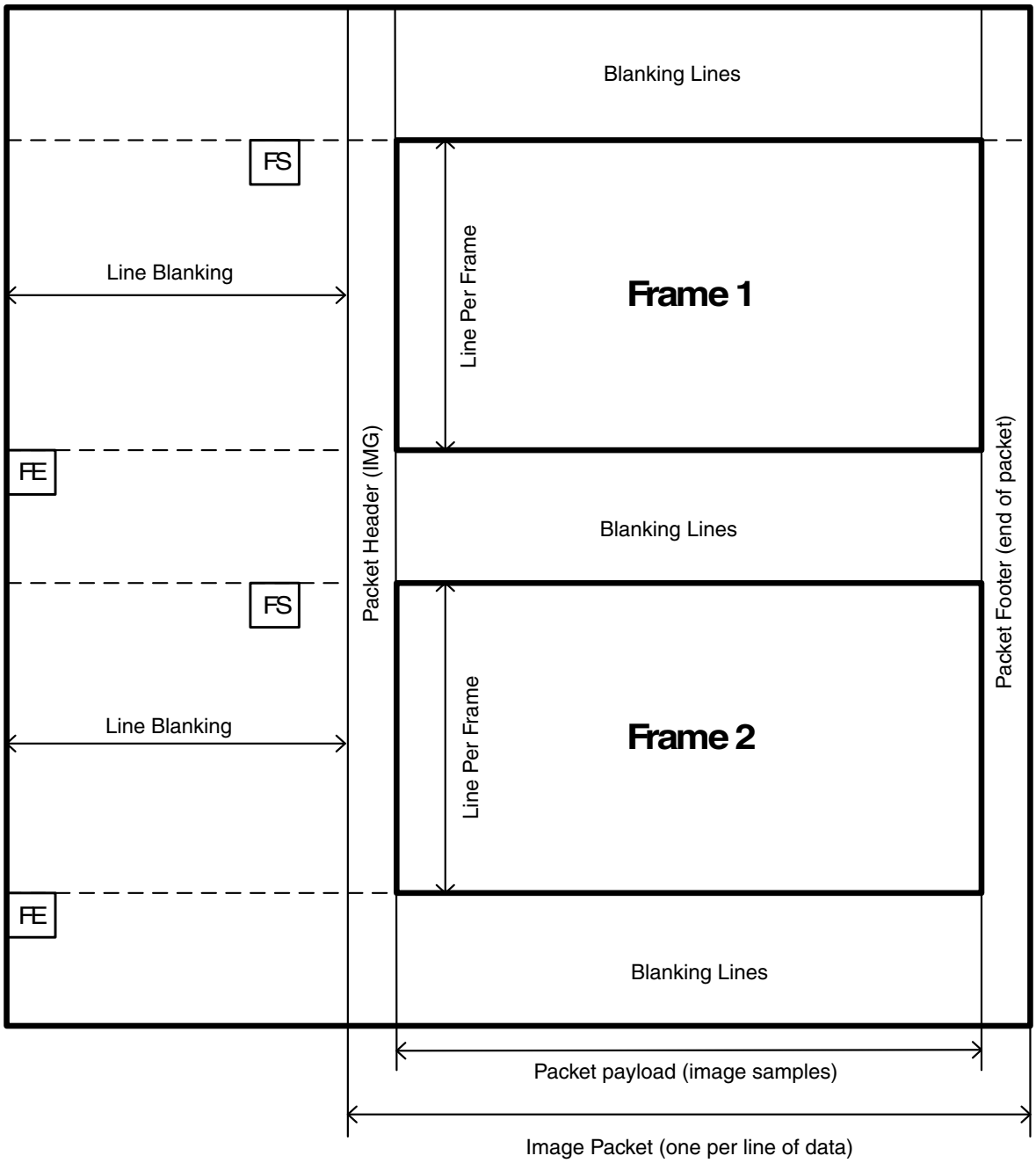


Figure 46-4. Rx Video Interface Raster Timing

NOTE

- FS- Frame Start Packet
- FE- Frame End Packet
- IMG- Image Data Packet

Note that the interface does not generate Horizontal sync pulses during the Blanking Lines.

The CSI-2 Rx Video controller generates a Vertical sync pulse upon the occurrence of every Frame Start packet. Each line of video sample data is preceded by a Horizontal sync pulse.

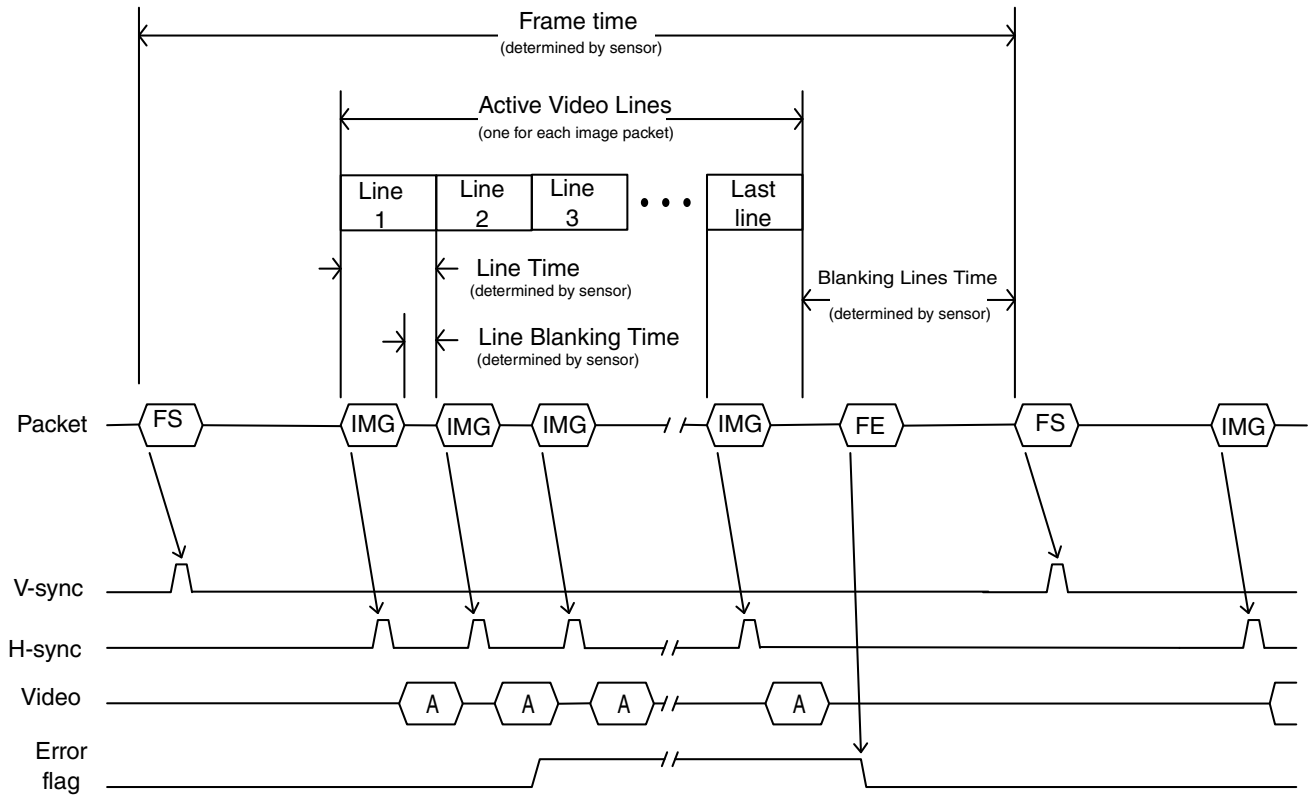


Figure 46-5. Relationship of Packet Stream to Video Interface

NOTE

- FS - Frame start Packet (causes V-sync)
- IMG - Image Data Packet(causes H-sync and Video after FIFO reaches send level)
- FE - Frame End Packet (causes any pending Error Flags which were set in the previous frame)
- A - Time period when video samples are valid (follows H-sync)

The video sample output bus (vid_out) is logically divided up into pixel lanes. The number of pixel lanes is determined by the pixel width of the CSI-2 Controller Core. This width may be one, two or four pixels wide (single, double or quad pixel interface). The bit width of each pixel lane is determined the widest pixel width supported by the core controller's Pixel Format block. When transferring image data whose pixels are narrower than the maximum required width unused more significant bits in each pixel lane will be output as zero.

To determine the validity of the data presented on each pixel lane the CSI-2 Rx Video Interface supplies a data enable output (vid_de) whose width is equal to the number of pixel lanes in the interface. When a data enable bit is high, its pixel lane is valid.

The only time at which all vid_de bits will not be asserted is on the final transfer of video pixels in a scan line. At this time only some of the lower order bits in the vid_de bus will be asserted only if the number of samples per line is not evenly divisible by the number of pixel lanes.

Thus as a design example:

1. Assume a double pixel interface which results in a two video data lanes. Image data packets are 1023 samples long.
2. This would result in 511 transfers of video out with both bits of vid_de set (1022 samples). The last transfer would only have the least significant vid_de bits set to transfer the final sample to achieve the 1023 samples in the scan line.

46.3.4.2 Video Output Port Waveforms

The vid_de port is used to enable capture of valid video pixels from the vid_out bus. The vid_de will have a single bit for each pixel lane in the vid_out bus. Thus for a single pixel interface (one pixel lane) this bus will be a single signal.

The vid_de port will assert all its signals high for all transfers except possibly the last transfer. This last transfer may only assert some number of lower order bits to one and the remaining higher order bits to zero. The number of valid pixel lanes in the last transfer is determined by the number of pixels transferred by the image packet which generated the video line.

The ending value may be predicted by taking number of pixels transferred in an image packet modulo number of pixel lanes. Any non-zero remainder indicates some number of vid_de bits will be zero on the last transfer.

The following table shows the last vid_de value as a function of number of pixel lanes and packet size.

Table 46-3. Last vid_de value depending on Number of Pixel Lanes and Packet Size

MODULO	Last vid_de value (binary) for Single Pixel Interface
0	1
1	NA
2	NA
3	NA

NOTE

MODULO = (Number of pixels in image packet) MOD (number of pixel lanes)

46.3.5 Rx Control and Status Register (CSR) APB Interface

The CSR provides an APB compatible interface that enables control of the controller's configuration inputs via registers accessible via the APB interface.

46.3.6 Clocks

Table 46-4. Clocks

Clock	Description
clk_ui	User interface clock The frequency of clk_ui must be such that the data received on the data_out output is greater than or equal to the total bandwidth of the physical MIPI interface. Clk_ui has no relationship requirement with regards to 'clk' other than the bandwidth requirement mentioned previously.
clk	Input clock to the Rx controller to process data received from D-PHY This clock must be exactly equal to or faster than the receive byte clock, RxByteClkHS_ln0, from the Rx D-PHY.
clk_esc	RX Escape Clock This must be the same escape clock that the Rx D-PHY receives.
pclk	Pixel Clock

The Rx controller core requires two clocks for proper operation clk_ui and clk. The clk_ui clock is the clock that the user interface is synchronous to, with all inputs and outputs referenced to the rising edge of clk_ui. The clk input is the clock that the CSI-2 Rx Controller Core uses to process data received from the D-PHY. The clock clk_ui must be of a sufficient frequency to keep up with the incoming data from the CSI-2 MIPI interface.

The clk_esc is Rx Escape Clock. This must be the same escape clock which the RX D-PHY receives.

The data from the D-PHY is clocked into shallow FIFOs (one per D-PHY lane) using a receive clock that the D-PHY generates. There is no direct timing requirements between the D-PHY generated receive clock and the CSI-2 Rx Controller Core's clk or clk_ui input. The frequency of the clock used for the Rx controller's clk input must be of a sufficient frequency that the Rx controller can keep up with the data from D-PHY. The

Rx Controller processes the Rx D-PHY data in a 32 bit wide data sufficient frequency that the Rx Controller can keep up with the receive data from the D-PHY. The Rx Controller processes the Rx D-PHY data in a 32 bit wide data path regardless of whether the CSI-2 MIPI receive data is coming across one, two lanes. The frequency of clk must be exactly equal to or greater than the Rx byte clock coming from the Rx D-PHY.

The CSI-2 Rx Controller Core can supply pixel data to the User application either one, two or four pixels per clock.

When in single Pixel Mode, the minimum frequency for clk_ui must be high enough to keep up with the incoming data rate of the CSI-2 MIPI interface. It is important to realize that the user interface is pixel based and therefore may be wider than the incoming byte data from the MIPI Rx D-PHY. As an example, if RGB888 data is being received, the user interface data width will be 24 bits wide (1 pixel per clk_ui rising edge and each pixel is 24 bits). As an example, let the MIPI interface run at 1000Mbps with a single lane. The data rate into the CSI-2 Rx controller will be 125MBps so clk must run at 125MHz or higher. Since the user interface is 3 bytes wide, clk_ui must run at 125MHz/3 or higher to keep up with the incoming data.

The figure below shows one possible clocking methodology. In this example, the user Clk_Gen module generates the edge aligned clk_ui and clk clocks.

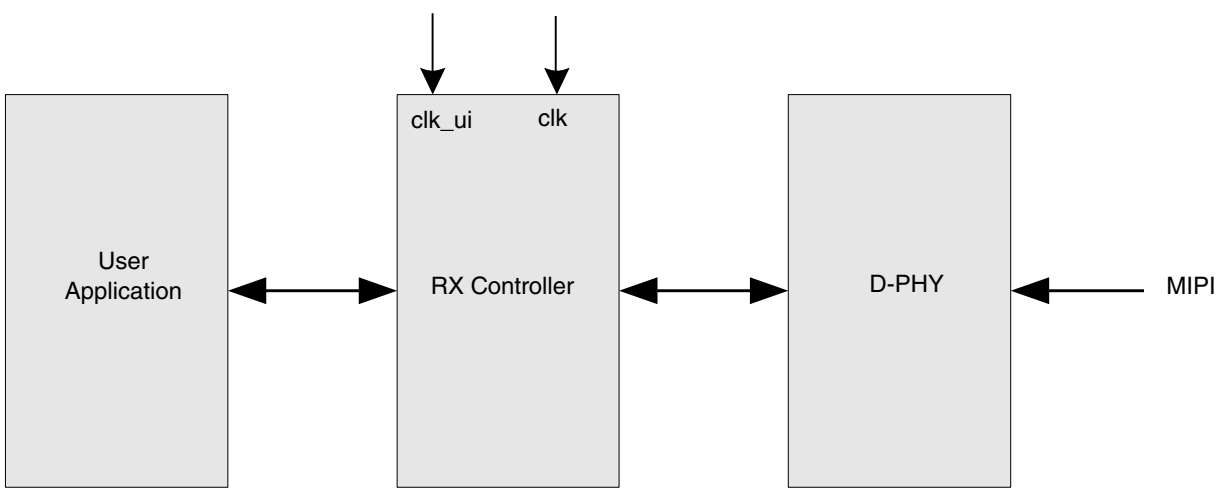


Figure 46-6. CSI-2 Rx Controller Core clocking example

46.3.7 Reset

The MIPI CSI2 contains the following reset signals:

Reset Signal	Description
reset_n	Async reset (active-low) This reset applies to all logic in the Controller Core that is in the ‘clk’ clock domain.
clk_vid_reset_n	Asynchronous active low reset for all logic clocked by clk_vid
clk_ui_reset_n	Async reset (active-low) This reset applies to all logic in the Controller Core that is in the ‘clk_ui’ clock domain.
clk_esc_reset_n	Async reset (active-low) This reset applies to all logic in the Controller Core that is in the ‘clk_esc’ clock domain.
pclk_reset_n	Async reset (active-low) This reset input for all logic in the pclk clock domain.

46.4 External Signals

The table below shows the external signals associated with this block:

Table 46-5. External Signals

Signal	Description	Direction
CKN	Negative MIPI Rx Differential Clock	I
CKP	Positive MIPI Rx Differential Clock	I
DNn	Negative MIPI Rx Differential Data	I
DPn	Positive MIPI Rx Differential Data	I
VSYNC	Video Vsync pulse	O
HSYNC	Video Hsync pulse	O
vid_out	Video Data bus	O
vid_de	Video data pixel lane valid	O

46.5 Initialization

The Initialization procedure for the CSI-2 Rx Controller is as follows:

1. Assert all resets.
2. Wait until clocks to the Rx Controller are stable and ensure D-PHY interface is idle (Stop State on all lanes)
3. De-assert reset to CSR, pclk_reset_n, if CSR is included with the controller. If no CSR go to step 4.

4. Program CSR registers if CSR is present. If CSR is not present, set values to all configuration ports (cfg_* inputs) to appropriate values.
5. De-assert all remaining resets.
6. CSI-2 Rx Controller is ready for use by the next rising edge of clk_ui.

46.6 Memory Map and register definition

This section includes the memory map and detailed descriptions of all registers.

46.6.1 register descriptions

46.6.1.1 CSI2RX memory map

MIPI_CSI2RX base address: 4081_0000h

Offset	Register	Width (In bits)	Access	Reset value
100h	Lane Configuration Register (CFG_NUM_LANES)	32	Table 46-5	0000_0000h
104h	Disable Data Lane Register (CFG_DISABLE_DATA_LANES)	32	Table 46-5	0000_000Fh
108h	ECC and CRC Error Status Register (BIT_ERR)	32	Table 46-5	0000_0000h
10Ch	IRQ Status Register (IRQ_STATUS)	32	Table 46-5	0000_0000h
110h	IRQ Mask Setting Register (IRQ_MASK)	32	Table 46-5	0000_0000h
114h	Ultra Low Power State (ULPS) Status Register (ULPS_STATUS)	32	Table 46-5	0000_0000h
118h	ERRSotHS Status Register (PPI_ERRSOT_HS)	32	Table 46-5	0000_0000h
11Ch	ErrSotSync HS Status Register (PPI_ERRSOTSYNC_HS)	32	Table 46-5	0000_0000h
120h	ErrEsc Status Register (PPI_ERRESC)	32	Table 46-5	0000_0000h
124h	ErrSyncEsc Status Register (PPI_ERRSYNCESC)	32	Table 46-5	0000_0000h
128h	ErrControl Status Register (PPI_ERRCONTROL)	32	Table 46-5	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
12Ch	Disable Payload 0 Register (CFG_DISABLE_PAYLOAD_0)	32	Table 46-5	0000_0000h
130h	Disable Payload 1 Register (CFG_DISABLE_PAYLOAD_1)	32	Table 46-5	0000_0000h

46.6.1.2 Lane Configuration Register (CFG_NUM_LANES)

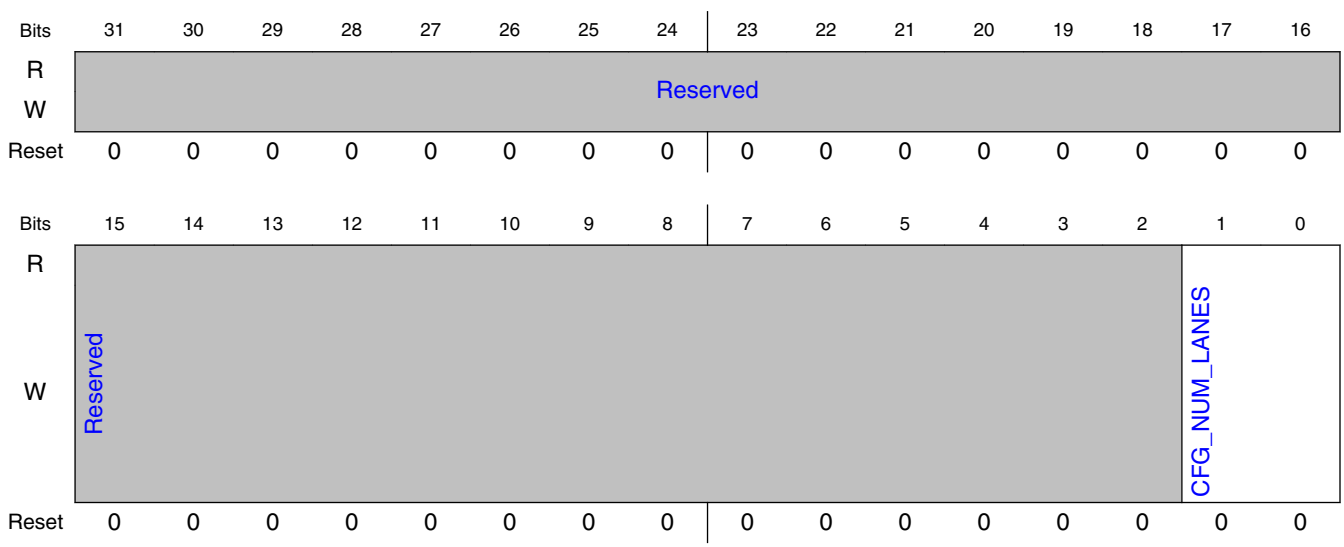
46.6.1.2.1 Offset

Register	Offset
CFG_NUM_LANES	100h

46.6.1.2.2 Function

Configures the number of active lanes for receiving data.

46.6.1.2.3 Diagram



46.6.1.2.4 Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Field	Function
—	
1-0 CFG_NUM_LANES	This field is used to set the number of active lanes for receiving data. 00b - 1 Lane 01b - 2 Lane 10b-11b - Reserved

46.6.1.3 Disable Data Lane Register (CFG_DISABLE_DATA_LANES)

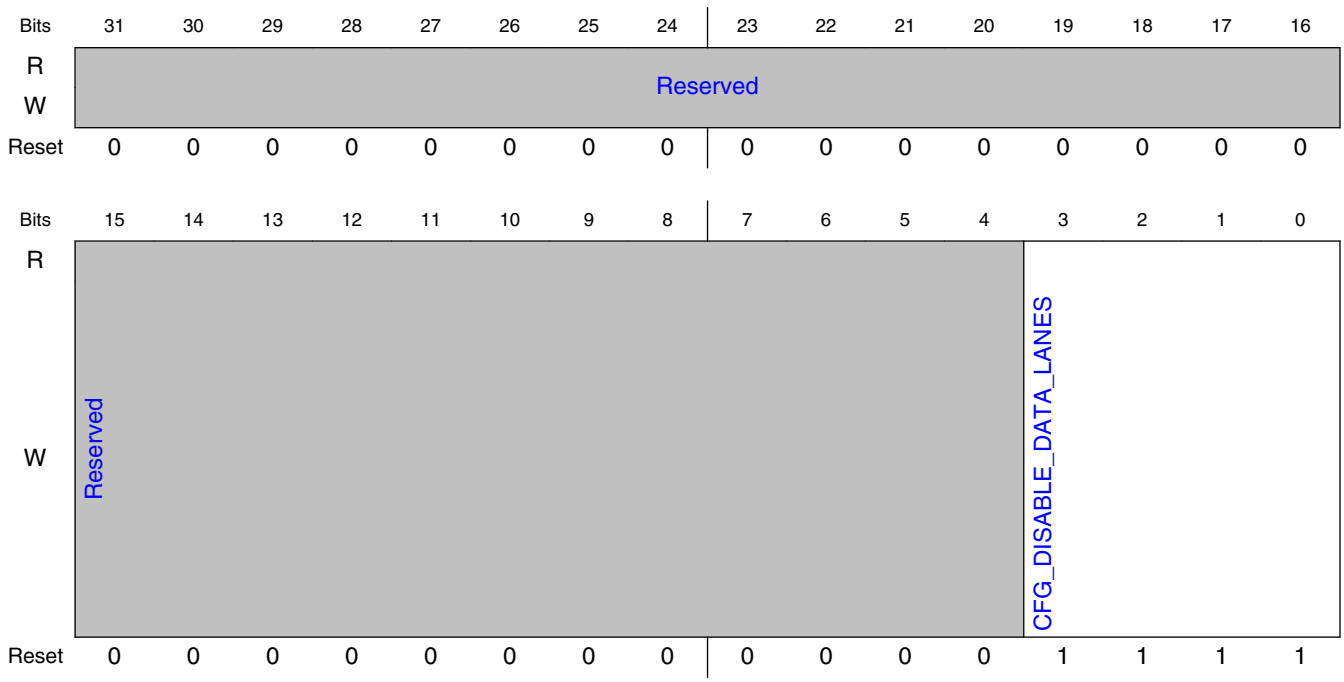
46.6.1.3.1 Offset

Register	Offset
CFG_DISABLE_DATA_LANES	104h

46.6.1.3.2 Function

Disables data lanes.

46.6.1.3.3 Diagram



46.6.1.3.4 Fields

Field	Function
31-4 —	Reserved
3-0 CFG_DISABLE_DATA_LANES	Used to disable data lanes. Setting bits to '1' will cause the D-PHY Enable signal to de-assert. [0] - Data Lane 0 [1] - Data Lane 1

46.6.1.4 ECC and CRC Error Status Register (BIT_ERR)

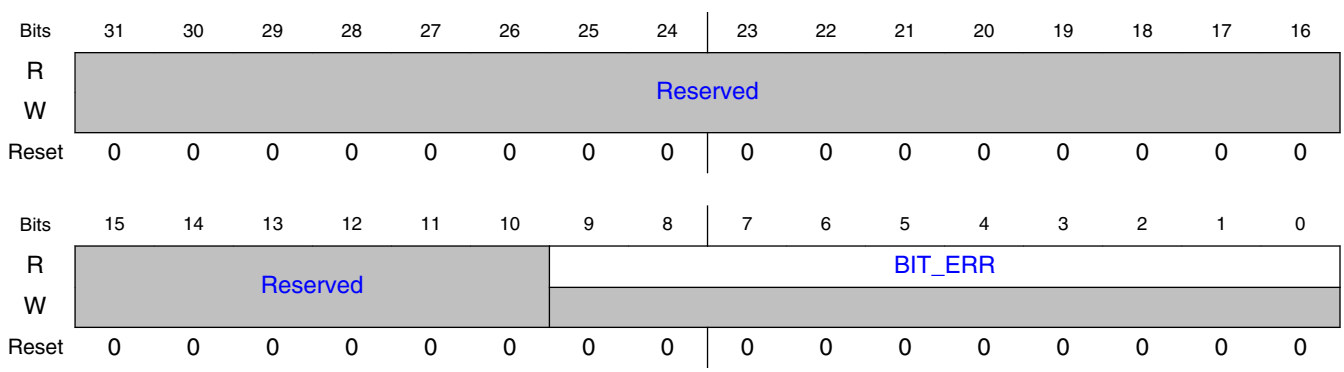
46.6.1.4.1 Offset

Register	Offset
BIT_ERR	108h

46.6.1.4.2 Function

Indicates error status of ECC and CRC.

46.6.1.4.3 Diagram



46.6.1.4.4 Fields

Field	Function
31-10	Reserved

Table continues on the next page...

Field	Function
—	
9-0 BIT_ERR	This field shows the error status of ECC and CRC Bit[0] – value of '1' indicates ECC two bit error has occurred since the last read of this register Bit[1] – value of '1' indicates ECC one bit error has occurred since the last read of this register Bit[6:2] – ECC one bit error bit position that occurred since the last read of this register Bit[7] – value of '1' indicates a CRC error has occurred since the last read of this register Bit[8] – value of '1' indicates that Rx video interface signal vid_err_send_level has asserted since the last read of this register. Register reads 0 when Rx video interface included. Bit[9] – value of '1' indicates that Rx video interface signal vid_err_fifo_wr_ovfl has asserted since the last read of this register. Register reads 0 when Rx video interface included.

46.6.1.5 IRQ Status Register (IRQ_STATUS)

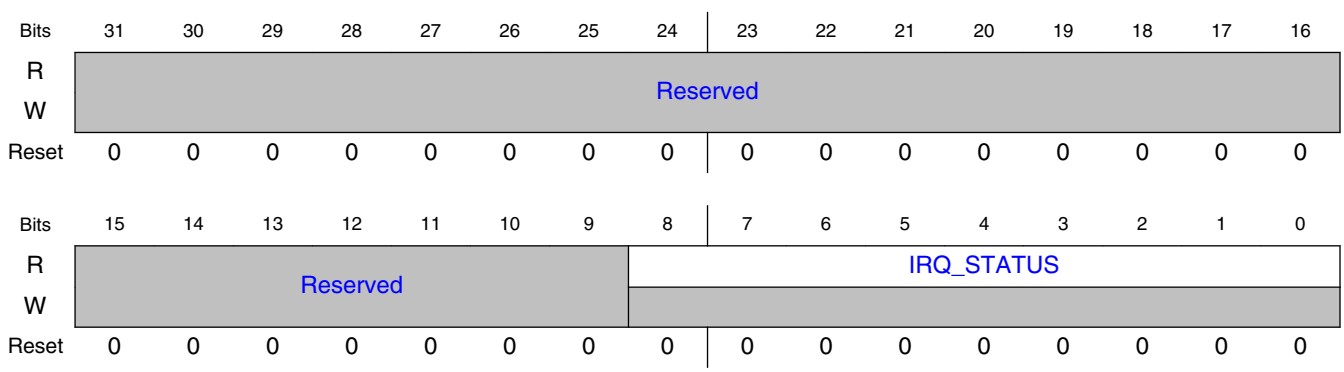
46.6.1.5.1 Offset

Register	Offset
IRQ_STATUS	10Ch

46.6.1.5.2 Function

Indicates interrupt status.

46.6.1.5.3 Diagram



46.6.1.5.4 Fields

Field	Function
31-9 —	Reserved
8-0 IRQ_STATUS	This field shows the IRQ status [0] – crc error Bit[1] – one bit ecc error Bit[2] – two bit ecc error Bit[3] – ULPS status change Bit[4] – D-PHY ErrSotHS has occurred Bit[5] – D-PHY ErrSotSync_HS has occurred Bit[6] – D-PHY ErrEsc has occurred Bit[7] – D-PHY ErrSyncEsc has occurred Bit[8] – D-PHY ErrControl has occurred

46.6.1.6 IRQ Mask Setting Register (IRQ_MASK)

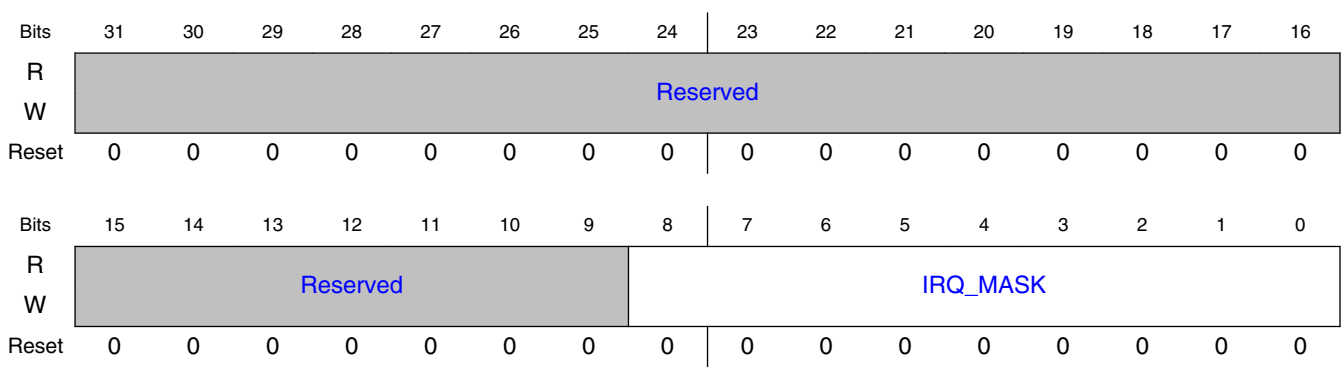
46.6.1.6.1 Offset

Register	Offset
IRQ_MASK	110h

46.6.1.6.2 Function

Indicates interrupt mask setting.

46.6.1.6.3 Diagram



46.6.1.6.4 Fields

Field	Function
31-9 —	Reserved
8-0 IRQ_MASK	This field shows the IRQ Mask setting Each bit in IRQ_MASK corresponds to a bit in IRQ_STATUS. Setting a bit in IRQ_MASK to '1' will mask the corresponding bit in IRQ_STATUS from causing irq_out to assert.

46.6.1.7 Ultra Low Power State (ULPS) Status Register (ULPS_STATUS)

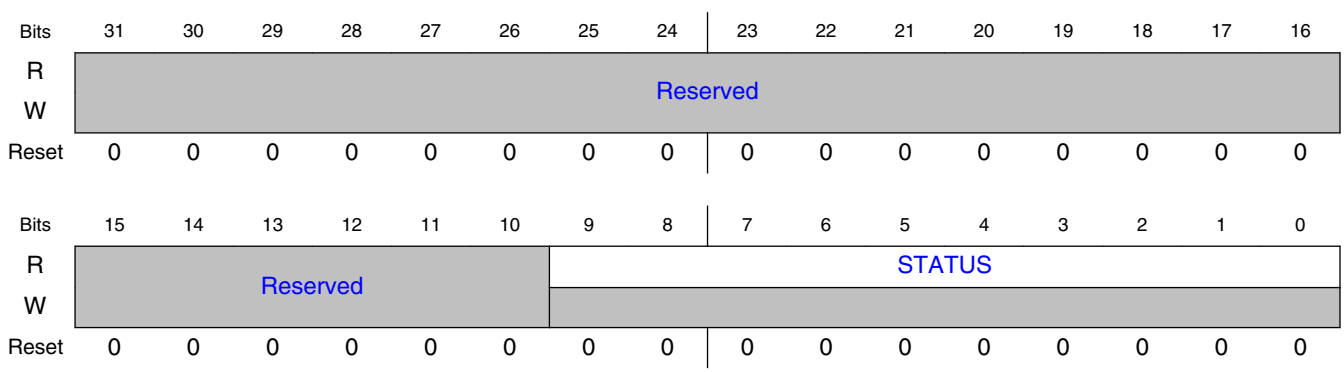
46.6.1.7.1 Offset

Register	Offset
ULPS_STATUS	114h

46.6.1.7.2 Function

Indicates status of Rx D-PHY ULPS state.

46.6.1.7.3 Diagram



46.6.1.7.4 Fields

Field	Function
31-10	Reserved

Table continues on the next page...

Field	Function
—	
9-0 STATUS	This field shows the status of Rx D-PHY ULPS state Bit[0] – Value '1' indicates Clock Lane in ULPS state Bit[1] – Value '1' indicates Data Lane 0 in ULPS state Bit[2] – Value '1' indicates Data Lane 1 in ULPS state Bit[5] – Value '1' indicates Clock Lane in Mark state Bit[6] – Value '1' indicates Data Lane 0 in Mark state Bit[7] – Value '1' indicates Data Lane 1 in Mark state Values not mentioned are 'Reserved'

46.6.1.8 ERRSotHS Status Register (PPI_ERRSOT_HS)

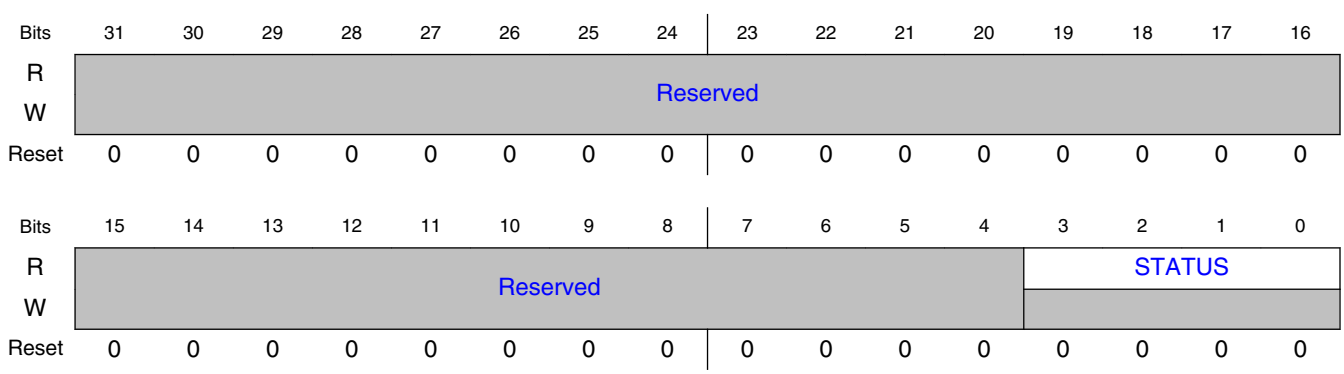
46.6.1.8.1 Offset

Register	Offset
PPI_ERRSOT_HS	118h

46.6.1.8.2 Function

Indicates PPI ErrSotHS captured status from D-PHY.

46.6.1.8.3 Diagram



46.6.1.8.4 Fields

Field	Function
31-4 —	Reserved
3-0 STATUS	This field indicates PPI ErrSotHS captured status from D-PHY Bit[0] – Data Lane0 Bit[1] – Data Lane1 Values not mentioned are 'Reserved'

46.6.1.9 ErrSotSync HS Status Register (PPI_ERRSOTSYNC_HS)

46.6.1.9.1 Offset

Register	Offset
PPI_ERRSOTSYNC_HS	11Ch

46.6.1.9.2 Function

Indicates PPI ErrSotSync_HS captured status from D-PHY.

46.6.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												STATUS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

46.6.1.9.4 Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Field	Function
—	
3-0 STATUS	This field indicates PPI ErrSotSync_HS captured status from D-PHY Bit[0] – Data Lane0 Bit[1] – Data Lane1 Values not mentioned are 'Reserved'

46.6.1.10 ErrEsc Status Register (PPI_ERRESC)

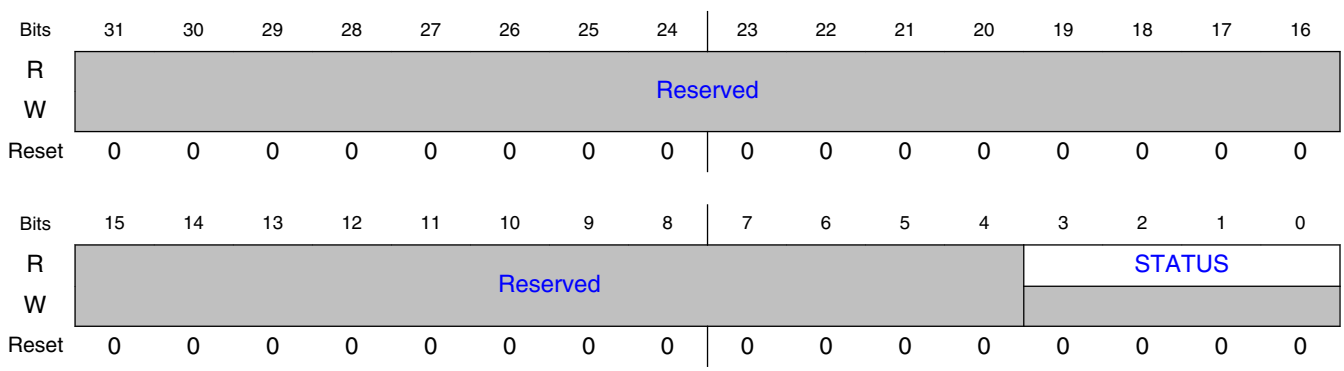
46.6.1.10.1 Offset

Register	Offset
PPI_ERRESC	120h

46.6.1.10.2 Function

Indicates PPI ErrEsc captured status from D-PHY.

46.6.1.10.3 Diagram



46.6.1.10.4 Fields

Field	Function
31-4 —	Reserved
3-0 STATUS	This field indicates PPI ErrEsc captured status from D-PHY [0] – Data Lane0

Field	Function
	[1] – Data Lane1 Values not mentioned are 'Reserved'

46.6.1.11 ErrSyncEsc Status Register (PPI_ERRSYNCESC)

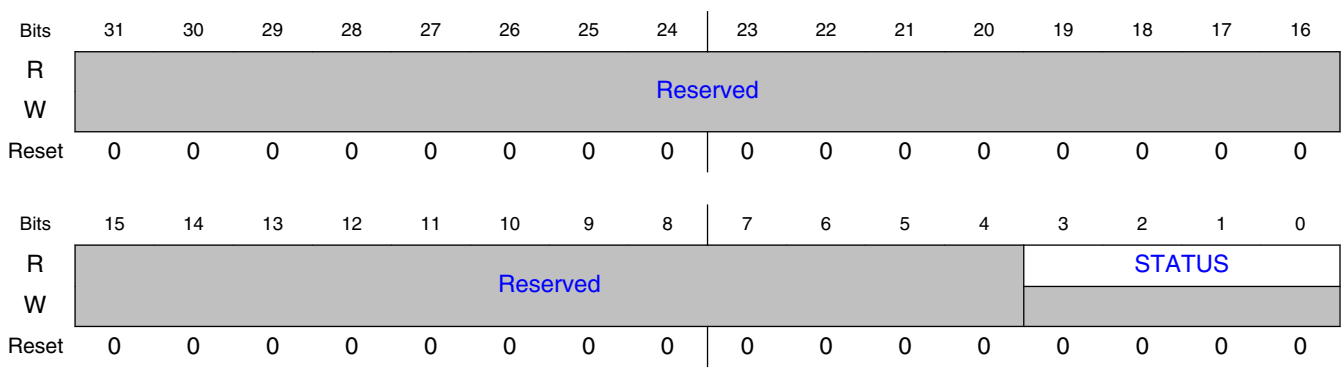
46.6.1.11.1 Offset

Register	Offset
PPI_ERRSYNCESC	124h

46.6.1.11.2 Function

Indicates PPI ErrSyncEsc captured status from D-PHY.

46.6.1.11.3 Diagram



46.6.1.11.4 Fields

Field	Function
31-4 —	Reserved
3-0 STATUS	This field indicates PPI ErrSyncEsc captured status from D-PHY [0] – Data Lane0 [1] – Data Lane1 Values not mentioned are 'Reserved'

46.6.1.12 ErrControl Status Register (PPI_ERRCONTROL)

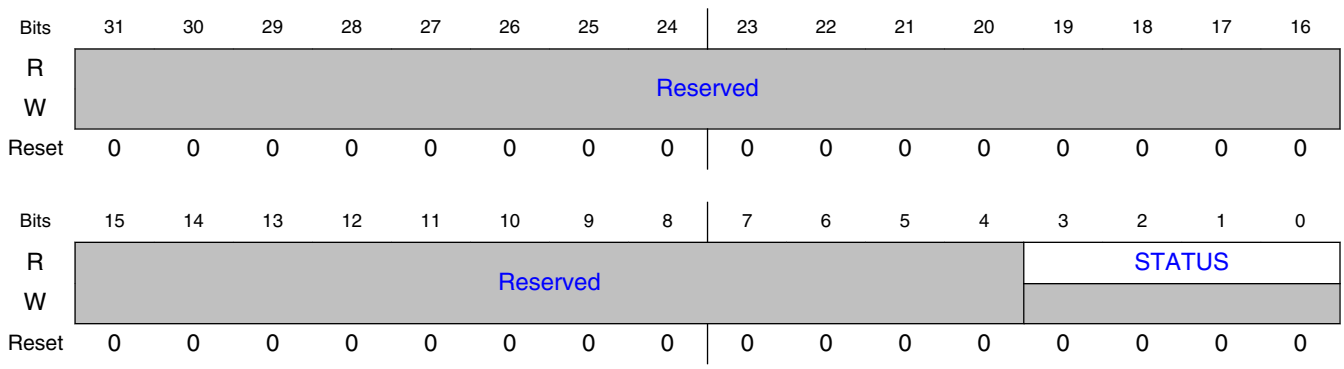
46.6.1.12.1 Offset

Register	Offset
PPI_ERRCONTROL	128h

46.6.1.12.2 Function

Indicates PPI ErrControl captured status from D-PHY.

46.6.1.12.3 Diagram



46.6.1.12.4 Fields

Field	Function
31-4 —	Reserved
3-0 STATUS	This field indicates PPI ErrControl captured status from D-PHY [0] – Data Lane0 [1] – Data Lane1 Values not mentioned are 'Reserved'

46.6.1.13 Disable Payload 0 Register (CFG_DISABLE_PAYLOAD_0)

46.6.1.13.1 Offset

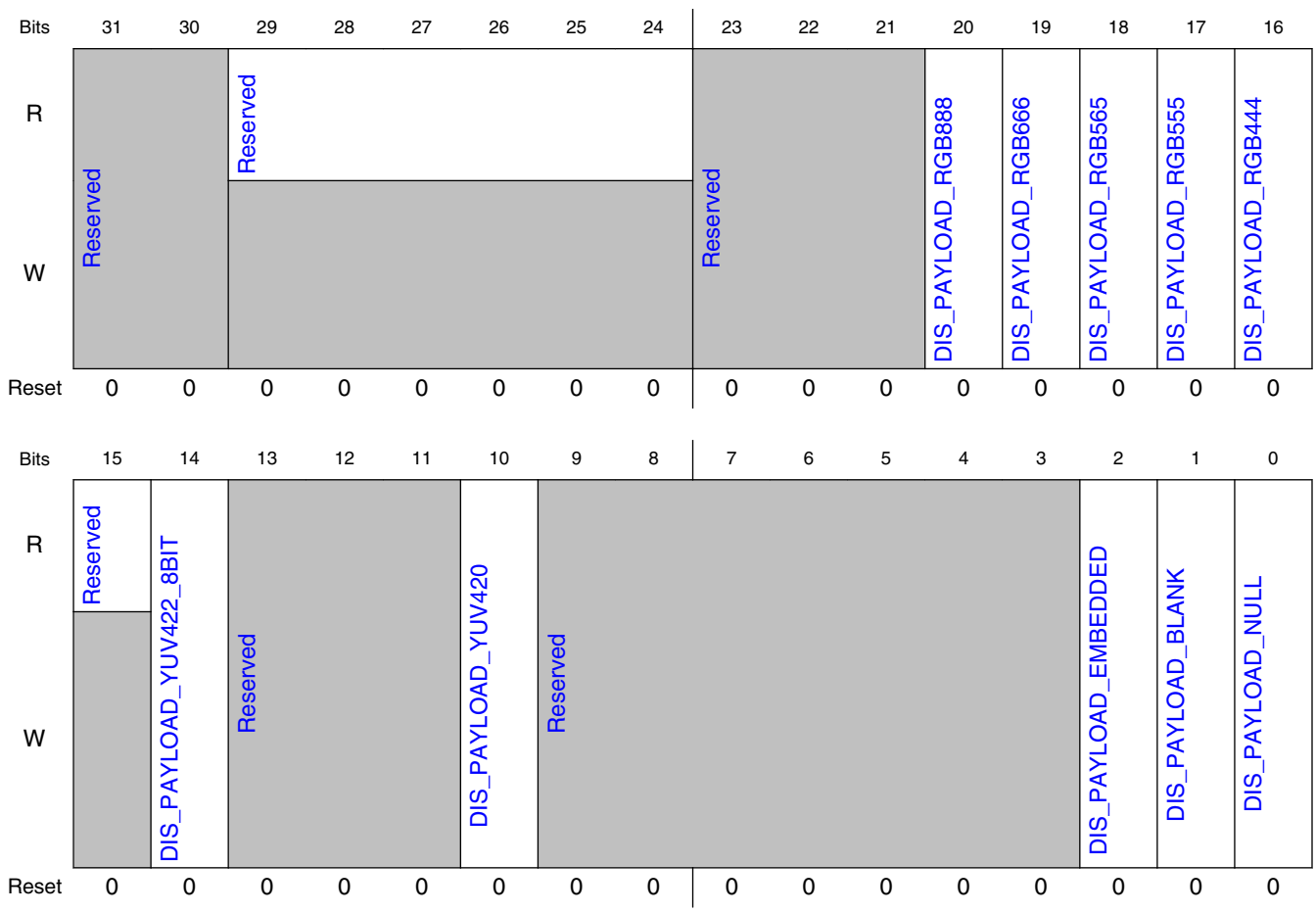
Register	Offset
CFG_DISABLE_PAYLOAD_0	12Ch

46.6.1.13.2 Function

Disables payload data for the selected data type.

When a bit is set that represents a supported data type, only the packet header will be presented at the user interface, along with the sop and eop indication where the payload data would have appeared if enabled.

46.6.1.13.3 Diagram



46.6.1.13.4 Fields

Field	Function
31-30 —	Reserved
29-24 —	Reserved
23-21 —	Reserved
20 DIS_PAYLOAD_RGB888	RGB888
19 DIS_PAYLOAD_RGB666	RGB666
18 DIS_PAYLOAD_RGB565	RGB565
17 DIS_PAYLOAD_RGB555	RGB555
16 DIS_PAYLOAD_RGB444	RGB444
15 —	Reserved
14 DIS_PAYLOAD_YUV422_8BIT	YUV422 8 bit
13-11 —	Reserved
10 DIS_PAYLOAD_YUV420	Legacy YUV 420 8 bit
9-3 —	Reserved
2 DIS_PAYLOAD_EMBEDDED	Embedded
1 DIS_PAYLOAD_BLANK	Blank
0 DIS_PAYLOAD_NULL	Null

46.6.1.14 Disable Payload 1 Register (CFG_DISABLE_PAYLOAD_1)

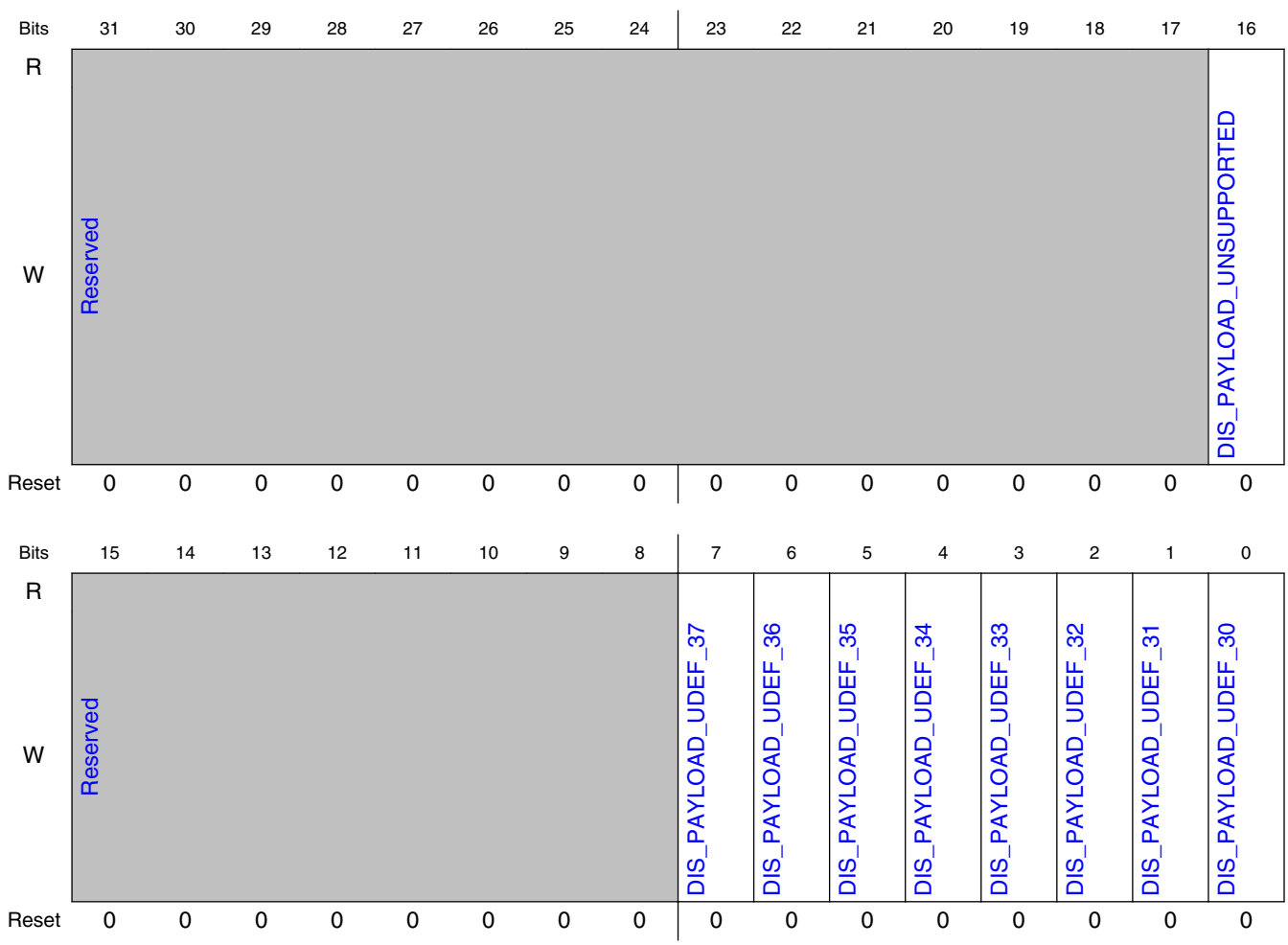
46.6.1.14.1 Offset

Register	Offset
CFG_DISABLE_PAYLOAD_1	130h

46.6.1.14.2 Function

Disables payload for user defined types.

46.6.1.14.3 Diagram



46.6.1.14.4 Fields

Field	Function
31-17 —	Reserved
16 DIS_PAYLOAD_UNSUPPORTED	Unsupported Data Types
15-8 —	Reserved
7 DIS_PAYLOAD_UDEF_37	User defined type 0x37
6 DIS_PAYLOAD_UDEF_36	User defined type 0x36
5 DIS_PAYLOAD_UDEF_35	User defined type 0x35
4 DIS_PAYLOAD_UDEF_34	User defined type 0x35
3 DIS_PAYLOAD_UDEF_33	User defined type 0x34
2 DIS_PAYLOAD_UDEF_32	User defined type 0x33
1 DIS_PAYLOAD_UDEF_31	User defined type 0x32
0 DIS_PAYLOAD_UDEF_30	User defined type 0x31

Chapter 47
Enhanced LCD Interface (eLCDIF)

47.1 Chip-specific eLCDIF information

Table 47-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

eLCDIF do not support VSYNC mode in this SoC.

47.2 Overview

The enhanced Liquid Crystal Display Interface (eLCDIF) is a general purpose display controller

The eLCDIF block supports the following:

- Displays that support moving pictures and require the RGB interface mode (DOTCLK interface).

47.2.1 Block Diagram

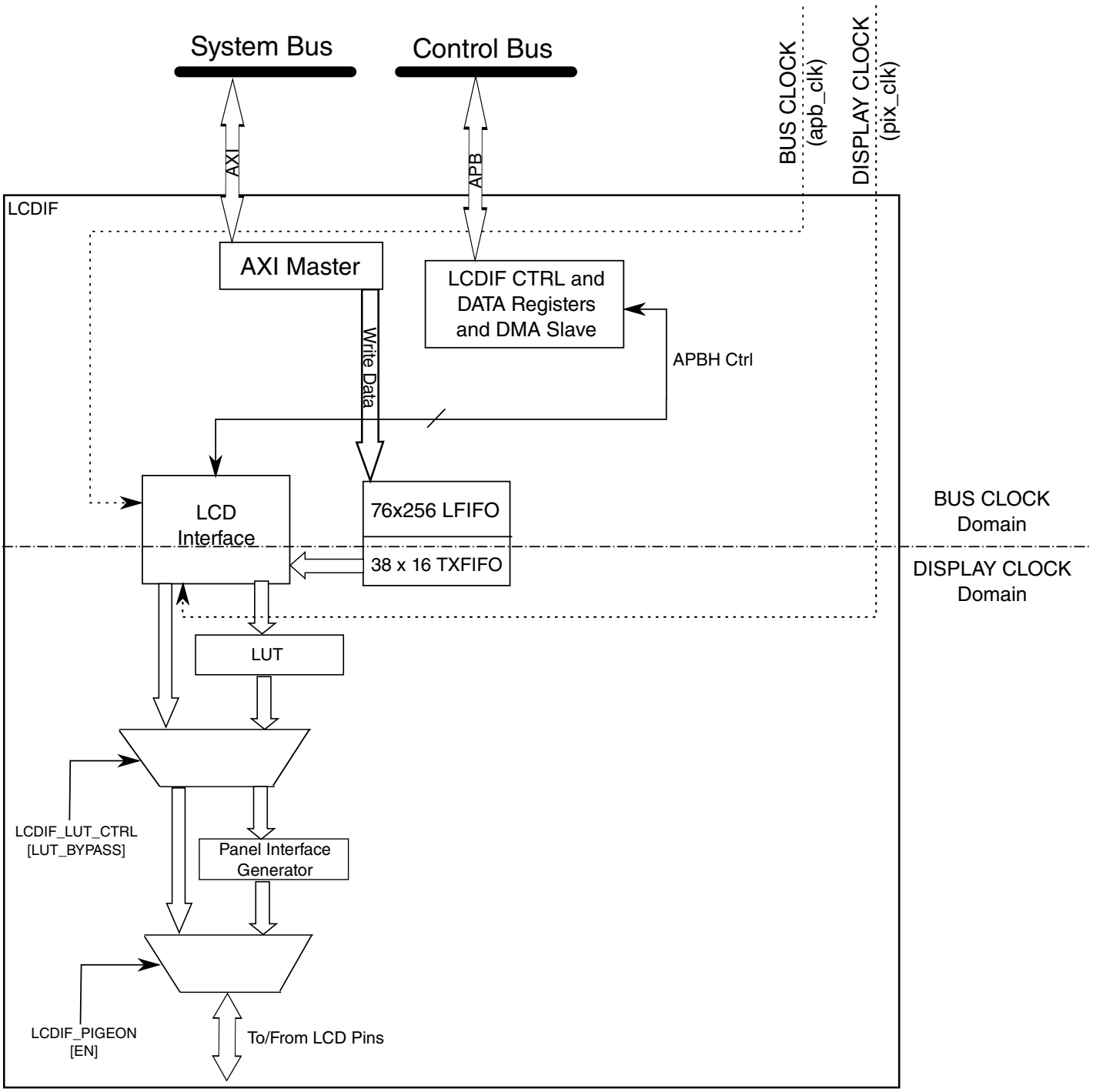


Figure 47-1. Top-Level Block Diagram of eLCDIF subsystem

47.2.2 Features

The eLCDIF provides fully programmable functionality to supported interfaces:

- Bus master interface to source frame buffer data for display refresh. This interface can also be used to drive data for "Smart" displays.
- DMA interface to manage data transfers between "Smart" displays and SoC.
- 8/16/18/24 bit LCD data bus support available depending on I/O mux options.
- Programmable timing and parameters for DOTCLK LCD interfaces.

47.3 Functional Description

The sections listed below describe the pipeline for the eLCDIF interface:

- Bus Interface Mechanisms
- Write Data Path
- Initialization

The differences for each modes are described in the sections listed below:

- DOTCLK Interface

LCDIF pin usage by interface mode is described in eLCDIF Pin Usage by Interface Mode.

47.3.1 Bus Interface Mechanisms

The eLCDIF module has memory-mapped control, data and status registers. It provides several interfaces to transfer data between the display and SoC.

The bus master interface is used to initiate the requests to transfer data from external memory to the display. It is completely autonomous, or no CPU intervention is required, to manage the cyclical nature of refreshing standard display types.

The DMA interface requires the SoC integrated APBDMA engine to manage data transfers on behalf of the eLCDIF. APBDMA descriptor chains are created to manage control and data flow between the eLCDIF and external display. This interface is typically used for "Smart" display types that integrate their own frame buffer and control interface. The APBDMA engine is used to reduce the CPU compute requirements to complete the display solution.

The host CPU executes display drivers to manage the display solution. The following sections describe the system bus interface mechanisms.

47.3.1.1 Bus Master Operation in Write/Display Modes

The eLCDIF block has a bus master interface that initiates requests for data to drive the display. The LCDIF_CTRL[MASTER] bit must be set to 1 to enable the bus master interface. Software should program all control registers required to transfer the frame sequence.

The DOTCLK mode is used to refresh the display at the desired refresh rate and resolution, and drive displays that don't integrate a display buffer memory. When the display is refreshed, the eLCDIF will automatically update the LCDIF_CUR_BUF register with the value in LCDIF_NEXT_BUF at the end of current frame and start fetching the next frame from the new address. If the LCDIF_NEXT_BUF register was not updated within a frame refresh cycle, eLCDIF will keep transmitting the last frame until a new value is programmed into that register.

47.3.1.2 System Bus Master Performance

The performance of the eLCDIF block can be controlled by changing the burst length and the outstanding cycle issuing capability depending on the memory bandwidth requirements. Two fields in the LCDIF_CTRL2 register will throttle system memory requests. The LCDIF_CTRL2[OUTSTANDING_REQS] bitfield will control how many requests the eLCDIF can have in flight on any given clock cycle. This should be programmed based on the expected system bus latency for returned read data. Also, the LCDIF_CTRL2[BURST_LEN_8] bit will set the number of 64 bit words requested for each eLCDIF system bus request to either 8 or 16 QWORDS. Generally, 4 outstanding requests of length 16 will provide enough performance to drive any standard display resolution. These configuration bits are intended to change the access pattern of the eLCDIF to optimize system bus throughput when other system masters will contend for system memory resources.

The LCDIF_THRES register can also be used to optimize bus throughput and power consumption.

The features available with the LCDIF_THRES register require support from system clocking and dynamic priority control.

47.3.1.3 DMA Operation in MPU Read Mode

Data can be transferred between an external display and the SoC using the APBDMA interface. This mode is enabled by setting the LCDIF_CTRL[MASTER] bit to 0 and is typically used with "Smart" displays. The APBDMA module executes DMA descriptor chains to process the desired data flow sequence. The data can be either frame buffer data or display control data.

47.3.2 Write Data Path

eLCDIF supports raster based frame buffers and there is no support for tiled buffers.

There are several options to accommodate endianness of display buffers in memory before the data is processed for the external display. The LCDIF_CTRL[INPUT_DATA_SWIZZLE] field provides the following options for data word multiplexing:

```
00 (0): No swizzle (little-endian)
01 (1): Swap bytes 0 and 3, swap bytes 1 and 2 (big-endian)
10 (2): Swap half-words
11 (3): Swap bytes within each half-word
```

The LCDIF_CTRL[WORD_LENGTH] field indicates the input data/pixel format. LCDIF_TRANSFER_COUNT register denotes how much data is contained in each frame. The LCDIF_TRANSFER_COUNT[H_COUNT] field indicates the number of pixels per line and LCDIF_TRANSFER_COUNT[V_COUNT] indicates the total number of lines per frame. The LCDIF_CTRL1[BYTE_PACKING_FORMAT] field can be used to specify which bytes within the 32-bit word are going to be valid. For example, if the entire 32-bit word is valid, LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be set to 0xF, if only lower 3 bytes of each word in the frame buffer are valid, then LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be set to 0x7.

The LCDIF_CTRL[LCD_DATABUS_WIDTH] field suggests the width of the bus going to the display controller. There is an option to source all 32 bits of the input word and transfer it to the output I/O display interface. If the LCDIF_CTRL[LCD_DATABUS_WIDTH] is not the same as LCDIF_CTRL[WORD_LENGTH], eLCDIF will perform RGB to RGB color space conversion. For example, if the input frame has fewer bits per pixel than the display, as in a 16 bpp input frame going to 24 bpp LCD, eLCDIF will pad the MSBs of each color to the LSBs of the same color for each pixel. If the input frame has more bits per pixel than the display, for example, 24 bpp input frame going to 16 bpp LCD, eLCDIF will drop the LSBs of each color channel to convert to the lower color depth. eLCDIF also has the capability to support delta pixel displays by swizzling the R, G and B colors of each pixel in the odd and even lines of the frame separately by programming the

LCDIF_CTRL2[ODD_LINE_PATTERN] and the LCDIF_CTRL2[EVEN_LINE_PATTERN] bit fields. This operation occurs after the RGB-to-RGB color space conversion operation.

The following list shows how the different input/output combinations can be obtained:

- LCDIF_CTRL[WORD_LENGTH]=1 indicates that the input is 8-bit data. Any combination of LCDIF_CTRL1[BYTE_PACKING_FORMAT] is permissible.

  Limitation: LCDIF_TRANSFER_COUNT[H_COUNT] must be a multiple of the sum of BYTE_PACKING_FORMAT [3], BYTE_PACKING_FORMAT [2], BYTE_PACKING_FORMAT [1] and BYTE_PACKING_FORMAT [0]. LCDIF_CTRL[LCD_DATABUS_WIDTH] must be 1, indicating an 8-bit data bus.

- LCDIF_CTRL[WORD_LENGTH]=0 implies the input frame buffer is RGB 16 bits per pixel. LCDIF_CTRL[DATA_FORMAT_16_BIT] field determines the pixels are RGB 555 or RGB 565.

  Limitation: LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be 0x3 or 0xC if there is only one pixel per word. If there are two pixels per word, it should be 0xF and LCDIF_TRANSFER_COUNT[H_COUNT] will be restricted to be a multiple of 2 pixels.

- LCDIF_CTRL[WORD_LENGTH]=2 indicates that input frame buffer is RGB 18 bits per pixel, that is, RGB 666. The valid RGB values can be left-aligned or right-aligned within a 32-bit word. The alignment of the valid 18 bits within a word is indicated by the LCDIF_CTRL[DATA_FORMAT_18_BIT] bit.

  Limitation: LCDIF_CTRL1[BYTE_PACKING_FORMAT] can be 0x7, 0xE or 0xF. Packed pixels are not supported in this case. LCDIF_TRANSFER_COUNT[H_COUNT] can be any number.

- LCDIF_CTRL[WORD_LENGTH]=3 indicates that the input frame-buffer is RGB 24 bits per pixel (RGB 888). If LCDIF_CTRL1[BYTE_PACKING_FORMAT] is 0x7, it indicates that there is only one pixel per 32-bit word and there is no restriction on LCDIF_TRANSFER_COUNT[H_COUNT]. This is also the option that provides 32 bit output depending on the I/O muxing options available. The fourth byte, or bits [31:24], and connected to the I/Os if this muxing is available in the chip package.

  Limitation: If LCDIF_CTRL1[BYTE_PACKING_FORMAT] is 0xF, it indicates that the pixels are packed, that is, there are 4 pixels in 3 words or 12 bytes and LCDIF_TRANSFER_COUNT[H_COUNT] must be a multiple of 4 pixels.

After the RGB to RGB color space conversions, there is one more opportunity to swizzle the data before sending it out to the display or the encoder. This can be done with the LCDIF_CTRL[CSC_DATA_SWIZZLE] field, and it provides the same options as the LCDIF_CTRL[INPUT_DATA_SWIZZLE] register.

Finally, there is an option to shift the output data before sending it out to the display. This is done based on the LCDIF_CTRL[SHIFT_DIR] and LCDIF_CTRL[SHIFT_NUM_BITS] fields.

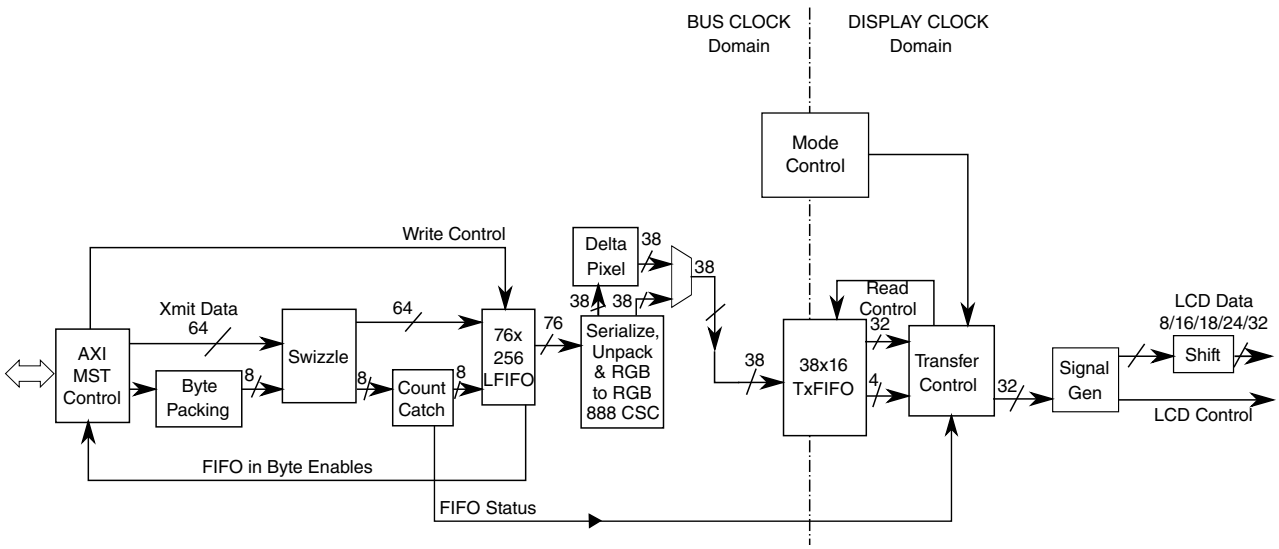


Figure 47-2. General Operations in Write Data Path

The examples in the following figures illustrate some different combinations of register programming for write mode. Assume that the data transferred over the system bus within a 32 bit word is organized as {A7-A0, B7-B0, C7-C0, D7-D0} in 8-bit mode and {A15-A0, B15-B0} in 16-bit mode.

In this example, all 32 bits of the input word are transferred out over an 8 bit display bus. Each byte within the 32 bit word is shifted to the right with zeros appended to bits D[7:6]. The input data bits [7:2] are shifted to the right by 2 bits and presented on the D[5:0].

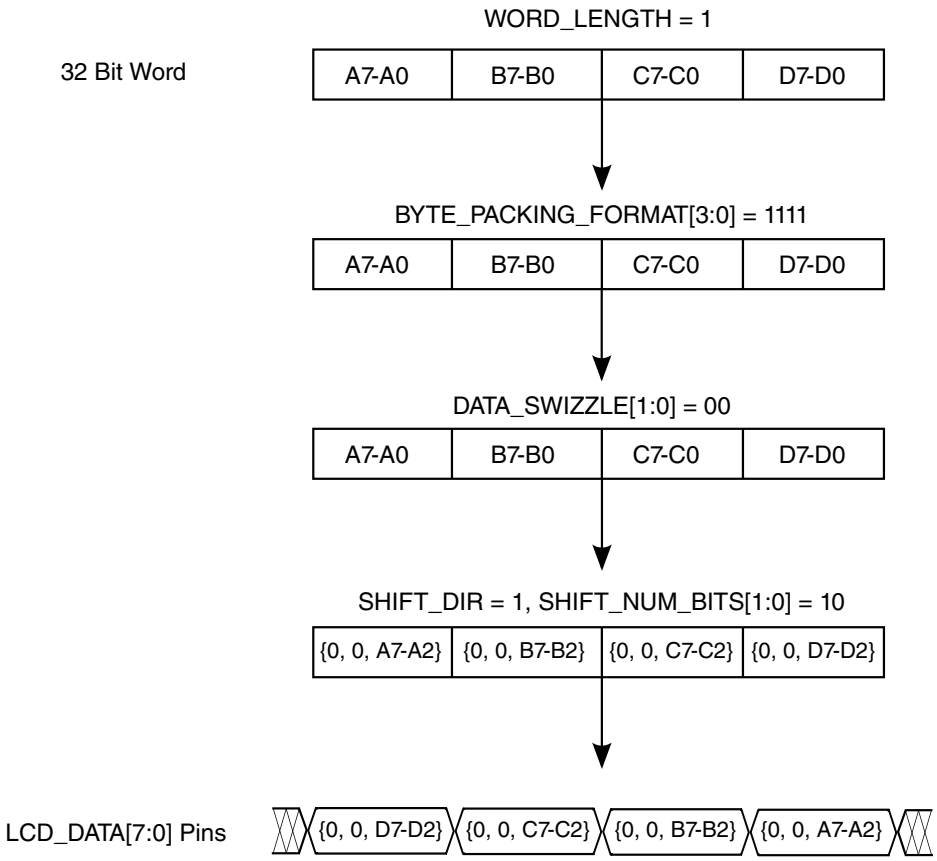


Figure 47-3. Register programming for write mode

In this 8 bit display interface example, one byte of the input word is deleted and not transferred over the external 8 bit display interface. This mode could be used to transfer 24bpp pixels over the 8 bit interface. In this case, the 4th unused byte is not transferred.

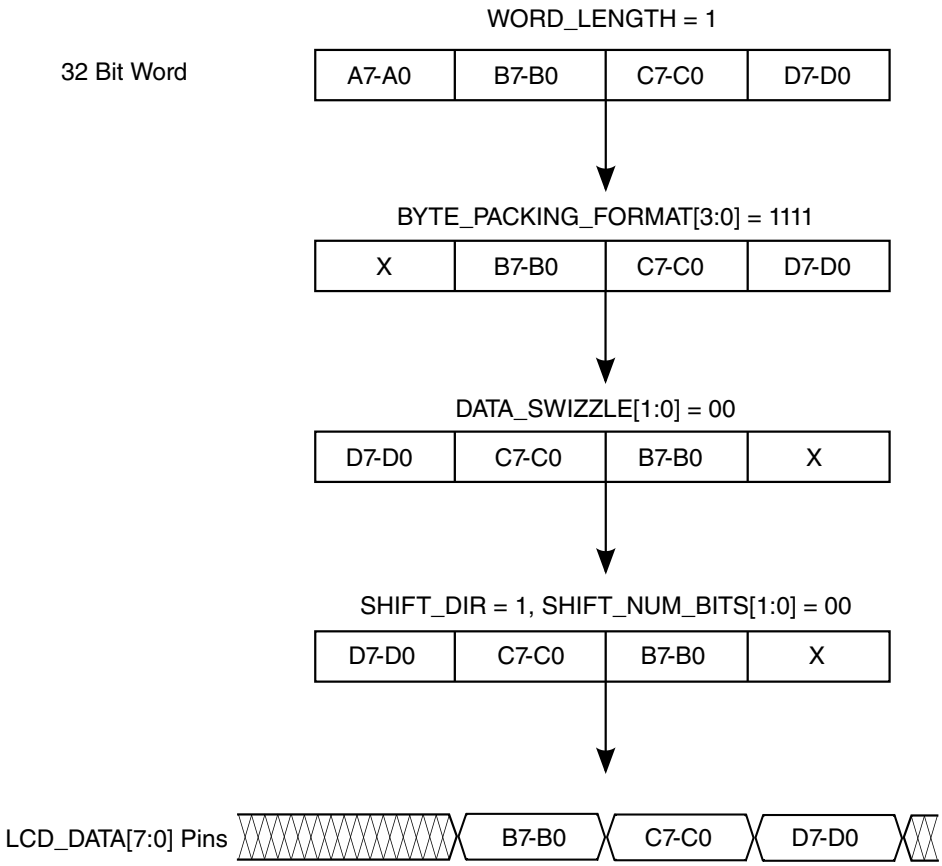


Figure 47-4. Register programming for write mode

The following example uses a 16 bit display interface. Each 16 bit half word is shifted to the right by two bits with zeros appended to the most significant two bits.

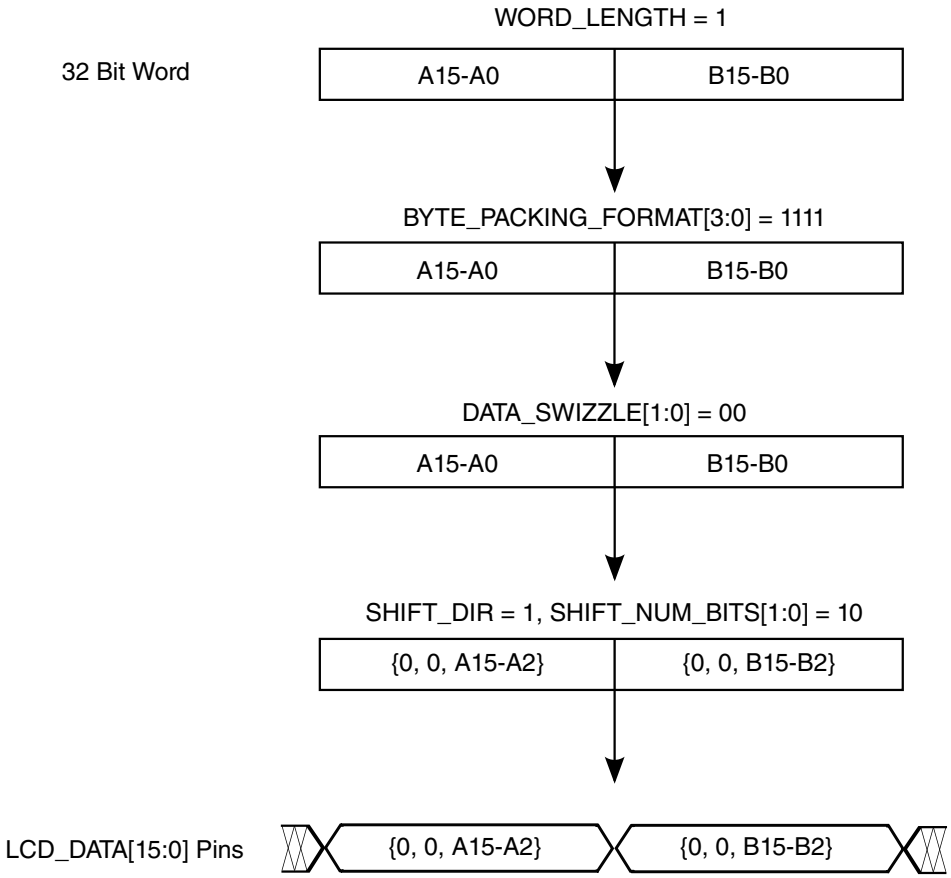


Figure 47-5. Register programming for write mode

This example indicates how an unpacked frame buffer can sourced for display. Only a single 16 bit half word within the 32 bit word is tranferred out via the 16 display bus.

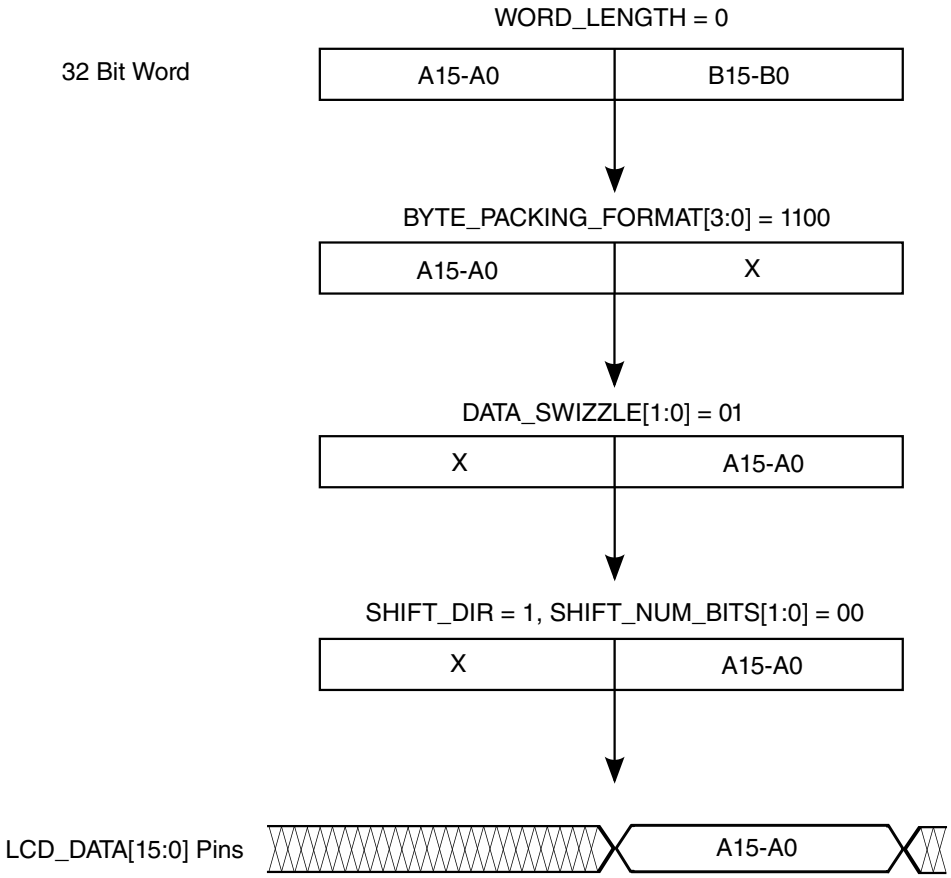


Figure 47-6. Register programming for write mode

47.3.3 DOTCLK Interface

The DOTCLK interface is another mode used in moving picture displays.

It includes the VSYNC, HSYNC, DOTCLK and (optional) ENABLE signals. The interface is popularly called the RGB interface if the ENABLE signal is present.

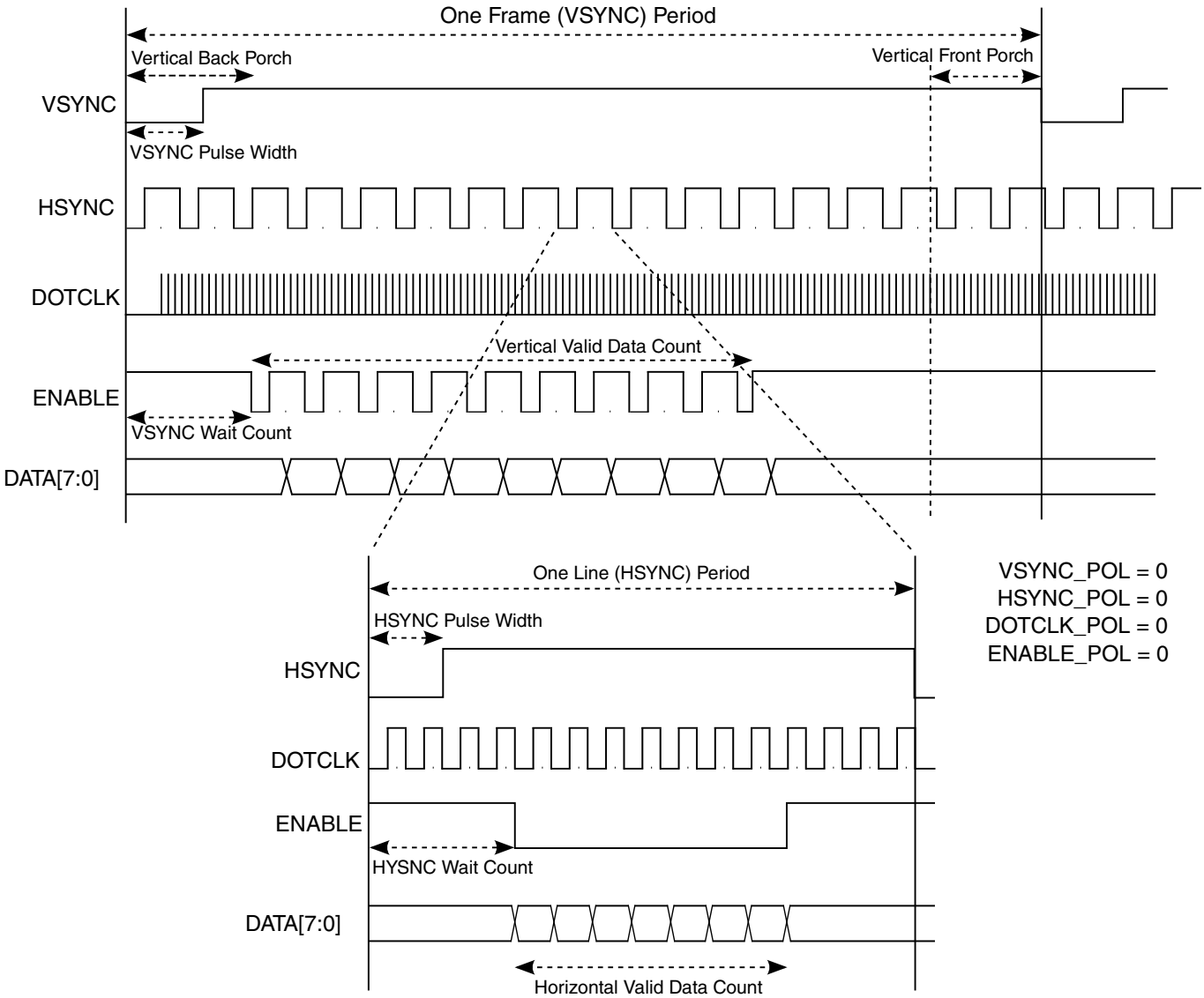


Figure 47-7. DOTCLK protocol with programmable parameters

The DOTCLK mode writes data at high speed to the LCD, and the display operation is synchronized with the VSYNC, HSYNC, ENABLE and DOTCLK signals. The polarities, periods and pulse-widths of the sync signals are programmable using the LCDIF_VDCTRL0-4 registers. The units for the VSYNC signal must be number of horizontal lines and can be selected using the VSYNC_PULSE_WIDTH_UNIT and VSYNC_PERIOD_UNIT bit fields. The VERTICAL_WAIT_CNT is by default given the same unit as the VSYNC_PERIOD. The DISPLAY CLOCK (pix_clk) frequency is managed by the CCM.

In DOTCLK mode, LCDIF_CTRL_BYPASS_COUNT bit must be set to 1. To end the current transfer, the software should make the DOTCLK_MODE bit 0, so that all data that is currently in the LCDIF LFIFO and TXFIFO is transmitted. Once that transfer is complete, the block will automatically clear the RUN bit and issue the cur_frame_done interrupt.

Refer to Write Data Path for more details.

47.3.3.1 Code Example

The following code shows an example for programming a 320x240 display.

NOTE

Setting up the display must be done via SPI.

```
// Note: Common initialization steps in Initialization must also be
// executed along with the following code
BF_CS1 (LCDIF_CTRL, DOTCLK_MODE, 1);
BF_CS1 (LCDIF_CTRL, BYPASS_COUNT, 1); //Always for DOTCLK mode
BF_CS1 (LCDIF_VDCTRL0, VSYNC_OEB, 0); //Vsync is always an output in the DOTCLK mode
BF_CS4 (LCDIF_VDCTRL0, VSYNC_POL, 0, HSYNC_POL, 0, DOTCLK_POL, 0, ENABLE_POL, 0);
BF_CS1 (LCDIF_VDCTRL0, ENABLE_PRESENT, 1);
BF_CS2 (LCDIF_VDCTRL0, VSYNC_PERIOD_UNIT, 1, VSYNC_PULSE_WIDTH_UNIT, 1);
BF_CS1 (LCDIF_VDCTRL0, VSYNC_PULSE_WIDTH, 2);
BF_CS1 (LCDIF_VDCTRL1, VSYNC_PERIOD, 280);
BF_CS2 (LCDIF_VDCTRL2, HSYNC_PULSE_WIDTH, 10, HSYNC_PERIOD, 360); //Assuming
                                                    //  LCD_DATABUS_WIDTH is 24bit
BF_CS2 (LCDIF_VDCTRL3, VSYNC_ONLY, 0);
BF_CS2 (LCDIF_VDCTRL3, HORIZONTAL_WAIT_CNT, 20, VERTICAL_WAIT_CNT, 20);
BF_CS1 (LCDIF_VDCTRL4, DOTCLK_H_VALID_DATA_CNT, 320);//Note that DOTCLK_V_VALID_DATA_CNT is
                                //implicitly assumed to be HW_LCDIF_TRANSFER_COUNT_V_COUNT
BF_CS1 (LCDIF_VDCTRL4, SYNC_SIGNALS_ON, 1);
BF_CS1 (LCDIF_CTRL, RUN, 1);
```

To stop the transfer completely, the ideal way is to make DOTCLK_MODE = 0. In that case, the block will transmit the contents in the FIFO and reset the RUN bit.

47.3.4 LUT

The Lookup Table (LUT) is used to expand the 8 bits pixel to 24 bits pixel before output to external displayer. This module is used when input data from memory is 8-bits in width under limited bandwidth system and external displayer has a 24 bits requirement.

There are two 256x24 bits LUT memory in eLCDIF module in case of changing LUT during active display. When using LUT function, it should be initialized through single PIO register writing first. Writing data to register LCDIF_LUT0_DATA will trigger storing data to lut0 memory and increase the lut0 memory address LCDIF_LUT0_ADDR

automatically. Writing data again to register LCDIF_LUT0_DATA will store the data to memory and increase the memory address accordingly. The same operation is performed while writing to LCDIF_LUT1_DATA and LCDIF_LUT1_ADDR.

After initialization, LCDIF_CUR_BUF_ADDR bit 0 determines which LUT is used during active display.

Lastly, when there is active display, it will use the lowest 8 bits eLCDIF data as LUT address and output the 24 bits from the lookup table.

47.3.5 Panel Interface Generator (Pigeon Mode)

There are several panel interface signal outputs (CLK signal not included), each of them with dedicated timing purpose as default. This is called "Legacy Mode". Pigeon Mode is a timing mode which can be independently enabled on any of the timing signals, with a unified flexible configuration. Signals within pigeon mode can be programmed into any supported signals, and are interchangeable. The following are the legacy timing signals which support pigeon mode:

- PIGEON[00] – LCD_DATA16
- PIGEON[01] – LCD_DATA17
- PIGEON[02] – LCD_DATA18
- PIGEON[03] – LCD_DATA19
- PIGEON[04] – LCD_DATA20
- PIGEON[05] – LCD_DATA21
- PIGEON[06] – LCD_DATA22
- PIGEON[07] – LCD_DATA23
- PIGEON[08] – LCD_ENABLE
- PIGEON[09] – LCD_HSYNC
- PIGEON[10] – LCD_VSYNC
- PIGEON[11] – LCD_CLK

Each pigeon signal has one local counter with a configurable start point and incremental condition. It will be compared to configuration register value for signal assertion/de-assertion control, plus delta offset for data alignment and other options like polarity/logic operation. A detailed running scenario is as follows:

1. Start local counter on the MASK rising edge (reference point/start point)
2. Increment on event selected through INC_SEL
3. Count and match SET_CNT: assert signal, reset counter (SET_CNT==0 means assert immediately on MASK's rising edge)
4. Count and match CLR_CNT: de-assert signal, stop counter (CLR_CNT==0 means de-assert on MASK's falling edge)

NOTE

When local counter is running, further changes to MASK are not cared unless CLR_CNT==0

MASK is the start point for local counter created using a combination any of the options (ANDed) from below:

1. STATE_MASK= (FS|FB|FD|FE) AND (LS|LB|LD|LE)

   In this case you have 8 bits to select in which vertical/horizontal state your counter starts ticking. This is the most common use-case because timing signals generally relate to scan states.

   For example, for a line timing signal start on Line Begin phase during Frame Begin/ Frame Data lines, use the configuration below:

Table 47-2. State Mask Combinations

STATE_MASK	LS	LB=1	LD	LE
FS				
FB=1		X		
FD=1		X		
FE				

2. MASK_CNT/MASK_CNT_SEL (global counter)

   Sometimes a more accurate reference point is required, such as "line 20 in a frame", or "line 12 in Frame Begin state" or "pixel 23 in LD phase". For such use-cases, several Global Counters shared by all pigeons are provided. Global counter type (line counter/frame counter/state counter, etc.) is selected using MASK_CNT_SEL, when global counter matches the MASK_CNT value. The pigeon local counter will start ticking.

3. Use another pigeon signal as mask (SIG_LOGIC=MASK)

   For some tightly coupled signals it is possible to use one as a reference to generate another.

   INC_SEL- select local counter tick event
   a. pclk - pixel clk
   b. line - line start pulse
   c. frame - frame start pulse
   d. another - another pigeon signal

OFFSET- offset on pclk basis: Some signals need to come out slightly earlier or later than programmed. For example, the CE type signal usually aligns with the Line Data phase, but some panels need it as one cycle pulse before Line Data. The user can set OFFSET to a negative value to achieve this.

Global counters (selectable through MASK_CNT_SEL):

- HCNT / VCNT : normal pclk counter / line counter
- HSTATE_CNT / VSTATE_CNT: similar to above, but reset when state changes
- HSTATE_CYCLE / VSTATE_CYCLE: (see figure below for definition of CYCLE/ PERIOD/CNT)

  Some panels have multiple Gate Drivers/Source Drivers, so Frame Data / Line Data state may be further split to match each driver, and signals such as CE[n] are only valid during part n of Line Data. For such signals, use LCDIF_PIGEON_CTRL[*_PERIOD] to specify PERIOD where CNT is reset and CYCLE is included. Then the user can select CYCLE as MASK_CNT to generate mask for CE[n].

- FRAME_CNT / FRAME_CYCLE (only for frame-crossing signals): frame cycle counter doesn't have a reset condition; use LCDIF_PIGEON_CTRL1[FRAME_CNT_CYCLES] to reset it.

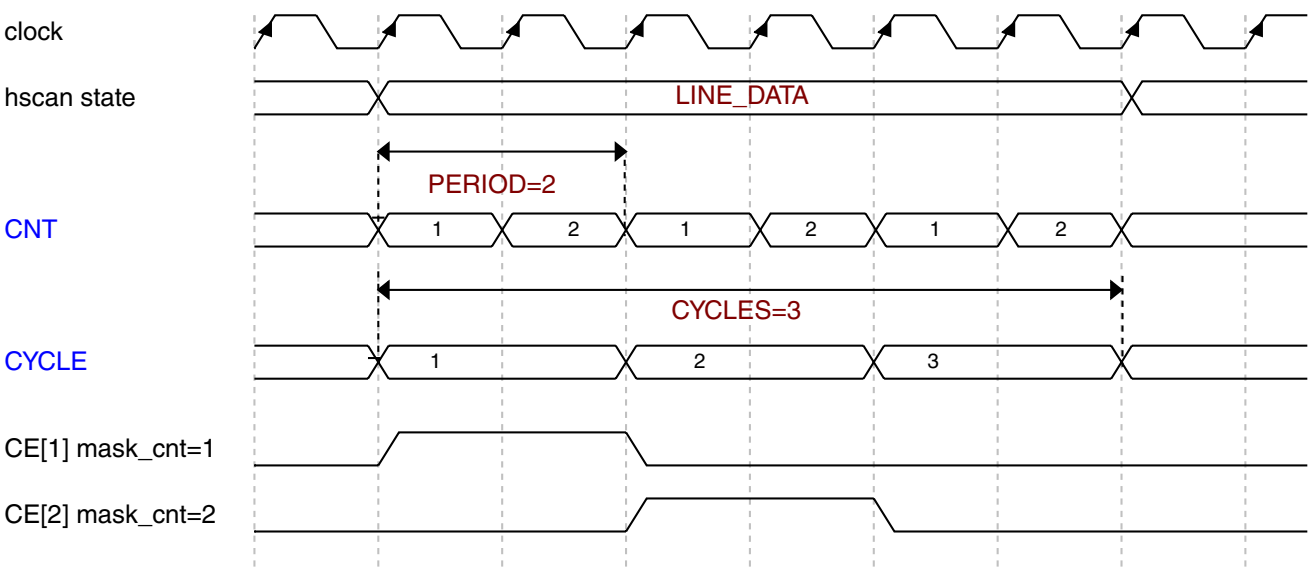


Figure 47-8. Definition of CNT, CYCLE, PERIOD

The following are register settings for the figure above:

- LCDIF_PIGEON_CTRL0[LD_PERIOD]=2
- MASK_CNT_SEL = 1 // HSTATE_CYCLE

- MASK_CNT = 1 // CE[1]
- MASK_CNT = 2 // CE[2]

47.3.6 eLCDIF Pin Usage by Interface Mode

The following tables detail how the eLCDIF level interface pins are used based on the desired mode of operation. The chip level I/Os should also be configured to be consistent with the desired eLCDIF operating mode.

NOTE

There is an option to internally mux the HSYNC, DOTCLK and ENABLE signals in the DOTCLK mode by setting the LCDIF_VDCTRL3[MUX_SYNC_SIGNALS] bit.

Table 47-3. Pin use in DOTCLK Mode

PIN NAME	8-bit DOTCLK LCD IF	16-bit DOTCLK LCD IF	18-bit DOTCLK LCD IF	24-bit DOTCLK LCD IF
LCD_VSYNC	LCD_VSYNC	LCD_VSYNC	LCD_VSYNC	LCD_VSYNC
LCD_HSYNC	LCD_HSYNC	LCD_HSYNC	LCD_HSYNC	LCD_HSYNC
LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK
LCD_ENABLE	LCD_ENABLE	LCD_ENABLE	LCD_ENABLE	LCD_ENABLE
LCD_DATA23 (LCD_D23)	X	X	X	LCD_DATA23
LCD_DATA22 (LCD_D22)	X	X	X	LCD_DATA22
LCD_DATA21 (LCD_D21)	X	X	X	LCD_DATA21
LCD_DATA20 (LCD_D20)	X	X	X	LCD_DATA20
LCD_DATA19 (LCD_D19)	X	X	X	LCD_DATA19
LCD_DATA18 (LCD_D18)	X	X	X	LCD_DATA18
LCD_DATA17 (LCD_D17)	X	X	LCD_DATA17	LCD_DATA17
LCD_DATA16 (LCD_D16)	X	X	LCD_DATA16	LCD_DATA16
LCD_DATA15 (LCD_D15)	X	LCD_DATA15	LCD_DATA15	LCD_DATA15
LCD_DATA14 (LCD_D14)	X	LCD_DATA14	LCD_DATA14	LCD_DATA14
LCD_DATA13 (LCD_D13)	X	LCD_DATA13	LCD_DATA13	LCD_DATA13

Table continues on the next page...

Table 47-3. Pin use in DOTCLK Mode (continued)

PIN NAME	8-bit DOTCLK LCD IF	16-bit DOTCLK LCD IF	18-bit DOTCLK LCD IF	24-bit DOTCLK LCD IF
LCD_DATA12 (LCD_D12)	X	LCD_DATA12	LCD_DATA12	LCD_DATA12
LCD_DATA11 (LCD_D11)	X	LCD_DATA11	LCD_DATA11	LCD_DATA11
LCD_DATA10 (LCD_D10)	X	LCD_DATA10	LCD_DATA10	LCD_DATA10
LCD_DATA09 (LCD_D9)	X	LCD_DATA09	LCD_DATA09	LCD_DATA09
LCD_DATA08 (LCD_D8)	X	LCD_DATA08	LCD_DATA08	LCD_DATA08
LCD_DATA07 (LCD_D7)	LCD_DATA07	LCD_DATA07	LCD_DATA07	LCD_DATA07
LCD_DATA06 (LCD_D6)	LCD_DATA06	LCD_DATA06	LCD_DATA06	LCD_DATA06
LCD_DATA05 (LCD_D5)	LCD_DATA05	LCD_DATA05	LCD_DATA05	LCD_DATA05
LCD_DATA04 (LCD_D4)	LCD_DATA04	LCD_DATA04	LCD_DATA04	LCD_DATA04
LCD_DATA03 (LCD_D3)	LCD_DATA03	LCD_DATA03	LCD_DATA03	LCD_DATA03
LCD_DATA02 (LCD_D2)	LCD_DATA02	LCD_DATA02	LCD_DATA02	LCD_DATA02
LCD_DATA01 (LCD_D1)	LCD_DATA01	LCD_DATA01	LCD_DATA01	LCD_DATA01
LCD_DATA00 (LCD_D0)	LCD_DATA00	LCD_DATA00	LCD_DATA00	LCD_DATA00
LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET

47.3.7 CRC Polynomial

The eLCDIF uses the CRC32 polynomial to calculate the data signature. The initial value for the CRC calculation is 0xFFFFFFFF.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Figure 47-9. Polynomial function for CRC32

The CRC generator uses the XOR-ed combination of the input data bits and the previous result bits.

47.3.8 Clocks

The following table describes the clock sources for eLCDIF. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 47-4. Clocks

Clock Name	Description
apb_clk	AXI Clock
pix_clk	Pixel Clock

47.3.9 Reset

The Bus Clock (apb_clk) and Display Clock (pix_clk) must be running before making any changes to LCDIF_CTRL[SFTRST] or LCDIF_CTRL[CLKGATE] bits. A soft reset (SFTRST) can take multiple clock periods to complete, so do not set CLKGATE when setting SFTRST. The reset process gates the clocks automatically.

To reset the eLCDIF module, follow the steps shown below:

1. Ungate the clock by setting LCDIF_CTRL[CLKGATE] =0
2. Set LCDIF_CTRL[SFTRST] =1 to trigger reset
3. Wait until LCDIF_CTRL[CLKGATE] asserts
4. Repeat step 1
5. Set LCDIF_CTRL[SFTRST]= 0 to release reset

47.3.10 Interrupts

The eLCDIF module supports a number of interrupts to aid controlling and status reporting of the block.

All the interrupts have individual mask bits for enabling or disabling each of them. They all get funneled through a single interrupt line connected to the interrupt collector (ICOLL).

The following list describes the different interrupts supported by eLCDIF:

- Underflow interrupt is asserted when the clock domain crossing FIFO (TXFIFO) becomes empty but the block is in active display portion during that time. Software should take corrective action to make sure that this does not happen.

- In the bus master mode, the overflow interrupt will be asserted if the block has requested more data than it's FIFOs could hold.
- Cur_frame_done interrupt occurs at the end of every frame .

47.4 External Signals

The following table describes the external signals of eLCDIF:

Table 47-5. External Signals

Signal	Description	Direction
BUSY	Busy signal	I
DOTCLK	Clock Signal	I
CS	Chip Select signal	O
DATAn	Data signal	I/O
ENABLE	Enable signal	I/O
HSYNC	Horizontal Sync	I
VSYNC	Vertical Sync	I

47.5 Initialization

This section describes write modes.

47.5.1 Write Modes

The following initialization steps are common to all eLCDIF write modes of operation before entering any particular mode.

Initialization steps:

1. Configure the external I/Os to correctly interface the external display, when required.
2. Start the Display Clock (pix_clk) clock and set the appropriate frequency by programming the registers in CCM.
3. Start the Bus Clock (apb_clk) and set the appropriate frequency by programming the registers in CCM.
4. Start the Display Clock (pix_clk) clock and set the appropriate frequency by programming the registers in CCM.
5. Start the Bus Clock (apb_clk) and set the appropriate frequency by programming the registers in CCM.

6. Bring the eLCDIF out of soft reset and disable the clock gate bit.
7. Set the transfer mode of operation to bus master. The LCDIF_CTRL[MASTER] bit determines the transfer mode selected. To select bus master mode, set LCDIF_CTRL[MASTER] =1 (to select APBDMA, set LCDIF_CTRL[MASTER] =0).
8. Set the LCDIF_CTRL[INPUT_DATA_SWIZZLE] according to the endianness of the eLCDIF controller. Also, set the LCDIF_CTRL[DATA_SHIFT_DIR] and LCDIF_CTRL[SHIFT_NUM_BITS] if it is required to shift the data left or right before it is output.
9. Set the LCDIF_CTRL[WORD_LENGTH] field appropriately: 0 = 16-bit input, 1 = 8-bit input, 2 = 18-bit input, 3 = 24/32-bit input. Also, select the correct 16/18/24 bit data format with the corresponding fields in LCDIF_CTRL register.
10. Set the LCDIF_CTRL1[BYTE_PACKING_FORMAT] field according to the input frame.
11. Set the LCDIF_CTRL[LCD_DATABUS_WIDTH] appropriately: 0 = 16-bit output, 1 = 8-bit output, 2 = 18-bit output, 3 = 24/32-bit output.
12. Enable the necessary IRQs.

47.6 eLCDIF Memory Map/Register Definition

Some of the eLCDIF registers (XXX_SET, XXX_CLR, and XXX_TOG) allow direct bit field masking and access.

- When writing 1 to XXX_SET bit fields, these registers allow setting the masked 1 bit fields, while keeping unchanged all bit fields which remain on 0 logic state.
- When writing 1 to XXX_CLR bit fields, these registers allow clearing the masked 1 bit fields, while keeping unchanged all other bit fields which remained on 0 logic state.
- When writing 1 to XXX_TOG bit fields, these registers allow inverting the logic state of all masked 1 bit fields, while they keep unchanged the remaining bit fields which were kept on 0 logic state.

LCDIF Hardware Register Format Summary

LCDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_4000	LCDIF General Control Register (LCDIF_CTRL)	32	R/W	C000_0000h	47.6.1/3064

Table continues on the next page...

LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_4004	LCDIF General Control Register (LCDIF_CTRL_SET)	32	R/W	C000_0000h	47.6.1/3064
4080_4008	LCDIF General Control Register (LCDIF_CTRL_CLR)	32	R/W	C000_0000h	47.6.1/3064
4080_400C	LCDIF General Control Register (LCDIF_CTRL_TOG)	32	R/W	C000_0000h	47.6.1/3064
4080_4010	LCDIF General Control1 Register (LCDIF_CTRL1)	32	R/W	000F_0000h	47.6.2/3066
4080_4014	LCDIF General Control1 Register (LCDIF_CTRL1_SET)	32	R/W	000F_0000h	47.6.2/3066
4080_4018	LCDIF General Control1 Register (LCDIF_CTRL1_CLR)	32	R/W	000F_0000h	47.6.2/3066
4080_401C	LCDIF General Control1 Register (LCDIF_CTRL1_TOG)	32	R/W	000F_0000h	47.6.2/3066
4080_4020	LCDIF General Control2 Register (LCDIF_CTRL2)	32	R/W	0020_0000h	47.6.3/3069
4080_4024	LCDIF General Control2 Register (LCDIF_CTRL2_SET)	32	R/W	0020_0000h	47.6.3/3069
4080_4028	LCDIF General Control2 Register (LCDIF_CTRL2_CLR)	32	R/W	0020_0000h	47.6.3/3069
4080_402C	LCDIF General Control2 Register (LCDIF_CTRL2_TOG)	32	R/W	0020_0000h	47.6.3/3069
4080_4030	LCDIF Horizontal and Vertical Valid Data Count Register (LCDIF_TRANSFER_COUNT)	32	R/W	0001_0000h	47.6.4/3071
4080_4040	LCD Interface Current Buffer Address Register (LCDIF_CUR_BUF)	32	R/W	0000_0000h	47.6.5/3071
4080_4050	LCD Interface Next Buffer Address Register (LCDIF_NEXT_BUF)	32	R/W	0000_0000h	47.6.6/3072
4080_4070	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0)	32	R/W	0000_0000h	47.6.7/3072
4080_4074	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_SET)	32	R/W	0000_0000h	47.6.7/3072
4080_4078	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_CLR)	32	R/W	0000_0000h	47.6.7/3072
4080_407C	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_TOG)	32	R/W	0000_0000h	47.6.7/3072
4080_4080	LCDIF VSYNC Mode and Dotclk Mode Control Register1 (LCDIF_VDCTRL1)	32	R/W	0000_0000h	47.6.8/3074
4080_4090	LCDIF VSYNC Mode and Dotclk Mode Control Register2 (LCDIF_VDCTRL2)	32	R/W	0000_0000h	47.6.9/3074
4080_40A0	LCDIF VSYNC Mode and Dotclk Mode Control Register3 (LCDIF_VDCTRL3)	32	R/W	0000_0000h	47.6.10/ 3075
4080_40B0	LCDIF VSYNC Mode and Dotclk Mode Control Register4 (LCDIF_VDCTRL4)	32	R/W	0000_0000h	47.6.11/ 3076
4080_4190	Bus Master Error Status Register (LCDIF_BM_ERROR_STAT)	32	R/W	0000_0000h	47.6.12/ 3077
4080_41A0	CRC Status Register (LCDIF_CRC_STAT)	32	R/W	0000_0000h	47.6.13/ 3077
4080_41B0	LCD Interface Status Register (LCDIF_STAT)	32	R	9500_0000h	47.6.14/ 3078
4080_4200	LCDIF Threshold Register (LCDIF_THRES)	32	R/W	0100_000Fh	47.6.15/ 3080
4080_4380	LCDIF Pigeon Mode Control0 Register (LCDIF_PIGEONCTRL0)	32	R/W	0000_0000h	47.6.16/ 3081

Table continues on the next page...

LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_4384	LCDIF Pigeon Mode Control0 Register (LCDIF_PIGEONCTRL0_SET)	32	R/W	0000_0000h	47.6.16/ 3081
4080_4388	LCDIF Pigeon Mode Control0 Register (LCDIF_PIGEONCTRL0_CLR)	32	R/W	0000_0000h	47.6.16/ 3081
4080_438C	LCDIF Pigeon Mode Control0 Register (LCDIF_PIGEONCTRL0_TOG)	32	R/W	0000_0000h	47.6.16/ 3081
4080_4390	LCDIF Pigeon Mode Control1 Register (LCDIF_PIGEONCTRL1)	32	R/W	0000_0000h	47.6.17/ 3081
4080_4394	LCDIF Pigeon Mode Control1 Register (LCDIF_PIGEONCTRL1_SET)	32	R/W	0000_0000h	47.6.17/ 3081
4080_4398	LCDIF Pigeon Mode Control1 Register (LCDIF_PIGEONCTRL1_CLR)	32	R/W	0000_0000h	47.6.17/ 3081
4080_439C	LCDIF Pigeon Mode Control1 Register (LCDIF_PIGEONCTRL1_TOG)	32	R/W	0000_0000h	47.6.17/ 3081
4080_43A0	LCDIF Pigeon Mode Control2 Register (LCDIF_PIGEONCTRL2)	32	R/W	0000_0000h	47.6.18/ 3082
4080_43A4	LCDIF Pigeon Mode Control2 Register (LCDIF_PIGEONCTRL2_SET)	32	R/W	0000_0000h	47.6.18/ 3082
4080_43A8	LCDIF Pigeon Mode Control2 Register (LCDIF_PIGEONCTRL2_CLR)	32	R/W	0000_0000h	47.6.18/ 3082
4080_43AC	LCDIF Pigeon Mode Control2 Register (LCDIF_PIGEONCTRL2_TOG)	32	R/W	0000_0000h	47.6.18/ 3082
4080_4800	Panel Interface Signal Generator Register (LCDIF_PIGEON_0_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4810	Panel Interface Signal Generator Register (LCDIF_PIGEON_0_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4820	Panel Interface Signal Generator Register (LCDIF_PIGEON_0_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4840	Panel Interface Signal Generator Register (LCDIF_PIGEON_1_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4850	Panel Interface Signal Generator Register (LCDIF_PIGEON_1_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4860	Panel Interface Signal Generator Register (LCDIF_PIGEON_1_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4880	Panel Interface Signal Generator Register (LCDIF_PIGEON_2_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4890	Panel Interface Signal Generator Register (LCDIF_PIGEON_2_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_48A0	Panel Interface Signal Generator Register (LCDIF_PIGEON_2_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_48C0	Panel Interface Signal Generator Register (LCDIF_PIGEON_3_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_48D0	Panel Interface Signal Generator Register (LCDIF_PIGEON_3_1)	32	R/W	0000_0000h	47.6.20/ 3084

Table continues on the next page...

LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_48E0	Panel Interface Signal Generator Register (LCDIF_PIGEON_3_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4900	Panel Interface Signal Generator Register (LCDIF_PIGEON_4_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4910	Panel Interface Signal Generator Register (LCDIF_PIGEON_4_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4920	Panel Interface Signal Generator Register (LCDIF_PIGEON_4_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4940	Panel Interface Signal Generator Register (LCDIF_PIGEON_5_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4950	Panel Interface Signal Generator Register (LCDIF_PIGEON_5_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4960	Panel Interface Signal Generator Register (LCDIF_PIGEON_5_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4980	Panel Interface Signal Generator Register (LCDIF_PIGEON_6_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4990	Panel Interface Signal Generator Register (LCDIF_PIGEON_6_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_49A0	Panel Interface Signal Generator Register (LCDIF_PIGEON_6_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_49C0	Panel Interface Signal Generator Register (LCDIF_PIGEON_7_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_49D0	Panel Interface Signal Generator Register (LCDIF_PIGEON_7_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_49E0	Panel Interface Signal Generator Register (LCDIF_PIGEON_7_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4A00	Panel Interface Signal Generator Register (LCDIF_PIGEON_8_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4A10	Panel Interface Signal Generator Register (LCDIF_PIGEON_8_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4A20	Panel Interface Signal Generator Register (LCDIF_PIGEON_8_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4A40	Panel Interface Signal Generator Register (LCDIF_PIGEON_9_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4A50	Panel Interface Signal Generator Register (LCDIF_PIGEON_9_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4A60	Panel Interface Signal Generator Register (LCDIF_PIGEON_9_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4A80	Panel Interface Signal Generator Register (LCDIF_PIGEON_10_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4A90	Panel Interface Signal Generator Register (LCDIF_PIGEON_10_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4AA0	Panel Interface Signal Generator Register (LCDIF_PIGEON_10_2)	32	R/W	0000_0000h	47.6.21/ 3085

Table continues on the next page...

LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4080_4AC0	Panel Interface Signal Generator Register (LCDIF_PIGEON_11_0)	32	R/W	0000_0000h	47.6.19/ 3083
4080_4AD0	Panel Interface Signal Generator Register (LCDIF_PIGEON_11_1)	32	R/W	0000_0000h	47.6.20/ 3084
4080_4AE0	Panel Interface Signal Generator Register (LCDIF_PIGEON_11_2)	32	R/W	0000_0000h	47.6.21/ 3085
4080_4B00	Look Up Table Control Register (LCDIF_LUT_CTRL)	32	R/W	0000_0001h	47.6.22/ 3086
4080_4B10	Lookup Table 0 Index Register (LCDIF_LUT0_ADDR)	32	R/W	0000_0000h	47.6.23/ 3086
4080_4B20	Lookup Table 0 Data Register (LCDIF_LUT0_DATA)	32	R/W	0000_0000h	47.6.24/ 3087
4080_4B30	Lookup Table 1 Index Register (LCDIF_LUT1_ADDR)	32	R/W	0000_0000h	47.6.25/ 3087
4080_4B40	Lookup Table 1 Data Register (LCDIF_LUT1_DATA)	32	R/W	0000_0000h	47.6.26/ 3088

47.6.1 LCDIF General Control Register (LCDIF_CTRL*n*)

The LCD Interface Control Register provides overall control of the LCDIF block. The LCDIF Control Register provides a variety of control functions to the programmer. These functions allow the interface to be very flexible to work with a variety of LCD controllers, and to minimize overhead and increase performance of LCD programming. The register has been organized such that switching between the different LCD modes can be done with minimum PIO writes.

Address: 4080_4000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFTRST	CLKGATE	-	Reserved	Reserved	DATA_SHIFT_DIR	SHIFT_NUM_BITS					Reserved	BYPASS_COUNT	Reserved	DOTCLK_MODE	Reserved
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INPUT_DATA_SWIZZLE		CSC_DATA_SWIZZLE		LCD_DATABUS_WIDTH		WORD_LENGTH		Reserved	ENABLE_PXP_HANDSHAKE	MASTER	Reserved	DATA_FORMAT_16_BIT	DATA_FORMAT_18_BIT	DATA_FORMAT_24_BIT	RUN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CTRL*n* field descriptions

Field	Description
31 SFTRST	This bit must be set to zero to enable normal operation of the LCDIF. When set to one, it forces a block level reset.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 -	Reserved
28 -	This field is reserved. Reserved

Table continues on the next page...

LCDIF_CTRL*n* field descriptions (continued)

Field	Description
27 - Reserved	This field is reserved.
26 DATA_SHIFT_DIR	Use this bit to determine the direction of shift of transmit data. 0x0 **TXDATA_SHIFT_LEFT** — Data to be transmitted is shifted LEFT by SHIFT_NUM_BITS bits. 0x1 **TXDATA_SHIFT_RIGHT** — Data to be transmitted is shifted RIGHT by SHIFT_NUM_BITS bits.
25–21 SHIFT_NUM_BITS	The data to be transmitted is shifted left or right by this number of bits.
20 - Reserved	This field is reserved.
19 BYPASS_COUNT	When this bit is 0, it means that LCDIF will stop the block operation and turn off the RUN bit after the amount of data indicated by the LCDIF_TRANSFER_COUNT register has been transferred out. When this bit is set to 1, the block will continue normal operation indefinitely until it is told to stop. This bit must be 0 in MPU and VSYNC modes, and must be 1 in DOTCLK and DVI modes of operation.
18 - Reserved	This field is reserved.
17 DOTCLK_MODE	Set this bit to 1 to make the hardware go into the DOTCLK mode, i.e. VSYNC/HSYNC/DOTCLK/ENABLE interface mode. ENABLE is optional, selected by the ENABLE_PRESENT bit. Toggle this bit from 1 to 0 to make the hardware go out of DOTCLK mode after completing all data transfer and deasserting the RUN bit.
16 - Reserved	This field is reserved.
15–14 INPUT_DATA_SWIZZLE	This field specifies how to swap the bytes fetched by the bus master interface. The swizzle function is independent of the WORD_LENGTH bit. The supported swizzle configurations are: 0x0 **NO_SWAP** — No byte swapping.(Little endian) 0x0 **LITTLE_ENDIAN** — Little Endian byte ordering (same as NO_SWAP). 0x1 **BIG_ENDIAN_SWAP** — Big Endian swap (swap bytes 0,3 and 1,2). 0x1 **SWAP_ALL_BYTES** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
13–12 CSC_DATA_SWIZZLE	This field specifies how to swap the bytes after the data has been converted into an internal representation of 24 bits per pixel and before it is transmitted over the LCD interface bus. The data is always transmitted with the least significant byte/hword (half word) first after the swizzle takes place. So, INPUT_DATA_SWIZZLE takes place first on the incoming data, and then CSC_DATA_SWIZZLE is applied. The swizzle function is independent of the WORD_LENGTH or the LCD_DATABUS_WIDTH fields. If RGB_TO_YCRCB422_CSC bit is set, the swizzle occurs on the Y, Cb, Cr values. The supported swizzle configurations are: 0x0 **NO_SWAP** — No byte swapping.(Little endian) 0x0 **LITTLE_ENDIAN** — Little Endian byte ordering (same as NO_SWAP). 0x1 **BIG_ENDIAN_SWAP** — Big Endian swap (swap bytes 0,3 and 1,2). 0x1 **SWAP_ALL_BYTES** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
11–10 LCD_DATABUS_WIDTH	LCD Data bus transfer width. **NOTE:** When LUT enabled, this field should be set to 0x01.

Table continues on the next page...

LCDIF_CTRL*n* field descriptions (continued)

Field	Description
	0x0 **16_BIT** — 16-bit data bus mode. 0x1 **8_BIT** — 8-bit data bus mode. 0x2 **18_BIT** — 18-bit data bus mode. 0x3 **24_BIT** — 24-bit data bus mode.
9–8 WORD_LENGTH	Input data format. 0x0 **16_BIT** — Input data is 16 bits per pixel. 0x1 **8_BIT** — Input data is 8 bits wide. 0x2 **18_BIT** — Input data is 18 bits per pixel. 0x3 **24_BIT** — Input data is 24 bits per pixel.
7 -	This field is reserved. Reserved
6 ENABLE_PXP_ HANDSHAKE	If this bit is set and LCDIF_MASTER bit is set, the LCDIF will act as bus master and the handshake mechanism between LCDIF and PXP will be turned on. If LCDIF_MASTER bit is not set, this bit becomes a don't care.
5 MASTER	Set this bit to make the LCDIF act as a bus master. If this bit is reset, the LCDIF will support APBDMA or PIO mode.
4 RSRVD0	This field is reserved. Reserved bits. Write as 0.
3 DATA_ FORMAT_16_ BIT	When this bit is 1 and WORD_LENGTH = 0, it implies that the 16-bit data is in ARGB555 format. When this bit is 0 and WORD_LENGTH = 0, it implies that the 16-bit data is in RGB565 format. When WORD_LENGTH is not 0, this bit does not care.
2 DATA_ FORMAT_18_ BIT	Used only when WORD_LENGTH = 2, i.e. 18-bit. 0x0 **LOWER_18_BITS_VALID** — Data input to the block is in 18 bpp format, such that lower 18 bits contain RGB 666 and upper 14 bits do not contain any useful data. 0x1 **UPPER_18_BITS_VALID** — Data input to the block is in 18 bpp format, such that upper 18 bits contain RGB 666 and lower 14 bits do not contain any useful data.
1 DATA_ FORMAT_24_ BIT	Used only when WORD_LENGTH = 3, i.e. 24-bit. Note that this applies to both packed and unpacked 24-bit data. 0x0 **ALL_24_BITS_VALID** — Data input to the block is in 24 bpp format, such that all RGB 888 data is contained in 24 bits. 0x1 **DROP_UPPER_2_BITS_PER_BYTE** — Data input to the block is actually RGB 18 bpp, but there is 1 color per byte, hence the upper 2 bits in each byte do not contain any useful data, and should be dropped.
0 RUN	When this bit is set by software, the LCDIF will begin transferring data between the SoC and the display. This bit must remain set until the operation is complete.

47.6.2 LCDIF General Control1 Register (LCDIF_CTRL1*n*)

The LCDIF Control Register provides overall control of the LCDIF block.

The LCDIF Control1 Register provides additional programming to the LCDIF. It implements some bits which are unlikely to change often in a particular application. It also carries interrupt-related bits which are common across more than one mode of operation.

Address: 4080_4000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMAGE_DATA_SELECT	CS_OUT_SELECT	Reserved		Reserved	BM_ERROR_IRQ_EN	BM_ERROR_IRQ	RECOVER_ON_UNDERFLOW	INTERLACE_FIELDS	START_INTERLACE_FROM_SECOND_FIELD	FIFO_CLEAR	IRQ_ON_ALTERNATE_FIELDS	BYTE_PACKING_FORMAT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OVERFLOW_IRQ_EN	UNDERFLOW_IRQ_EN	CUR_FRAME_DONE_IRQ_EN	VSYNC_EDGE_IRQ_EN	OVERFLOW_IRQ	UNDERFLOW_IRQ	CUR_FRAME_DONE_IRQ	VSYNC_EDGE_IRQ	Reserved					Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CTRL1*n* field descriptions

Field	Description
31 IMAGE_DATA_SELECT	Command Mode MIPI image data select bit. This bit will control MIPI DSI SYS_ADDR[1]. This bit should only be changed when RUN is 0.
30 CS_OUT_SELECT	This bit is CS0/CS1 valid select signals. When set 0x0, LCDIF_CSn0 output valid, and LCSIF_CSn1 will alway output 1. WHne set to 0x1, LCDIF_CSn1 output valid, and LCSIF_CSn0 will alway output 1.
29–28 -	This field is reserved. Reserved bits. Write as 0.
27 -	This field is reserved. Reserved.
26 BM_ERROR_IRQ_EN	This bit is set to enable bus master error interrupt in the LCDIF master mode.
25 BM_ERROR_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. This bit will be set when the LCDIF is in master mode and an error response was returned by the slave.

Table continues on the next page...

LCDIF_CTRL1*n* field descriptions (continued)

Field	Description
	0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
24 RECOVER_ON_ UNDERFLOW	Set this bit to enable the LCDIF block to recover in the next field/frame if there was an underflow in the current field/frame.
23 INTERLACE_ FIELDS	Set this bit if it is required that the LCDIF block fetches odd lines in one field and even lines in the other field. It will work only in LCDIF_MASTER is set to 1.
22 START_ INTERLACE_ FROM_ SECOND_FIELD	The default is to grab the odd lines first and then the even lines. Set this bit if it is required to grab the even lines first and then the odd lines. (Line numbers start from 1, so odd lines are 1,3,5,etc. and even lines are 2,4,6, etc.)
21 FIFO_CLEAR	Set this bit to clear all the data in the latency FIFO (LFIFO), TXFIFO and the RXFIFO.
20 IRQ_ON_ ALTERNATE_ FIELDS	If this bit is set, the LCDIF block will assert the cur_frame_done interrupt only on alternate fields, otherwise it will issue the interrupt on both odd and even field. This bit is mostly relevant if INTERLACE_FIELDS is set.
19–16 BYTE_ PACKING_ FORMAT	This bitfield is used to show which data bytes in a 32-bit word are valid. Default value 0xf indicates that all bytes are valid. For 8-bit transfers, any combination in this bitfield will mean valid data is present in the corresponding bytes. In the 16-bit mode, a 16-bit half-word is valid only if adjacent bits [1:0] or [3:2] or both are 1. A value of 0x0 will mean that none of the bytes are valid and should not be used. For example, set the bit field value to 0x7 if the display data is arranged in the 24-bit unpacked format (A-R-G-B where A value does not have be transmitted).
15 OVERFLOW_ IRQ_EN	This bit is set to enable an overflow interrupt in the TXFIFO in the write mode.
14 UNDERFLOW_ IRQ_EN	This bit is set to enable an underflow interrupt in the TXFIFO in the write mode.
13 CUR_FRAME_ DONE_IRQ_EN	This bit is set to 1 enable an interrupt every time the hardware enters in the vertical blanking state.
12 VSYNC_EDGE_ IRQ_EN	This bit is set to enable an interrupt every time the hardware encounters the leading VSYNC edge in the VSYNC and DOTCLK modes, or the beginning of every field in DVI mode.
11 OVERFLOW_ IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. A latency FIFO (LFIFO) overflow in the write mode (MPU/VSYNC/DOTCLK/DVI mode) was detected, data samples have been lost. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
10 UNDERFLOW_ IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. A TXFIFO underflow in the write mode (MPU/VSYNC/DOTCLK/DVI mode) was detected. Could produce an error in the DOTCLK / DVI modes. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.

Table continues on the next page...

LCDIF_CTRL1*n* field descriptions (continued)

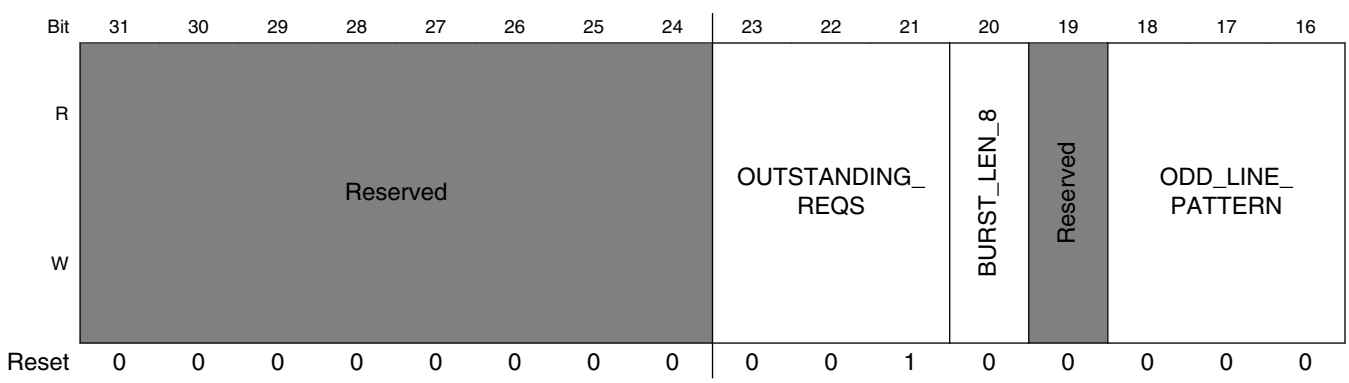
Field	Description
9 CUR_FRAME_DONE_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. It indicates that the hardware has completed transmitting the current frame and is in the vertical blanking period in the DOTCLK/DVI modes. In the MPU and VSYNC modes, this IRQ is asserted at the end of the data transfer indicated by LCDIF_TRANSFER_COUNT register. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
8 VSYNC_EDGE_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. It is set whenever the leading VSYNC edge is detected in the VSYNC and DOTCLK modes. In the DVI mode, it is asserted every time the block enters a new field. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
7–3 RSRVD0	This field is reserved. Reserved bits. Write as 0.
2 -	This field is reserved. Reserved
1 -	This field is reserved. Reserved.
0 -	This field is reserved. Reserved.

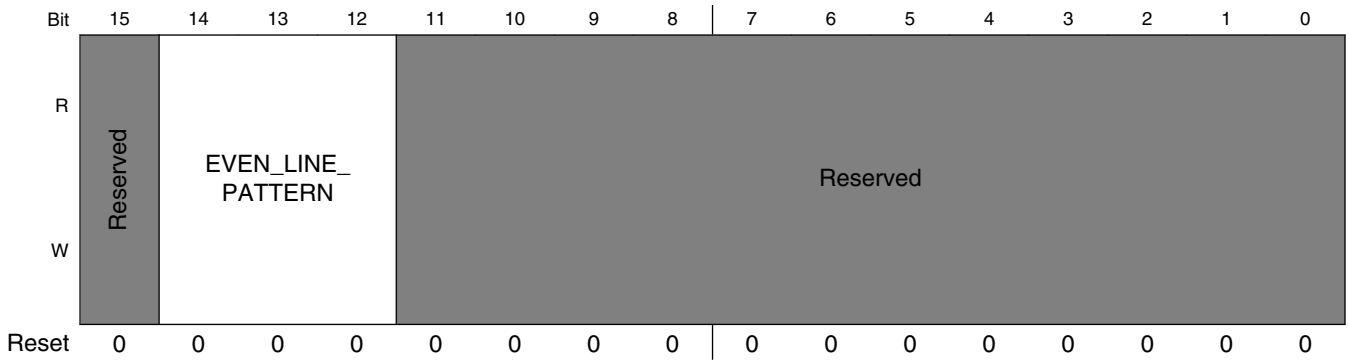
47.6.3 LCDIF General Control2 Register (LCDIF_CTRL2*n*)

The LCDIF Control Register provides overall control of the LCDIF block.

The LCDIF Control2 Register provides additional programming to the LCDIF. It implements some bits which are unlikely to change often in a particular application.

Address: 4080_4000h base + 20h offset + (4d × i), where i=0d to 3d





LCDIF_CTRL2*n* field descriptions

Field	Description
31–24 RSRVD5	This field is reserved. Reserved bits. Write as 0.
23–21 OUTSTANDING_ REQS	This bitfield indicates the maximum number of outstanding transactions that LCDIF should request when it is acting as a bus master. Default is 2 outstanding transactions. 0x0 **REQ_1** — 0x1 **REQ_2** — 0x2 **REQ_4** — 0x3 **REQ_8** — 0x4 **REQ_16** —
20 BURST_LEN_8	By default, when the LCDIF is in the bus master mode, it will issue AXI bursts of length 16 (except when in packed 24 bpp mode, it will issue bursts of length 15). When this bit is set to 1, the block will issue bursts of length 8 (except when in packed 24 bpp mode, it will issue bursts of length 9). Note that this bitfield is only applicable when LCDIF_MASTER is set to 1.
19 RSRVD4	This field is reserved. Reserved bits. Write as 0.
18–16 ODD_LINE_ PATTERN	This field determines the order of the RGB components of each pixel in ODD lines (line numbers 1,3,5,..). 0x0 **RGB** — 0x1 **RBG** — 0x2 **GBR** — 0x3 **GRB** — 0x4 **BRG** — 0x5 **BGR** —
15 RSRVD3	This field is reserved. Reserved bits. Write as 0.
14–12 EVEN_LINE_ PATTERN	This field determines the order of the RGB components of each pixel in EVEN lines (line numbers 2,4,6,..). 0x0 **RGB** — 0x1 **RBG** — 0x2 **GBR** — 0x3 **GRB** — 0x4 **BRG** — 0x5 **BGR** —

Table continues on the next page...

LCDIF_CTRL2*n* field descriptions (continued)

Field	Description
RSRVD0	This field is reserved. Reserved bits. Write as 0.

47.6.4 LCDIF Horizontal and Vertical Valid Data Count Register (LCDIF_TRANSFER_COUNT)

This register tells the LCDIF how much data will be sent for this frame, or transaction. The total number of words is a product of the V_COUNT and H_COUNT fields. The word size is specified by the WORD_LENGTH field.

This register gives the dimensions of the input frame. For normal operation, but V_COUNT and H_COUNT should be non-zero.

Address: 4080_4000h base + 30h offset = 4080_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V_COUNT																H_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_TRANSFER_COUNT field descriptions

Field	Description
31–16 V_COUNT	Number of horizontal lines per frame which contain valid data. In DOTCLK mode, V_COUNT should be the same as the number of active horizontal lines in a progressive frame.
H_COUNT	Total valid data (pixels) in each horizontal line. The data size is given by the WORD_LENGTH. In 24-bit packed format (WORD_LENGTH=0x3, BYTE_PACKING_FORMAT=0xF), the H_COUNT must be a multiple of 4 pixels. In 16-bit packed format (WORD_LENGTH=0x0, BYTE_PACKING_FORMAT=0xF), the H_COUNT must be a multiple of 2 pixels.

47.6.5 LCD Interface Current Buffer Address Register (LCDIF_CUR_BUF)

This register indicates the address of the current frame being transmitted by LCDIF.

When the LCDIF is behaving as a master, this address points to the address of the current frame of data being sent out via the LCDIF. When the current frame is done, the LCDIF block will assert the cur_frame_done interrupt for software to take action. The block will also copy the LCDIF_NEXT_BUF_ADDR into this bitfield so that the software can program the next frame address into the LCDIF_NEXT_BUF_ADDR bitfield. This address must always be double-word aligned.

Address: 4080_4000h base + 40h offset = 4080_4040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CUR_BUF field descriptions

Field	Description
ADDR	Address of the current frame being transmitted by LCDIF.

47.6.6 LCD Interface Next Buffer Address Register (LCDIF_NEXT_BUF)

This register indicates the address of next frame that will be transmitted by LCDIF.

When the LCDIF is behaving as a master, this address points to the address of the next frame of data that will be sent out via the LCDIF. It is up to the software to make sure that this register is programmed before the end of the current frame, otherwise it might result in old data going out the LCDIF. This address must always be double-word aligned.

Address: 4080_4000h base + 50h offset = 4080_4050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_NEXT_BUF field descriptions

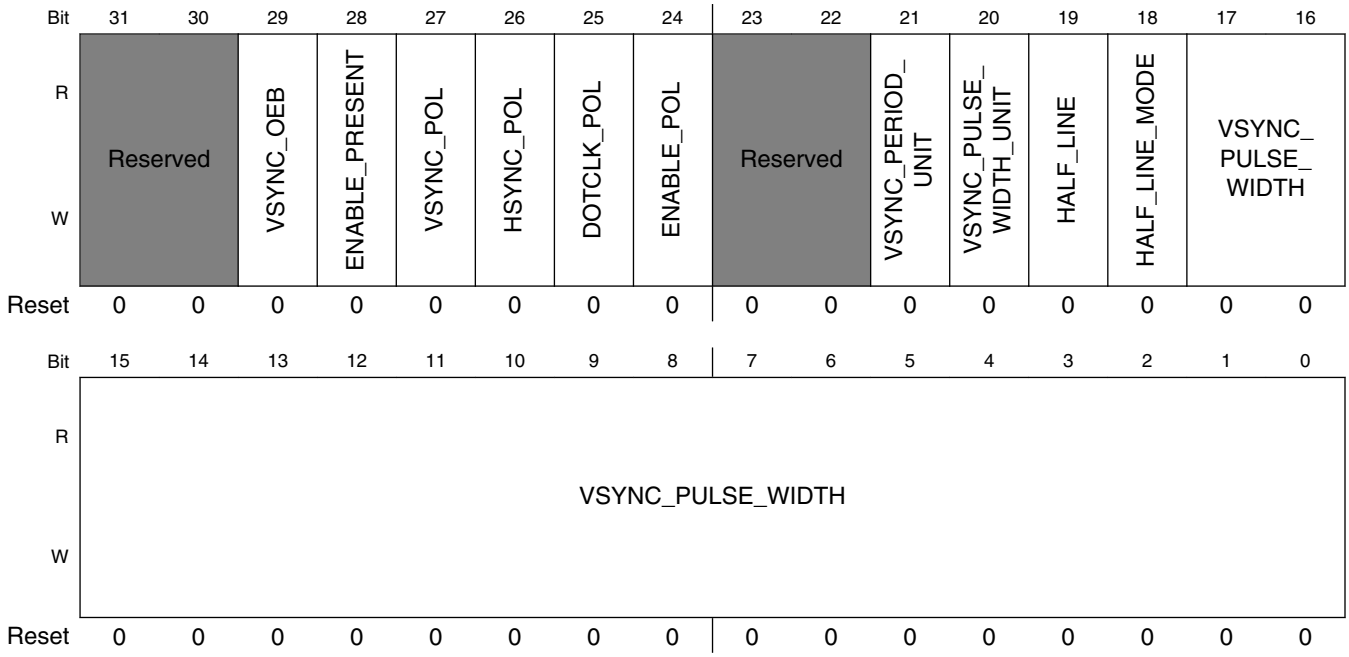
Field	Description
ADDR	Address of the next frame that will be transmitted by LCDIF.

47.6.7 LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0*n*)

This register is used to control the VSYNC and DOTCLK modes of the LCDIF so as to work with different types of LCDs like moving picture displays and delta pixel displays.

This register gives general programmability to the VSYNC signal including polarity, direction, pulse width, etc.

Address: 4080_4000h base + 70h offset + (4d × i), where i=0d to 3d



LCDIF_VDCTRL0*n* field descriptions

Field	Description
31–30 RSRVD2	This field is reserved. Reserved bits. Write as 0.
29 VSYNC_OEB	0 means the VSYNC signal is an output, 1 means it is an input. Should be set to 0 in the DOTCLK mode. 0x0 **VSYNC_OUTPUT** — The VSYNC pin is in the output mode and the VSYNC signal has to be generated by the LCDIF block. 0x1 **VSYNC_INPUT** — The VSYNC pin is in the input mode and the LCD controller sends the VSYNC signal to the block.
28 ENABLE_ PRESENT	Setting this bit to 1 will make the hardware generate the ENABLE signal in the DOTCLK mode, thereby making it the true RGB interface along with the remaining three signals VSYNC, HSYNC and DOTCLK.
27 VSYNC_POL	Default 0 active low during VSYNC_PULSE_WIDTH time and will be high during the rest of the VSYNC period. Set it to 1 to invert the polarity.
26 HSYNC_POL	Default 0 active low during HSYNC_PULSE_WIDTH time and will be high during the rest of the HSYNC period. Set it to 1 to invert the polarity.
25 DOTCLK_POL	Default is data launched at negative edge of DOTCLK and captured at positive edge. Set it to 1 to invert the polarity. Set it to 0 in DVI mode.
24 ENABLE_POL	Default 0 active low during valid data transfer on each horizontal line.
23–22 RSRVD1	This field is reserved. Reserved bits. Write as 0.
21 VSYNC_ PERIOD_UNIT	Default 0 for counting VSYNC_PERIOD in terms of DISPLAY CLOCK (pix_clk) cycles. Set it to 1 to count in terms of complete horizontal lines. DISPLAY CLOCK (pix_clk) cycles should be used in the VSYNC mode, while horizontal line should be used in the DOTCLK mode.
20 VSYNC_PULSE_ WIDTH_UNIT	Default 0 for counting VSYNC_PULSE_WIDTH in terms of DISPLAY CLOCK (pix_clk) cycles. Set it to 1 to count in terms of complete horizontal lines.

Table continues on the next page...

LCDIF_VDCTRL0*n* field descriptions (continued)

Field	Description
19 HALF_LINE	Setting this bit to 1 will make the total VSYNC period equal to the VSYNC_PERIOD field plus half the HORIZONTAL_PERIOD field (i.e. VSYNC_PERIOD field plus half horizontal line), otherwise it is just VSYNC_PERIOD. Should be only used in the DOTCLK mode, not in the VSYNC interface mode.
18 HALF_LINE_ MODE	When this bit is 0, the first field (VSYNC period) will end in half a horizontal line and the second field will begin with half a horizontal line. When this bit is 1, all fields will end with half a horizontal line, and none will begin with half a horizontal line.
VSYNC_PULSE_ WIDTH	Number of units for which VSYNC signal is active. For the DOTCLK mode, the unit is determined by the VSYNC_PULSE_WIDTH_UNIT. If the VSYNC_PULSE_WIDTH_UNIT is 0 for DOTCLK mode, VSYNC_PULSE_WIDTH must be less than HSYNC_PERIOD. For the VSYNC interface mode, it should be in terms of number of DISPLAY CLOCK (pix_clk) cycles only.

47.6.8 LCDIF VSYNC Mode and Dotclk Mode Control Register1 (LCDIF_VDCTRL1)

This register is used to control the VSYNC signal in the VSYNC and DOTCLK modes of the block.

This register determines the period and duty cycle of the VSYNC signal when it is generated in the block.

Address: 4080_4000h base + 80h offset = 4080_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VSYNC_PERIOD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL1 field descriptions

Field	Description
VSYNC_PERIOD	Total number of units between two positive or two negative edges of the VSYNC signal. If HALF_LINE is set, it is implicitly calculated to be VSYNC_PERIOD plus half HSYNC_PERIOD.

47.6.9 LCDIF VSYNC Mode and Dotclk Mode Control Register2 (LCDIF_VDCTRL2)

This register is used to control the HSYNC signal in the DOTCLK mode of the block.

This register determines the period and duty cycle of the HSYNC signal when it is generated in the block.

Address: 4080_4000h base + 90h offset = 4080_4090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSYNC_PULSE_WIDTH																HSYNC_PERIOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL2 field descriptions

Field	Description
31–18 HSYNC_PULSE_ WIDTH	Number of DISPLAY CLOCK (pix_clk) cycles for which HSYNC signal is active.
HSYNC_PERIOD	Total number of DISPLAY CLOCK (pix_clk) cycles between two positive or two negative edges of the HSYNC signal.

47.6.10 LCDIF VSYNC Mode and Dotclk Mode Control Register3 (LCDIF_VDCTRL3)

This register is used to determine the vertical and horizontal wait counts.

This register determines the back porches of HSYNC and VSYNC signals when they are generated by the block.

Address: 4080_4000h base + A0h offset = 4080_40A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		MUX_SYNC_ SIGNALS	VSYNC_ONLY	HORIZONTAL_WAIT_CNT											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VERTICAL_WAIT_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL3 field descriptions

Field	Description
31–30 RSRVD0	This field is reserved. Reserved bits, write as 0.

Table continues on the next page...

LCDIF_VDCTRL3 field descriptions (continued)

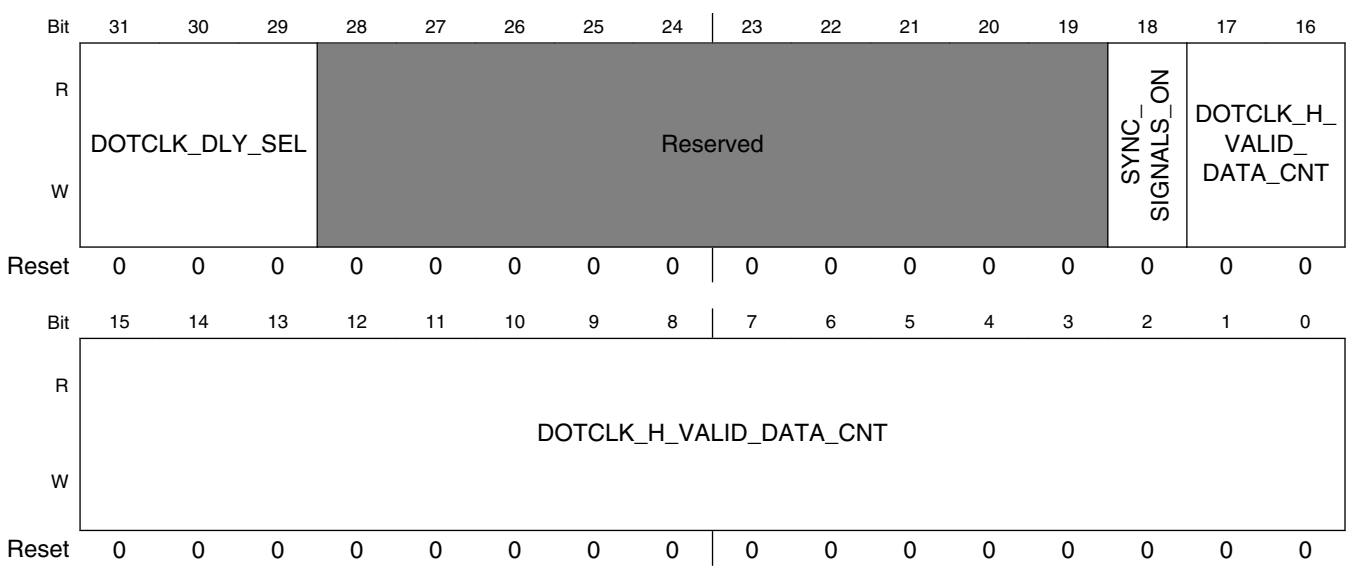
Field	Description
29 MUX_SYNC_ SIGNALS	When this bit is set, the LCDIF block will internally mux HSYNC with LCD_D14, DOTCLK with LCD_D13 and ENABLE with LCD_D12, otherwise these signals will go out on separate pins. This feature can be used to maintain backward compatible with 37xx.
28 VSYNC_ONLY	This bit must be set to 1 in the VSYNC mode of operation, and 0 in the DOTCLK mode of operation.
27–16 HORIZONTAL_ WAIT_CNT	In the DOTCLK mode, wait for this number of clocks from falling edge (or rising if HSYNC_POL is 1) of HSYNC signal to account for horizontal back porch plus the number of DOTCLKs before the moving picture information begins.
VERTICAL_ WAIT_CNT	In the VSYNC interface mode, wait for this number of DISPLAY CLOCK (pix_clk) cycles from the falling VSYNC edge (or rising if VSYNC_POL is 1) before starting LCD transactions and is applicable only if WAIT_FOR_VSYNC_EDGE is set. Minimum is CMD_SETUP+5. In the DOTCLK mode, it accounts for the vertical back porch lines plus the number of horizontal lines before the moving picture begins. The unit for this parameter is inherently the same as the VSYNC_PERIOD_UNIT.

47.6.11 LCDIF VSYNC Mode and Dotclk Mode Control Register4 (LCDIF_VDCTRL4)

This register is used to control the DOTCLK mode of the block.

This register determines the active data in each horizontal line in the DOTCLK mode. Note that the total number of active horizontal lines in the DOTCLK mode is the same as the V_COUNT bitfield in the LCDIF_TRANSFER_COUNT register.

Address: 4080_4000h base + B0h offset = 4080_40B0h



LCDIF_VDCTRL4 field descriptions

Field	Description
31–29 DOTCLK_DLY_SEL	This bitfield selects the amount of time by which the DOTCLK signal should be delayed before coming out of the LCD_DOTCK pin. 0 = 2ns; 1=4ns;2=6ns;3=8ns. Remaining values are reserved.
28–19 RSRVD0	This field is reserved. Reserved bits, write as 0.
18 SYNC_SIGNALS_ON	Set this field to 1 if the LCD controller requires that the VSYNC or VSYNC/HSYNC/DOTCLK control signals should be active at least one frame before the data transfers actually start and remain active at least one frame after the data transfers end. The hardware does not count the number of frames automatically. Rather, the VSYNC edge interrupt can be monitored by software to count the number of frames that have occurred after this bit is set and then the RUN bit can be set to start the data transactions. This bit must always be set in the DOTCLK mode of operation, and it must be set in the VSYNC mode of operation when VSYNC signal is an output.
DOTCLK_H_VALID_DATA_CNT	Total number of DISPLAY CLOCK (pix_clk) cycles on each horizontal line that carry valid data in DOTCLK mode.

47.6.12 Bus Master Error Status Register (LCDIF_BM_ERROR_STAT)

This register reflects the virtual address at which the AXI master received an error response from the slave.

When the BM_ERROR_IRQ is asserted, the address of the bus error is updated in the register.

Address: 4080_4000h base + 190h offset = 4080_4190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_BM_ERROR_STAT field descriptions

Field	Description
ADDR	Virtual address at which bus master error occurred.

47.6.13 CRC Status Register (LCDIF_CRC_STAT)

This register reflects the CRC value of each frame sent out by LCDIF. The CRC is done on the final output bus, so the value will be dependent on the LCD_DATABUS_WIDTH bitfield even if the input data is the same.

This register will be updated when the CUR_FRAME_DONE_IRQ is asserted.

Address: 4080_4000h base + 1A0h offset = 4080_41A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC_VALUE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CRC_STAT field descriptions

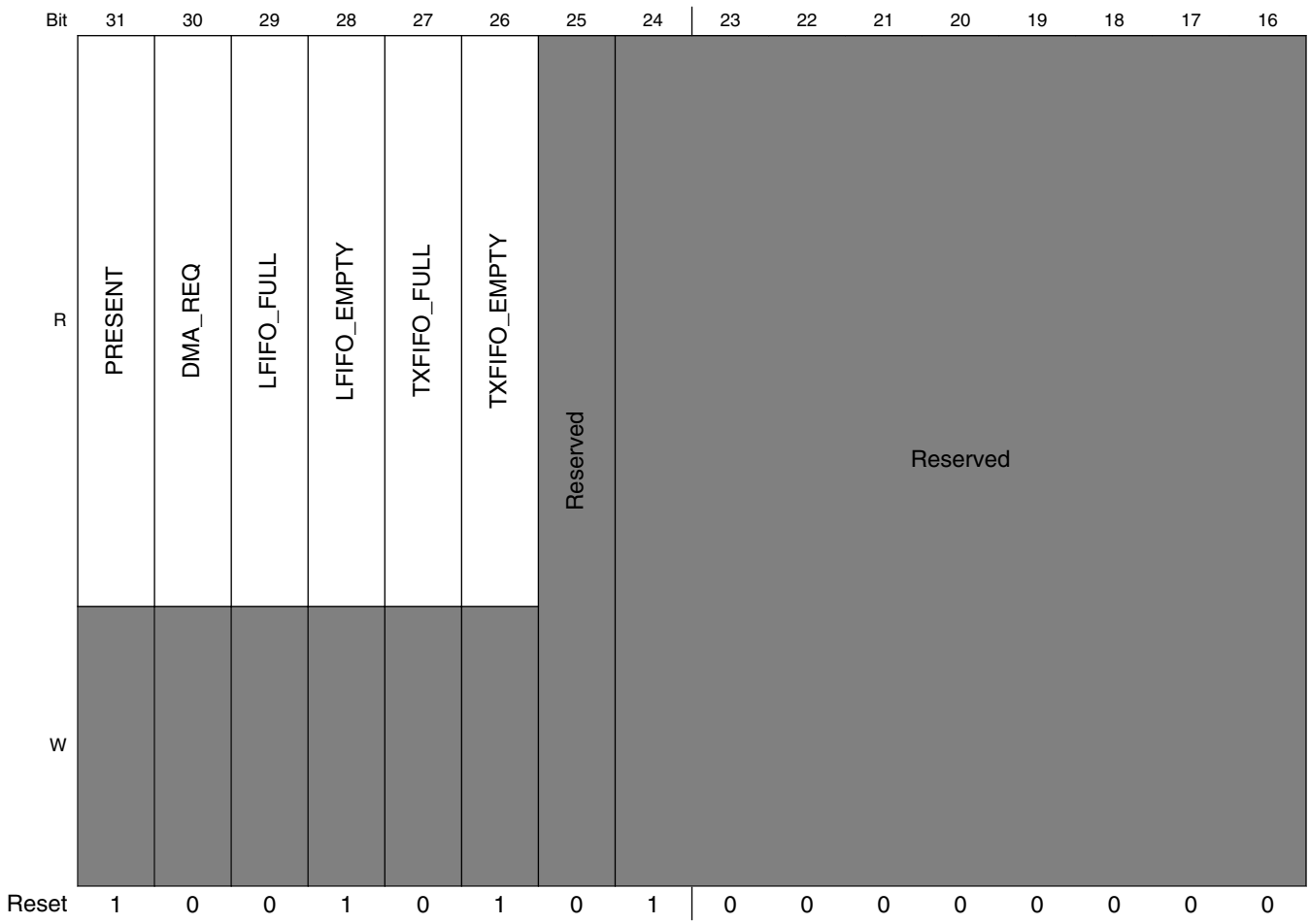
Field	Description
CRC_VALUE	Calculated CRC value. Display output valid data will be calculated by CRC32 and updated during each frame. Refer to CRC Polynomial for details.

47.6.14 LCD Interface Status Register (LCDIF_STAT)

The LCD interface status register can be used to check the current status of the LCDIF block.

The LCD interface status register that contains read only views of some parameters or current state of the block.

Address: 4080_4000h base + 1B0h offset = 4080_41B0h



LCDIF_STAT field descriptions

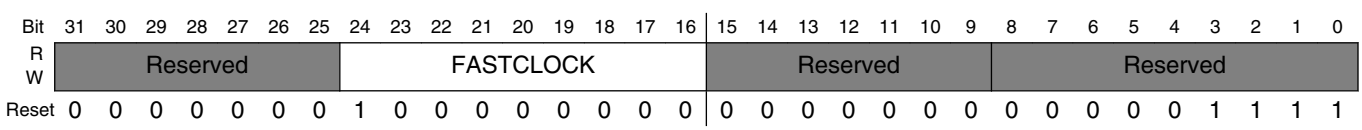
Field	Description
31 PRESENT	0: LCDIF not present on this product 1: LCDIF is present.
30 DMA_REQ	Reflects the current state of the DMA Request line for the LCDIF. The DMA Request line toggles for each new request.
29 LFIFO_FULL	Read only view of the signals that indicates LCD LFIFO is full.
28 LFIFO_EMPTY	Read only view of the signals that indicates LCD LFIFO is empty.
27 TXFIFO_FULL	Read only view of the signals that indicates LCD TXFIFO is full.
26 TXFIFO_EMPTY	Read only view of the signals that indicates LCD TXFIFO is empty.
25 -	This field is reserved. Reserved
24–9 RSRVD0	This field is reserved. Reserved bits. Write as 0.
LFIFO_COUNT	Read only view of the current count in Latency buffer (LFIFO).

47.6.15 LCDIF Threshold Register (LCDIF_THRES)

This register is used to activate control signals when the number of pixels reaches the programmed threshold. These control signals, in turn, can be used to manipulate access priority or dynamically change the input clock frequency to meet the required pixel throughput.

Memory request priority threshold register.

Address: 4080_4000h base + 200h offset = 4080_4200h



LCDIF_THRES field descriptions

Field	Description
31–25 RSRVD2	This field is reserved. Reserved bits. Write as 0.
24–16 FASTCLOCK	This value should be set to a value of pixels, from 0 to 511. When the number of pixels in the input pixel FIFO is LESS than this value, the fast clock control output will be raised. This signal can be used to reduce the system bus clock frequency to save power during horizontal or vertical blanking intervals.
15–9 RSRVD1	This field is reserved. Reserved bits. Write as 0.

Table continues on the next page...

LCDIF_THRES field descriptions (continued)

Field	Description
RSRVD	This field is reserved. Reserved bits. Write as 0.

47.6.16 LCDIF Pigeon Mode Control0 Register (LCDIF_PIGEONCTRL0*n*)

This register contains global counter settings for Pigeon Mode also houses general purpose timing adjustment registers

Address: 4080_4000h base + 380h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				LD_PERIOD												Reserved				FD_PERIOD											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_PIGEONCTRL0*n* field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–16 LD_PERIOD	Period of pclk counter during LD phase
15–12 -	This field is reserved. Reserved
FD_PERIOD	Period of line counter during FD phase

47.6.17 LCDIF Pigeon Mode Control1 Register (LCDIF_PIGEONCTRL1*n*)

This register contains global counter settings for Pigeon Mode also houses general purpose timing adjustment registers

Address: 4080_4000h base + 390h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				FRAME_CNT_CYCLES												Reserved				FRAME_CNT_PERIOD											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

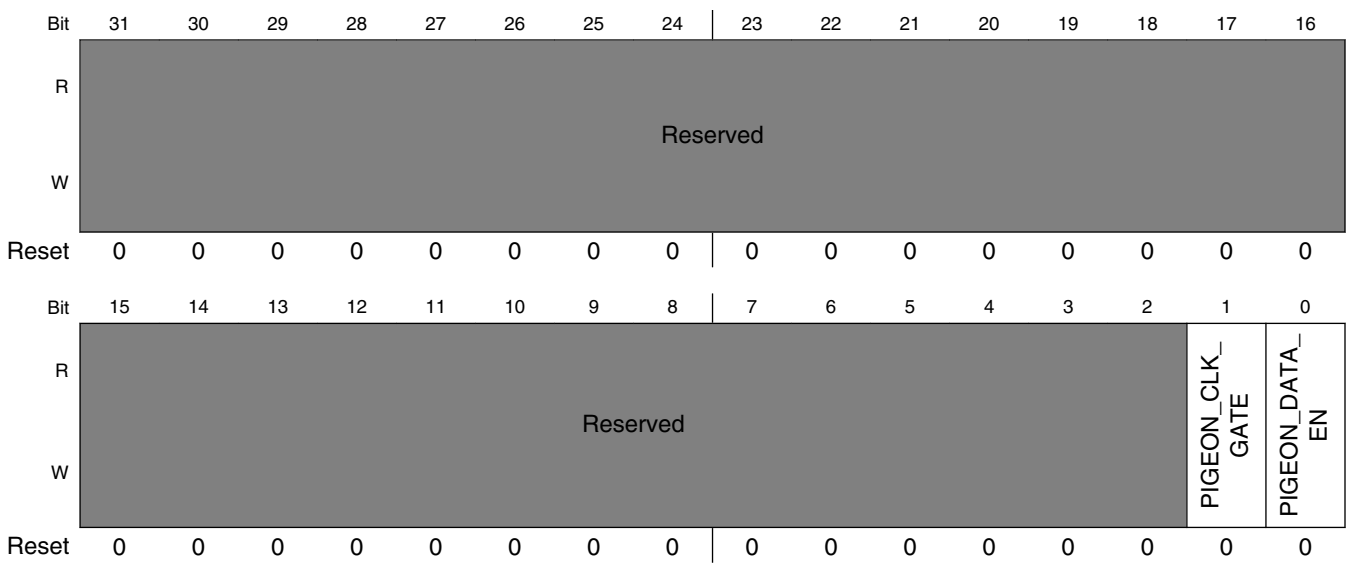
LCDIF_PIGEONCTRL1*n* field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–16 FRAME_CNT_CYCLES	Max cycles of frame counter
15–12 -	This field is reserved. Reserved
FRAME_CNT_PERIOD	Period of frame counter

47.6.18 LCDIF Pigeon Mode Control2 Register (LCDIF_PIGEONCTRL2*n*)

This register contains global counter settings for Pigeon Mode also houses clock gating and data enable registers

Address: 4080_4000h base + 3A0h offset + (4d × i), where i=0d to 3d



LCDIF_PIGEONCTRL2*n* field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 PIGEON_CLK_GATE	Pigeon mode dot clock gate enable

Table continues on the next page...

LCDIF_PIGEONCTRL2*n* field descriptions (continued)

Field	Description
0 PIGEON_DATA_EN	Pigeon mode data enable

47.6.19 Panel Interface Signal Generator Register (LCDIF_PIGEON*n*)

This register contains parameters for timing signal generation

Address: 4080_4000h base + 800h offset + (64d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STATE_MASK								MASK_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_CNT				MASK_CNT_SEL				OFFSET				INC_SEL		POL	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_PIGEON*n* field descriptions

Field	Description
31–24 STATE_MASK	state_mask = (FS|FB|FD|FE) and (LS|LB|LD|LE) , select any combination of scan states as reference point for local counter to start ticking 0x1 **FS** — FRAME SYNC 0x2 **FB** — FRAME BEGIN 0x4 **FD** — FRAME DATA 0x8 **FE** — FRAME END 0x10 **LS** — LINE SYNC 0x20 **LB** — LINE BEGIN 0x40 **LD** — LINE DATA 0x80 **LE** — LINE END
23–12 MASK_CNT	When the global counter selected through MASK_CNT_SEL matches value in this reg, pigeon local counter start ticking. 0=disable
11–8 MASK_CNT_SEL	select global counters as mask condition, use together with MASK_CNT 0x0 **HSTATE_CNT** — pclk counter within one hscan state 0x1 **HSTATE_CYCLE** — pclk cycle within one hscan state 0x2 **VSTATE_CNT** — line counter within one vscan state 0x3 **VSTATE_CYCLE** — line cycle within one vscan state 0x4 **FRAME_CNT** — frame counter 0x5 **FRAME_CYCLE** — frame cycle

Table continues on the next page...

LCDIF_PIGEON*n* field descriptions (continued)

Field	Description
	0x6 **HCNT** — horizontal counter (pclk counter within one line) 0x7 **VCNT** — vertical counter (line counter within one frame)
7–4 OFFSET	offset on pclk unit. 0=align with data, positive value means delay, minus value mean ahead. Supported range depends on panel mode
3–2 INC_SEL	Event to incrment local counter 0x0 **PCLK** — pclk 0x1 **LINE** — Line start pulse 0x2 **FRAME** — Frame start pulse 0x3 **SIG_ANOTHER** — Use another signal as tick event
1 POL	Polarity of signal output 0x0 **ACTIVE_HIGH** — Normal Signal (Active high) 0x1 **ACTIVE_LOW** — Inverted signal (Active low)
0 EN	Enable pigeon Mode on this signal

47.6.20 Panel Interface Signal Generator Register (LCDIF_PIGEON*n*)

This register contains parameters for timing signal generation

Address: 4080_4000h base + 810h offset + (64d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLR_CNT																SET_CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_PIGEON*n* field descriptions

Field	Description
31–16 CLR_CNT	Deassert signal output when counter match this value 0x0 **CLEAR_USING_MASK** — Keep active until mask off
SET_CNT	Assert signal output when counter match this value 0x0 **START_ACTIVE** — Start as active

47.6.21 Panel Interface Signal Generator Register (LCDIF_PIGEON*n*)

This register contains parameters for timing signal generation

Address: 4080_4000h base + 820h offset + (64d × i), where i=0d to 11d

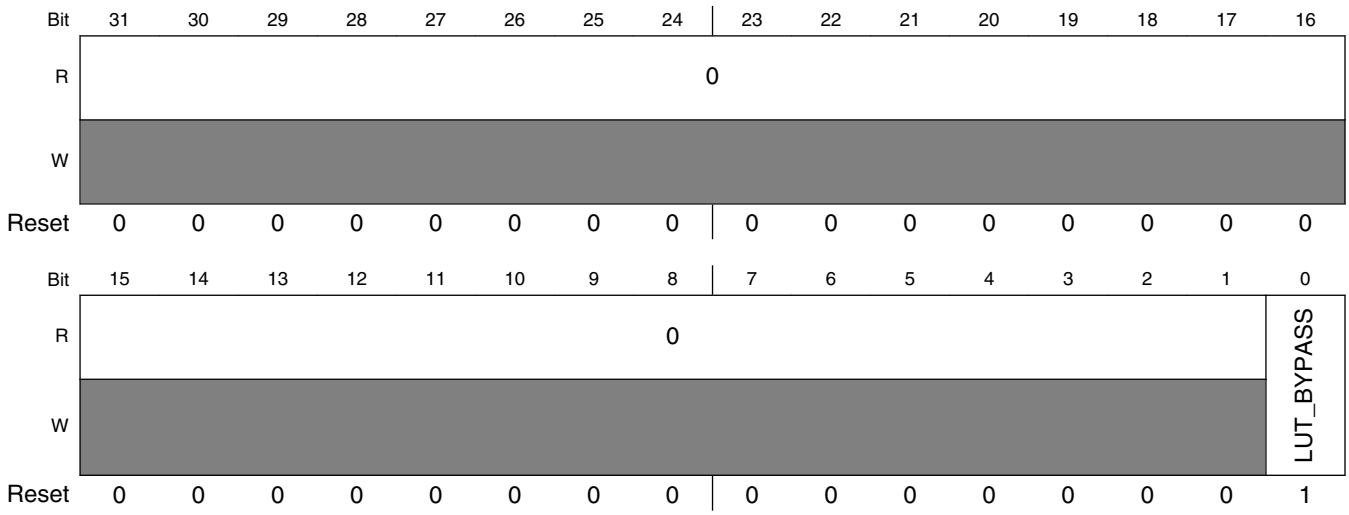
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																							SIG_ANOTHER					SIG_LOGIC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_PIGEON*n* field descriptions

Field	Description
31–9 RSVD	This field is reserved. Reserved
8–4 SIG_ANOTHER	Select another signal for logic operation or as mask or counter tick event 0 **CLEAR_USING_MASK** — Keep active until mask off
SIG_LOGIC	Logic operation with another signal: DIS/AND/OR/COND 0 **DIS** — No logic operation 1 **AND** — sigout = sig_another AND this_sig 2 **OR** — sigout = sig_another OR this_sig 3 **MASK** — mask = sig_another AND other_masks

47.6.22 Look Up Table Control Register (LCDIF_LUT_CTRL)

Address: 4080_4000h base + B00h offset = 4080_4B00h

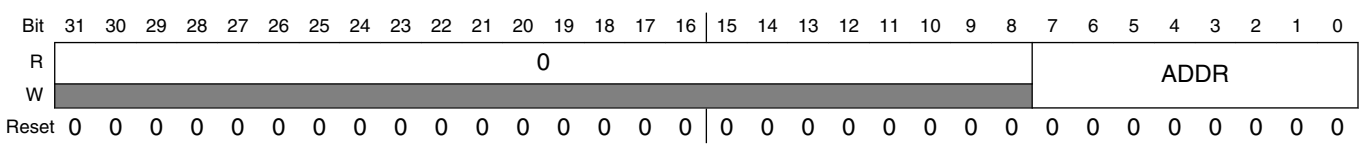


LCDIF_LUT_CTRL field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 LUT_BYPASS	Setting this bit will bypass the LUT memory resource completely. No pixel transformations will occur at last stage before output to external displayer.

47.6.23 Lookup Table 0 Index Register (LCDIF_LUT0_ADDR)

Address: 4080_4000h base + B10h offset = 4080_4B10h



LCDIF_LUT0_ADDR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.

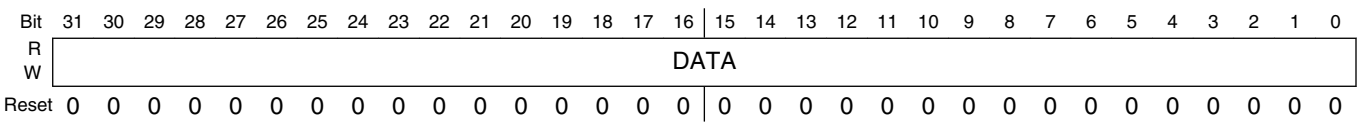
Table continues on the next page...

LCDIF_LUT0_ADDR field descriptions (continued)

Field	Description
ADDR	LUT indexed address pointer. This address into the LUT memory is always four byte aligned for PIO access. The least two significant bits are not used to drive the LUT memory array. For PIO LUT access, when the LUT data register is written, the contents of the LUT at the address specified by this address field will be loaded with a 32-bit DWORD. This address pointer will be incremented after the LUT data is written. This will provide recursive writes to the LUT data register to initialize the entire LUT array with recursive writes to the LUT data register.

47.6.24 Lookup Table 0 Data Register (LCDIF_LUT0_DATA)

Address: 4080_4000h base + B20h offset = 4080_4B20h

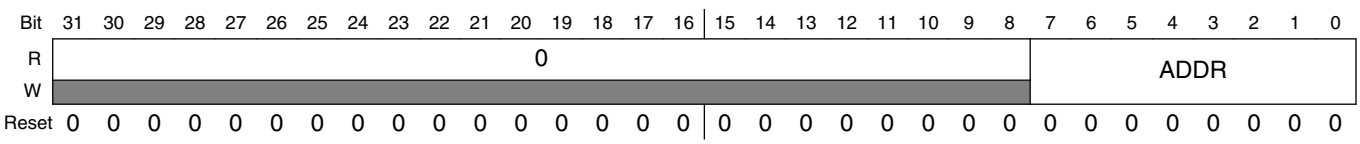


LCDIF_LUT0_DATA field descriptions

Field	Description
DATA	Writing this field will load 4 bytes, aligned to four byte boundaries, of data indexed by the ADDR field of the REG_LUT_CTRL register.

47.6.25 Lookup Table 1 Index Register (LCDIF_LUT1_ADDR)

Address: 4080_4000h base + B30h offset = 4080_4B30h



LCDIF_LUT1_ADDR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
ADDR	LUT indexed address pointer. This address into the LUT memory is always four byte aligned for PIO access. The least two significant bits are not used to drive the LUT memory array. For PIO LUT access, when the LUT data register is written, the contents of the LUT at the address specified by this address field will be loaded with a 32-bit DWORD. This address pointer will be incremented after the LUT data is

Table continues on the next page...

LCDIF_LUT1_ADDR field descriptions (continued)

Field	Description
	written. This will provide recursive writes to the LUT data register to initialize the entire LUT array with recursive writes to the LUT data register.

47.6.26 Lookup Table 1 Data Register (LCDIF_LUT1_DATA)

Address: 4080_4000h base + B40h offset = 4080_4B40h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_LUT1_DATA field descriptions

Field	Description
DATA	Writing this field will load 4 bytes, aligned to four byte boundaries, of data indexed by the ADDR field of the REG_LUT_CTRL register.

Chapter 48
LCDIF Interface v2 (LCDIF v2)

48.1 Chip-specific LCDIFv2 information

Table 48-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

Global Alpha Blending is limited to 2-layer Porter-Duff Alpha Blending, but works well with other blending modes.

48.2 Overview

The LCD Interface v2 (LCDIFv2) is a system master that fetches graphics stored in memory and displays them on a TFT LCD panel or connects to a display interface depending on the chip configuration. A wide range of panel sizes are supported and the timing of the interface signals is highly configurable. Graphics are read directly from memory and blended in real-time, which allows for dynamic content creation with minimal CPU intervention. Graphics can be encoded in a variety of formats for optimum memory usage.

48.2.1 Block diagram

The block diagram for LCDIFv2 is shown below:

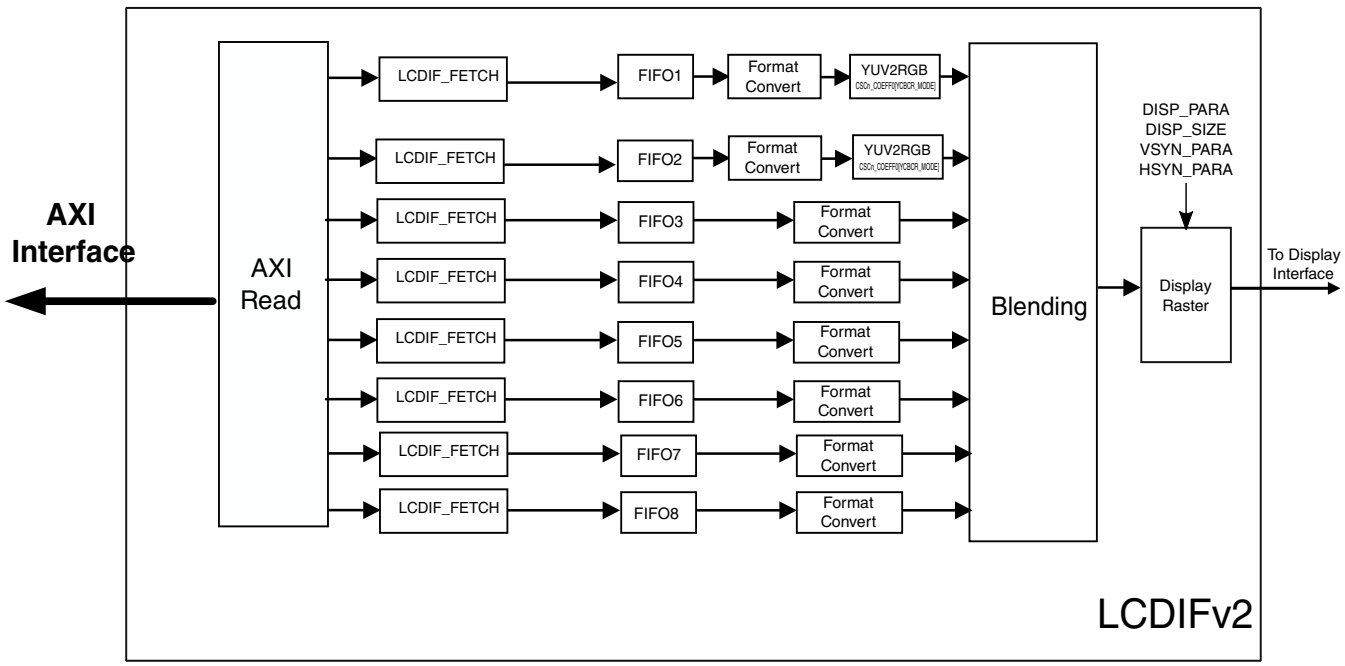


Figure 48-1. LCDIFv2 block diagram

The LCDIFv2 manages the graphical content of the panel through separate registers for each layers. There are 8 layers available in the LCDIFv2 and each can be configured with:

- Horizontal and vertical size of graphic (CTRLDESCLn_1)
- Position of graphic on the panel (CTRLDESCLn_2)
- Address of graphic in memory (CTRLDESCLn_4)
- Color formats or Color palettes, and type of Alpha Blending mode (CTRLDESCLn_5)
- Background colors for each layer if not active (CTRLDESCLn_6)

The values in these registers may be changed at any time, the panel content will be updated automatically in the panel refresh cycle when CTRLDESCLx_5[SHADOW_LOAD_EN] is set. The Alpha Blending Mode can be set as either global/embedded Alpha or Porter Duff mode (For the Porter Duff mode, all the active layers for blending should in the same mode).

NOTE

The Color Space Conversion (CSC) is only supported in Layer1 and Layer2.

48.2.2 Features

The LCDIFv2 includes the following features:

- Support for DOTCLK mode only in display mode
- Display layers can support up to maximum 8 layers of alpha blending:
  - 1 Background (BG) layer for static background image
  - 1 Foreground (FG) layer for video
  - 6 User Interface (UI) layers for the icons, test, moving pointers etc.
  - The UI layers are for small objects which can be stored in OCRAM
  - The index color (1/2/4/8 bpp) support for each layer and lookup with separate Color Look-Up Table (CLUT) memory to 32 bits ARGB pixel
- Each layer supports:
  - Programmable plane size (Width/Height/Pitch) and X/Y offset on the panel
  - Background color for plane graphics
  - Embedded alpha and global alpha
  - Index color (1/2/4/8 bpp)
  - Other layer encoding formats:
    - 16bpp(RGB565)
    - 16bpp(ARGB1555)
    - 16bpp(ARGB4444)
    - YCbCr422(Can support up to two YCbCr layer in blending operation)
    - 24bpp(RGB888)
    - 32bpp(ARGB8888)
    - 32bpp(ABGR8888)
- Support for one parallel camera interface input and the following data formats of CSI-2:
  - 16bpp(YUV422 8-bit)
  - 24bpp(RGB888)
  - 18bpp(RGB666)
  - 16bpp(RGB565)
  - 15bpp(RGB555)
  - 12bpp(RGB444)

48.3 Functional description

The LCDIFv2 is a master on the bus fabric; it fetches graphic source information directly from memory and dynamically performs blending operation before send data to a TFT LCD panel.

As the LCDIFv2 is a master on the NIC, it can access directly any memory or device connected to the NIC as a slave. Therefore, any compatible graphic stored anywhere on-chip can be displayed on the connected TFT LCD panel with no further intervention from CPU.

48.3.1 Graphics and data format

The memory location of the graphic that is displayed on the layer is defined by register in the control descriptor for the layer. This 32-bit value can contain the address of any memory location in the memory map of the system.

The format of the data that describes the graphic is defined by the BPP bit field in the control descriptor for the layer. By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

The LCDIFv2 can support the conversion of YUV422 data to RGB888 data on layer 0 and 1. If YCbCr422 input is present, software must change the coefficient registers appropriately, and set the CSC_COEFF0[YCBCR_MODE] bitfield to ensure proper conversion of YUV versus YCBCR data.

The table below indicates the expected coefficients for YUV and YCbCr modes of operation:

Table 48-2. Coefficient Table

Coefficient	YUV	YCbCr
Yoffset	0x000	0x1F0 (-16)
UVoffset	0x000	0x180 (-128)
C0	0x100 (1.00)	0x12A (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x730 (-0.813)
C3	0x79B (-0.394)	0x79C (-0.392)
C4	0x208 (2.032)	0x204 (2.017)

There are six modes where the RGB values of the pixels are stored directly in the graphic. In these modes, the LCDIFv2 treats the data as describing a direct RGB color. The modes are:

- ABGR8888, where the data defines 8-bit values for the blue, green, red, and alpha components of the image.
- ARGB8888, where the data defines 8-bit values for the red, green, blue, and alpha components of the image.
- RGB888, where the data defines 8-bit values for the red, green, and blue components of the image.
- RGB565 where the data defines 5-bit values for the red and blue components, and 6-bit values for the green component of the image.
- ARGB1555 where the data defines 5-bit values for the red, green, and blue components, and a 1-bit value for the alpha component of the image.
- ARGB4444, where the data defines 4-bit values for the red, green, blue, and alpha components of the image.

The three 16-bit formats (RGB565, ARGB1555, and ARGB4444) are promoted to full 8 bit per component format by shifting the bits left so that the MSB of the component in the 16-bit format becomes the MSB of the 24/32 bpp (bit per pixel) format, and the LSB is filled with the value of the MSBs. Please check the table below for different modes:

Table 48-3. Modes

Format	Original	After promoting
RGB565	10000:010000:11011	10000**100**:01000**001**:11011**110**
ARGB4444	1010:0011:1100:0101	1010**1010**:00110011:1100**1100**:0101**0101**
ARGB1555	1:10100:01000:11011	1**1111111**:10100**101**:01000**010**:1101**110**

There are four indexed color formats (1/2/4/8 bpp) where the data in the graphic does not define the RGB color to display. Instead, the data defines the entry in a color look-up table (CLUT) that contains a palette of ARGB colors. The maximum number of colors in the CLUT is defined by the size of the data stored in the graphic. For 1 bpp graphics, there is a maximum of two colors in the CLUT. For 2 bpp, there is a maximum of four colors. For 4 bpp and 8 bpp data, the maximums are 16 and 256 colors.

There is a set storage format for each data format provided by the LCDIFv2. These formats can be seen in the following tables.

Table 48-4. Data Layout for ARGB8888

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	A1	R1	G1	B1	A0	R0	G0	B0
0x08	A3	R3	G3	B3	A2	R2	G2	B2

In the following table, the YCbCr422 format encodes chroma information across two pixels. Therefore, the chroma values apply to the even pixel denoted in the table and its adjacent odd pixel.

Table 48-5. Data layout for YCbCr422 format

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	Y3	Cr2	Y2	Cb2	Y1	Cr0	Y0	Cb0
0x08	Y7	Cr6	Y6	Cb6	Y5	Cr4	Y4	Cb4

Table 48-6. Data Layout for 24 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	G2	B2	R1	G1	B1	R0	G0	B0
0x08	B5	R4	G4	B4	R3	G3	B3	R2

For 16 bpp, data expected is in the form of RGB565, ARGB1555 or ARGB4444.

Table 48-7. Generic data layout for 16 bpp

Address offset	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p3[15:8]	p3[7:0]	p2[15:8]	p2[7:0]	p1[15:8]	p1[7:0]	p0[15:8]	p0[7:0]
0x08	p7[15:8]	p7[7:0]	p6[15:8]	p6[7:0]	p5[15:8]	p5[7:0]	p4[15:8]	p4[7:0]

Table 48-8. Data layout for RGB565 format

Address offset	[31:27]	[26:21]	[20:16]	[15:11]	[10:15]	[4:0]
0x0	R1	G1	B1	R0	G0	B0
0x4	R3	G3	B3	R2	G2	B2

Table 48-9. Data layout for ARGB4444 format

Address offset	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0x0	A1	R1	G1	B1	A0	R0	G0	B0
0x4	A3	R3	G3	B3	A2	R2	G2	B2

Table 48-10. Data layout for ARGB1555 format

Address offset	[31]	[30:26]	[25:21]	[20:16]	[15]	[14:10]	[9:5]	[4:0]
0x00	A1	R1	G1	B1	A0	R0	G0	B0
0x4	A3	R3	G3	B3	A2	R2	G2	B2

Table 48-11. Data Layout for 8 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p7	p6	p5	p4	p3	p2	p1	p0
0x08	p15	p14	p13	p12	p11	p10	p9	p8

Table 48-12. Data layout for 4 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p15-p14	p13-p12	p11-p10	p9-p8	p7-p6	p5-p4	p3-p2	p1-p0
0x08	p31-p30	p29-p28	p27-p26	p25-p24	p23-p22	p21-p20	p19-p18	p17-p16

Table 48-13. Data Layout for 2 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p31-28	p27-24	p23-20	p19-16	p15-p12	p11-p8	p7-p4	p3-p0
0x08	p63-p60	p59-p56	p55-p52	p51-p48	p47-p44	p43-p40	p39-p36	p35-p32

Table 48-14. Data Layout for 1 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p63-p56	p55-48	p47-40	p39-p32	p31-p24	p23-p16	p15-p8	p7-p0
0x08	p127-p120	p119-p112	p111-p104	p103-p96	p95-p88	p87-p80	p79-p72	p71-p64

The LCDIFv2 includes interrupt status bit that indicates when it has completed fetching graphics from memory for the current frame refresh. If required, this flag (INT_STATUS_Dx[DMA_DONE]) can used to determine when changes can be made to the source graphic content.

48.3.2 Alpha blending modes

This section describes the Alpha blending modes of the LCDIFv2 module.

The below figure shows how the pixel blending takes place inside the LCDIFv2. When there are 8 layers overlap in the FIFOs for a given pixel output, the alpha blending takes place from the FIFO1(layer 0) and FIFO2 (layer 1) first, and then the output result takes alpha blending with FIFO3(layer2), and so on until FIFO8(layer(7)).

Users control the graphical content of the TFT panel by manipulating the configuration of elements in the LCDIFv2 called layers. The LCDIFv2 provides 8 independent layers that all display graphics with fixed blending pipeline, the blending setting on each layer allows the pixels on that layer to be opaque, partially transparent, or fully transparent, which allows them to combine with pixels on other layers that they overlap.

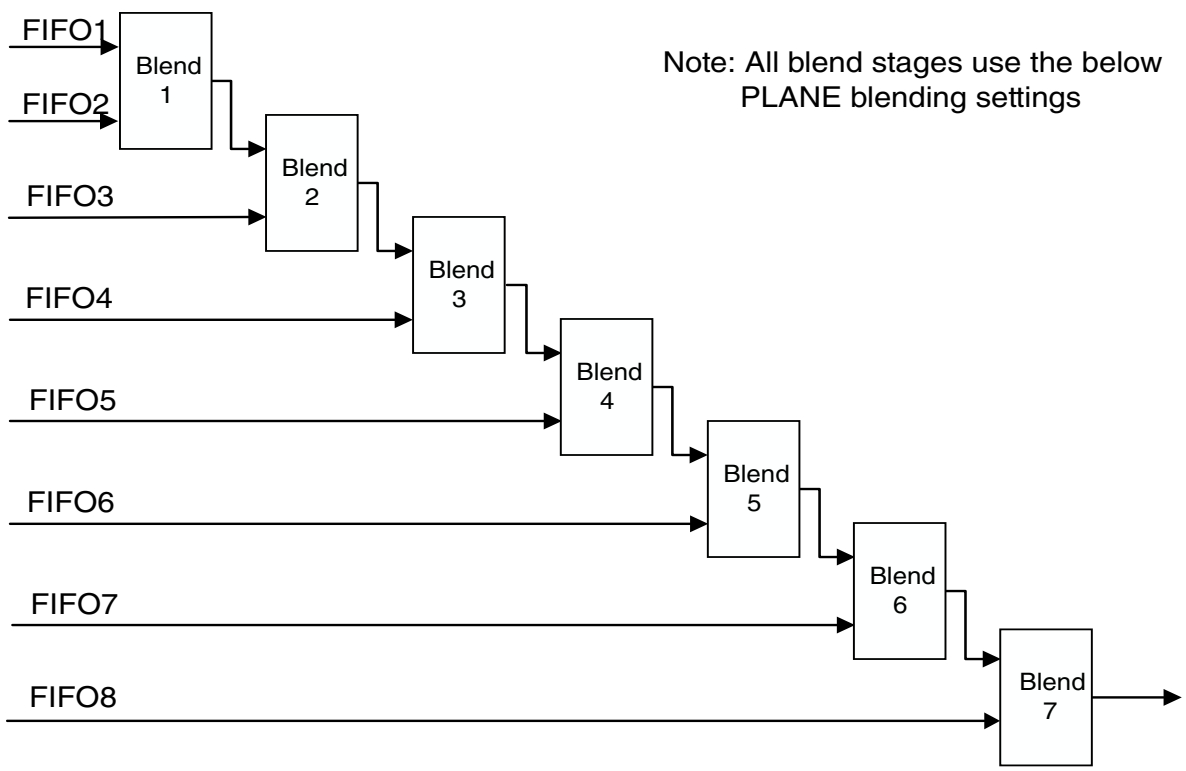


Figure 48-2. Alpha blending diagram

There are two alpha blending modes in LCDIFv2, one is the normal blending mode using single alpha parameter (Embedded Alpha or Global Alpha), another one is the Porter Duff blending mode used in the 2D engine. The Porter Duff blending includes 12 blending modes to describe the digital image composite. These processes include Clear, Source Only, Destination Only, Source Over, Source In, Source Out, Source Atop, Destination Over, Destination In, Destination Out, Destination Atop and XOR. Through these processes, it can achieve any 2D image composite.

The figure below shows the Porter Duff alpha blending flow. All the control bits are included in the CTRLDESCLn_5 register for the layer.

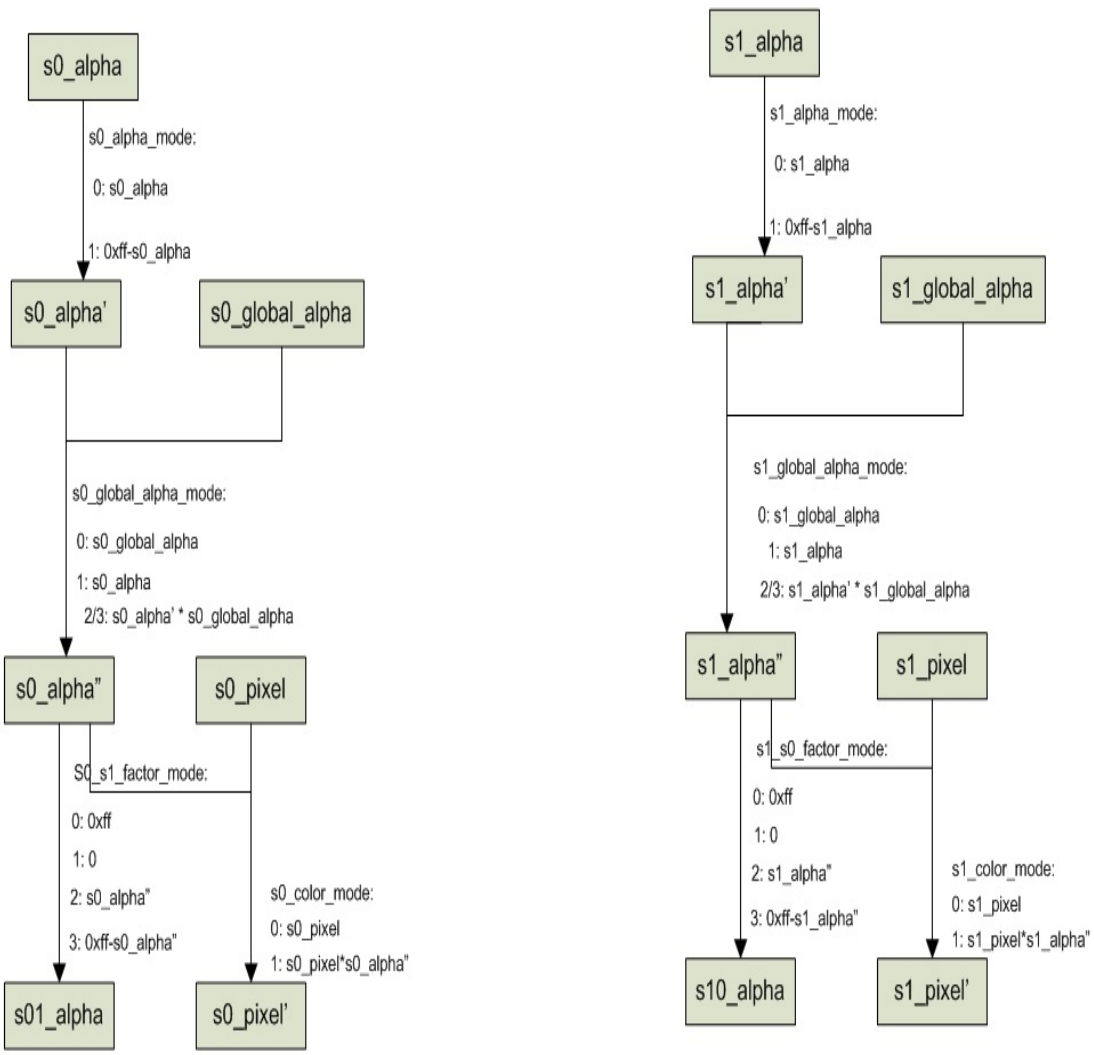


Figure 48-3. Porter Duff Alpha Blending Flow

48.3.3 Clocks

The table below shows the clocks described in this module:

Table 48-15. Clocks

Clocks	Description
pix_clk	Pixel Clock This clock should be configured according to TFT panel specification for supported display resolution
apb_clk	System Bus access clock
b_clk	LCDIFv2 peripheral access bus clock connected to apb_clk

NOTE

There is no frequency restriction between pix_clk and b_clk in the LCDIFv2, but for better performance for multiple layers data fetch, higher b_clk can provide more available bandwidth from system memory to avoid FIFO under-run.

48.3.4 Reset

There is one hardware reset input to LCDIFv2 (reset_b). The reset is asynchronous and forces all internal registers into their reset state when it is active-low.

There is one software reset bit (CTRL[SW_RESET]), which can reset all the internal registers, except the user configured registers.

48.3.5 Interrupts

There are two output interrupt signals (lcdif_d0_irq, and lcdif_d1_irq) in this module. They have similar functions and can be used for different CPU domains in the SOC.

- The INT_ENABLE_Dx[VSYNC] interrupt flag indicates that the LCDIFv2 has begun the next panel refresh period.
- The INT_ENABLE_Dx[UNDERRUN] interrupt flag indicates the TXFIFO for display is in an underrun condition.
- The INT_ENABLE_Dx[VS_BLANK] interrupt flag indicates the start of the vertical blanking period.
- The INT_ENABLE_Dx[FIFO_EMPTY]/INT_ENABLE_Dx[DMA_DONE]/ INT_ENABLE_Dx[DMA_ERR] interrupt flags for each layer indicate the status of pixel FIFO, frame buffer fetch, and AXI interface.

48.4 External Signals

Table 48-16. Signal Descriptions

Signal	Direction	Description
DOTCLK	OUTPUT	Pixel clock used to drive the display panel
VSYNC	OUTPUT	Vertical sync signal, indicating the beginning of a new frame
HSYNC	OUTPUT	Horizontal sync signal, indicating the beginning of a new line
ENABLE	OUTPUT	Data Enable. Indicates when there is valid pixel data
DATA[23:0]	OUTPUT	Red, green and blue data output

48.5 Application information

The figure below shows the details of software flow for the LCDIFv2.

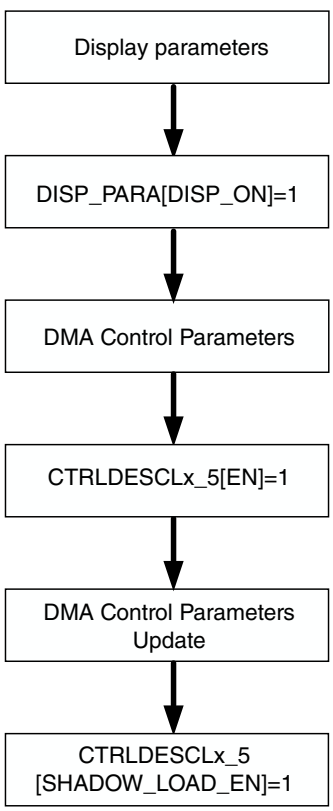


Figure 48-4. Recommended Software flow diagram

48.6 Memory Map and register definition

This section includes the LCDIFv2 module memory map and detailed descriptions of all registers.

48.6.1 LCDIF_V2 register descriptions

Table 48-17. Memory map

Parameter	Address Range
Register address space	0x0000 - 0x1FFF
CLUT RAM address space	0x2000 - 0x3FFF

48.6.1.1 LCDIF_V2 memory map

LCDIFV2 base address: 4080_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	LCDIFv2 display control Register (CTRL)	32	RW	8000_0000h
4h	LCDIFv2 display control Register (CTRL_SET)	32	RW	8000_0000h
8h	LCDIFv2 display control Register (CTRL_CLR)	32	RW	8000_0000h
Ch	LCDIFv2 display control Register (CTRL_TOG)	32	RW	8000_0000h
10h	Display Parameter Register (DISP_PARA)	32	RW	0000_0000h
14h	Display Size Register (DISP_SIZE)	32	RW	0000_0000h
18h	Horizontal Sync Parameter Register (HSYN_PARA)	32	RW	00C0_1803h
1Ch	Vertical Sync Parameter Register (VSYN_PARA)	32	RW	00C0_1803h
20h	Interrupt Status Register for domain 0 (INT_STATUS_D0)	32	RW	0000_0000h
24h	Interrupt Enable Register for domain 0 (INT_ENABLE_D0)	32	RW	0000_0000h
30h	Interrupt Status Register for domain 1 (INT_STATUS_D1)	32	RW	0000_0000h
34h	Interrupt Enable Register for domain 1 (INT_ENABLE_D1)	32	RW	0000_0000h
40h	Reserved (PDI_PARA)	32	RW	0000_1000h
200h	Control Descriptor Layer 1 Register (CTRLDESCL0_1)	32	RW	0000_0000h
204h	Control Descriptor Layer 2 Register (CTRLDESCL0_2)	32	RW	0000_0000h
208h	Control Descriptor Layer 3 Register (CTRLDESCL0_3)	32	RW	0000_0000h
20Ch	Control Descriptor Layer 4 Register (CTRLDESCL0_4)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
210h	Control Descriptor Layer 5 Register (CTRLDESCL0_5)	32	RW	0000_0000h
214h	Control Descriptor Layer 6 Register (CTRLDESCL0_6)	32	RW	0000_0000h
218h	Color Space Conversion Coefficient Register 0 (CSC0_COEF0)	32	RW	0400_0000h
21Ch	Color Space Conversion Coefficient Register 1 (CSC0_COEF1)	32	RW	0123_0208h
220h	Color Space Conversion Coefficient Register 2 (CSC0_COEF2)	32	RW	076B_079Ch
240h	Control Descriptor Layer 1 Register (CTRLDESCL1_1)	32	RW	0000_0000h
244h	Control Descriptor Layer 2 Register (CTRLDESCL1_2)	32	RW	0000_0000h
248h	Control Descriptor Layer 3 Register (CTRLDESCL1_3)	32	RW	0000_0000h
24Ch	Control Descriptor Layer 4 Register (CTRLDESCL1_4)	32	RW	0000_0000h
250h	Control Descriptor Layer 5 Register (CTRLDESCL1_5)	32	RW	0000_0000h
254h	Control Descriptor Layer 6 Register (CTRLDESCL1_6)	32	RW	0000_0000h
258h	Color Space Conversion Coefficient Register 0 (CSC1_COEF0)	32	RW	0400_0000h
25Ch	Color Space Conversion Coefficient Register 1 (CSC1_COEF1)	32	RW	0123_0208h
260h	Color Space Conversion Coefficient Register 2 (CSC1_COEF2)	32	RW	076B_079Ch
280h	Control Descriptor Layer 1 Register (CTRLDESCL2_1)	32	RW	0000_0000h
284h	Control Descriptor Layer 2 Register (CTRLDESCL2_2)	32	RW	0000_0000h
288h	Control Descriptor Layer 3 Register (CTRLDESCL2_3)	32	RW	0000_0000h
28Ch	Control Descriptor Layer 4 Register (CTRLDESCL2_4)	32	RW	0000_0000h
290h	Control Descriptor Layer 5 Register (CTRLDESCL2_5)	32	RW	0000_0000h
294h	Control Descriptor Layer 6 Register (CTRLDESCL2_6)	32	RW	0000_0000h
2C0h	Control Descriptor Layer 1 Register (CTRLDESCL3_1)	32	RW	0000_0000h
2C4h	Control Descriptor Layer 2 Register (CTRLDESCL3_2)	32	RW	0000_0000h
2C8h	Control Descriptor Layer 3 Register (CTRLDESCL3_3)	32	RW	0000_0000h
2CCh	Control Descriptor Layer 4 Register (CTRLDESCL3_4)	32	RW	0000_0000h
2D0h	Control Descriptor Layer 5 Register (CTRLDESCL3_5)	32	RW	0000_0000h
2D4h	Control Descriptor Layer 6 Register (CTRLDESCL3_6)	32	RW	0000_0000h
300h	Control Descriptor Layer 1 Register (CTRLDESCL4_1)	32	RW	0000_0000h
304h	Control Descriptor Layer 2 Register (CTRLDESCL4_2)	32	RW	0000_0000h
308h	Control Descriptor Layer 3 Register (CTRLDESCL4_3)	32	RW	0000_0000h
30Ch	Control Descriptor Layer 4 Register (CTRLDESCL4_4)	32	RW	0000_0000h
310h	Control Descriptor Layer 5 Register (CTRLDESCL4_5)	32	RW	0000_0000h
314h	Control Descriptor Layer 6 Register (CTRLDESCL4_6)	32	RW	0000_0000h
340h	Control Descriptor Layer 1 Register (CTRLDESCL5_1)	32	RW	0000_0000h
344h	Control Descriptor Layer 2 Register (CTRLDESCL5_2)	32	RW	0000_0000h
348h	Control Descriptor Layer 3 Register (CTRLDESCL5_3)	32	RW	0000_0000h
34Ch	Control Descriptor Layer 4 Register (CTRLDESCL5_4)	32	RW	0000_0000h
350h	Control Descriptor Layer 5 Register (CTRLDESCL5_5)	32	RW	0000_0000h
354h	Control Descriptor Layer 6 Register (CTRLDESCL5_6)	32	RW	0000_0000h
380h	Control Descriptor Layer 1 Register (CTRLDESCL6_1)	32	RW	0000_0000h
384h	Control Descriptor Layer 2 Register (CTRLDESCL6_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
388h	Control Descriptor Layer 3 Register (CTRLDESCL6_3)	32	RW	0000_0000h
38Ch	Control Descriptor Layer 4 Register (CTRLDESCL6_4)	32	RW	0000_0000h
390h	Control Descriptor Layer 5 Register (CTRLDESCL6_5)	32	RW	0000_0000h
394h	Control Descriptor Layer 6 Register (CTRLDESCL6_6)	32	RW	0000_0000h
3C0h	Control Descriptor Layer 1 Register (CTRLDESCL7_1)	32	RW	0000_0000h
3C4h	Control Descriptor Layer 2 Register (CTRLDESCL7_2)	32	RW	0000_0000h
3C8h	Control Descriptor Layer 3 Register (CTRLDESCL7_3)	32	RW	0000_0000h
3CCh	Control Descriptor Layer 4 Register (CTRLDESCL7_4)	32	RW	0000_0000h
3D0h	Control Descriptor Layer 5 Register (CTRLDESCL7_5)	32	RW	0000_0000h
3D4h	Control Descriptor Layer 6 Register (CTRLDESCL7_6)	32	RW	0000_0000h
400h	LCDIFv2 CLUT load Register (CLUT_LOAD)	32	RW	0000_0000h

48.6.1.2 LCDIFv2 display control Register (CTRL)

48.6.1.2.1 Offset

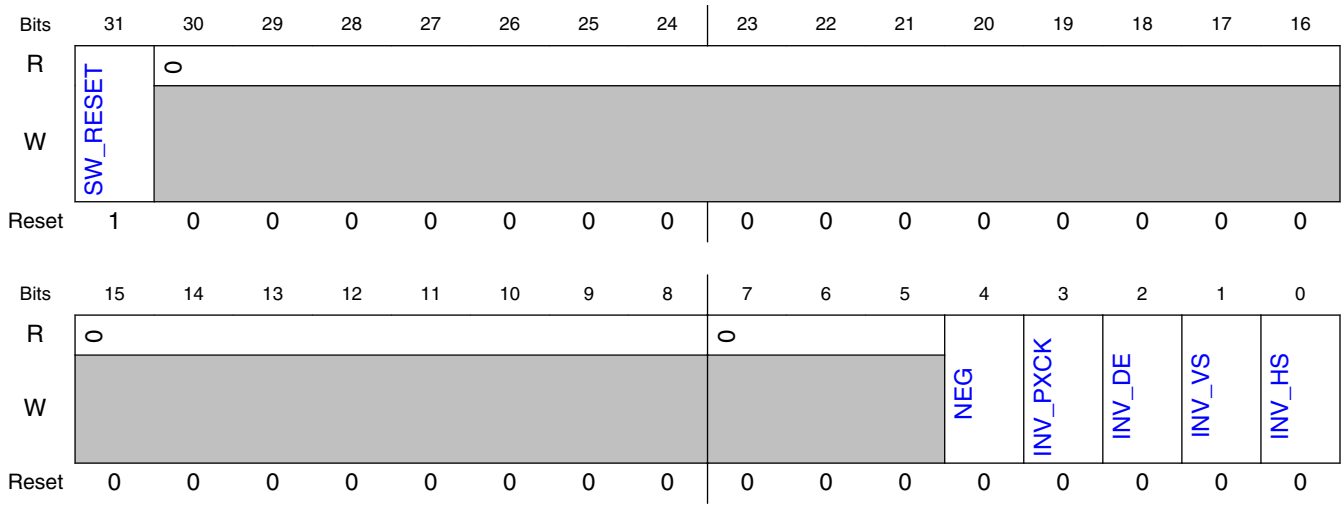
This type of register has supplemental _SET, _CLR, and _TOG registers at adjacent offsets.

Register	Offset	Description
CTRL	0h	LCDIFv2 display control Register
CTRL_SET	4h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 1
CTRL_CLR	8h	Writing 1 to a bit in this register ensures that the corresponding bit in CTRL is 0
CTRL_TOG	Ch	Writing 1 to a bit in this register inverts the value of the corresponding bit in CTRL

48.6.1.2.2 Function

This register control the polarity for HSYNC/VSYNC/DE of display

48.6.1.2.3 Diagram



48.6.1.2.4 Fields

Field	Function
31 SW_RESET	Software Reset This bit is high active. If it is set to 1, all logical will be reset. 0b - No action 1b - All LCDIFv2 internal registers are forced into their reset state. User registers are not affected
30-16 —	Reserved
15-8 —	Reserved
7-5 —	Reserved
4 NEG	Indicates if value at the output (pixel data output) needs to be negated 0b - Output is to remain same 1b - Output to be negated
3 INV_PXCK	Polarity change of Pixel Clock 0b - Display samples data on the falling edge 1b - Display samples data on the rising edge
2 INV_DE	Invert Data Enable polarity 0b - Data enable is active high 1b - Data enable is active low
1 INV_VS	Invert Vertical synchronization signal 0b - VSYNC signal not inverted (active HIGH) 1b - Invert VSYNC signal (active LOW)
0 INV_HS	Invert Horizontal synchronization signal 0b - HSYNC signal not inverted (active HIGH) 1b - Invert HSYNC signal (active LOW)

48.6.1.3 Display Parameter Register (DISP_PARA)

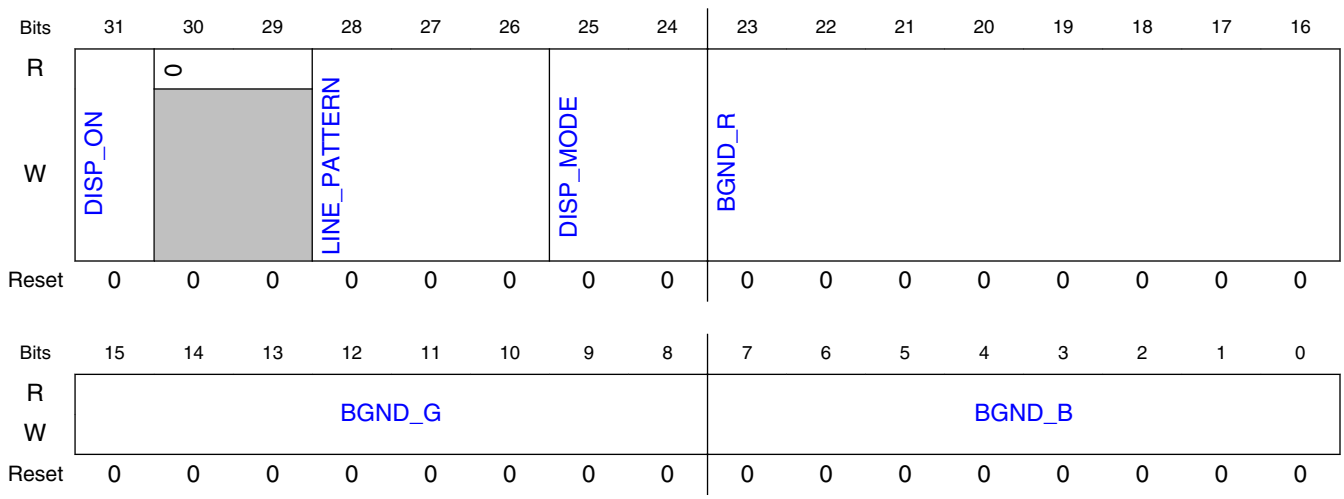
48.6.1.3.1 Offset

Register	Offset
DISP_PARA	10h

48.6.1.3.2 Function

Display Parameter Register

48.6.1.3.3 Diagram



48.6.1.3.4 Fields

Field	Function
31 DISP_ON	Display panel On/Off mode 0b - Display Off 1b - Display On
30-29 —	Reserved
28-26 LINE_PATTERN	LCDIFv2 line output order 000b - RGB 001b - RBG 010b - GBR 011b - GRB

Table continues on the next page...

Field	Function
	100b - BRG 101b - BGR
25-24 DISP_MODE	LCDIFv2 operating mode 00b - Normal mode. Panel content controlled by layer configuration 01b - Test Mode1(BGND Color Display) 10b - Test Mode2(Column Color Bar) 11b - Test Mode3(Row Color Bar)
23-16 BGND_R	Red component of the default color displayed in Test Mode 1.
15-8 BGND_G	Green component of the default color displayed in Test Mode 1.
7-0 BGND_B	Blue component of the default color displayed in Test Mode 1.

48.6.1.4 Display Size Register (DISP_SIZE)

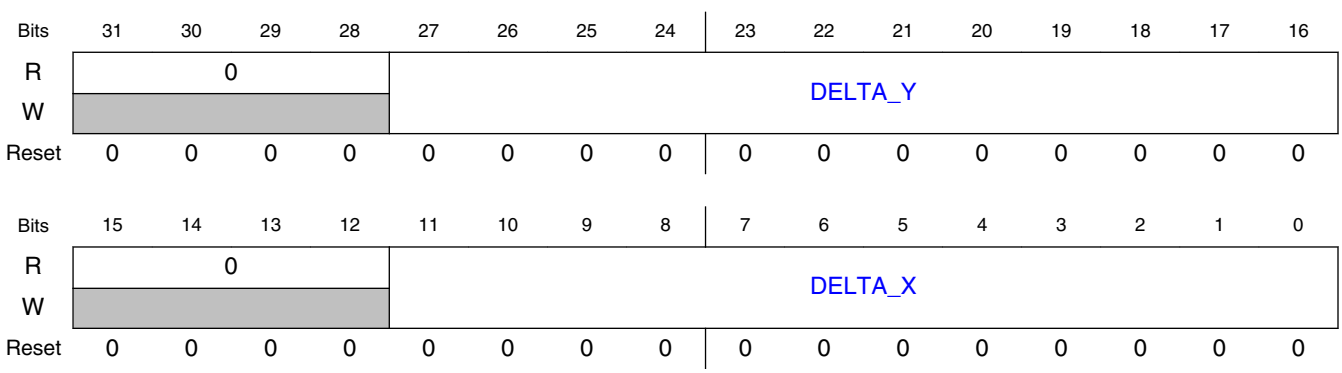
48.6.1.4.1 Offset

Register	Offset
DISP_SIZE	14h

48.6.1.4.2 Function

Display Size Register

48.6.1.4.3 Diagram



48.6.1.4.4 Fields

Field	Function
31-28 —	Reserved
27-16 DELTA_Y	Sets the display size vertical resolution in pixels
15-12 —	Reserved
11-0 DELTA_X	Sets the display size horizontal resolution in pixels

48.6.1.5 Horizontal Sync Parameter Register (HSYN_PARA)

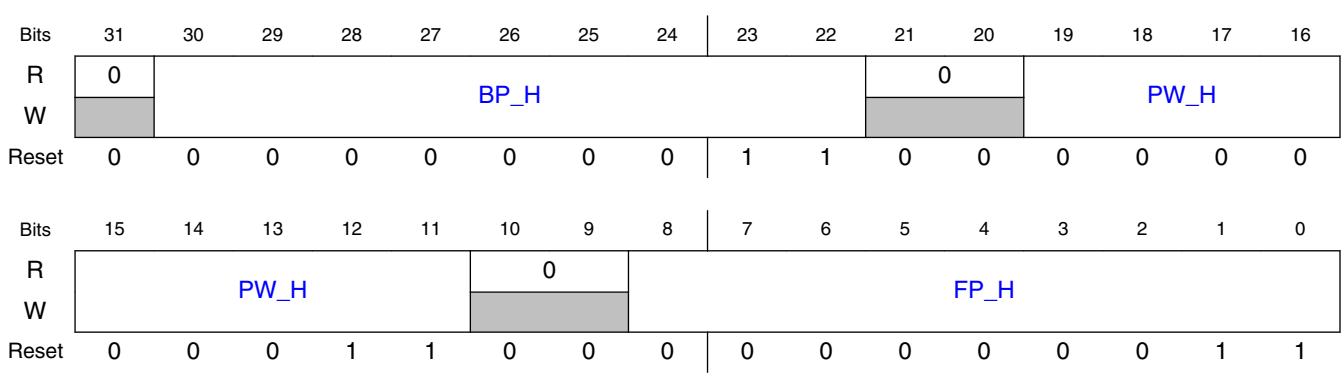
48.6.1.5.1 Offset

Register	Offset
HSYN_PARA	18h

48.6.1.5.2 Function

HSYN_PARA register sets timing parameters related to the horizontal synchronization signal generation for the TFT. The fields FP_H, BP_H, and PW_H stand for HSYNC signal front-porch, back-porch, and active pulse width, respectively

48.6.1.5.3 Diagram



48.6.1.5.4 Fields

Field	Function
31 —	Reserved
30-22 BP_H	HSYNC back-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1
21-20 —	Reserved
19-11 PW_H	HSYNC active pulse width (in pixel clock cycles). Pulse width has a minimum value of 1
10-9 —	Reserved
8-0 FP_H	HSYNC front-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1

48.6.1.6 Vertical Sync Parameter Register (VSYN_PARA)

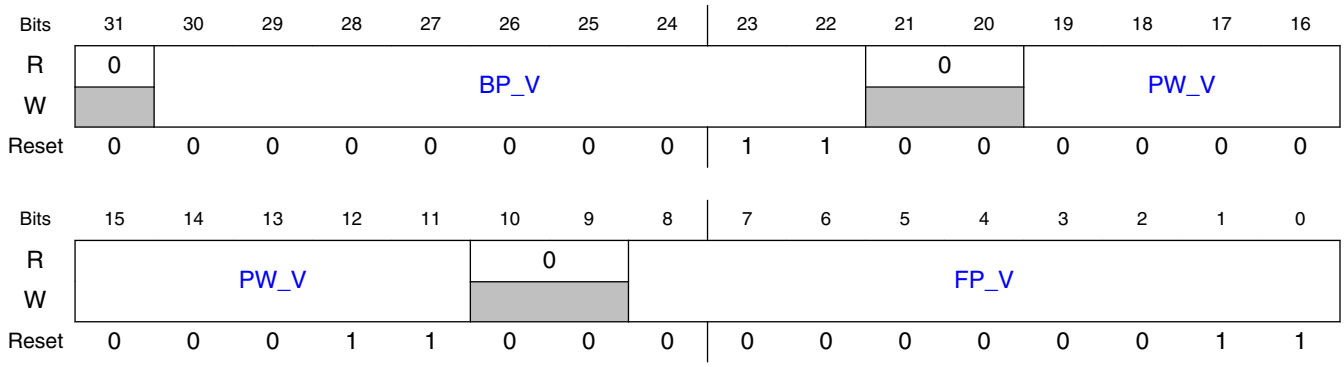
48.6.1.6.1 Offset

Register	Offset
VSYN_PARA	1Ch

48.6.1.6.2 Function

VSYN_PARA register sets timing parameters of the TFT related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively

48.6.1.6.3 Diagram



48.6.1.6.4 Fields

Field	Function
31 —	Reserved
30-22 BP_V	VSYNC back-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1
21-20 —	Reserved
19-11 PW_V	VSYNC active pulse width (in horizontal line cycles). Pulse width has a minimum value of 1
10-9 —	Reserved
8-0 FP_V	VSYNC front-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1

48.6.1.7 Interrupt Status Register for domain 0 (INT_STATUS_D0)

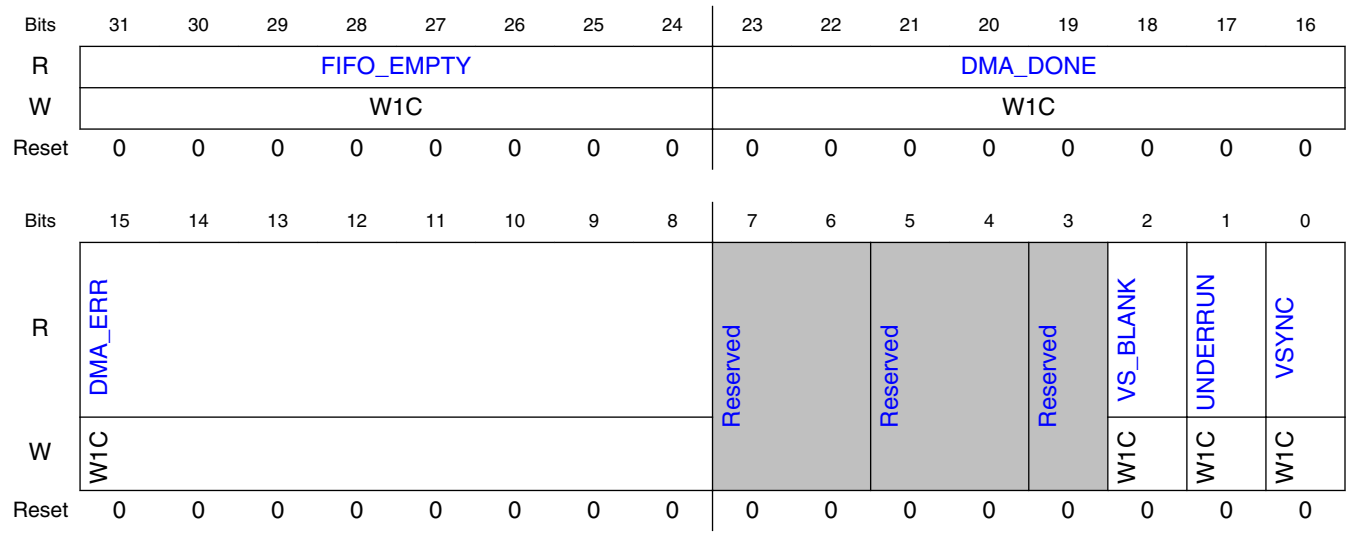
48.6.1.7.1 Offset

Register	Offset
INT_STATUS_D0	20h

48.6.1.7.2 Function

Interrupt Status Register for domain 0

48.6.1.7.3 Diagram



48.6.1.7.4 Fields

Field	Function
31-24 FIFO_EMPTY	Interrupt flag to indicate that which FIFO in the pixel blending underflowed • bit 0: FIFO1 underflow • bit 1: FIFO2 underflow • bit 2: FIFO3 underflow • bit 3: FIFO4 underflow • bit 4: FIFO5 underflow • bit 5: FIFO6 underflow • bit 6: FIFO7 underflow • bit 7: FIFO8 underflow
23-16 DMA_DONE	Interrupt flag to indicate that which PLANE has fetched the last pixel from memory • bit 0: PLANE1 • bit 1: PLANE2 • bit 2: PLANE3 • bit 3: PLANE4 • bit 4: PLANE5 • bit 5: PLANE6 • bit 6: PLANE7 • bit 7: PLANE8
15-8 DMA_ERR	Interrupt flag to indicate that which PLANE has Read Error on the AXI interface • bit 0: PLANE1 Read error • bit 1: PLANE2 Read error • bit 2: PLANE3 Read error • bit 3: PLANE4 Read error • bit 4: PLANE5 Read error • bit 5: PLANE6 Read error • bit 6: PLANE7 Read error • bit 7: PLANE8 Read error

Table continues on the next page...

Field	Function
7-6 —	Reserved
5-4 —	Reserved
3 —	Reserved
2 VS_BLANK	Interrupt flag to indicate vertical blanking period 0b - Vertical blanking period has not started 1b - Vertical blanking period has started
1 UNDERRUN	Interrupt flag to indicate the output buffer underrun condition 0b - Output buffer not underrun 1b - Output buffer underrun
0 VSYNC	Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame) 0b - VSYNC has not started 1b - VSYNC has started

48.6.1.8 Interrupt Enable Register for domain 0 (INT_ENABLE_D0)

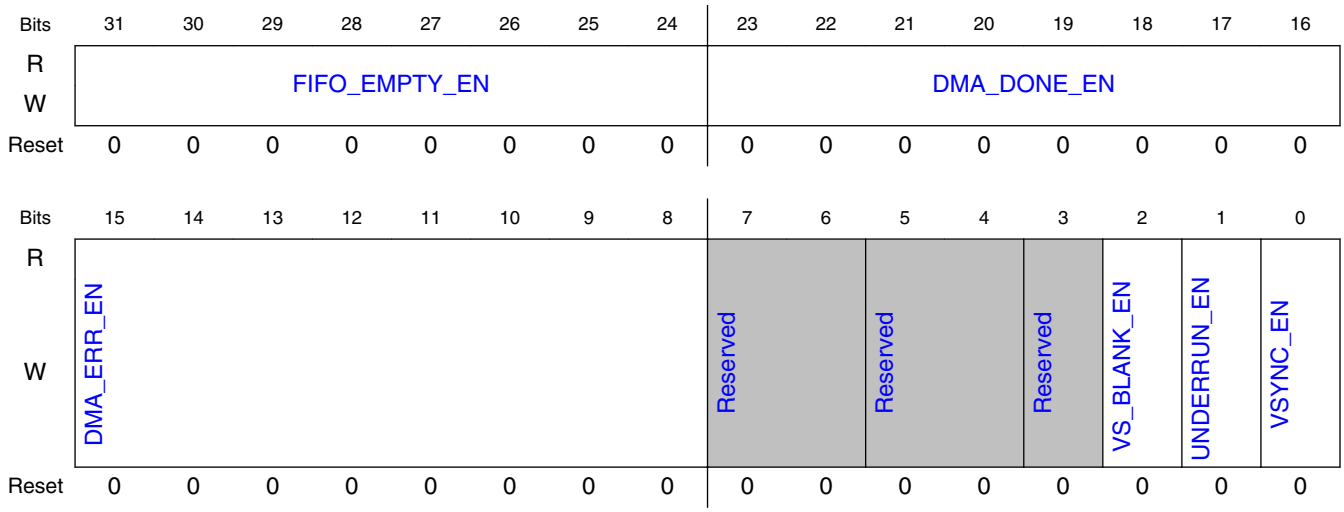
48.6.1.8.1 Offset

Register	Offset
INT_ENABLE_D0	24h

48.6.1.8.2 Function

Interrupt Enable Register for domain 0

48.6.1.8.3 Diagram



48.6.1.8.4 Fields

Field	Function
31-24 FIFO_EMPTY_EN	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed • bit 0: FIFO1 underflow enable • bit 1: FIFO2 underflow enable • bit 2: FIFO3 underflow enable • bit 3: FIFO4 underflow enable • bit 4: FIFO5 underflow enable • bit 5: FIFO6 underflow enable • bit 6: FIFO7 underflow enable • bit 7: FIFO8 underflow enable
23-16 DMA_DONE_EN	Enable Interrupt flag to indicate that which PLANE has fetched the last pixel from memory • bit 0: PLANE1 fetch done enable • bit 1: PLANE2 fetch done enable • bit 2: PLANE3 fetch done enable • bit 3: PLANE4 fetch done enable • bit 4: PLANE5 fetch done enable • bit 5: PLANE6 fetch done enable • bit 6: PLANE7 fetch done enable • bit 7: PLANE8 fetch done enable
15-8 DMA_ERR_EN	Enable Interrupt flag to indicate that which PLANE has Read Error on the AXI interface • bit 0: PLANE1 Read Error enable • bit 1: PLANE2 Read Error enable • bit 2: PLANE3 Read Error enable • bit 3: PLANE4 Read Error enable • bit 4: PLANE5 Read Error enable • bit 5: PLANE6 Read Error enable • bit 6: PLANE7 Read Error enable • bit 7: PLANE8 Read Error enable
7-6	Reserved

Table continues on the next page...

Field	Function
—	
5-4 —	Reserved
3 —	Reserved
2 VS_BLANK_EN	Enable Interrupt flag to indicate vertical blanking period 0b - Vertical blanking start interrupt disable 1b - Vertical blanking start interrupt enable
1 UNDERRUN_EN	Enable Interrupt flag to indicate the output buffer underrun condition 0b - Output buffer underrun disable 1b - Output buffer underrun enable
0 VSYNC_EN	Enable Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame) 0b - VSYNC interrupt disable 1b - VSYNC interrupt enable

48.6.1.9 Interrupt Status Register for domain 1 (INT_STATUS_D1)

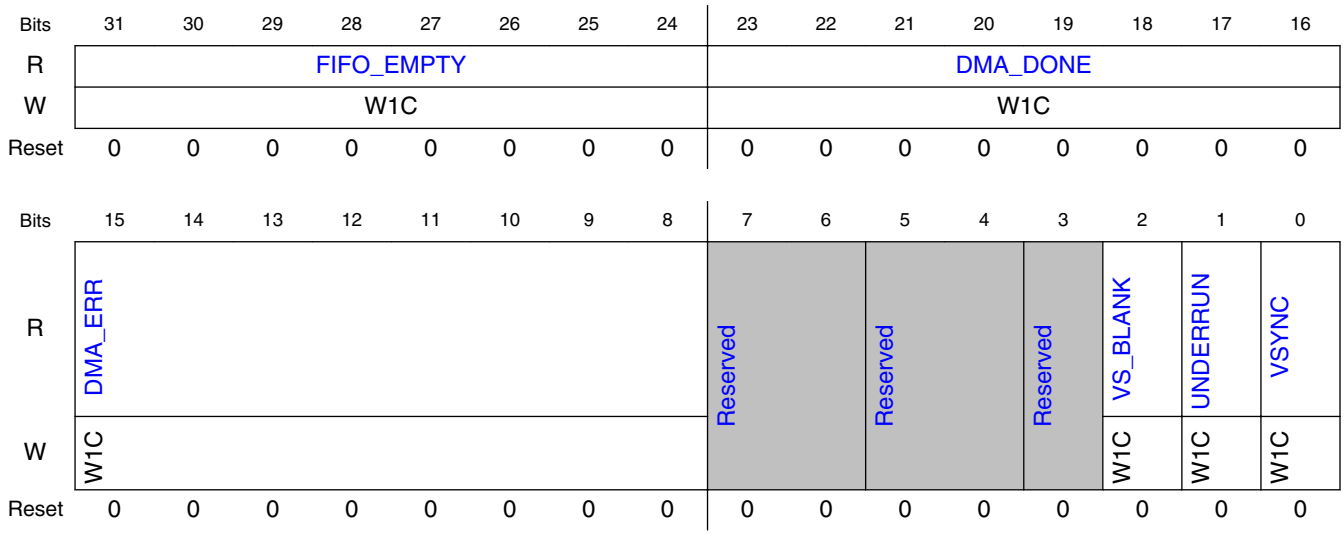
48.6.1.9.1 Offset

Register	Offset
INT_STATUS_D1	30h

48.6.1.9.2 Function

Interrupt Status Register for domain 1

48.6.1.9.3 Diagram



48.6.1.9.4 Fields

Field	Function
31-24 FIFO_EMPTY	Interrupt flag to indicate that which FIFO in the pixel blending underflowed • bit 0: FIFO1 • bit 1: FIFO2 • bit 2: FIFO3 • bit 3: FIFO4 • bit 4: FIFO5 • bit 5: FIFO6 • bit 6: FIFO7 • bit 7: FIFO8
23-16 DMA_DONE	Interrupt flag to indicate that which PLANE has fetched the last pixel from memory • bit 0: PLANE1 • bit 1: PLANE2 • bit 2: PLANE3 • bit 3: PLANE4 • bit 4: PLANE5 • bit 5: PLANE6 • bit 6: PLANE7 • bit 7: PLANE8
15-8 DMA_ERR	Interrupt flag to indicate that which PLANE has Read Error on the AXI interface • bit 0: PLANE1 Read error • bit 1: PLANE2 Read error • bit 2: PLANE3 Read error • bit 3: PLANE4 Read error • bit 4: PLANE5 Read error • bit 5: PLANE6 Read error • bit 6: PLANE7 Read error • bit 7: PLANE8 Read error
7-6	Reserved

Table continues on the next page...

Field	Function
—	
5-4 —	Reserved
3 —	Reserved
2 VS_BLANK	Interrupt flag to indicate vertical blanking period
1 UNDERRUN	Interrupt flag to indicate the output buffer underrun condition
0 VSYNC	Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame)

48.6.1.10 Interrupt Enable Register for domain 1 (INT_ENABLE_D1)

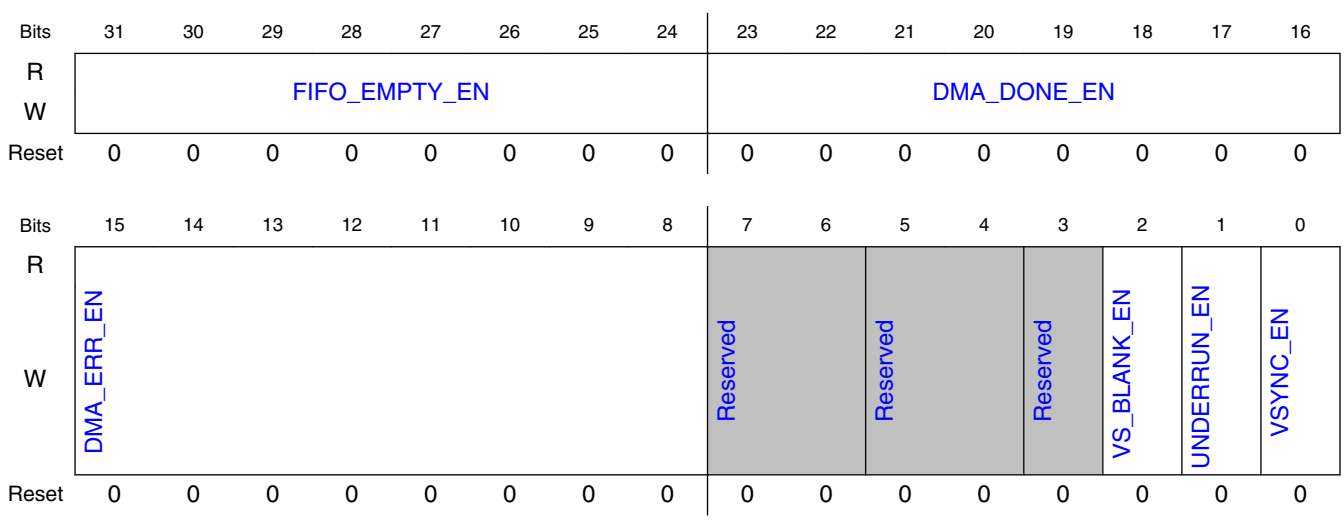
48.6.1.10.1 Offset

Register	Offset
INT_ENABLE_D1	34h

48.6.1.10.2 Function

Interrupt Enable Register for domain 1

48.6.1.10.3 Diagram



48.6.1.10.4 Fields

Field	Function
31-24 FIFO_EMPTY_EN	Enable Interrupt flag to indicate that which FIFO in the pixel blending underflowed • bit 0: FIFO1 underflowed enable • bit 1: FIFO2 underflowed enable • bit 2: FIFO3 underflowed enable • bit 3: FIFO4 underflowed enable • bit 4: FIFO5 underflowed enable • bit 5: FIFO6 underflowed enable • bit 6: FIFO7 underflowed enable • bit 7: FIFO8 underflowed enable
23-16 DMA_DONE_EN	Enable Interrupt flag to indicate that which PLANE has fetched the last pixel from memory • bit 0: PLANE1 fetched done enable • bit 1: PLANE2 fetched done enable • bit 2: PLANE3 fetched done enable • bit 3: PLANE4 fetched done enable • bit 4: PLANE5 fetched done enable • bit 5: PLANE6 fetched done enable • bit 6: PLANE7 fetched done enable • bit 7: PLANE8 fetched done enable
15-8 DMA_ERR_EN	Enable Interrupt flag to indicate that which PLANE has Read Error on the AXI interface • bit 0: PLANE1 Read Error enable • bit 1: PLANE2 Read Error enable • bit 2: PLANE3 Read Error enable • bit 3: PLANE4 Read Error enable • bit 4: PLANE5 Read Error enable • bit 5: PLANE6 Read Error enable • bit 6: PLANE7 Read Error enable • bit 7: PLANE8 Read Error enable
7-6 —	Reserved
5-4 —	Reserved
3 —	Reserved
2 VS_BLANK_EN	Enable Interrupt flag to indicate vertical blanking period
1 UNDERRUN_EN	Enable Interrupt flag to indicate the output buffer underrun condition
0 VSYNC_EN	Enable Interrupt flag to indicate that the vertical synchronization phase(The beginning of a frame)

48.6.1.11 Reserved (PDI_PARA)

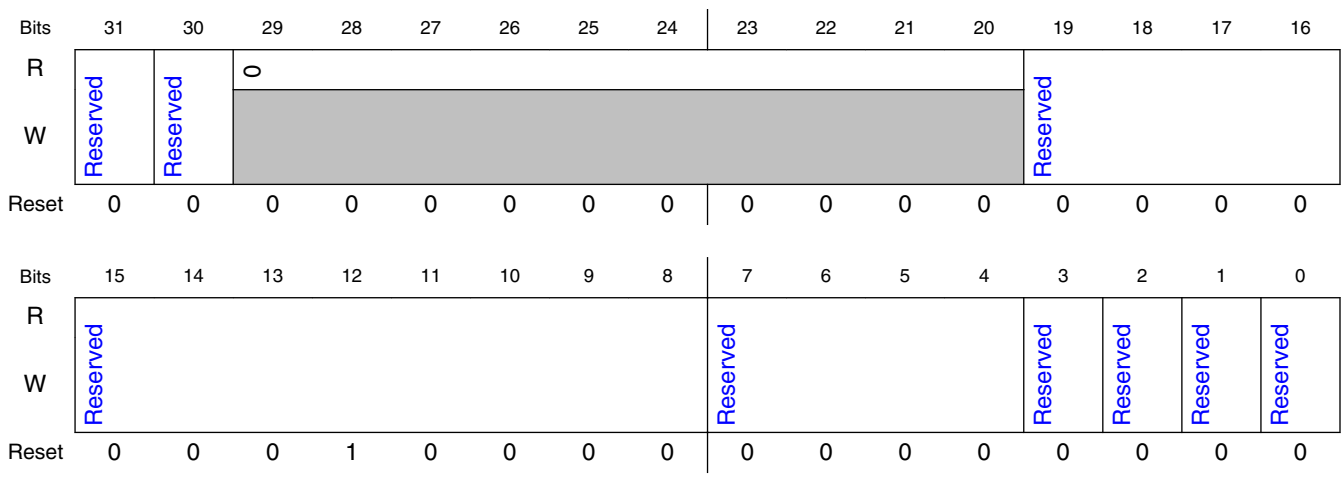
48.6.1.11.1 Offset

Register	Offset
PDI_PARA	40h

48.6.1.11.2 Function

This register is reserved for this SoC

48.6.1.11.3 Diagram



48.6.1.11.4 Fields

Field	Function
31 —	Reserved Software must not change the value of this field
30 —	Reserved Software must not change the value of this field
29-20 —	Reserved
19-8 —	Reserved Software must not change the value of this field
7-4 —	Reserved Software must not change the value of this field

Table continues on the next page...

Field	Function
3 —	Reserved Software must not change the value of this field
2 —	Reserved Software must not change the value of this field
1 —	Reserved Software must not change the value of this field
0 —	Reserved Software must not change the value of this field

48.6.1.12 Control Descriptor Layer 1 Register (CTRLDESCL0_1 - CTRLDESCL7_1)

48.6.1.12.1 Offset

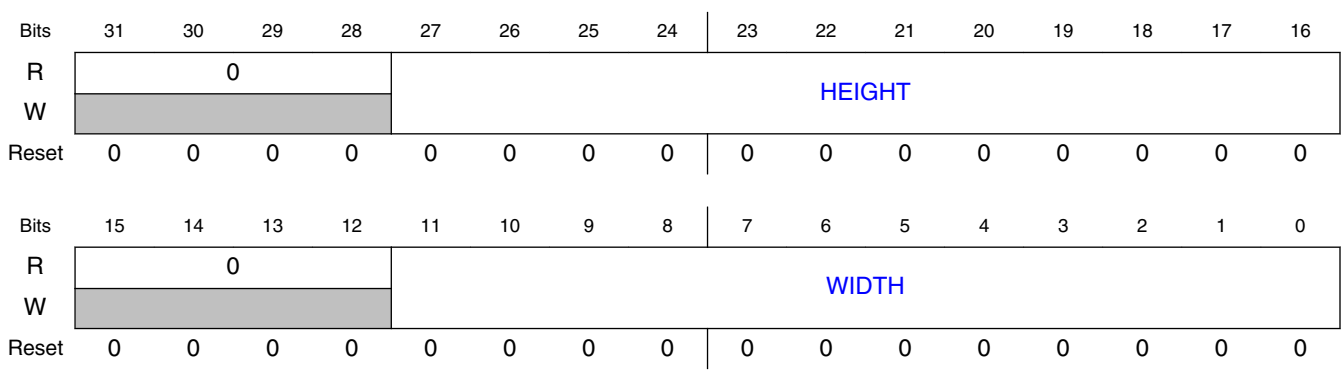
For a = 0 to 7:

Register	Offset
CTRLDESCLa_1	200h + (a × 40h)

48.6.1.12.2 Function

This register sets the height and width of the layer associated with the register

48.6.1.12.3 Diagram



48.6.1.12.4 Fields

Field	Function
31-28 —	Reserved
27-16 HEIGHT	Height of the layer in pixels
15-12 —	Reserved
11-0 WIDTH	Width of the layer in pixels The layer width must be in multiples of the number of pixels that can be stored in 32 bits, and therefore differs depending on color encoding. For example, if 2 bits per pixel format is used, then the layer width must be configured in multiples of 16

48.6.1.13 Control Descriptor Layer 2 Register (CTRLDESCL0_2 - CTRLDESCL7_2)

48.6.1.13.1 Offset

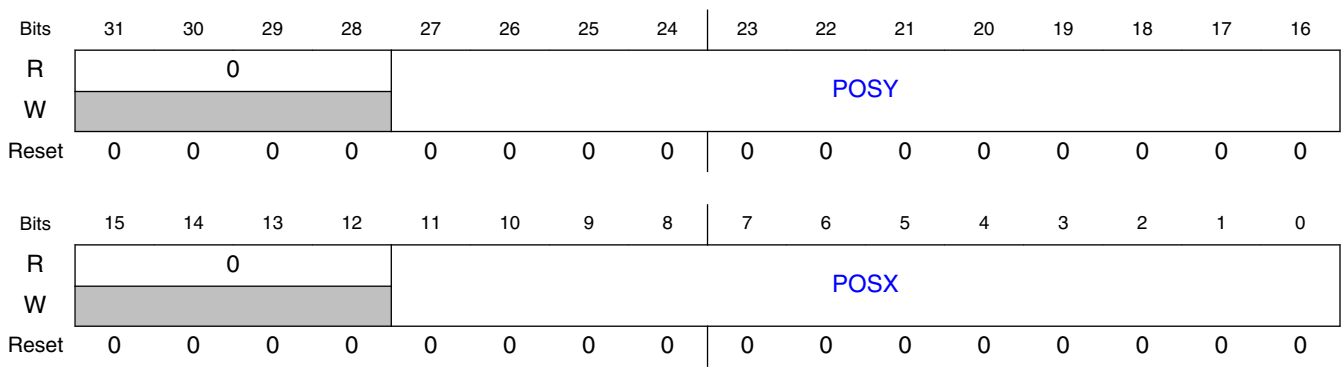
For a = 0 to 7:

Register	Offset
CTRLDESCLa_2	204h + (a × 40h)

48.6.1.13.2 Function

This register sets the origin (top/left) of the layer associated with the register

48.6.1.13.3 Diagram



48.6.1.13.4 Fields

Field	Function
31-28 —	Reserved
27-16 POSY	The vertical position of top row of the layer, where 0 is the top row of the panel, only positive values are below the top row of the panel
15-12 —	Reserved
11-0 POSX	POS X The horizontal position of left-hand column of the layer, where 0 is the left-hand column of the panel, only positive values are to the right the left-hand column of the panel

48.6.1.14 Control Descriptor Layer 3 Register (CTRLDESCL0_3 - CTRLDESCL7_3)

48.6.1.14.1 Offset

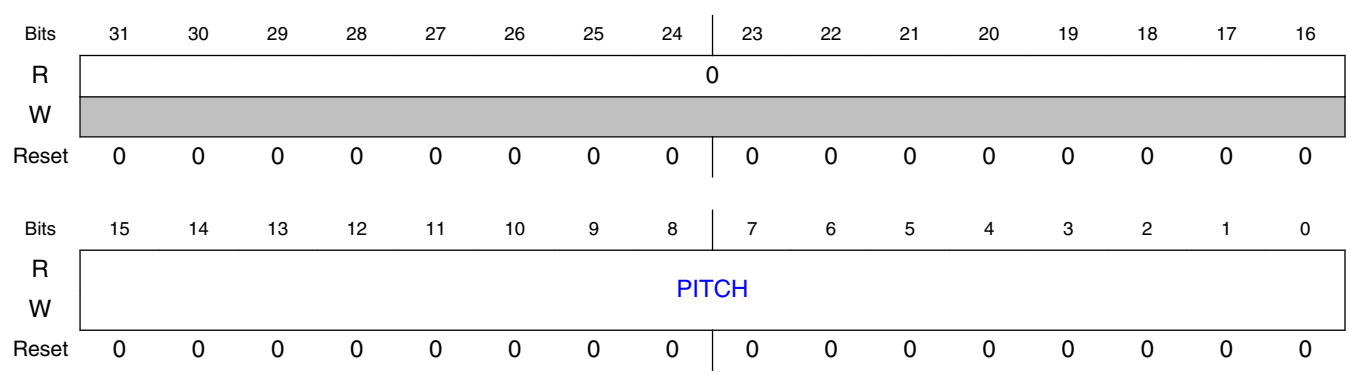
For a = 0 to 7:

Register	Offset
CTRLDESCLa_3	208h + (a × 40h)

48.6.1.14.2 Function

This register sets the buffer line pitch of layer data

48.6.1.14.3 Diagram



48.6.1.14.4 Fields

Field	Function
31-16 —	Reserved
15-0 PITCH	Number of bytes between 2 vertically adjacent pixels in system memory. Byte granularity is supported, but SW should align to 64B boundry

48.6.1.15 Control Descriptor Layer 4 Register (CTRLDESCL0_4 - CTRLDESCL7_4)

48.6.1.15.1 Offset

For a = 0 to 7:

Register	Offset
CTRLDESCLa_4	20Ch + (a × 40h)

48.6.1.15.2 Function

This register sets the beginning address of layer data

48.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

48.6.1.15.4 Fields

Field	Function
31-0 ADDR	Address of layer data in the memory. The address programmed should be 64-bit aligned

48.6.1.16 Control Descriptor Layer 5 Register (CTRLDESCL0_5 - CTRLDESCL7_5)

48.6.1.16.1 Offset

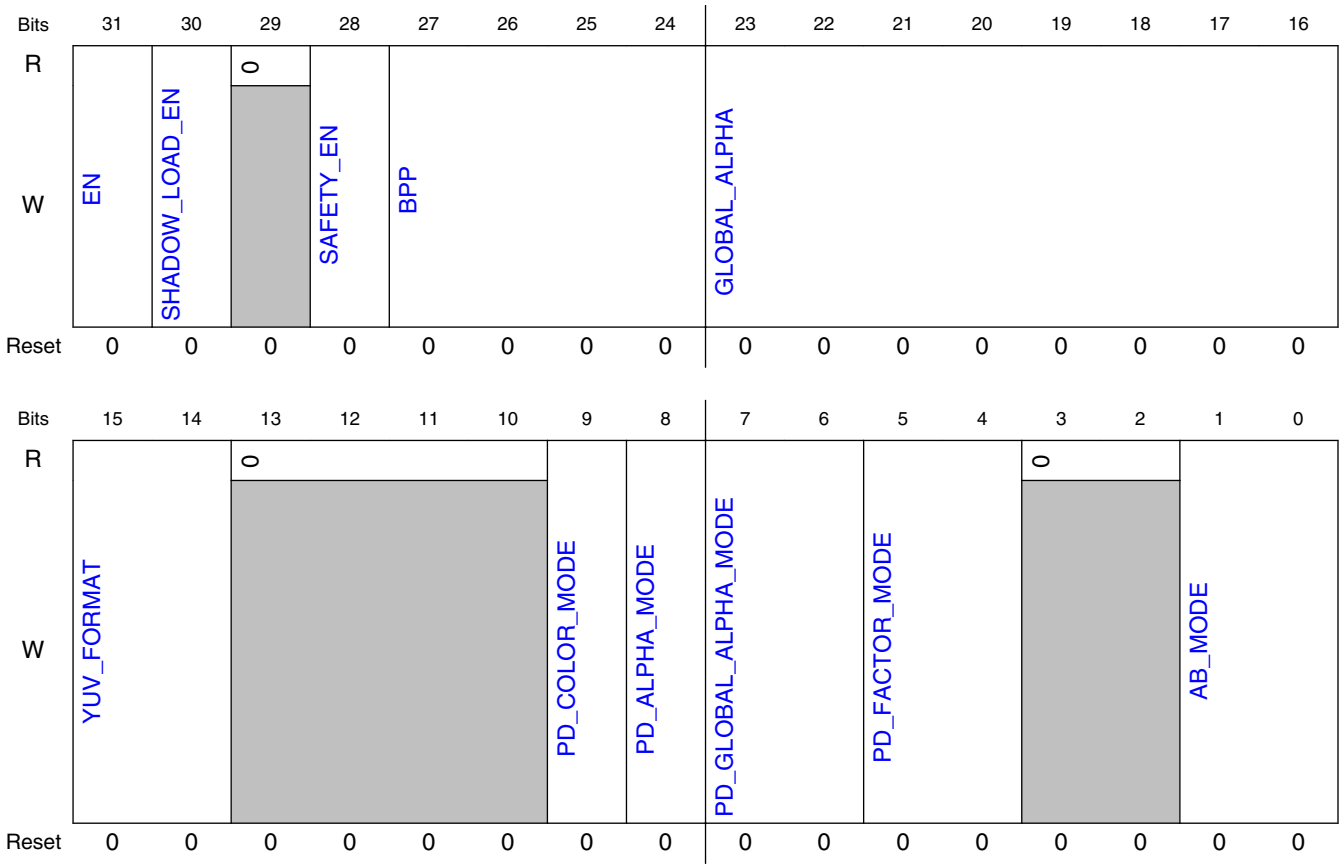
For a = 0 to 7:

Register	Offset
CTRLDESCLa_5	210h + (a × 40h)

48.6.1.16.2 Function

This register controls various graphics options and whether the layer is enabled

48.6.1.16.3 Diagram



48.6.1.16.4 Fields

Field	Function
31 EN	Enable the layer for DMA 0b - OFF 1b - ON
30 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all DMA control registers are written. If set to 1, shadowed control registers are updated to the active control registers on VSYNC of next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters
29 —	Reserved
28 SAFETY_EN	Safety Mode Enable Bit 0b - Safety Mode is disabled 1b - Safety Mode is enabled for this layer
27-24 BPP	Layer encoding format (bit per pixel) 0000b - 1 bpp 0001b - 2 bpp

Table continues on the next page...

Field	Function
	0010b - 4 bpp 0011b - 8 bpp 0100b - 16 bpp (RGB565) 0101b - 16 bpp (ARGB1555) 0110b - 16 bpp (ARGB4444) 0111b - YCbCr422 (Only layer 0/1 can support this format) 1000b - 24 bpp (RGB888) 1001b - 32 bpp (ARGB8888) 1010b - 32 bpp (ABGR8888)
23-16 GLOBAL_ALPHA	Global Alpha When AB_MODE=01b, this is the legacy global alpha When AB_MODE=11b, this is the global alpha used in Porter Duff function
15-14 YUV_FORMAT	The YUV422 input format selection 00b - The YVYU422 8bit sequence is U1,Y1,V1,Y2 01b - The YVYU422 8bit sequence is V1,Y1,U1,Y2 10b - The YVYU422 8bit sequence is Y1,U1,Y2,V1 11b - The YVYU422 8bit sequence is Y1,V1,Y2,U1
13-10 —	Reserved
9 PD_COLOR_MODE	Porter Duff alpha mode 0b - Straight mode for Porter Duff color 1b - Inversed mode for Porter Duff color
8 PD_ALPHA_MODE	Porter Duff alpha mode 0b - Straight mode for Porter Duff alpha 1b - Inversed mode for Porter Duff alpha
7-6 PD_GLOBAL_ALPHA_MODE	Porter Duff global alpha mode 00b - Using global alpha 01b - Using local alpha 10b - Using scaled alpha 11b - Using scaled alpha
5-4 PD_FACTOR_MODE	Porter Duff factor mode 00b - Using 1 01b - Using 0 10b - Using straight alpha 11b - Using inverse alpha
3-2 —	Reserved
1-0 AB_MODE	Alpha Blending Mode 00b - No alpha Blending (The SAFETY_EN bit need set to 1) 01b - Blend with global ALPHA 10b - Blend with embedded ALPHA 11b - Blend with Porter Duff enable

48.6.1.17 Control Descriptor Layer 6 Register (CTRLDESCL0_6 - CTRLDESCL7_6)

48.6.1.17.1 Offset

For a = 0 to 7:

Register	Offset
CTRLDESCLa_6	214h + (a × 40h)

48.6.1.17.2 Function

Control Descriptor Layer 6 Register

48.6.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								BCLR_R							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BCLR_G								BCLR_B							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

48.6.1.17.4 Fields

Field	Function
31-24 —	Reserved
23-16 BCLR_R	Background R component value
15-8 BCLR_G	Background G component value
7-0 BCLR_B	Background B component value

48.6.1.18 Color Space Conversion Coefficient Register 0 (CSC0_COEF0 - CSC1_COEF0)

48.6.1.18.1 Offset

Register	Offset
CSC0_COEF0	218h
CSC1_COEF0	258h

48.6.1.18.2 Function

The Coefficient 0 register contains coefficients used in the color space conversion algorithm. The Y and UV offsets are added to the source buffer to normalize them before the conversion. C0 is the coefficient that is used to multiply the luma component of the data for all three RGB components

// The equations used for Colorspace conversion are:

// R = C0*(Y+YOFFSET) + C1(V+UV_OFFSET)

// G = C0*(Y+YOFFSET) + C3(U+UV_OFFSET) + C2(V+UV_OFFSET)

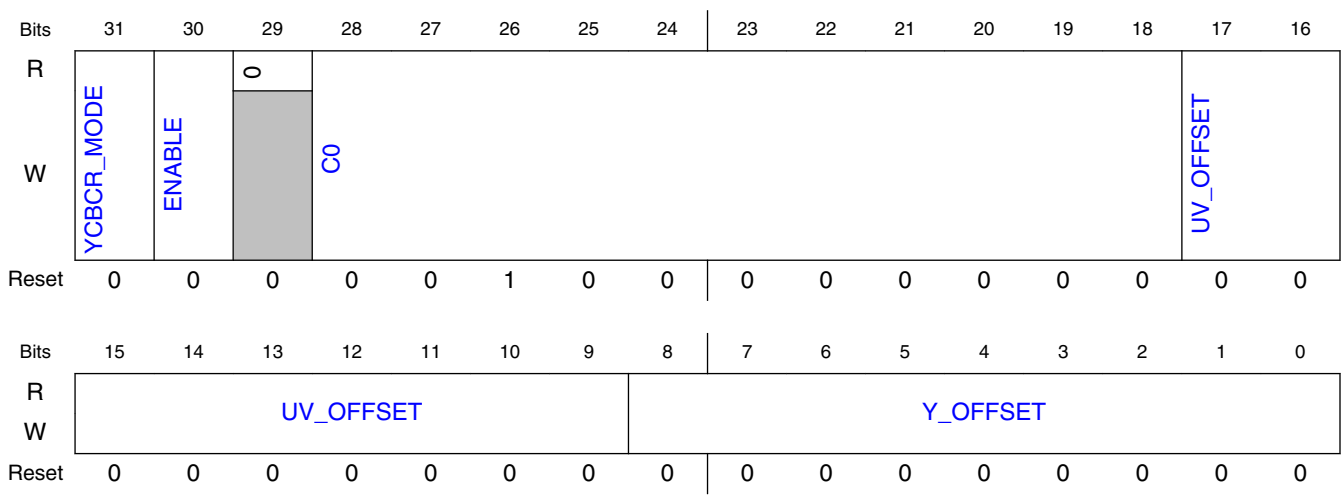
// B = C0*(Y+YOFFSET) + C4(U+UV_OFFSET)

REG_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset

REG_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4

REG_CSCCOEF2_WR(0x076B079C); // YUV coefficients: C2, C3

48.6.1.18.3 Diagram



48.6.1.18.4 Fields

Field	Function
31 YCBCR_MODE	This bit changes the behavior when performing U/V converting 0b - Converting YUV to RGB data 1b - Converting YCbCr to RGB data
30 ENABLE	Enable the CSC unit in the LCDIFv2 plane data path 0b - The CSC is bypassed and the input pixels are RGB data already 1b - The CSC is enabled and the pixels will be converted to RGB data
29 —	Reserved
28-18 C0	Two's compliment Y multiplier coefficient. YUV=0x100 (1.000) YCbCr=0x12A (1.164)
17-9 UV_OFFSET	UV OFFSET Two's compliment phase offset implicit for CbCr data. Generally used for YCbCr to RGB conversion. YCbCr=0x180, YUV=0x000 (typically -128 or 0x180 to indicate normalized -0.5 to 0.5 range)
8-0 Y_OFFSET	Y OFFSET Two's compliment amplitude offset implicit in the Y data. For YUV, this is typically 0 and for YCbCr, this is typically -16 (0x1F0)

48.6.1.19 Color Space Conversion Coefficient Register 1 (CSC0_COEF1 - CSC1_COEF1)

48.6.1.19.1 Offset

Register	Offset
CSC0_COEF1	21Ch
CSC1_COEF1	25Ch

48.6.1.19.2 Function

The Coefficient 1 register contains coefficients used in the color space conversion algorithm. C1 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the red component. C4 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the blue component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal

48.6.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					C1										
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					C4										
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

48.6.1.19.4 Fields

Field	Function
31-27 —	Reserved
26-16 C1	Two's compliment Red V/Cr multiplier coefficient. YUV=0x123 (1.140) YCbCr=0x198 (1.596)
15-11 —	Reserved
10-0 C4	Two's compliment Blue U/Cb multiplier coefficient. YUV=0x208 (2.032) YCbCr=0x204 (2.017)

48.6.1.20 Color Space Conversion Coefficient Register 2 (CSC0_COEF2 - CSC1_COEF2)

48.6.1.20.1 Offset

Register	Offset
CSC0_COEF2	220h
CSC1_COEF2	260h

48.6.1.20.2 Function

The Coefficient 2 register contains coefficients used in the color space conversion algorithm. C2 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the green component. C3 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the green component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal

48.6.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					C2										
W																
Reset	0	0	0	0	0	1	1	1	0	1	1	0	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					C3										
W																
Reset	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0

48.6.1.20.4 Fields

Field	Function
31-27 —	Reserved
26-16 C2	Two's compliment Green V/Cr multiplier coefficient. YUV=0x76B (-0.581) YCbCr=0x730 (-0.813)
15-11 —	Reserved
10-0 C3	Two's compliment Green U/Cb multiplier coefficient. YUV=0x79C (-0.394) YCbCr=0x79C (-0.392)

48.6.1.21 LCDIFv2 CLUT load Register (CLUT_LOAD)

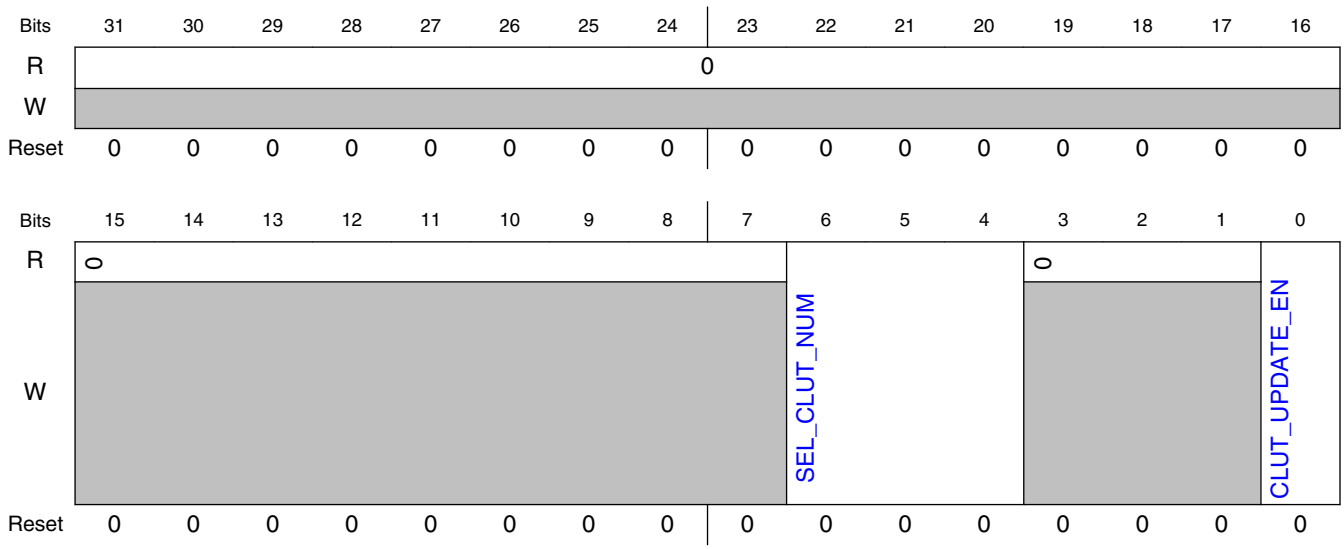
48.6.1.21.1 Offset

Register	Offset
CLUT_LOAD	400h

48.6.1.21.2 Function

LCDIFv2 CLUT load Register

48.6.1.21.3 Diagram



48.6.1.21.4 Fields

Field	Function
31-7 —	Reserved
6-4 SEL_CLUT_NUM	Selected CLUT Number The SEL_CLUT_NUM is used to select which plane's CLUT need to be updated. The hardware can only backup one CLUT setting and load, so the SEL_CLUT_NUM can't be changed between setup of CLUT_UPDATE_EN and SHADOW_LOAD_EN. • 3'h0 - PLANE 0 • 3'h1 - PLANE 1 • ------ • 3'h7 - PLANE 7
3-1 —	Reserved
0 CLUT_UPDATE_EN	CLUT Update Enable The CLUT_UPDATE_EN bit is written to 1 when software want to update the Color Look Up Tables during display. If set to 1, software update selected CLUT due to SEL_CLUT_NUM setting, the table will be loaded during vertical blanking period after SHADOW_LOAD_EN is set to 1. If set to 0, software update CLUT directly according to the CLUT memory map. Hardware will automatically clear this bit when selected CLUT is updated according to SEL_CLUT_NUM.

Chapter 49
MIPI DSI Host Controller (MIPI_DSI)

49.1 Chip-specific MIPI DSI information

Table 49-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

The dpi_pclk can be sourced from either eLCDIF or LCDIFv2 depending on the configuration in the Video Multiplexer (VIDEO_MUX) block. For detailed information, refer to VID_MUX_CTRL[MIPI_DSI_SEL] field in Video Multiplexer Memory Map and register definition.

49.2 Overview

The Mobile Industry Processor Interface (MIPI) Display Serial Interface (DSI) controller implements all protocol functions defined in the MIPI DSI Specification. The MIPI DSI controller provides an interface that allows communication with MIPI DSI-compliant peripherals.

The MIPI DSI D-PHY is a source-synchronous physical layer supporting the MIPI Alliance standard for D-PHY.

49.2.1 Block Diagram

The figure below illustrates the DSI Host Controller Core structure. The DSI Host Controller Core operates on the host (transmit) side of a DSI link.

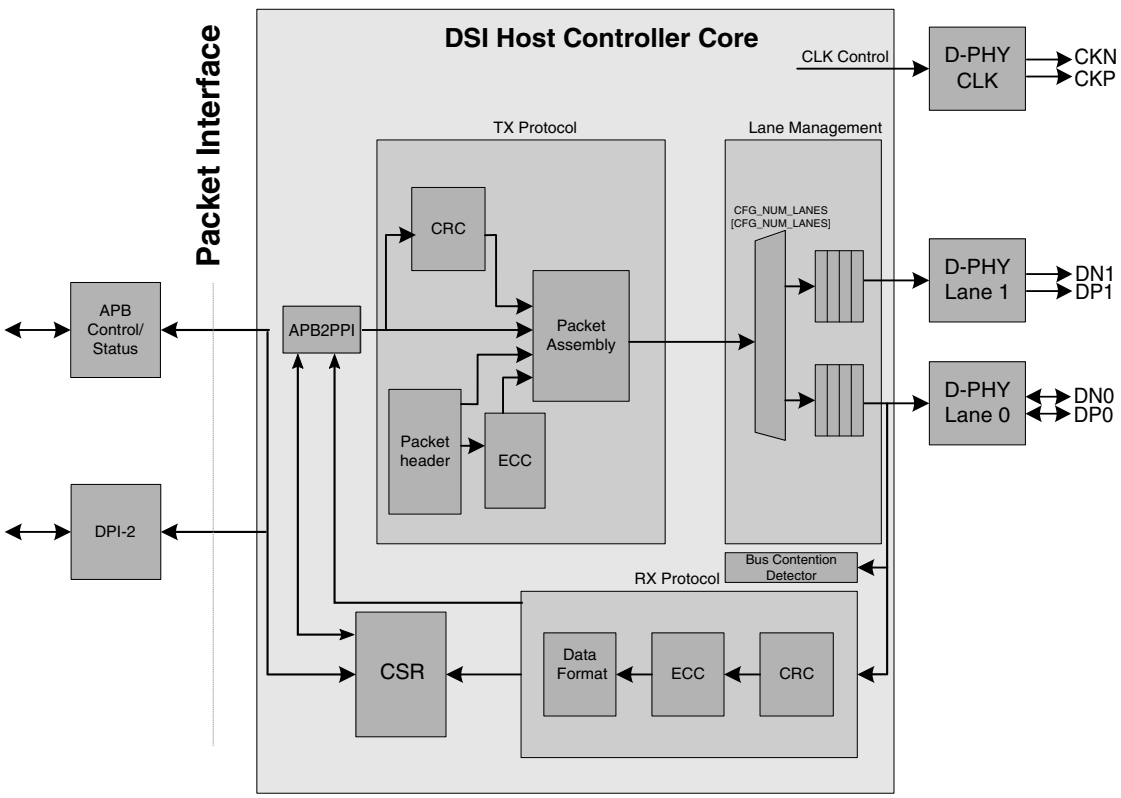


Figure 49-1. DSI Host Controller Core Block Diagram

The DSI Host Controller Core implements all three layers defined by the DSI Specification: Pixel to Byte Packing (handled by DPI Interface), Low Level Protocol, and Lane Management. The DSI Host Controller Core sends and receives DSI commands via the Packet Interface.

The Packet Interface is an easy-to-use data interface that accepts commands and data, and sends it over the DSI link. It supports the use of 2 D-PHY lanes. The DSI Host Controller Core takes care of all packet formatting details and transmission over the MIPI bus.

The DPI Translator connects to the DSI Host Controller Core via the Packet Interface. DPI masters may connect directly to the DPI Translator to send commands across the DSI link.

Sending commands and receiving information through the DSI link is also possible through the APB to PPI gasket. See MIPI DSI Host APB PKT IF Memory Map/Register section for details about the DSI registers which allow the APB to PPI setup and communication.

49.2.2 Features

Key features of the MIPI DSI Controller Core include:

- Implements all three DSI Layers (Pixel to Byte packing, Low Level Protocol, Lane Management)
- Support for Command and Video Modes
- Host Version
- Scalable data lane support, 1 to 2 Data Lanes
  - Bidirectional support on lane 0
- Supports High Speed and Low Power operation
- Support for all DSI data types and formats
- Supports ULPS mode
- Full Low-Level Protocol Error and Contention detection and reporting
- Supports continuous and non-continuous Clock Lane operation
- Supports multiple packets per transmission
- Support for all three Video Mode packet sequences
  - Non-Burst Mode with Sync Pulses
  - Non-Burst Mode with Sync Events
  - Burst mode
- Support for bus turnaround signaling
- Flexible packet based user interface
  - APB interface (status and control)
  - Display Pixel Interface Core (DPI-2)
- Supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs
- MIPI Alliance Specification for Display Serial Interface Version 1.1 compliant

The DSI Host Controller DPI-2 Interface Core provides the following features:

- Support for Type 2,3, and 4 displays
- Support for 16, 18 and 24 bit Pixel data and all alignment configurations
- Support for 16, 18, 24, and 18 bit loosely packed DSI data types
- Supports DSI video modes
  - Non-Burst Mode with Sync Pulses
  - Non-Burst Mode with Sync Events
  - Burst Mode
- Supports normal or inverted HSYNC and VSYNC signals

- Handles clock domain crossing from DPI Pixel clock to the Host controller TX Byte clock
- Interfaces directly to the Host Controller's DSI Packet Interface
- Comes already integrated with the DSI Host Controller

The MIPI DSI D-PHY supports the MIPI Standard for D-PHY. It features one clock lane, two data lanes, and high speed Serializers and Deserializers. The D-PHY also supports high speed and low power modes.

49.3 Functional description

49.3.1 Packet Interface

The DSI Host Controller Core Packet Interface consists of Transmit, Receive, and Control and Status sections. Through these interfaces, the user application can take complete control of the DSI interface, sending all video timing, sending DSI commands, receiving DSI reads, monitoring the status of the interface and responding to error reporting.

User application may have access to DSI Host Controller Core Packet Interface through APB_PKT_IF registers. For information on how to send a packet or receive a read request for a packet, refer to Application Information

49.3.1.1 Transmit Ports

The Transmit Packet Interface is the mechanism with which the user creates packets to send over the MIPI Interface.

For Long Packets, the user provides the Data Type (DT) and Word Count (WC) to the controller. The Controller then creates a packet header and pulls the packet data from the Packet Interface and out to the D-PHY to transmit.

For Short Packets, the user provides the Data Type (DT) and required parameters (if any) to the controller. The Controller then creates the short packet and sends it to the D-PHY to transmit. This interface enables the user application to transmit and receive any type of DSI packet.

49.3.1.2 Receive Ports

The Receive Packet Interface returns data from the Peripheral to the user. The user is provided the Data Type (DT), Word Count (WC), and Rx Payload Data.

This interface enables the user application to receive any type of DSI packet.

49.3.2 Status Interface Ports

The Status Interface Core provides for reading the Status of the DSI Host Controller Core via a simple interface. All signals in the Status Interface are synchronous to the clk_byte clock.

NOTE

The Packet Interface signals are accessible by user application through the following DSI_HOST device registers:

- CFG_STATUS_OUT
- CFG_HTX_TO_COUNT
- CFG_LRX_H_TO_COUNT

Refer to DSI_HOST device sections for details.

49.3.3 Interface Cores

There are several interface cores for the DSI Host Controller Core. These interface cores provide industry standard interfaces to the DSI Host Controller Core to simplify interfacing and integration the DSI Host Controller Core to your application. When selected, these interface cores provide additional ports to the DSI Host Controller core.

49.3.3.1 DPI-2 Interface Host Bridge Core

The DPI-2 Interface Host Bridge Core provides a DPI-2 compliant interface to the DSI Host controller. The DPI-2 Interface Host Bridge Core handles converting the DPI-2 interface into a packet format that is compatible with the DSI Host Controller Core's Packet Interface, and that adheres to the DSI specification. The DPI-2 Interface Core handles all pixel-to-byte packing for all DPI data types.

A block diagram of the DPI-2 Interface Host Bridge Core is shown below. The DPI-2 Host Bridge Core accepts a MIPI compliant DPI-2 set of signals (vsync, hsync, pixel data), creates DSI packets for the video timing events and video data and transmits those packets via the DSI Host Controller's Packet Interface.

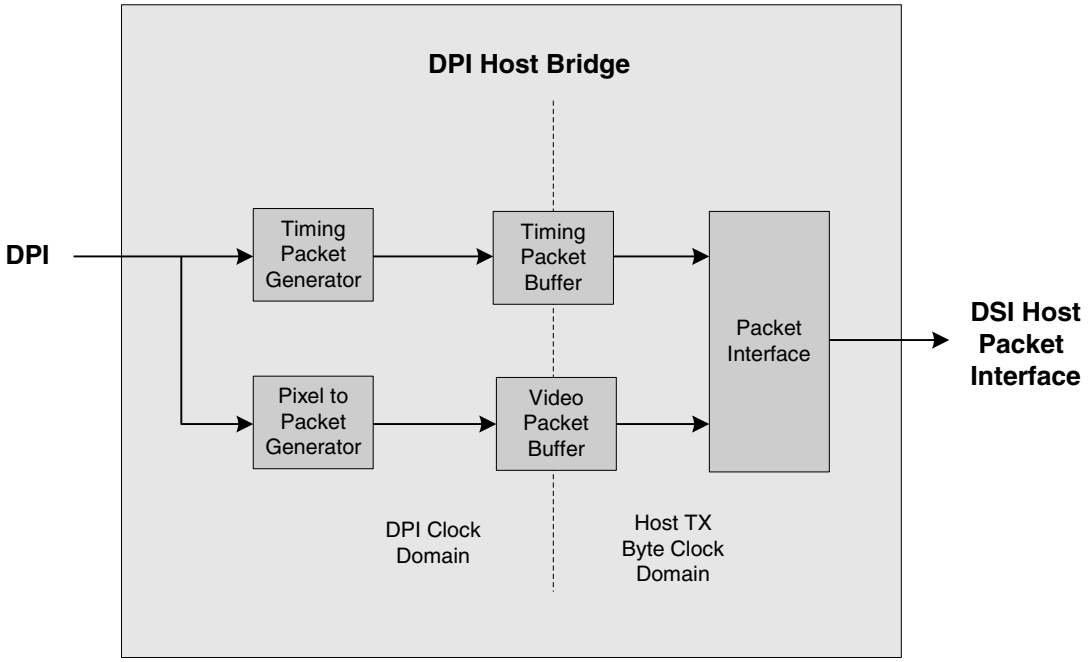


Figure 49-2. DPI-2 Host Bridge Core Block Diagram

49.3.3.1.1 DPI-2 Interface Ports

The DPI-2 Interface module provides a MIPI compliant DPI-2 interface to the DSI Host Controller Core. In addition to the standard DPI-2 interface signals there are several configuration and status ports that allow the user to adjust the behavior of the DPI-2 Interface module to better suit a particular application.

NOTE

The DPI configuration signals herein described are accessible through DSI_HOST_DPI_INTFC device registers (please refer to MIPI DSI Host Memory Map/Register section for details).

49.3.3.1.2 DPI-2 Video Timing Recreation over DSI

The DSI Host DPI-2 module is designed to preserve the receive video timing by properly spacing the DSI packets that carry the DPI video information and by inserting blanking packets into the packet stream. In order to achieve this goal, the DSI Host controller needs to have the cfg_host_dpi_hfp, cfg_host_dpi_hbp, and cfg_host_dpi_hsa ports properly set. The values of these ports reflect the size of the hfp, hbp, and has in terms of DSI packet bytes, values such that the number of DSI bytes when transmitted takes approximately the same amount of absolute time as the video event took on the DPI interface.

To relate pixel sizes on the DPI interface to DSI packet bytes, use the following relationship:

Time of DPI event - time to transmit x number of bytes on the DSI interface

dpi_event_size * dpi_pclk_period = number_of_dsi_bytes * 8 * dsi_hs_bit_period / cfg_num_lanes

Solving for number_of_dsi_bytes:

number_of_dsi_bytes = dpi_event_size * dpi_pclk_period * cfg_num_lanes / (8 * dsi_hs_bit_period)

number_of_dsi_bytes - value to set the cfg_host_dpi_hfp (hbp, or hsa) port to
dpi_event_size - size of the video event in pixels (hfp, hbp, hsa)
dpi_pclk_period - period of dpi_pclk
dsi_hs_bit_period - bit period of the mipi data lanes in High Speed mode
cfg_num_lanes - host controller setting for number of active lanes,1 = 2 lanes)

49.3.4 RAM Usage

The DSI Host Controller Core requires on-chip RAM arrays. These RAM arrays and their corresponding sizes are listed in the table below.

All RAM arrays used in this core are configured as two-port, having the following attributes:

- Two address buses, one for the write port and one for read port
- Two data buses, one for write data on the write port and one for read data on the read port
- Two clock inputs, one for the write port and one for the read port. (Note: In some cases, the core logic will connect the same clock to both the read and write clock inputs of the RAM.)

Ram Purpose	Size (Depth x Width)
DPI Core Pixel Data FIFO	2048 x 32
APB to Packet Interface TX FIFO	64 x 32
APB to Packet Interface RX FIFO	64 x 32

49.3.5 DSI DPHY

49.3.5.1 DPHY Overview

The DSI Host Controller Core D-PHY Interface connects directly to MIPI PPI compliant D-PHYs. The DSI Host Controller Core uses this interface to transmit data, send Escape sequences, receive and transmit triggers, and detect and report D-PHY error conditions.

The figure below shows the block diagram of the MIPI DSI D-PHY. It consists of a clock lane module and two data lane modules. Each of these PHY Lane Modules communicates through a differential line to a complementary PHY at the other side of the lane interconnect.

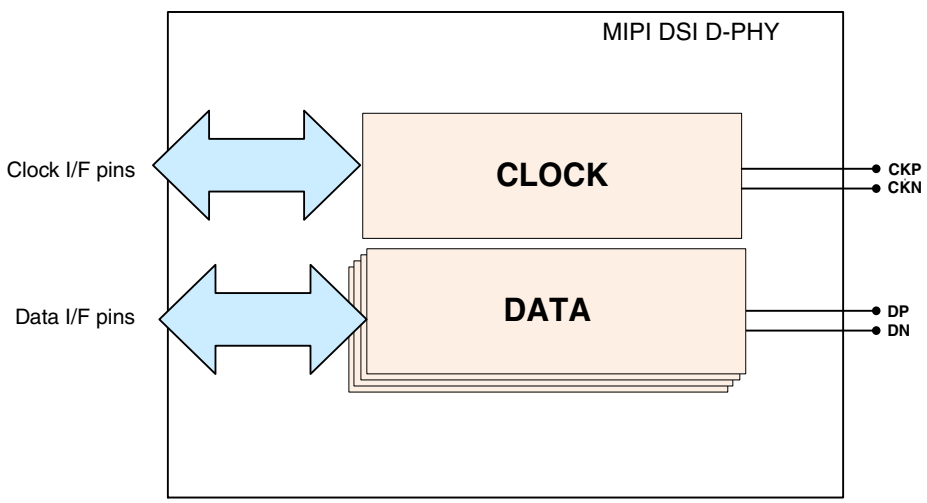


Figure 49-3. MIPI D-PHY Block Diagram

The figure below shows the D-PHY Lane components:

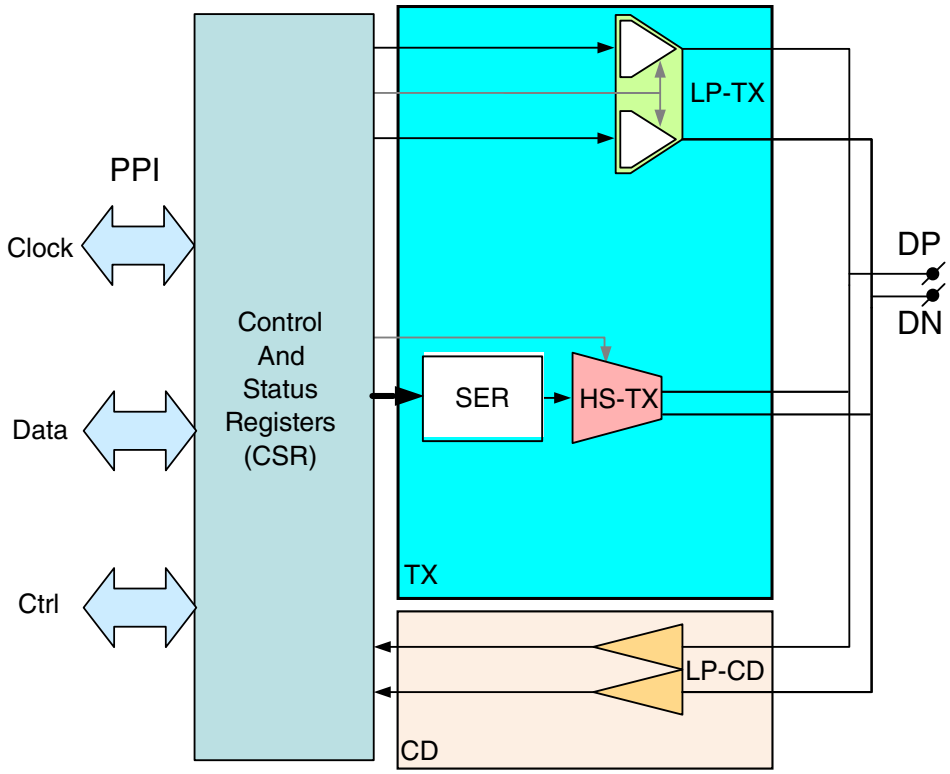


Figure 49-4. D-PHY Lane Diagram

The DATAy are bidirectional lanes and consist of High-Speed Transmitter (HS-TX), Low-Power Transmitter (LP-TX), Serializer (SER), and Low- Power Contention Detector (LP-CD).

The Clock lane is a bidirectional lane and consists of High-Speed Transmitter (HS-TX), and Low-Power Transmitter (LP-TX).

The Control and Interface Logic (CIL) module interfaces with the Protocol and determines the global operation of the Lane Module. The interface between the D-PHY and the protocol is called the PHY-Protocol Interface (PPI).

During normal operation, a Lane switches between Low-Power and High-Speed mode. Bidirectional Lanes can also switch communication direction. The change of operating mode or direction requires enabling and disabling of certain electrical functions. These enable and disable events do not cause glitches on the Lines that would result in a detection of an incorrect signal level. Therefore, all mode and direction changes are smooth to always ensure a proper detection of the Line signals.

49.3.5.1.1 High-Speed Transmitter (HS-TX)

The High-Speed Transmit interface is used to transmit High-Speed data across the MIPI interface. All signals are synchronous to the PPI TX byte clock, TxByteClkHS, and are PPI compliant.

The differential output voltage VOD is defined as the difference of the voltages V_{DP} and V_{DN} at the DP and DN pins, respectively.

VOD =VDP −VDN

The output voltages V_{DP} and V_{DN} at the DP and DN pins will not exceed the high-speed output high voltage V_{OHHS}. The common-mode voltage V_{CMTX} is defined as the arithmetic mean value of the voltages at the DP and DN pins:

VCMTX = (VDP + V DN)/2

V_{OD} and V_{CMTX} are graphically shown in Figure 4 for ideal HS signals. Figure 5 shows single ended HS signals with the possible kinds of distortion of the differential output and common mode voltages. V_{OD} and V_{CMTX} may be slightly different for driving a Differential-1 or a Differential-0 on the pins.

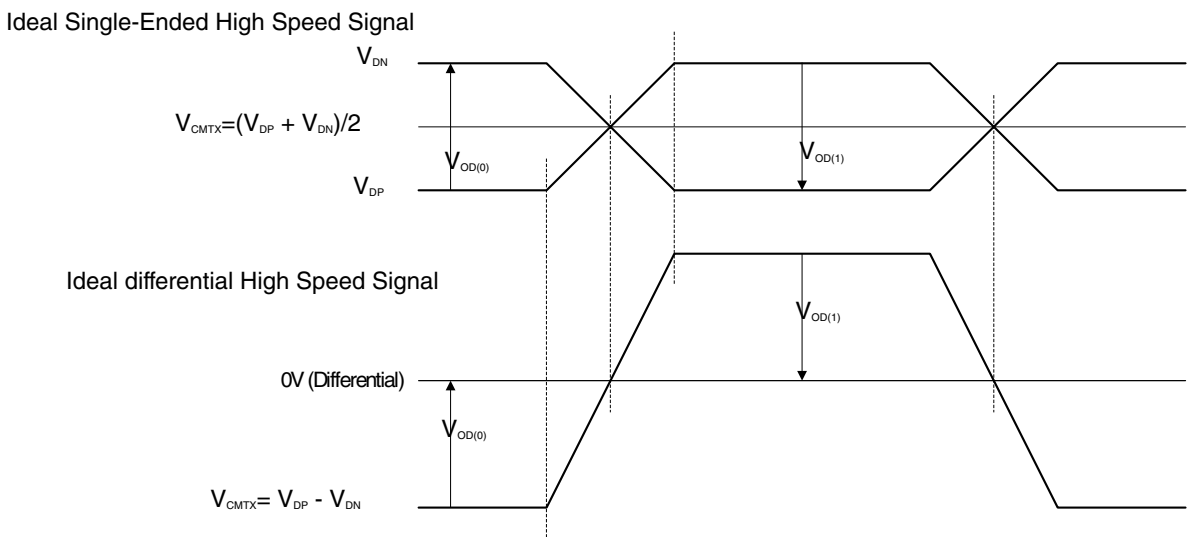


Figure 49-5. Ideal Single-ended and Resulting Differential HS Signals

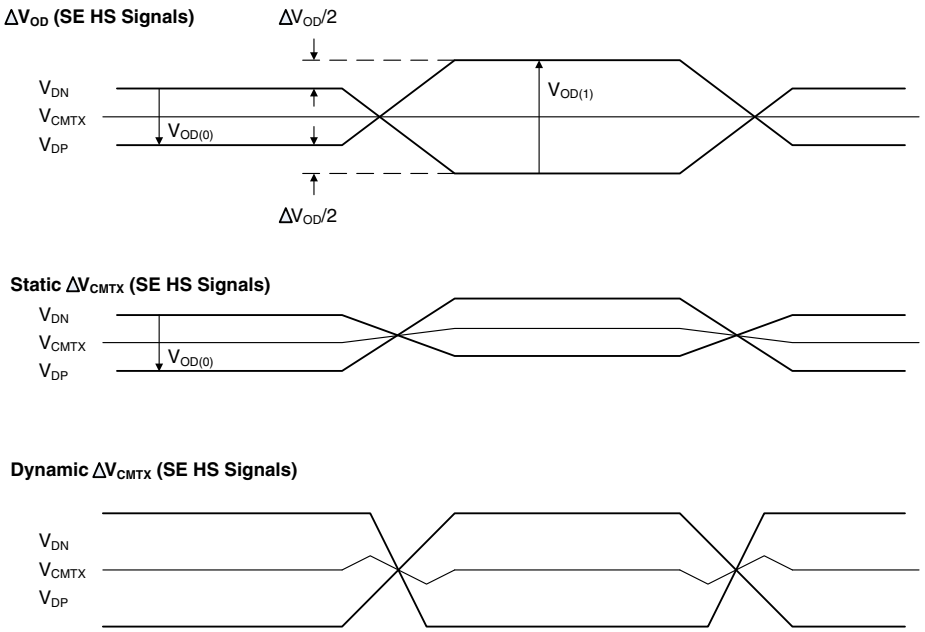


Figure 49-6. Possible ΔVcmtx and ΔVod Distortions of the Single-Ended HS Signals

The output differential voltage mismatch ΔVOD is defined as the difference of the absolute values of the differential output voltage in the Differential-1 state VOD(1) and the differential output voltage in the Differential-0 state VOD(0). This is expressed by:

$\Delta V = |V_{OD(1)}| - |V_{OD(0)}|$

If VCMTX(1) and VCMTX(0) are the common-mode voltages for static Differential-1 and Differential- 0 states respectively, then the common-mode reference voltage is defined by:

$V_{CMTX,\,REF} = (V_{CMTX(1)} + V_{CMTX(0)})/2$

The transient common-mode voltage variation is defined by:

$\Delta V_{CMTX\,(t)} = V_{CMTX\,(t)} - V_{CMTX,REF(t)}$

The static common-mode voltage mismatch between the Differential-1 and Differential-0 state is given by:

$\Delta V_{CMTX\,(1,0)} = (V_{CMTX\,(1)} - V_{CMTX\,(0)})/2$

A test circuit for the measurement of VOD and VCMTX is shown:

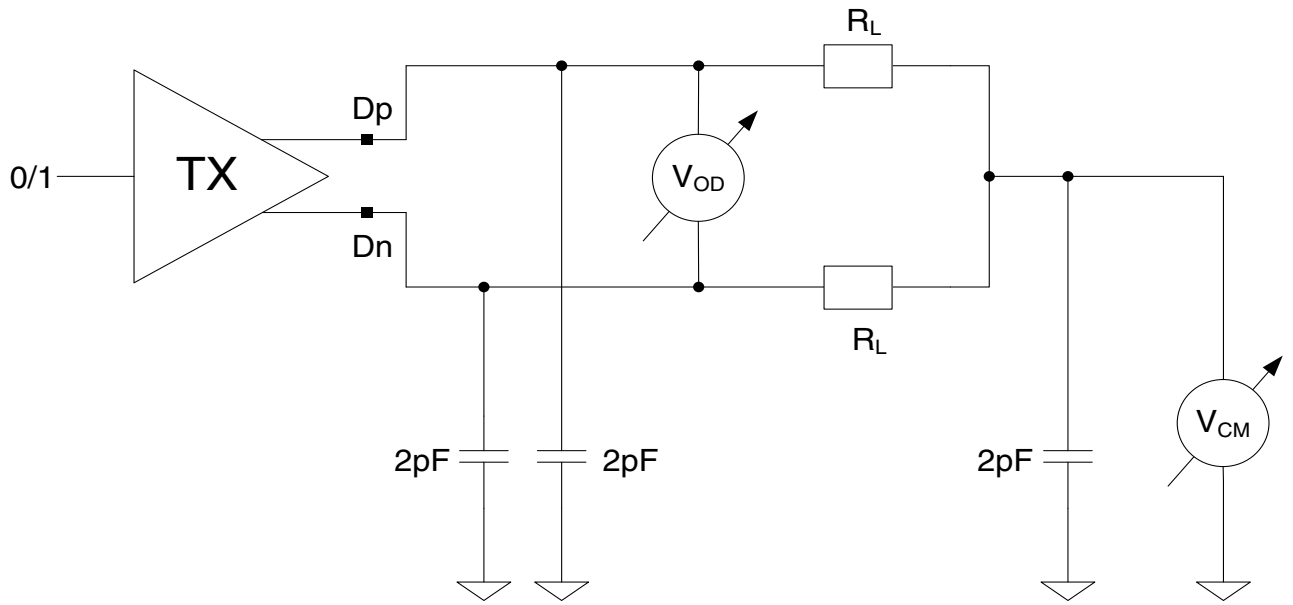


Figure 49-7. Test Circuit for Vcmtx and Vod measurement

The single-ended output impedance of the transmitter at both the DP and DN pins is denoted by Z_{OS}. ΔZ_{OS} is the mismatch of the single ended output impedances at the DP and DN pins, denoted by ZOSDP and ZOSDN respectively. This mismatch is defined as the ratio of the absolute value of the difference of Z_{OSDP} and Z_{OSDN} and the average of those impedances:

$$\Delta Z_{OS} = (2*|Z_{OSDP} - Z_{OSDN}|)/(Z_{OSDP} + Z_{OSDN})$$

The output impedance ZOS and the output impedance mismatch ΔZOS is compliant with High Speed Transmitter DC Specifications for both the Differential-0 and Differential-1 states for all allowed loading conditions. It is recommended the design keep the output impedance during state transitions as close as possible to the steady state value. The output impedance ZOS can be determined by injecting an AC current into the DP and DN pins and measuring the peak-to-peak voltage amplitude. The test circuit for such a measurement is shown in Figure 7, where RL are load resistors and VTEST,PP is a test signal applied at the common-mode node via an AC coupling capacitor CAC. The frequency of the test signal is fTEST.

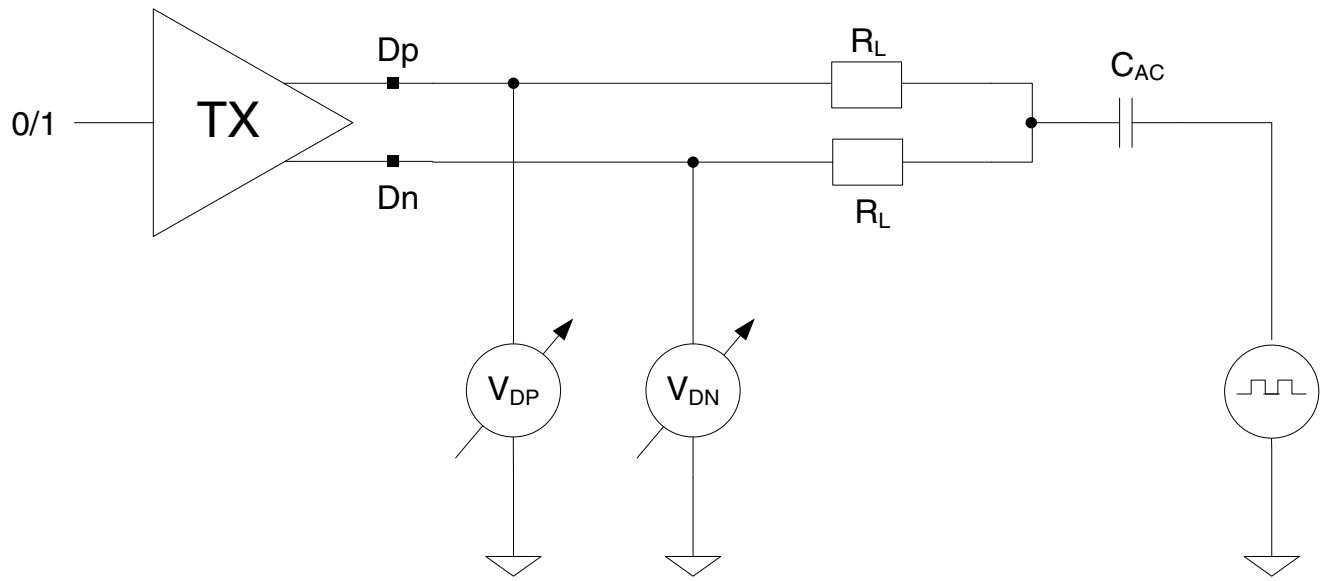


Figure 49-8. Test Circuit for Output Impedance measurements

Assuming negligible effect of the coupling capacitor, the example circuit gives the following relation between peak-to-peak pin voltage VPIN,PP and ZOS. This equation can be used for both DN and DP pins individually.

$Z_{OS} = (V_{PIN,PP} * RL)/(V_{TEST,PP} - Z_{PIN,PP})$

Where VPIN,PP is the peak-to-peak voltage at the pin DP or DN.

49.3.5.1.2 Low-Power Transmitter (LP-TX)

The Low-Power transmitter is a slew-rate controlled push-pull driver. It is used for driving the Lines in all Low-Power operating modes. The LP transmitter has been optimized to minimize static power consumption. The slew-rate of signal transitions is bounded in order to keep EMI low. The LP transmitter will not drive the pad pin potential statically beyond the maximum value of V_{OH}. V_{OH} is the thevenin output, high-level voltage in the high-level state, when the pad pin is not loaded. V_{OL} is the thevenin output, low-level voltage in the LP transmit mode. This is the voltage at an unloaded pad pin in the low-level state.

49.3.5.1.3 Low-Power Contention Detector (LP-CD)

Contention can be inferred from any of the following conditions:

- LP high fault is detected when the LP transmitter is driving high and the pin voltage is less than VIL.
- LP low fault is detected when the LP transmitter is driving low and the pad pin voltage is greater than VIHCD. An LP low fault will not be detected when the pin voltage is less than VILCD.

The general operation of a contention detector is similar to that of an LP receiver with lower threshold voltages. Although the DC specifications differ, the AC specifications of the LP-CD are defined to match those of the LP receiver. The LP-CD sufficiently filters the input signal to avoid false triggering on short events

The LP-CD threshold voltages (VILCD, VIHCD) are shown along with the normal signaling voltages:

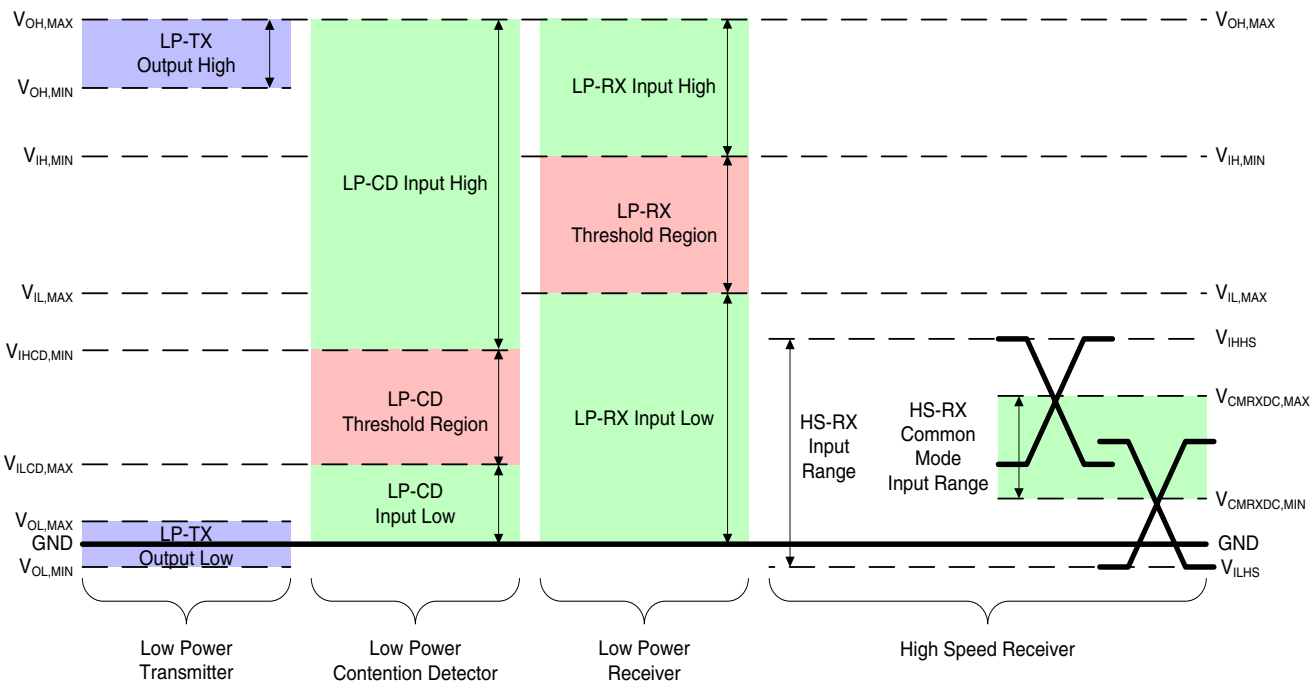


Figure 49-9. Signaling and MIPI Contention Voltage Levels

49.3.5.1.4 Input Characteristics

No structure within the PHY may be damaged when a DC signal that is within the signal voltage range VPIN is applied to a pad pin for an indefinite period of time. $V_{PIN(absmax)}$ is the maximum transient output voltage at the transmitter pin. The voltage on the transmitter's output pin will not excel $V_{PIN,MAX}$ for a period greater than $T_{VPIN(absmax)}$. When the PHY is in the low-power receive mode the pad pin leakage current will be I_{LEAK} when the pad signal voltage is within the signal voltage range of V_{PIN}. The

specification of ILEAK assures interoperability of any PHY in the LP mode by restricting the maximum load current of an LP transmitter. The test circuit for leakage current measurement is shown:

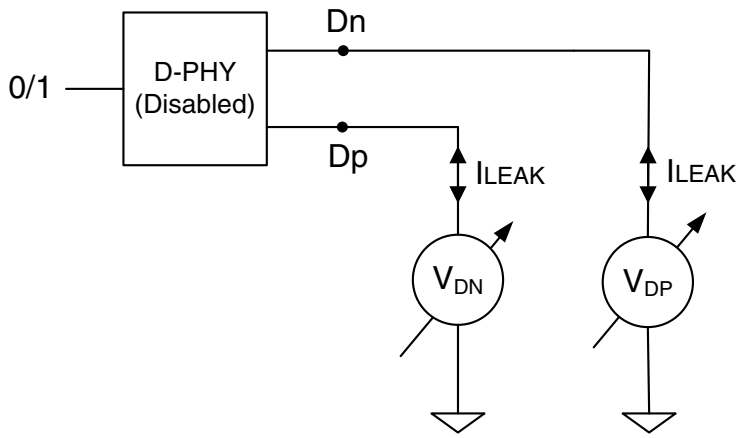


Figure 49-10. Pin-Leakage Measurement DC specification

49.3.5.1.5 High-Speed Serializer

The HS-Serializer converts 8-bits parallel data into 1 bit data stream to be transmitted by HSTX module. The CSR should clock out 8-bits parallel data, HSTX_DATA[7:0], on D0_HS_BYTE_CLKS rising edges. Then the Serializer will capture HSTX_DATA[7:0] on D0_HS_BYTE_CLKS rising edges and the signal connected to HSTX_DATA[0] is transmitted first. There is setup timing constraint and hold timing constraint designated to the rising edge of D0_HS_BYTE_CLKS.

The figure shows the relationship between HS_TXCLKP, HS_TXCLKN, HS_SER_LD, HS_BYTE_CLKS and HSTX_DATA[7:0]. It also shows how the 8-bits of parallel bits are mapped into the serial link:

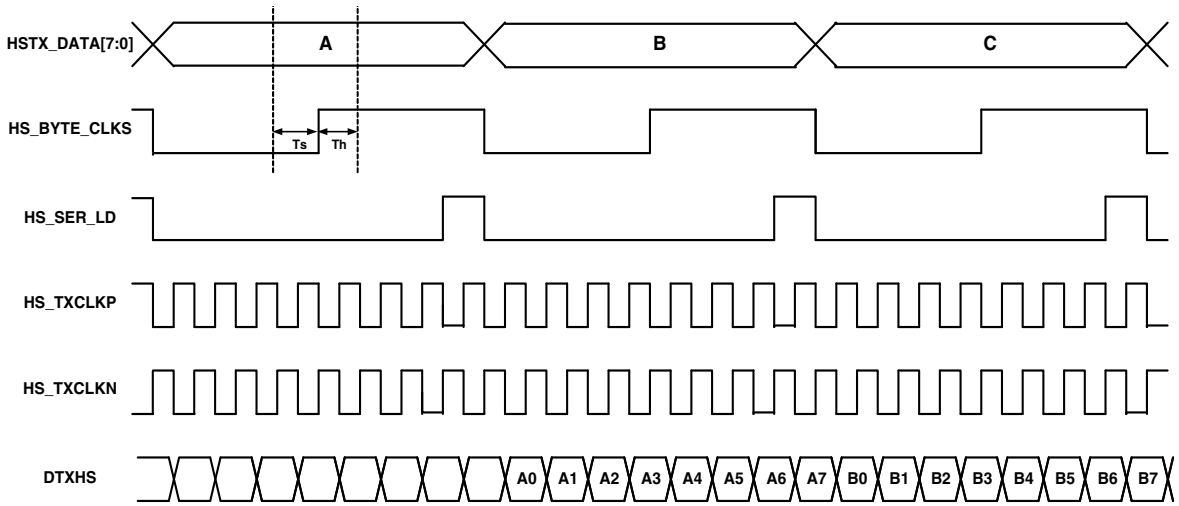


Figure 49-11. Serializer Timing Diagram

49.3.5.1.6 Operating Modes

49.3.5.1.6.1 High-Speed Transmit from the Master Side

The figure shows an example of a High-Speed transmission on the Master side. While TxRequestHS is low, the Lane Module ignores the value of TxDataHS. To begin transmission, the protocol drives TxDataHS with the first byte of data and asserts TxRequestHS. This data byte is accepted by the PHY on the first rising edge of TxByteClkHS with TxReadyHS also asserted. At this point, the protocol logic drives the next data byte onto TxDataHS. After every rising clock cycle with TxReadyHS active, the protocol supplies a new valid data byte or ends the transmission. After the last data byte has been transferred to the Lane Module, TxRequestHS is driven low to cause the Lane Module to stop the transmission and enter Stop state. The minimum number of bytes transmitted could be as small as one.

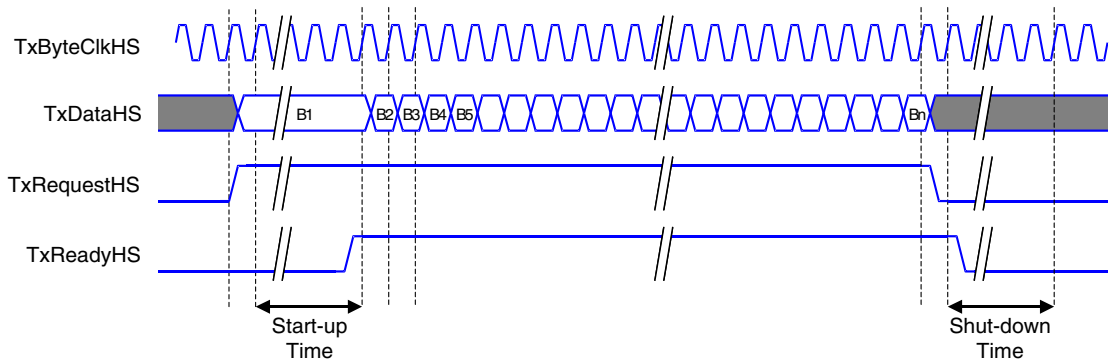


Figure 49-12. High-Speed Transmission from the Master Side

49.3.5.1.6.2 Low-Power Data Transmission

For Low-Power data transmission the TxClkEsc is used. The Protocol directs the Data Lane to enter Low-Power data transmission Escape mode by asserting TxRequestEsc with TxLpdtEsc high. The Low-Power transmit data is transferred on the TxDataEsc lines when TxValidEsc and TxReadyEsc are both active at a rising edge of TxClkEsc. The byte is transmitted in the time after the TxDataEsc is accepted by the Lane Module (TxValidEsc = TxReadyEsc = high). The Protocol knows the byte transmission is finished when TxReadyEsc is asserted. After the last byte has been transmitted, the protocol de-asserts TxRequestEsc to end the Low-Power data transmission. This causes TxReadyEsc to return low. Whenever TxRequestEsc transitions from high-to-low, it always remains in the low state for a minimum of two TxClkEsc clock cycles. The figure shows an example Low-Power data transmission operation.

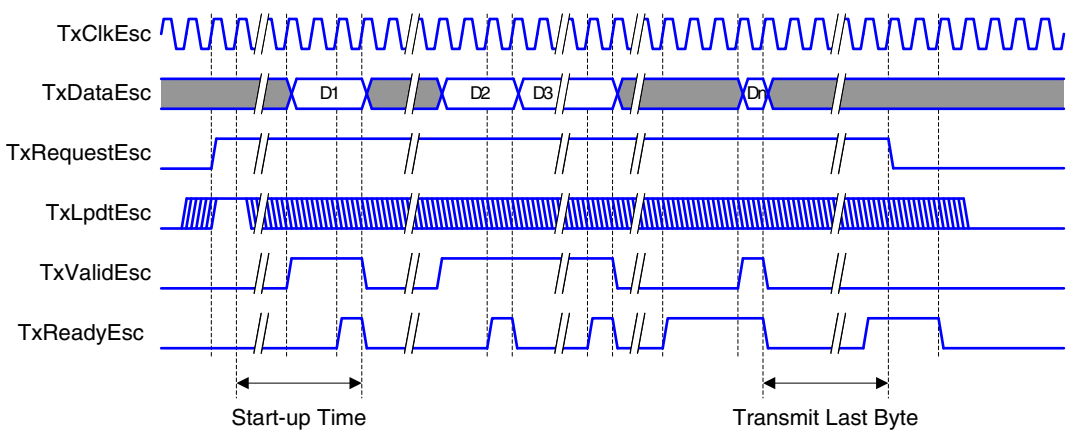


Figure 49-13. Low-Power Data Transmission

49.3.5.1.6.3 Turn-around

If the Master side and Slave side Lane Modules are both bi-directional, it is possible to turn around the Link for Escape mode signaling. Which side is allowed to transmit is determined by passing a “token” back and forth. That is, the side currently transmitting passes the token to the receiving side. If the receiving side acknowledges the turn-around request, as indicated by driving the appropriate line state, the direction is switched. The figure shows an example of two turn-around events. At the beginning, the local side is the transmitter, as shown by Direction=0. When the protocol on this side wishes to turn the Lane around (i.e. give the token to the other side), it asserts TurnRequest for at least one cycle of TxClkEsc. This initiates the turn around procedure. The remote side acknowledges the turn-around request by driving the appropriate states on the Lines. When this happens, the local Direction signal changes from transmit (0) to receive (1). The remote side initiates a turn-around request, passing the token back to the local side. When this happens, the local Direction signal changes back to transmit (0). The transmitter is in control of the Link direction and decides when to turn the Link around, passing control to the receiver. If the remote side does not acknowledge the turnaround request, the Direction signal does not change.

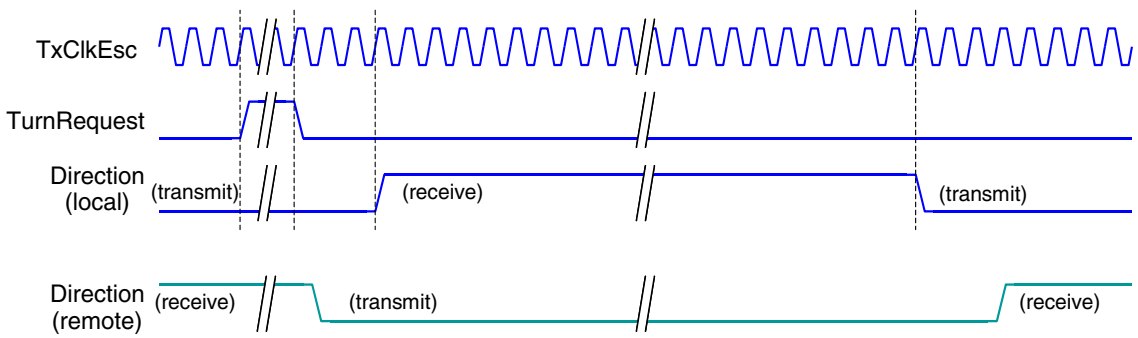


Figure 49-14. Turn-around Actions Transmit-to-Receive and Back to Transmit

49.3.5.2 DPHY PLL

The MIPI DSI DPHY PLL is a high performance PLL based frequency synthesizer that incorporates a lock detector, independent output divider, and supports power down modes. The PLL block diagram is shown below.

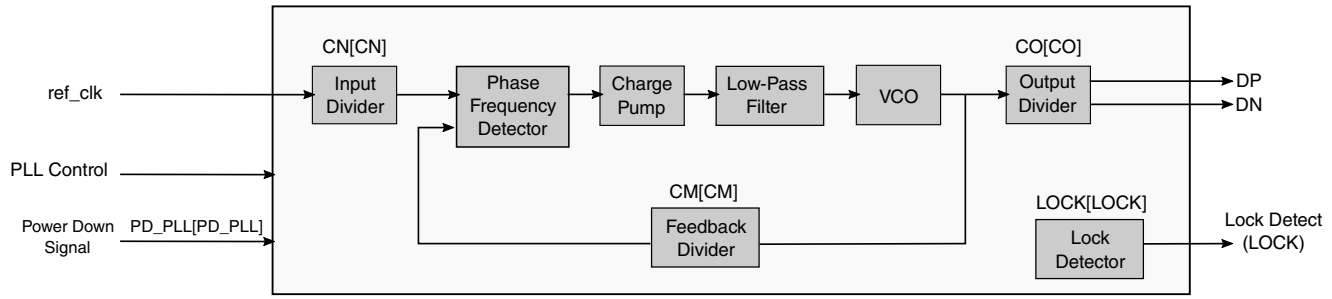


Figure 49-15. MIPI DSI DPHY PLL Block Diagram

The DPHY PLL input clock is ref_clk. The PLL output multiplies the input frequency by `(CM / (CN * CO))`.

NOTE

The input frequency ranges from 24 MHz till 200 MHz. The input divider has to be programmed such that the frequency after the input divider ranges from 24 MHz till 30 MHz.

NOTE

The VCO output frequency range is 640 MHz to 1500 MHz.

The power down signal (PD) is active-high and used to power down the PLL. Asserting PD will reset the PLL to it's initial state. To ensure proper PLL functionality, ref_clk needs to be stable before PLL power up. The expected power-on sequence for the DSI DPHY PLL is illustrated below.

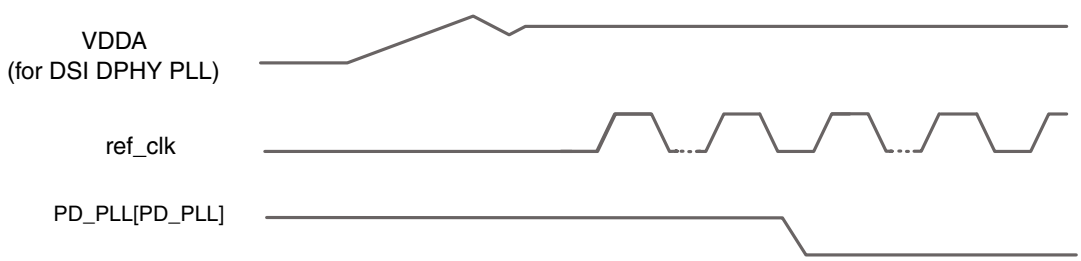


Figure 49-16. MIPI DSI DPHY PLL Power-on Sequence

49.3.5.3 DPHY Programming

The following sections detail the programming of the MIPI DSI DPHY high-speed timers.

49.3.5.3.1 High-Speed Prepare Timer ($T_{HS-PREPARE}$)

Calculate the value for clock lane high-speed prepare timer (mc_PRG_HS_PREPARE) using Table 49-2 and Table 49-3 :

Table 49-2. Clock Lane High-Speed prepare timer range

	Min (ns)	Max (ns)	PRG HS PREPARE	$T_{CLK-PREPARE}$ (ns)
$T_{CLK-PREPARE}$	38	95	1	75

1. Assuming TxClkEsc = 20 MHz (T_{per} = 50 ns)
2. When MC_PRG_HS_PREPARE[MC_PRG_HS_PREPARE] = 0, $T_{CLK-PREPARE}$ = 1 x TxClkEsc Period

   When MC_PRG_HS_PREPARE[MC_PRG_HS_PREPARE] = 1, $T_{CLK-PREPARE}$ = 1.5 x TxClkEsc Period
3. There is a calculation inaccuracy (of 0.5 x Tperiod of TxClkEsc) due to an expected mismatch coming from delays on the low power signals inside Hard-Macro before low power data begins transmission on the high-speed serial interface.

Calculate the value for data lane high-speed prepare timer (m_PRG_HS_PREPARE[1:0]) using Table 49-3 and Table 49-4 :

Table 49-3. Data Lane High-Speed prepare timer range

	Min	Max
$T_{HS-PREPARE}$	40 ns + 4 x UI	85 ns + 6 x UI

Table 49-4. Data Lane High-Speed prepare timer for different data rates

Data Rate	Min (ns)	Max (ns)	PRG HS PREPARE[1:0]	$T_{HS-PREPARE}$
1.5 Gbps	42.67	89	01	75
1.0 Gbps	44	91	01	75
500 Mbps	48	97	01	75
250 Mbps	56	109	10	100
80 Mbps	90	160	10	100

1. Assuming TxClkEsc = 20 MHz (T_{per} = 50 ns)
2. When M_PRG_HS_PREPARE[M_PRG_HS_PREPARE] = 00, $T_{HS-PREPARE}$ = 1 x TxClkEsc Period

   When M_PRG_HS_PREPARE[M_PRG_HS_PREPARE] = 01, $T_{HS-PREPARE}$ = 1.5 x TxClkEsc Period

When M_PRG_HS_PREPARE[M_PRG_HS_PREPARE] = 10, $T_{HS-PREPARE}$ = 2 x TxClkEsc Period

When M_PRG_HS_PREPARE[M_PRG_HS_PREPARE] = 11, $T_{HS-PREPARE}$ = 2.5 x TxClkEsc Period

3. There is a calculation inaccuracy (of 0.5 x Tperiod of TxClkEsc) due to an expected mismatch coming from delays on the low power signals inside Hard-Macro before low power data begins transmission on the high-speed serial interface.

49.3.5.3.2 High-Speed Zero Timer($T_{HS-ZERO}$)

Calculate the value for clock lane high-speed zero timer (mc_PRG_HS_ZERO) using the formula below:

- $T_{CLK-ZERO}$= (mc_PRG_HS_ZERO + 3)*(TxByteClkHS Period)
- TxByteClkHS = 1/8 Data Rate
- When MC_PRG_HS_ZERO[MC_PRG_HS_ZERO] = 000110 @ 250 Mb/s, $T_{CLK-ZERO}$ = (6+3) * 32 ns = 288 ns
- When MC_PRG_HS_ZERO[MC_PRG_HS_ZERO] = 100000 @ 1 Gb/s, $T_{CLK-ZERO}$ = (32+3) * 8 ns = 280 ns

Calculate the value for data lane high-speed zero timer (m_PRG_HS_ZERO) using the formula below:

- $T_{HS-ZERO}$ = (m_PRG_HS_ZERO + 6)*(TxByteClkHS Period)
- TxByteClkHS = 1/8 Data Rate
- When M_PRG_HS_ZERO[M_PRG_HS_ZERO] = 00001 @ 250 Mb/s, $T_{HS-ZERO}$ = (1+6) * 32 ns = 224 ns
- When M_PRG_HS_ZERO[M_PRG_HS_ZERO] = 01001 @ 1 Gb/s, $T_{HS-ZERO}$ = (9+6) * 8 ns = 120 ns

49.3.5.3.3 High-Speed Trail Timer ($T_{HS-TRAIL}$)

Calculate the value for the high-speed trail timer (PRG_HS_TRAIL) using the formula below:

- $T_{HS-TRAIL}$= (PRG_HS_TRAIL)*(TxByteClkHS Period)
- TxByteClkHS = 1/8 Data Rate
- When M_PRG_HS_TRAIL[M_PRG_HS_TRAIL] = 0100 @ 250 Mb/s, >$T_{HS-TRAIL}$ = 4 * 32 ns = 128 ns
- When M_PRG_HS_TRAIL[M_PRG_HS_TRAIL] = 1100 @ 1 Gb/s, >$T_{HS-TRAIL}$ = 12 * 8 ns = 96 ns

49.3.5.4 Packet Data and Pixel Formats

The DSI Host Controller Core and the external DSI Peripheral explicitly handle and exchanges 32-bit words encapsulated as packet data. The mapping between pixel formats and pixel data must be done on both sides by user application on both sides of the DSI link. The DSI spec gives many examples of common pixel formats. The DSI Host Controller Core and external DSI Peripheral have no preference or limitation on which pixel formats it will handle. The IP which provides video source packet data through DPI-2 interface (e.g. eLCDIF) onto DSI should also be configured by the user application to match the pixel formats expected by the DSI Host Controller.

The DPI-2 interface core handles pixel to byte packing and byte to pixel unpacking according to the MIPI DSI specification.

49.3.6 DSI Host Clocks

The DSI Host Controller contains the following clocks:

Table 49-5. Clocks

Clock	Description
ref_clk	Reference clock (Used by DPHY)
clk_byte	Byte clock (Generated by DPHY and used for packet generation)
clk_tx_esc	TX escape mode clock (Used for internal LPDT state machines and low-power transmissions)
dpi_pclk	DPI Pixel Clock
TxClkEsc	Escape mode Transmit Clock. This clock is directly used to generate escape sequences. The period of this clock determines the symbol time for low power signals. It is therefore constrained by the normative part of the DPHY specification. **NOTE:** The max frequency of TxClkEsc is 20 MHz and the Min frequency of TxClkEsc is 12 MHz.
TxByteClkHS	High-Speed Transmit Byte Clock. This is used to synchronize PPI signals in the High-Speed transmit clock domain. It is recommended that all transmitting Data Lane Modules share on TxByteClkHS signal. The frequency of TxByteClkHS is exactly 1/8 the High-Speed bit rate.

The figure below depicts the various clocks used in the DSI Host:

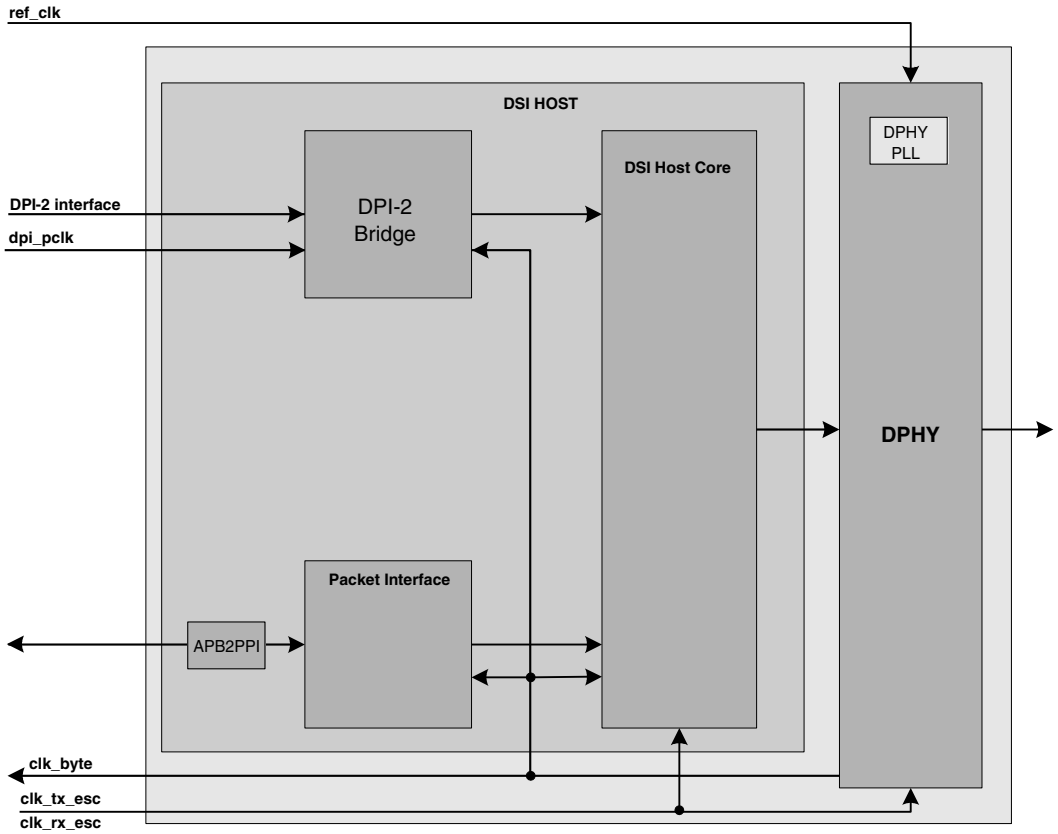


Figure 49-17. DSI Host Controller Clocking Diagram

49.3.6.1 ref_clk

The reference clock is used by the DPHY (DPHY PLL) to generate the High Speed MIPI clock and High Speed Data Lane signaling. The frequency of the resulting clock generated from DPHY PLL is typically equal to the high speed bit rate of the data lanes. The DPHY will also use this clock to generate the clk_byte clock output used by the Host DSI Controller and user application logic.

The MIPI DPHY PLL must be setup through the MIPI_DSI_HOST_FSL_IP1_DPHY INTFCx registers, prior to starting DSI Host Controller and DPHY operation. For details, please refer to MIPI DSI Host IP1 DPHY INTFC Memory section.

49.3.6.2 clk_byte

The clk_byte clock is generated by the DPHY (DPHY PLL) and the frequency is 1/8th the data rate of the DPHY Data Lane High Speed data rate. A DPHY configured to generate High Speed data at 1Gbps/lane would generate a 125MHz clk_byte clock.

The clk_byte clock is used by the DSI Host Controller to interface to the MIPI Tx DPHY High Speed PPI interface. The DSI Host Controller's Packet Interface is synchronous to the clk_byte clock.

49.3.6.3 dpi_pclk

The dpi_pclk clock is used on the Host DPI-2 interface. All of the Host DPI-2 signals are synchronous to this clock. The DSI Host Controller's DPI Bridge module handles transferring video data received in the dpi_pclk clock domain over to the clk_byte clock domain.

The dpi_pclk and clk_byte frequencies are related by the following formula:
clk_byte_freq >= dpi_pclk_freq * DPI_pixel_size / (8 * (cfg_num_lanes + 1))

cfg_num_lanes = the configuration port setting that selects the number of active MIPI DPHY data lanes clk_byte_freq = frequency of clk_byte which is 1/8th the High Speed data lane rate. dpi_pclk_freq = frequency of the dpi_pclk clock on the DPI-2 interface. DPI_pixel_size = size of pixels, in bits, on the DPI-2 interface

If the clk_byte frequency does not meet this requirement, the MIPI interface will not be able to keep up with the video stream on the DPI-2 interface resulting in dropped video lines.

NOTE

For more information on this clock, please see the Chip-specific bridge topic.

49.3.6.4 clk_tx_esc

The clk_tx_esc clock is used by the MIPI Tx DPHY for state control and low power data transmission. The DSI Host Controller also uses clk_tx_esc for the portion of controller logic that interfaces to the MIPI Tx DPHY that are synchronous to clk_tx_esc. The frequency of clk_tx_esc is defined by the requirements of the MIPI interface and the MIPI Tx DPHY. For details about the clk_tx_esc source, please refer to the chip-specific MIPI-DSI clocking section.

49.3.7 Reset

The MIPI DSI Host contains the following reset signals:

Reset Signal	Description
reset_dpi_n	Asynchronous reset in the dpi_pclk clock domain (Active-low signal)
reset_byte_n	Asynchronous reset to all logic in the clk_byte clock domain (Active-low signal)
reset_esc_n	Asynchronous reset to all logic in the clk_esc clock domain (Active-low signal)

49.4 External Signals

The table below shows the external signals in this module.

Table 49-6. External Signals

External Signal	Description	Direction
DNn	Differential Data Signal (Negative)	O
DPn	Differential Data Signal (Positive)	O
CKN	Differential Clock Signal (Negative)	O
CKP	Differential Clock Signal (Positive)	O

49.5 Application Information

49.5.1 Steps to Transmit a Packet

1. If the transmit packet is a long data type, write payload data to TX_PAYLOAD[PAYLOAD]
2. Write transmit packet information to PKT_CONTROL[CTRL]
3. Write any value to SEND_PACKET[TX_SEND] to start a packet transmission
4. Wait for packet transmit to complete. Either wait for an interrupt from the apb interface or monitor PKT_STATUS register.

49.5.2 Steps to perform a Packet Read Request

1. Follow the steps for transmitting a packet as shown above, but with a data type that is a read command. You won't have payload data for a read request so the first step can be skipped

2. Write 1 to PKT_CONTROL[25] to turn the bus around (BTA) after the transmission is complete so that the peripheral can send the read data back
3. Monitor PKT_STATUS[2] for the direction to check if the peripheral bus has control
4. Set PKT_CONTROL[25] bit to 0
5. Wait for receive packet by either waiting for the interrupt or monitoring PKT_STATUS[7] to indicate an Rx packet has been received
6. Read the receive packet header information and Rx payload data if read is a long data type

49.6 Initialization

The Initialization procedure for the DSI Host Controller is as follows:

1. Assert all resets through the chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n
2. Setup DSI related clocks through chip-specific registers provided for this purpose (refer to CCM for details).
3. Setup MIPI DSI application by programming all DSI Host Controller and DSI DPHY required parameters (refer to Memory Map and register definition for details).
4. De-assert all remaining resets through the Chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n

The DSI Host Controller should be ready for use now.

49.7 Memory Map and register definition

This section includes the DSI Host module memory map and detailed descriptions of all registers.

49.7.1 DSI HOST register descriptions

49.7.1.1 DSI_HOST memory map

MIPI_DSI.DSI_HOST base address: 4080_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Configure Number of Lanes (CFG_NUM_LANES)	32	RW	0000_0000h
4h	Configure Noncontinuous Clock (CFG_NONCONTINUOUS_CLK)	32	RW	0000_0000h
8h	Configure TCLK-PRE DPHY (CFG_T_PRE)	32	RW	0000_0000h
Ch	Configure TCLK Wait (CFG_T_POST)	32	RW	0000_0000h
10h	Configure Transmit Gap (CFG_TX_GAP)	32	RW	0000_0000h
14h	Configure Auto Insert EoTp (CFG_AUTOINSERT_EOTP)	32	RW	0000_0000h
18h	Configure Extra Packets after EOTP (CFG_EXTRA_CMDS_AFTER_EOTP)	32	RW	0000_0000h
1Ch	Configure HTX Timeout Count (CFG_HTX_TO_COUNT)	32	RW	0000_0000h
20h	Configure Low-Power Receive Timeout Count (CFG_LRX_H_TO_COUNT)	32	RW	0000_0000h
24h	Configure BTA_H Timeout Count (CFG_BTA_H_TO_COUNT)	32	RW	0000_0000h
28h	Configure Twakeup (CFG_TWAKEUP)	32	RW	0000_0000h
2Ch	Configure Status Out (CFG_STATUS_OUT)	32	R	0000_0000h
30h	Receive Error Status (RX_ERROR_STATUS)	32	R	0000_0000h

49.7.1.2 Configure Number of Lanes (CFG_NUM_LANES)

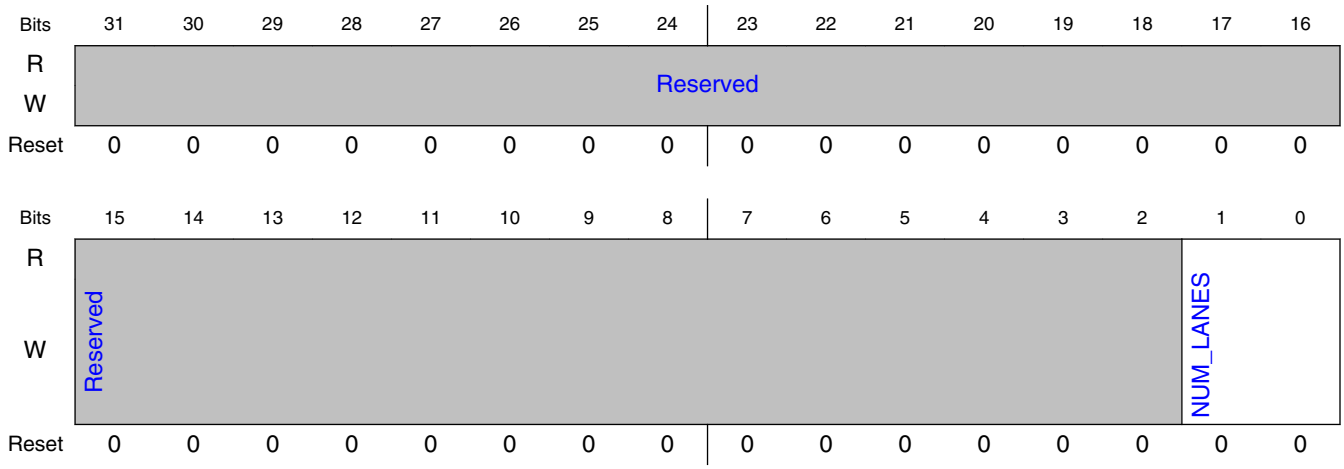
49.7.1.2.1 Offset

Register	Offset
CFG_NUM_LANES	0h

49.7.1.2.2 Function

Sets the number of active lanes that are to be used for transmitting data.

49.7.1.2.3 Diagram



49.7.1.2.4 Fields

Field	Function
31-2 —	Reserved
1-0 NUM_LANES	Number of active lanes. 00b - 1 lane 01b - 2 lanes

49.7.1.3 Configure Noncontinuous Clock (CFG_NONCONTINUOUS_CLK)

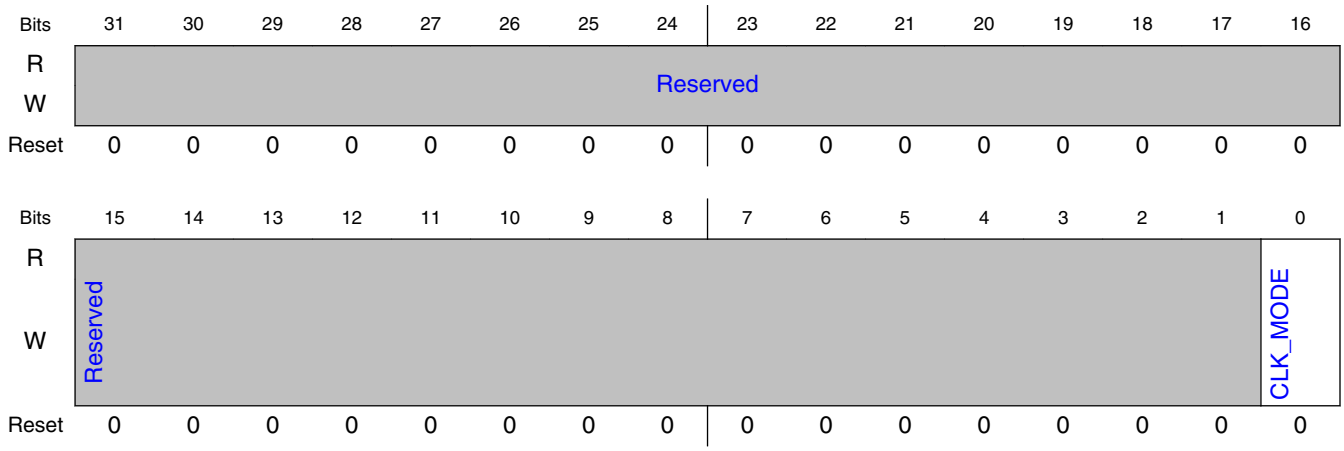
49.7.1.3.1 Offset

Register	Offset
CFG_NONCONTINUOUS_CLK	4h

49.7.1.3.2 Function

Sets the Host Controller into non-continuous MIPI clock mode. When in non-continuous clock mode, the high speed clock will transition into low power mode between transmissions.

49.7.1.3.3 Diagram



49.7.1.3.4 Fields

Field	Function
31-1 —	Reserved
0 CLK_MODE	Clock Mode 0b - Continuous high speed clock 1b - Non-Continuous high speed clock

49.7.1.4 Configure TCLK-PRE DPHY (CFG_T_PRE)

49.7.1.4.1 Offset

Register	Offset
CFG_T_PRE	8h

49.7.1.4.2 Function

Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after enabling the clock lane for HS operation before enabling the data lanes for HS operation. This setting represents the TCLK-PRE DPHY timing parameter. The minimum value for this port is 1.

49.7.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								NUM_PERIODS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.4.4 Fields

Field	Function
31-8 —	Reserved
7-0 NUM_PERIODS	Number of Periods 0 - Zero clock periods (Not supported) 1 - One clock period 2 - Two clock periods ... ..

49.7.1.5 Configure TCLK Wait (CFG_T_POST)

49.7.1.5.1 Offset

Register	Offset
CFG_T_POST	Ch

49.7.1.5.2 Function

Sets the number of byte clock periods ('clk_byte' input) to wait before putting the clock lane into LP mode after the data lanes have been detected to be in Stop State. This setting represents the DPHY timing parameters TLPX + TCLK-PREPARE + TCLK-ZERO + TCLK-PRE requirement for the clock lane before the data lane is allowed to change from LP11 to start a high speed transmission. The minimum value for this port is 1.

49.7.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								NUM_PERIODS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.5.4 Fields

Field	Function
31-8 —	Reserved
7-0 NUM_PERIODS	Number of Periods 0 - Zero clock periods (Not supported) 1 - One clock period 2 - Two clock periods ... ..

49.7.1.6 Configure Transmit Gap (CFG_TX_GAP)

49.7.1.6.1 Offset

Register	Offset
CFG_TX_GAP	10h

49.7.1.6.2 Function

Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after the clock lane has been put into LP mode before enabling the clock lane for HS mode again. This setting represents the THS-EXIT DPHY timing parameter. The minimum value for this port is 1.

49.7.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								NUM_PERIODS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.6.4 Fields

Field	Function
31-8 —	Reserved
7-0 NUM_PERIODS	Number of Periods 0 - Zero clock periods (Not supported) 1 - One clock period 2 - Two clock periods ... ..

49.7.1.7 Configure Auto Insert EoTp (CFG_AUTOINSERT_EOTP)

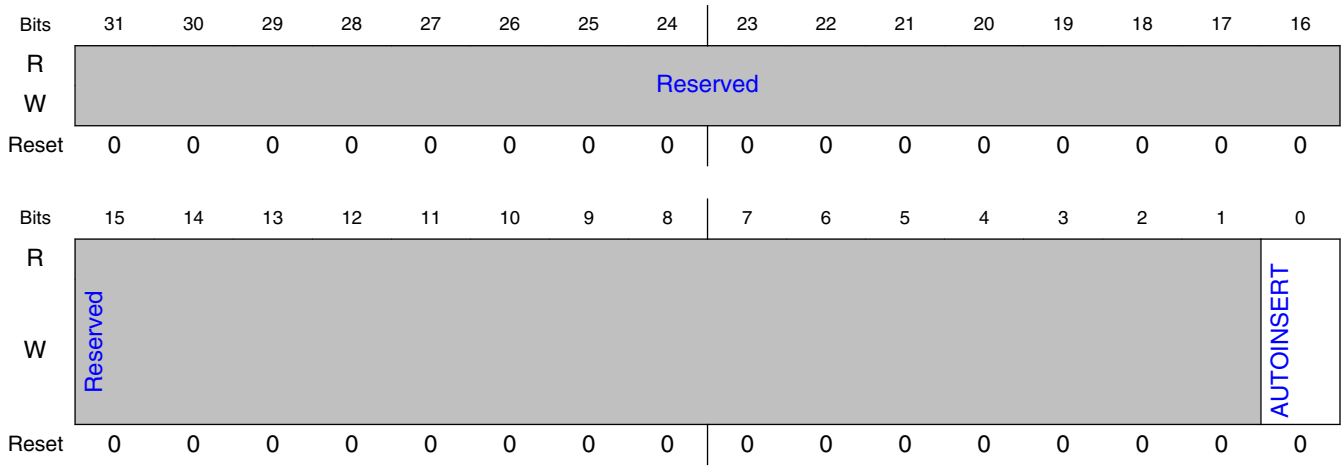
49.7.1.7.1 Offset

Register	Offset
CFG_AUTOINSERT_EOTP	14h

49.7.1.7.2 Function

Enables the Host Controller to automatically insert an EoTp short packet when switching from HS to LP mode.

49.7.1.7.3 Diagram



49.7.1.7.4 Fields

Field	Function
31-1 —	Reserved
0 AUTOINSERT	Auto Insert **NOTE:** AUTOINSERT must always equal 1 when using Video mode. 0b - EoTp is not automatically inserted 1b - EoTp is automatically inserted

49.7.1.8 Configure Extra Packets after EOTP (CFG_EXTRA_CMDS_AFTER_EOTP)

49.7.1.8.1 Offset

Register	Offset
CFG_EXTRA_CMDS_AFTER_EOTP	18h

49.7.1.8.2 Function

Configures the DSI Host Controller to send extra End Of Transmission Packets after the end of a packet. The value is the number of extra EOTP packets sent.

49.7.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EXTRA_EOTP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.8.4 Fields

Field	Function
31-8 —	Reserved
7-0 EXTRA_EOTP	Extra EOTP 0 - Zero packets 1 -1 packet ... ...

49.7.1.9 Configure HTX Timeout Count (CFG_HTX_TO_COUNT)

49.7.1.9.1 Offset

Register	Offset
CFG_HTX_TO_COUNT	1Ch

49.7.1.9.2 Function

Sets the value of the DSI Host High Speed TX timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification.

49.7.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.9.4 Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Count 0 - Zero clock periods 1 - One clock period 2 - Two clock periods ... ..

49.7.1.10 Configure Low-Power Receive Timeout Count (CFG_LRX_H_TO_COUNT)

49.7.1.10.1 Offset

Register	Offset
CFG_LRX_H_TO_COUNT	20h

49.7.1.10.2 Function

Sets the value of the DSI Host low power RX timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification.

49.7.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.10.4 Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Count 0 - Zero clock periods 1 - One clock period 2 - Two clock periods ... ..

49.7.1.11 Configure BTA_H Timeout Count (CFG_BTA_H_TO_COUNT)

49.7.1.11.1 Offset

Register	Offset
CFG_BTA_H_TO_COUNT	24h

49.7.1.11.2 Function

Sets the value of the DSI Host Bus Turn Around (BTA) timeout in clk_byte clock periods that once reached will initiate a timeout error.

49.7.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.11.4 Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Count 0 - Zero clock periods 1 - One clock period 2 - Two clock periods ... ..

49.7.1.12 Configure Twakeup (CFG_TWAKEUP)

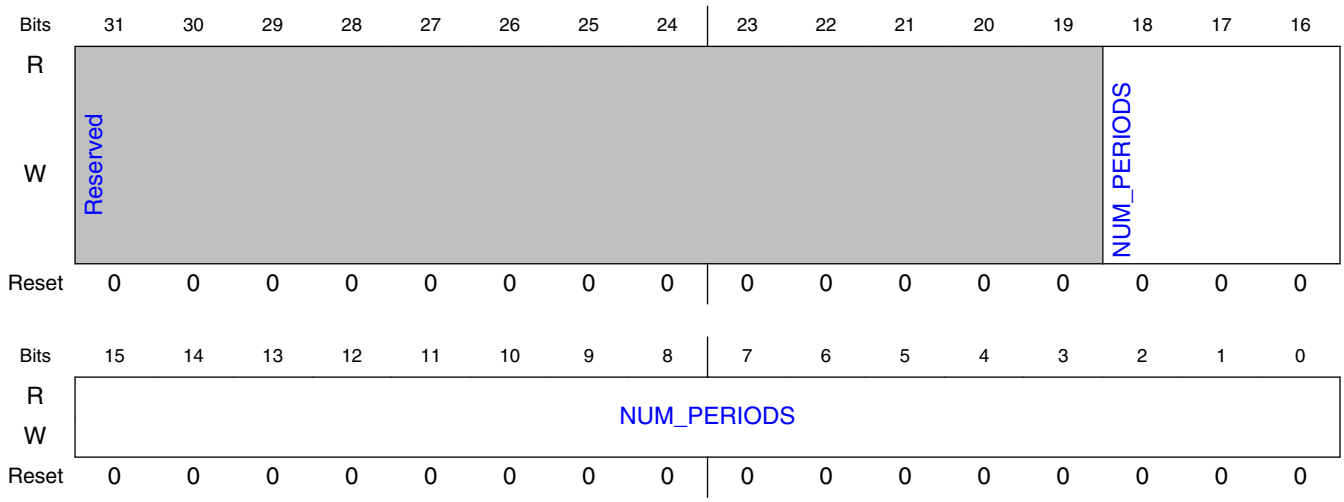
49.7.1.12.1 Offset

Register	Offset
CFG_TWAKEUP	28h

49.7.1.12.2 Function

DPHY Twakeup timing parameter. Sets the number of clk_esc clock periods to keep a clock or data lane in Mark-1 state after exiting ULPS. The MIPI DPHY spec requires a minimum of 1ms in Mark-1 state after leaving ULPS.

49.7.1.12.3 Diagram



49.7.1.12.4 Fields

Field	Function
31-19 —	Reserved
18-0 NUM_PERIODS	Number of Periods 0 - Zero clock periods 1 - One clock period 2 - Two clock periods ... ..

49.7.1.13 Configure Status Out (CFG_STATUS_OUT)

49.7.1.13.1 Offset

Register	Offset
CFG_STATUS_OUT	2Ch

49.7.1.13.2 Function

Shows status out.

49.7.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.1.13.4 Fields

Field	Function
31-0 STATUS	Status

49.7.1.14 Receive Error Status (RX_ERROR_STATUS)

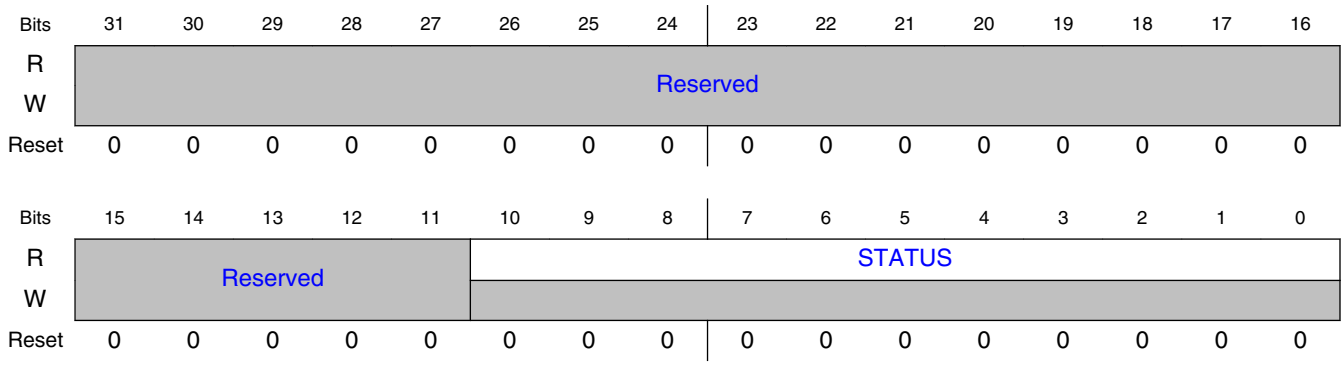
49.7.1.14.1 Offset

Register	Offset
RX_ERROR_STATUS	30h

49.7.1.14.2 Function

Shows status for host receive error detection, ECC errors, CRC errors and for timeout indicators

49.7.1.14.3 Diagram



49.7.1.14.4 Fields

Field	Function
31-11 —	Reserved
10-0 STATUS	Status 1b - Detected 0b - Not detected [0]: ECC single bit error detected [1]: ECC multi bit error detected [6:2]: Error bit position for single bit ECC error [7]: CRC error detected [8]: High Speed forward TX timeout detected [9]: Reverse Low power data receive timeout detected [10]: BTA timeout detected

49.7.2 DSI Host DPI Interface register descriptions

49.7.2.1 DPI_IF memory map

MIPI_DSI.DSI_HOST_DPI_INTFC base address: 4080_C200h

Offset	Register	Width (In bits)	Access	Reset value
0h	Pixel Payload Size (PIXEL_PAYLOAD_SIZE)	32	RW	0000_0000h
4h	Pixel FIFO Send Level (PIXEL_FIFO_SEND_LEVEL)	32	RW	0000_0000h
8h	Interface Color Coding (INTERFACE_COLOR_CODING)	32	RW	0000_0000h
Ch	Pixel Format (PIXEL_FORMAT)	32	RW	0000_0000h
10h	Vertical Sync Polarity (VSYNC_POLARITY)	32	RW	0000_0000h
14h	Horizontal Polarity (HSYNC_POLARITY)	32	RW	0000_0000h
18h	Video Mode (VIDEO_MODE)	32	RW	0000_0000h
1Ch	Horizontal Front Porch (HFP)	32	RW	0000_0000h
20h	Vertical Front Porch (HBP)	32	RW	0000_0000h
24h	Horizontal SYNC Blanking Packet Size (HSA)	32	RW	0000_0000h
28h	Enable Multiple Packets (ENABLE_MULT_PKTS)	32	RW	0000_0000h
2Ch	Vertical Back Porch (VBP)	32	RW	0000_0000h
30h	Vertical Front Porch (VFP)	32	RW	0000_0000h
34h	BLLP Mode (BLLP_MODE)	32	RW	0000_0000h
38h	Use Null Packet BLLP (USE_NULL_PKT_BLLP)	32	RW	0000_0000h
3Ch	Vertical Active (VACTIVE)	32	RW	0000_0000h

49.7.2.2 Pixel Payload Size (PIXEL_PAYLOAD_SIZE)

49.7.2.2.1 Offset

Register	Offset
PIXEL_PAYLOAD_SIZE	0h

49.7.2.2.2 Function

Sets maximum number of pixels that should be sent as one DSI packet. Recommended to be evenly divisible by the line size (in pixels).

49.7.2.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAYLOAD_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.2.4 Fields

Field	Function
31-16 —	Reserved
15-0 PAYLOAD_SIZE	Payload Size (in pixels). 0 - Zero pixels 1 - 1 pixel 2 - 2 pixel ... ...

49.7.2.3 Pixel FIFO Send Level (PIXEL_FIFO_SEND_LEVEL)

49.7.2.3.1 Offset

Register	Offset
PIXEL_FIFO_SEND_LEVEL	4h

49.7.2.3.2 Function

In order to optimize DSI utility, the DPI bridge buffers a certain number of DPI pixels before initiating a DSI packet. This configuration port controls the level at which the DPI Host bridge begins sending pixels.

49.7.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FIFO_SEND_LEVEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.3.4 Fields

Field	Function
31-16 —	Reserved
15-0 FIFO_SEND_LEVEL	FIFO Send Level 0 - Zero pixels 1 - 1 pixel 2 - 2 pixels ... ...

49.7.2.4 Interface Color Coding (INTERFACE_COLOR_CODING)

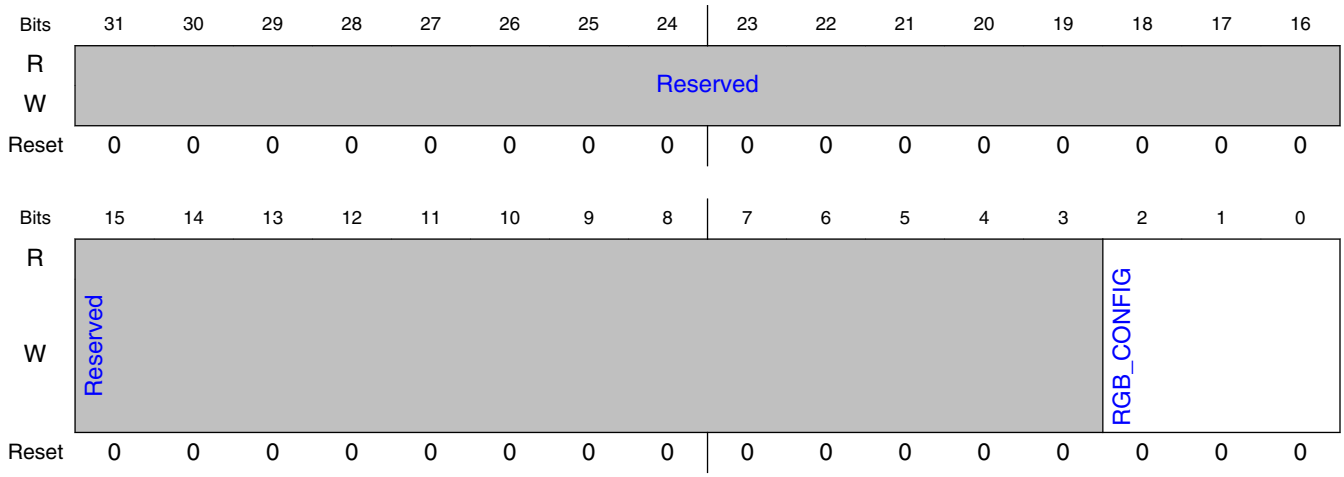
49.7.2.4.1 Offset

Register	Offset
INTERFACE_COLOR_CODING	8h

49.7.2.4.2 Function

Sets the distribution of RGB bits within the 24-bit d bus, as specified by the DPI specification.

49.7.2.4.3 Diagram



49.7.2.4.4 Fields

Field	Function
31-3 —	Reserved
2-0 RGB_CONFIG	RGB Bit Configuration. 000b - 16-bit Configuration 1 001b - 16-bit Configuration 2 010b - 16-bit Configuration 3 011b - 18-bit Configuration 1 100b - 18-bit Configuration 2 101b - 24-bit 110b,111b - Reserved

49.7.2.5 Pixel Format (PIXEL_FORMAT)

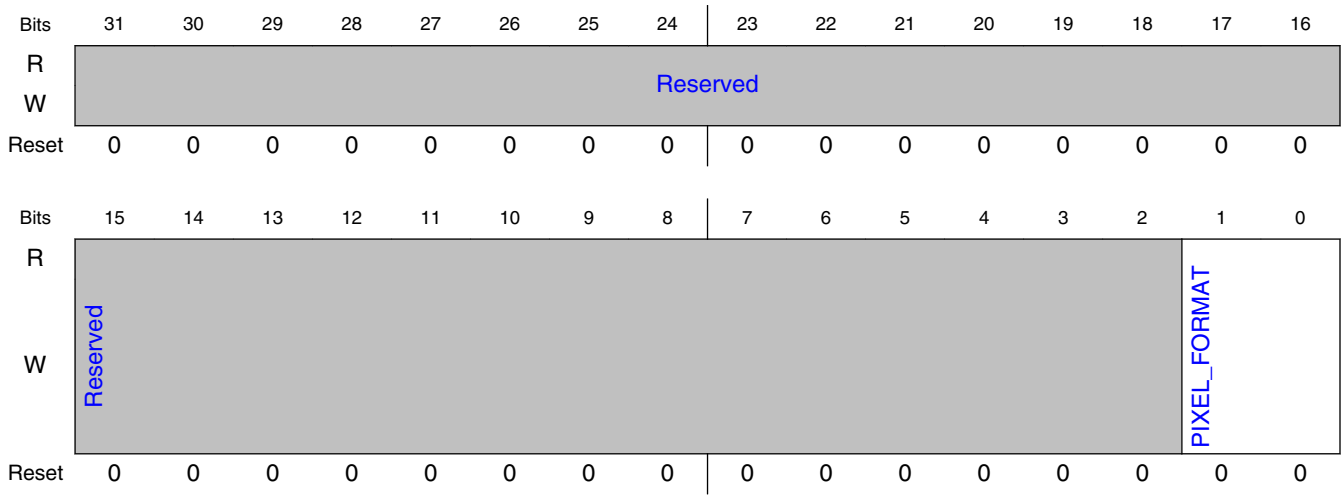
49.7.2.5.1 Offset

Register	Offset
PIXEL_FORMAT	Ch

49.7.2.5.2 Function

Sets the DSI packet type of the pixels.

49.7.2.5.3 Diagram



49.7.2.5.4 Fields

Field	Function
31-2 —	Reserved
1-0 PIXEL_FORMAT	Pixel Format 00b - 16 bit 01b - 18 bit 10b - 18 bit loosely packed 11b - 24 bit

49.7.2.6 Vertical Sync Polarity (VSYNC_POLARITY)

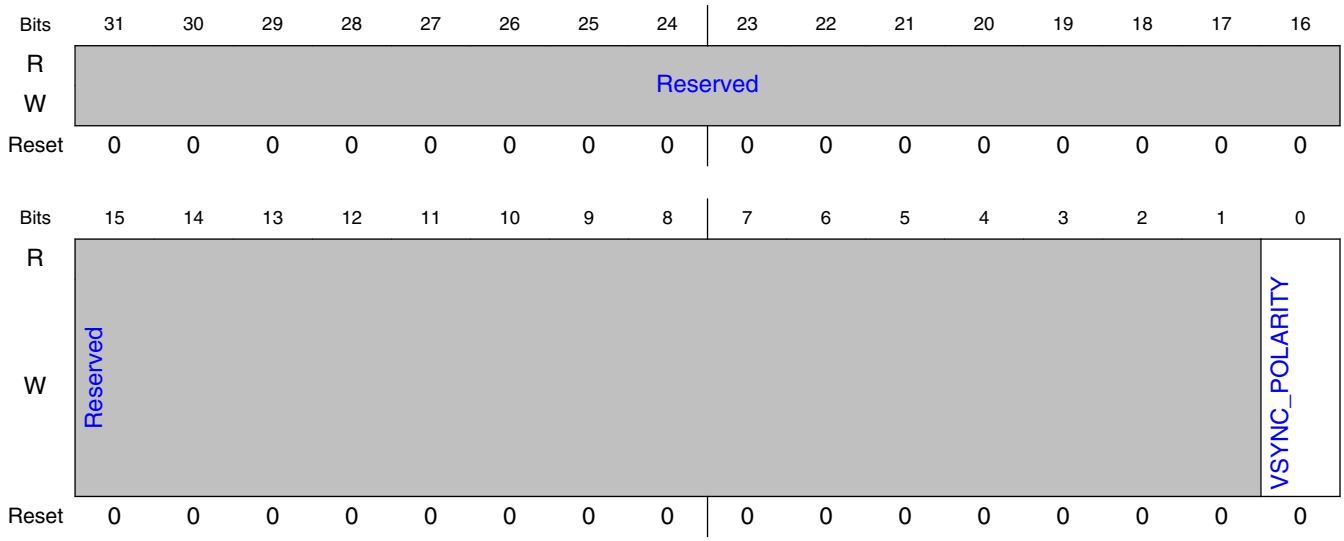
49.7.2.6.1 Offset

Register	Offset
VSYNC_POLARITY	10h

49.7.2.6.2 Function

Sets polarity of dpi_vsync_input.

49.7.2.6.3 Diagram



49.7.2.6.4 Fields

Field	Function
31-1 —	Reserved
0 VSYNC_POLARITY	VSYNC Polarity 0b - Active low 1b - Active high

49.7.2.7 Horizontal Polarity (HSYNC_POLARITY)

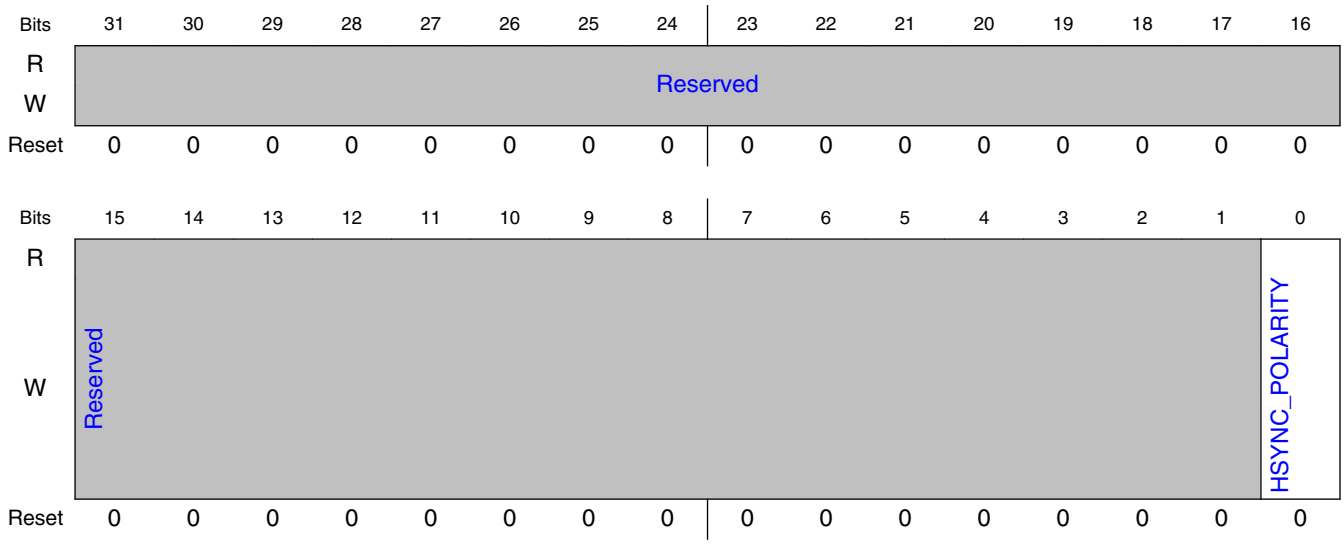
49.7.2.7.1 Offset

Register	Offset
HSYNC_POLARITY	14h

49.7.2.7.2 Function

Sets polarity of dpi_hsync_input.

49.7.2.7.3 Diagram



49.7.2.7.4 Fields

Field	Function
31-1 —	Reserved
0 HSYNC_POLARITY	HSYNC Polarity 0b - Active low 1b - Active high

49.7.2.8 Video Mode (VIDEO_MODE)

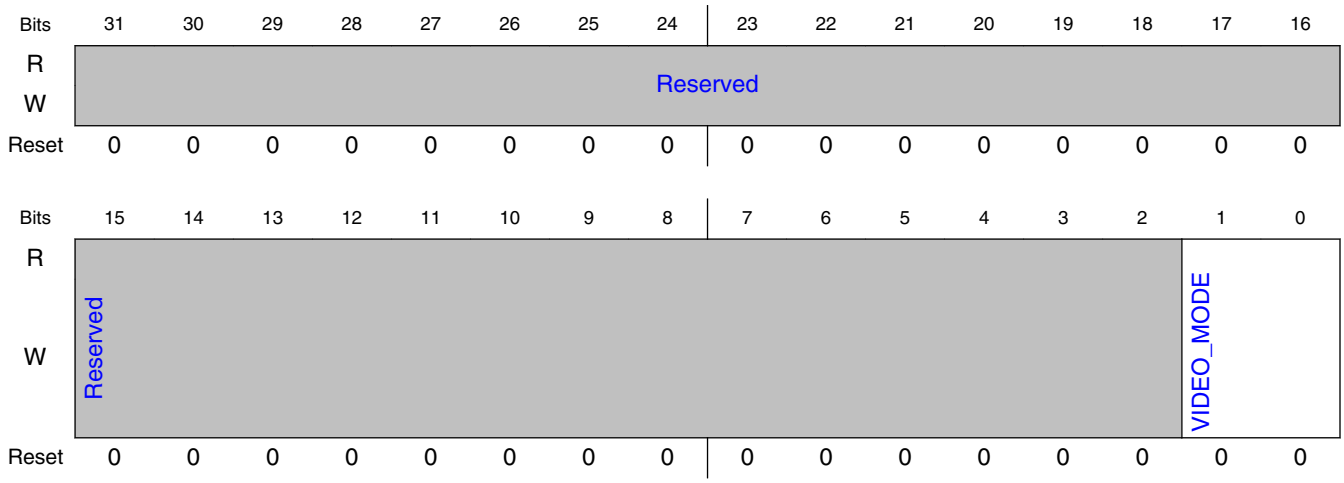
49.7.2.8.1 Offset

Register	Offset
VIDEO_MODE	18h

49.7.2.8.2 Function

Selects the DSI video mode for which the host DPI module generates packets.

49.7.2.8.3 Diagram



49.7.2.8.4 Fields

Field	Function
31-2 —	Reserved
1-0 VIDEO_MODE	Video Mode 00b - Non-Burst mode with Sync Pulses 01b - Non-Burst mode with Sync Events 10b - Burst mode 11b - Reserved

49.7.2.9 Horizontal Front Porch (HFP)

49.7.2.9.1 Offset

Register	Offset
HFP	1Ch

49.7.2.9.2 Function

Sets the DSI packet payload size, in bytes, of the horizontal front porch blanking packet.

49.7.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAYLOAD_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.9.4 Fields

Field	Function
31-16 —	Reserved
15-0 PAYLOAD_SIZE	Payload Size 0 - Zero bytes 1 - 1 byte 2 - 2 bytes ... ...

49.7.2.10 Vertical Front Porch (HBP)

49.7.2.10.1 Offset

Register	Offset
HBP	20h

49.7.2.10.2 Function

Sets the DSI packet payload size, in bytes, of the horizontal back porch blanking packet.

49.7.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAYLOAD_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.10.4 Fields

Field	Function
31-16 —	Reserved
15-0 PAYLOAD_SIZE	Payload Size 0 - Zero bytes 1 - 1 byte 2 - 2 bytes ... ...

49.7.2.11 Horizontal SYNC Blanking Packet Size (HSA)

49.7.2.11.1 Offset

Register	Offset
HSA	24h

49.7.2.11.2 Function

Sets the DSI packet payload size, in bytes, of the horizontal sync width filler blanking packet.

49.7.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAYLOAD_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.11.4 Fields

Field	Function
31-16 —	Reserved
15-0 PAYLOAD_SIZE	Payload Size 0 - Zero bytes 1 - 1 byte 2 - 2 bytes ... ...

49.7.2.12 Enable Multiple Packets (ENABLE_MULT_PKTS)

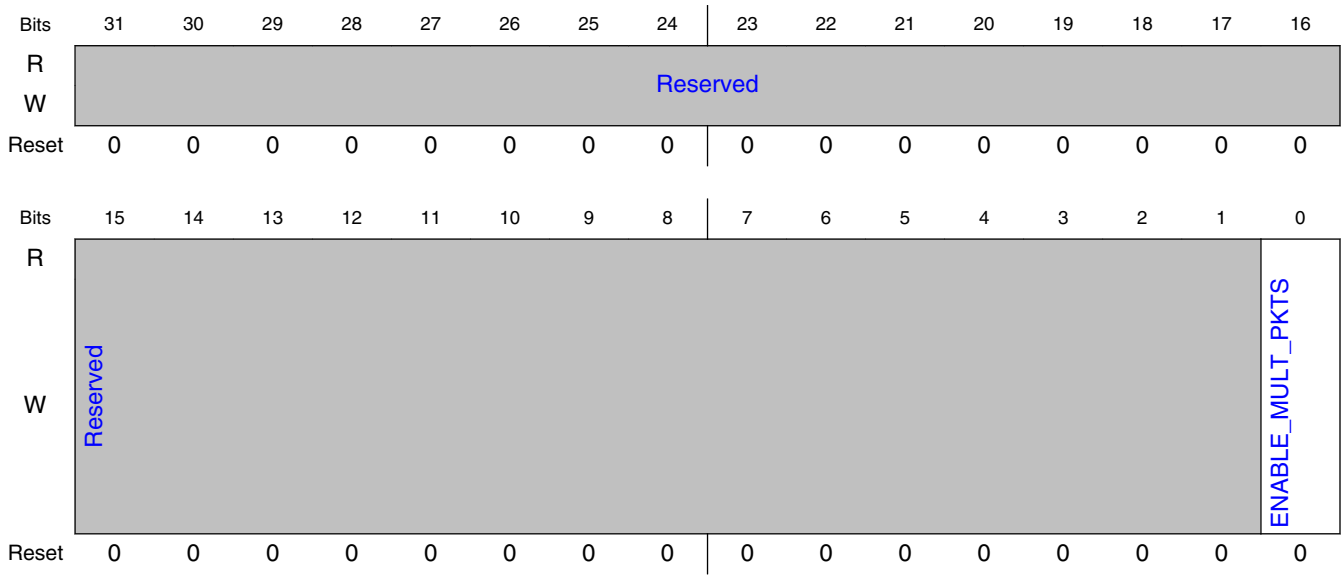
49.7.2.12.1 Offset

Register	Offset
ENABLE_MULT_PKTS	28h

49.7.2.12.2 Function

Enable Multiple packets per video line. When enabled, PIXEL_PAYLOAD_SIZE[PAYLOAD_SIZE] must be set to exactly half the size of the video line

49.7.2.12.3 Diagram



49.7.2.12.4 Fields

Field	Function
31-1 —	Reserved
0 ENABLE_MULT_PKTS	Enable Multiple Packets 0b - Video Line is sent in a single packet 1b - Video Line is sent in two packets

49.7.2.13 Vertical Back Porch (VBP)

49.7.2.13.1 Offset

Register	Offset
VBP	2Ch

49.7.2.13.2 Function

Sets the number of lines in the vertical back porch.

49.7.2.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								NUM_LINES							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.13.4 Fields

Field	Function
31-8 —	Reserved
7-0 NUM_LINES	Number of Lines 0 - Zero line 1 - 1 line 2 - 2 lines ... ...

49.7.2.14 Vertical Front Porch (VFP)

49.7.2.14.1 Offset

Register	Offset
VFP	30h

49.7.2.14.2 Function

Sets the number of lines in the vertical front porch.

49.7.2.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								NUM_LINES							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.2.14.4 Fields

Field	Function
31-8 —	Reserved
7-0 NUM_LINES	Number of lines 0 - Zero line 1 - 1 line 2 - 2 lines ... ...

49.7.2.15 BLLP Mode (BLLP_MODE)

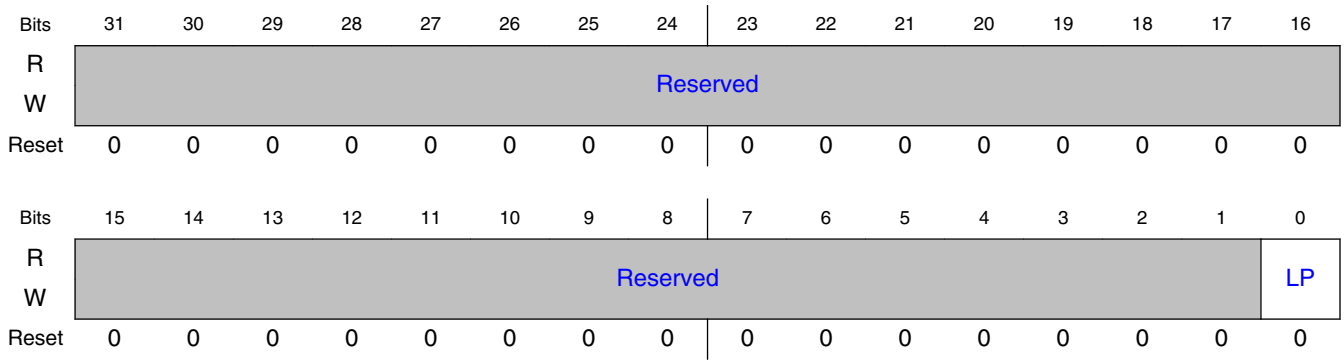
49.7.2.15.1 Offset

Register	Offset
BLLP_MODE	34h

49.7.2.15.2 Function

Optimize bllp periods to Low Power mode when possible.

49.7.2.15.3 Diagram



49.7.2.15.4 Fields

Field	Function
31-1 —	Reserved
0 LP	Low Power mode when possible. 0b - Blanking packets are sent during BLLP periods 1b - LP mode is used for BLLP periods

49.7.2.16 Use Null Packet BLLP (USE_NULL_PKT_BLLP)

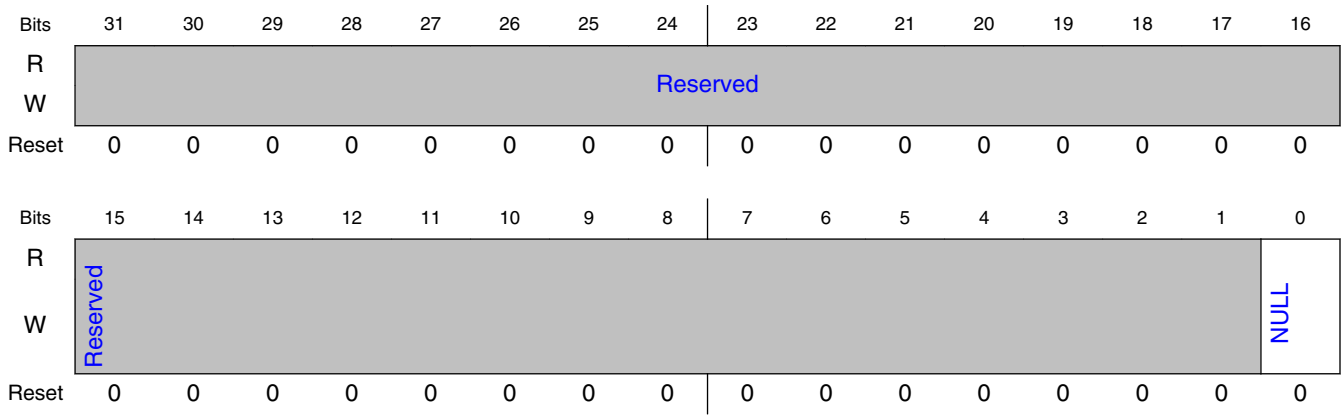
49.7.2.16.1 Offset

Register	Offset
USE_NULL_PKT_BLLP	38h

49.7.2.16.2 Function

Selects type of blanking packet to be sent during bllp.

49.7.2.16.3 Diagram



49.7.2.16.4 Fields

Field	Function
31-1 —	Reserved
0 NULL	Null 0b - Blanking packet used in bllp region 1 1b - Null packet used in bllp region

49.7.2.17 Vertical Active (VACTIVE)

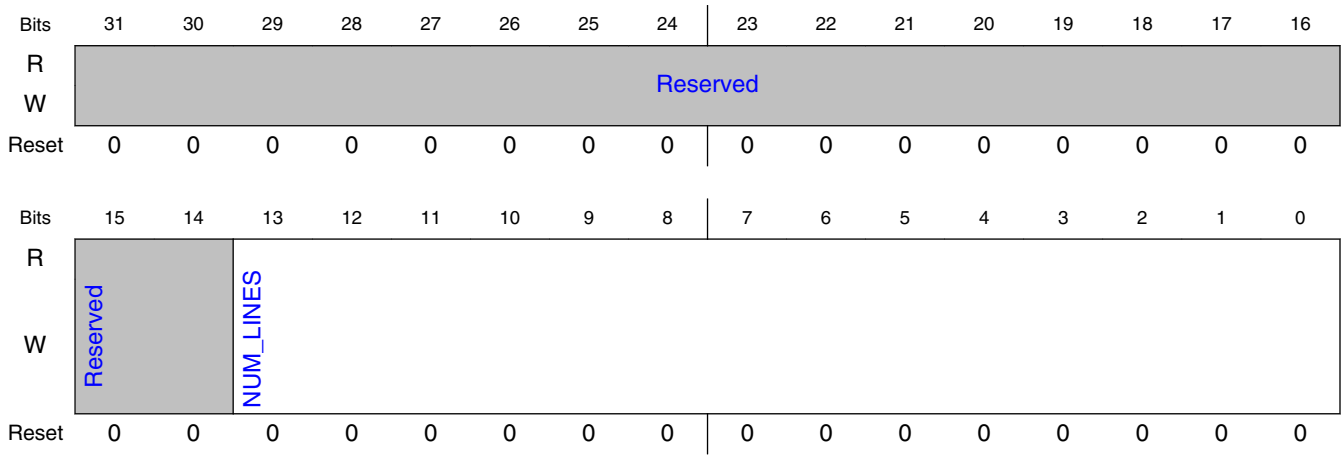
49.7.2.17.1 Offset

Register	Offset
VACTIVE	3Ch

49.7.2.17.2 Function

Sets the number of lines in the vertical active aread.

49.7.2.17.3 Diagram



49.7.2.17.4 Fields

Field	Function
31-14 —	Reserved
13-0 NUM_LINES	Number of lines 0 - Zero line 1 - 1 line 2 - 2 lines ... ...

49.7.3 DSI HOST APB PKT Interface register descriptions

49.7.3.1 APB_PKT_IF memory map

MIPI_DSI.DSI_HOST_APB_PKT_IF base address: 4080_C280h

Offset	Register	Width (In bits)	Access	Reset value
0h	Transmit Payload Data Write (TX_PAYLOAD)	32	RW	0000_0000h
4h	Packet Control (PKT_CONTROL)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8h	Send Packet (SEND_PACKET)	32	RW	0000_0000h
Ch	Packet Status (PKT_STATUS)	32	R	0000_0000h
10h	Packet FIFO Write Level (PKT_FIFO_WR_LEVEL)	32	R	0000_0000h
14h	Packet FIFO Read Level (PKT_FIFO_RD_LEVEL)	32	R	0000_0000h
18h	Packet Receive Payload (PKT_RX_PAYLOAD)	32	R	0000_0000h
1Ch	Packet Receive Packet Header (PKT_RX_PKT_HEADER)	32	R	0000_0000h
20h	Interrupt Status (IRQ_STATUS)	32	R	0000_0000h
24h	Interrupt Status 2 (IRQ_STATUS2)	32	R	0000_0000h
28h	Interrupt Mask (IRQ_MASK)	32	RW	0000_0000h
2Ch	Interrupt Mask 2 (IRQ_MASK2)	32	RW	0000_0000h

49.7.3.2 Transmit Payload Data Write (TX_PAYLOAD)

49.7.3.2.1 Offset

Register	Offset
TX_PAYLOAD	0h

49.7.3.2.2 Function

Writes to this register loads the payload FIFO with 32 bit values.

49.7.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PAYLOAD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PAYLOAD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.2.4 Fields

Field	Function
31-0 PAYLOAD	Payload

49.7.3.3 Packet Control (PKT_CONTROL)

49.7.3.3.1 Offset

Register	Offset
PKT_CONTROL	4h

49.7.3.3.2 Function

Transmit packet control.

49.7.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					CTRL										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.3.4 Fields

Field	Function
31-27 —	Reserved
26-0 CTRL	Control [15:0] - Packet word count [17:16] - Reserved

Field	Function
	[23:18] - Packet Header DSI Data Type [24] - Lp or HS select: 0 - LP mode, 1 - HS mode [25] - perform BTA after packet is sent [26] - perform BTA only, no packet Tx

49.7.3.4 Send Packet (SEND_PACKET)

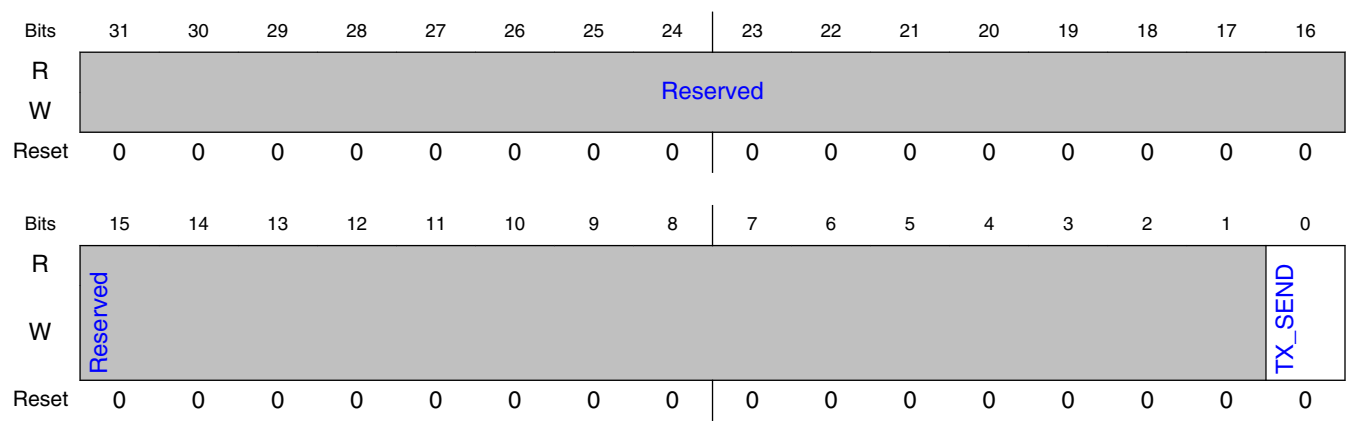
49.7.3.4.1 Offset

Register	Offset
SEND_PACKET	8h

49.7.3.4.2 Function

Writes to this register causes the packet described in dsi_host_pkt_control to be sent.

49.7.3.4.3 Diagram



49.7.3.4.4 Fields

Field	Function
31-1 —	Reserved
0 TX_SEND	Transmit send packet 0b - Packet not sent 1b - Packet is sent

49.7.3.5 Packet Status (PKT_STATUS)

49.7.3.5.1 Offset

Register	Offset
PKT_STATUS	Ch

49.7.3.5.2 Function

Status of APB to packet interface.

49.7.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.5.4 Fields

Field	Function
31-9 —	Reserved
8-0 STATUS	Status 0b - event hasn't occurred 1b - event occurred [0] - state machine not idle [1] - Tx packet done [2] - dphy direction: 0 - Tx had control, 1 - Rx has control [3] - Tx FIFO overflow [4] - Tx FIFO underflow

Field	Function
	[5] - Rx FIFO overflow [6] - Rx FIFO underflow [7] - Rx packet header has been received [8] - all Rx packet payload data has been received

49.7.3.6 Packet FIFO Write Level (PKT_FIFO_WR_LEVEL)

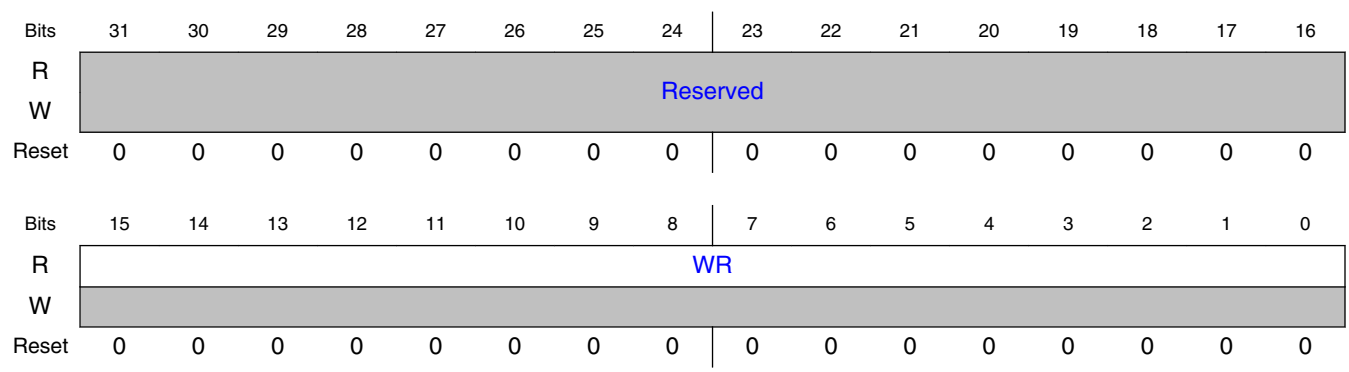
49.7.3.6.1 Offset

Register	Offset
PKT_FIFO_WR_LEVEL	10h

49.7.3.6.2 Function

Write level of APB to packet interface FIFO.

49.7.3.6.3 Diagram



49.7.3.6.4 Fields

Field	Function
31-16 —	Reserved
15-0 WR	Write level

49.7.3.7 Packet FIFO Read Level (PKT_FIFO_RD_LEVEL)

49.7.3.7.1 Offset

Register	Offset
PKT_FIFO_RD_LEVEL	14h

49.7.3.7.2 Function

Read level of APB to packet interface FIFO.

49.7.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.7.4 Fields

Field	Function
31-16 —	Reserved
15-0 RD	Read level

49.7.3.8 Packet Receive Payload (PKT_RX_PAYLOAD)

49.7.3.8.1 Offset

Register	Offset
PKT_RX_PAYLOAD	18h

49.7.3.8.2 Function

APB to packet interface receive payload read.

49.7.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAYLOAD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAYLOAD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.8.4 Fields

Field	Function
31-0 PAYLOAD	Payload

49.7.3.9 Packet Receive Packet Header (PKT_RX_PKT_HEADER)

49.7.3.9.1 Offset

Register	Offset
PKT_RX_PKT_HEADER	1Ch

49.7.3.9.2 Function

APB to packet interface receive packet header.

49.7.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								HEADER							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HEADER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.9.4 Fields

Field	Function
31-24 —	Reserved
23-0 HEADER	Header [15:0] - Word count [21:16] - Data type [23:22] - Reserved

49.7.3.10 Interrupt Status (IRQ_STATUS)

49.7.3.10.1 Offset

Register	Offset
IRQ_STATUS	20h

49.7.3.10.2 Function

Status of APB to packet interface.

49.7.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.10.4 Fields

Field	Function
31-0 STATUS	Status 0b - Event has not occurred 1b - event occurred [0] - State machine not idle [1] - Tx packet done [2] - dphy direction: 0 - Tx had control, 1 - Rx has control [3] - Tx FIFO overflow [4] - Tx FIFO underflow [5] - Rx FIFO overflow [6] - Rx FIFO underflow [7] - Rx packet header has been received [8] - All Rx packet payload data has been received [28:9] - Map directory to dsi host controller status_out port bit descriptions [29] - High speed Tx timeout [30] - Low power Rx timeout [31] - Host BTA timeout

49.7.3.11 Interrupt Status 2 (IRQ_STATUS2)

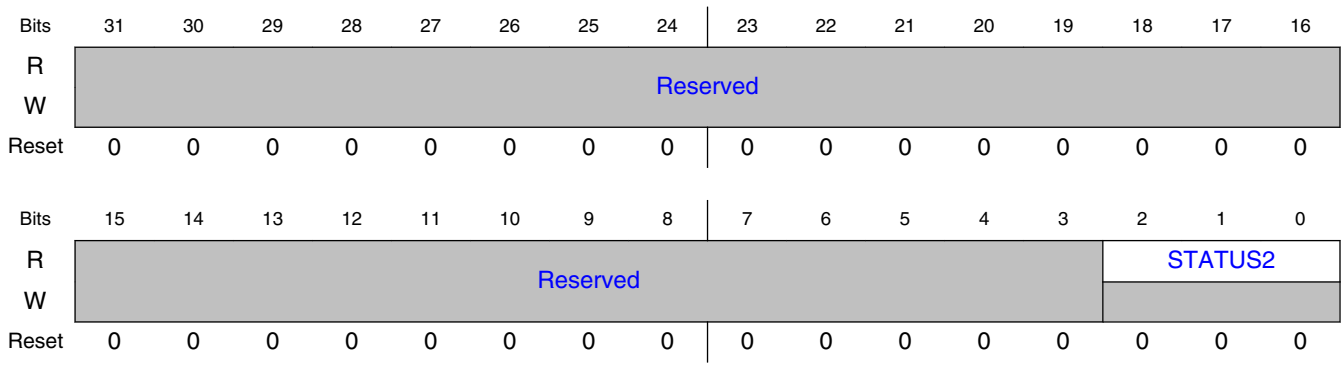
49.7.3.11.1 Offset

Register	Offset
IRQ_STATUS2	24h

49.7.3.11.2 Function

Reads status of APB to packet interface part 2. Read part 2 first then dsi_host_irq_status. Reading dsi_host_irq_status clears both status and status2.

49.7.3.11.3 Diagram



49.7.3.11.4 Fields

Field	Function
31-3 —	Reserved
2-0 STATUS2	Status 2 0b - event hasn't occurred 1b - event occurred [0] - Single bit ecc error [1] - Multi bit ecc error [2] - CRC error

49.7.3.12 Interrupt Mask (IRQ_MASK)

49.7.3.12.1 Offset

Register	Offset
IRQ_MASK	28h

49.7.3.12.2 Function

Sets Mask.

49.7.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.3.12.4 Fields

Field	Function
31-0 MASK	IRQ Mask 0b - Not mask 1b - Mask [0] - state machine not idle [1] - Tx packet done [2] - dphy direction: 0 - Tx had control, 1 - Rx has control [3] - Tx FIFO overflow [4] - Tx FIFO underflow [5] - Rx FIFO overflow [6] - Rx FIFO underflow [7] - Rx packet header has been received [8] - All Rx packet payload data has been received [28:9] - Map directory to dsi host controller status_out port bit descriptions [29] - High speed Tx timeout [30] - Low power Rx timeout [31] - Host BTA timeout

49.7.3.13 Interrupt Mask 2 (IRQ_MASK2)

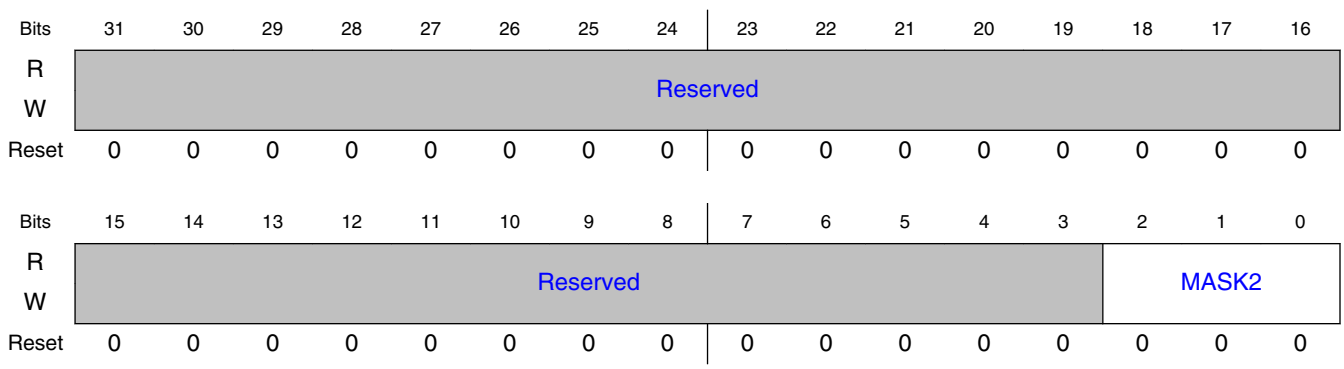
49.7.3.13.1 Offset

Register	Offset
IRQ_MASK2	2Ch

49.7.3.13.2 Function

Sets Mask.

49.7.3.13.3 Diagram



49.7.3.13.4 Fields

Field	Function
31-3 —	Reserved
2-0 MASK2	IRQ Mask 2 0b - Not mask 1b - Mask [0] - Single bit ecc error [1] - Multi bit ecc error [2] - CRC error

49.7.4 DSI HOST DPHY INTFC register descriptions

49.7.4.1 DSI_HOST_DPHY_INTFC memory map

MIPI_DSI.DSI_HOST_DPHY_INTFC base address: 4080_C300h

Offset	Register	Width (In bits)	Access	Reset value
0h	Powerdown Transmit (PD_TX)	32	RW	0000_0001h
4h	M Program High Speed Prepare (M_PRG_HS_PREPARE)	32	RW	0000_0000h
8h	MC Program High Speed Prepare (MC_PRG_HS_PREPARE)	32	RW	0000_0000h
Ch	M Program High Speed Zero (M_PRG_HS_ZERO)	32	RW	0000_0000h
10h	MC Program High Speed Zero (MC_PRG_HS_ZERO)	32	RW	0000_0000h
14h	M Program High Speed Trail (M_PRG_HS_TRAIL)	32	RW	0000_0000h
18h	MC Program High Speed Trail (MC_PRG_HS_TRAIL)	32	RW	0000_0000h
1Ch	Powerdown PLL (PD_PLL)	32	RW	0000_0001h
20h	Test (TST)	32	RW	0000_0025h
24h	Control N Divider (CN)	32	RW	0000_0000h
28h	Control M Divider (CM)	32	RW	0000_0000h
2Ch	Control Divider (CO)	32	RW	0000_0000h
30h	Lock (LOCK)	32	R	0000_0000h
34h	Lock BYP (LOCK_BYP)	32	RW	0000_0000h
38h	TX_RCAL (TX_RCAL)	32	RW	0000_0000h
3Ch	AUTO_PD_EN (AUTO_PD_EN)	32	RW	0000_0000h
40h	RXLPRP (RXLPRP)	32	RW	0000_0000h
44h	RXCDRP (RXCDRP)	32	RW	0000_0000h

49.7.4.2 Powerdown Transmit (PD_TX)

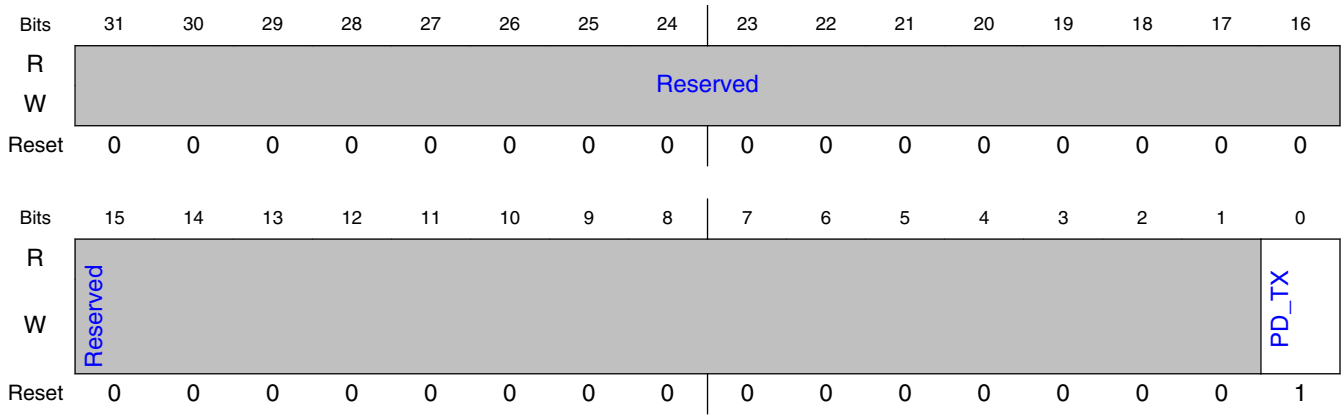
49.7.4.2.1 Offset

Register	Offset
PD_TX	0h

49.7.4.2.2 Function

Power Down input for D-PHY

49.7.4.2.3 Diagram



49.7.4.2.4 Fields

Field	Function
31-1 —	Reserved
0 PD_TX	Power Down Transmit 0b - Power Up 1b - Power Down

49.7.4.3 M Program High Speed Prepare (M_PRG_HS_PREPARE)

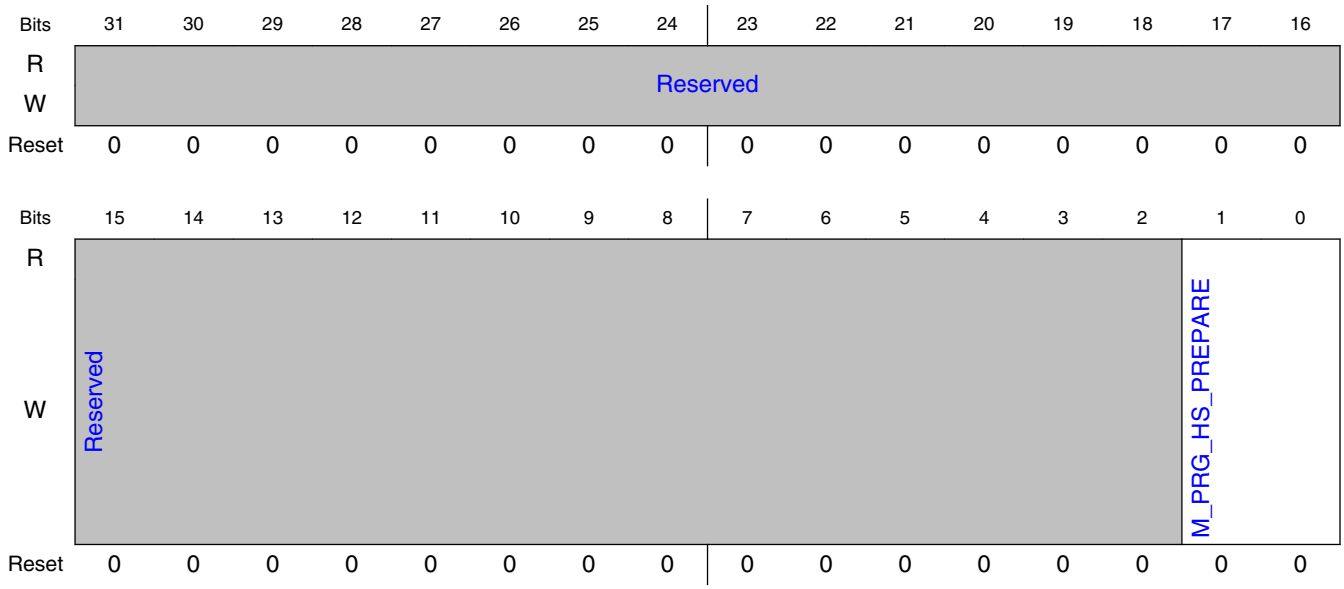
49.7.4.3.1 Offset

Register	Offset
M_PRG_HS_PREPARE	4h

49.7.4.3.2 Function

Bits used to program T_HS_PREPARE time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming

49.7.4.3.3 Diagram



49.7.4.3.4 Fields

Field	Function
31-2 —	Reserved
1-0 M_PRG_HS_PREPARE	DPHY m_PRG_HS_PREPARE input

49.7.4.4 MC Program High Speed Prepare (MC_PRG_HS_PREPARE)

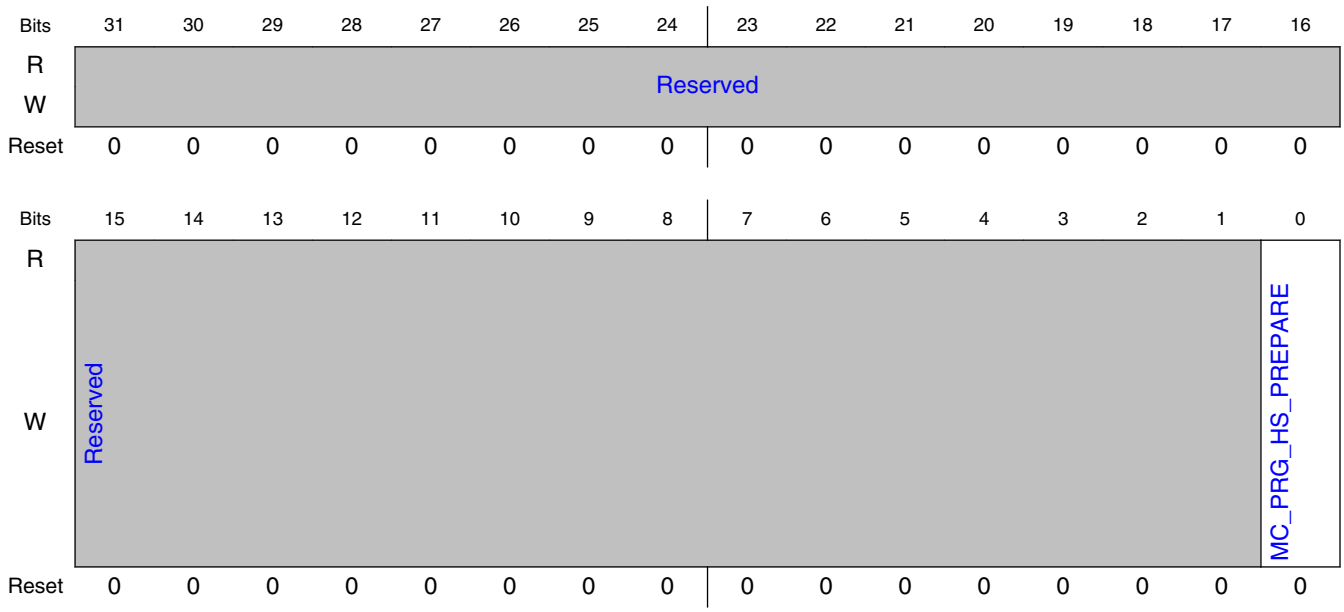
49.7.4.4.1 Offset

Register	Offset
MC_PRG_HS_PREPARE	8h

49.7.4.4.2 Function

Bits used to program T_CLK_PREPARE time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming

49.7.4.4.3 Diagram



49.7.4.4.4 Fields

Field	Function
31-1 —	Reserved
0 MC_PRG_HS_PREPARE	DPHY mc_PRG_HS_PREPARE input

49.7.4.5 M Program High Speed Zero (M_PRG_HS_ZERO)

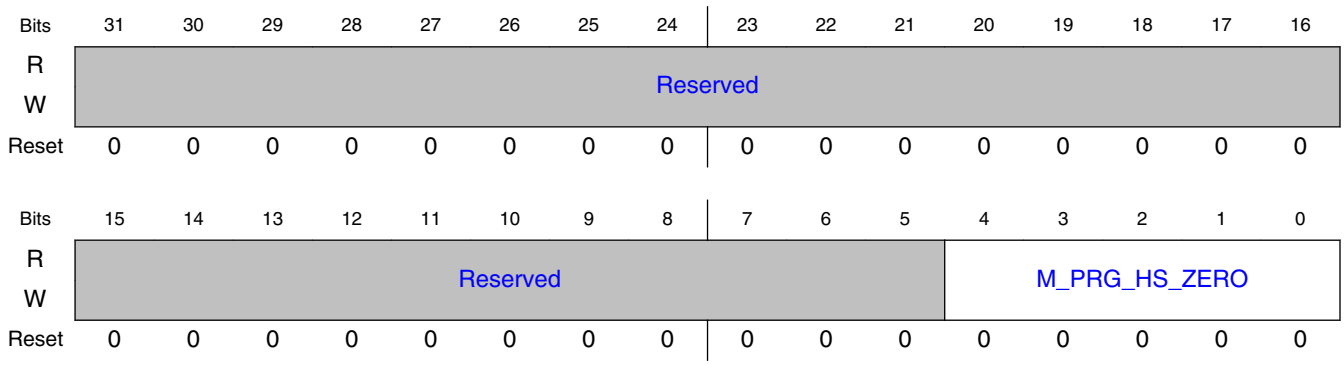
49.7.4.5.1 Offset

Register	Offset
M_PRG_HS_ZERO	Ch

49.7.4.5.2 Function

Bits used to program T_HS_ZERO time in the beginning of high speed transmission mode. For more details on how to program this field, see DPHY Programming

49.7.4.5.3 Diagram



49.7.4.5.4 Fields

Field	Function
31-5 —	Reserved
4-0 M_PRG_HS_ZERO	DPHY m_PRG_HS_ZERO input

49.7.4.6 MC Program High Speed Zero (MC_PRG_HS_ZERO)

49.7.4.6.1 Offset

Register	Offset
MC_PRG_HS_ZERO	10h

49.7.4.6.2 Function

Bits used to program T_CLK_ZERO time in the beginning of high speed transmission mode.For more details on how to program this field, see DPHY Programming

49.7.4.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										MC_PRG_HS_ZERO					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.4.6.4 Fields

Field	Function
31-6 —	Reserved
5-0 MC_PRG_HS_ZERO	DPHY mc_PRG_HS_ZERO input

49.7.4.7 M Program High Speed Trail (M_PRG_HS_TRAIL)

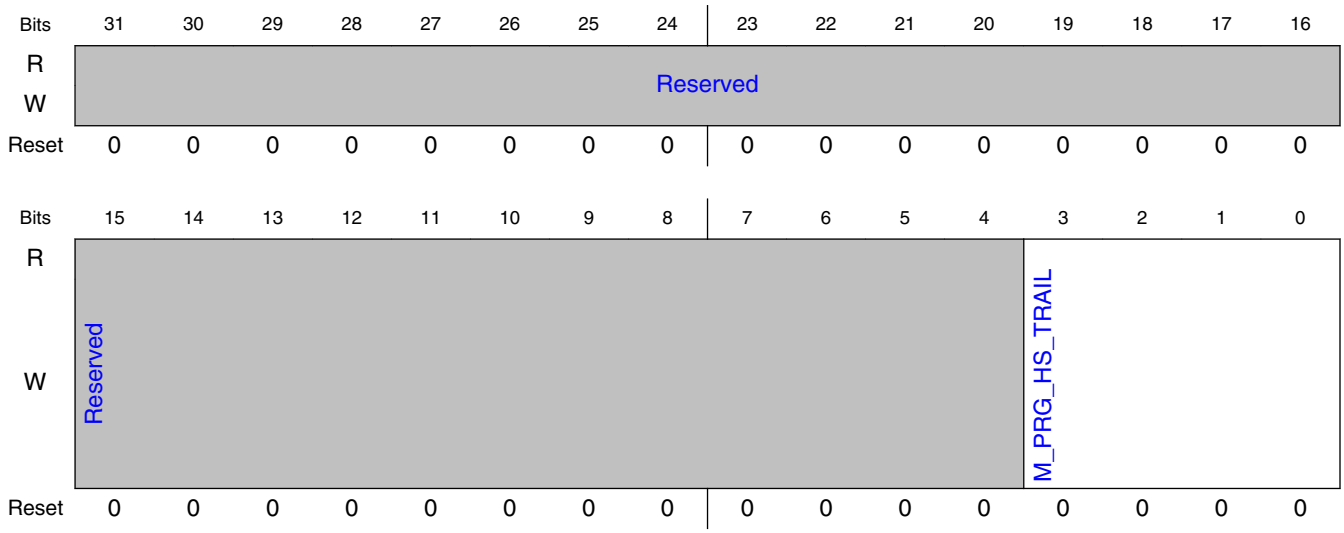
49.7.4.7.1 Offset

Register	Offset
M_PRG_HS_TRAIL	14h

49.7.4.7.2 Function

Bits used to program T_HS_TRAIL time in the end of high speed transmission mode. For more details on how to program this field, see DPHY Programming

49.7.4.7.3 Diagram



49.7.4.7.4 Fields

Field	Function
31-4 —	Reserved
3-0 M_PRG_HS_TRAIL	DPHY m_PRG_HS_TRAIL input

49.7.4.8 MC Program High Speed Trail (MC_PRG_HS_TRAIL)

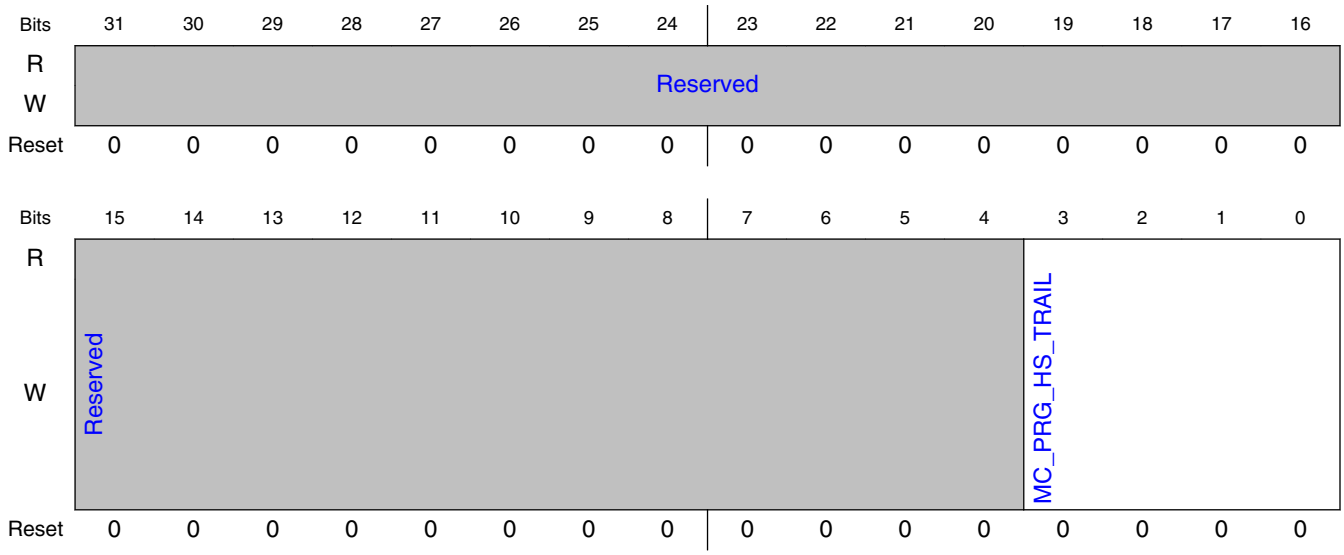
49.7.4.8.1 Offset

Register	Offset
MC_PRG_HS_TRAIL	18h

49.7.4.8.2 Function

Bits used to program T_CLK_TRAIL time in the end of high speed transmission mode.

49.7.4.8.3 Diagram



49.7.4.8.4 Fields

Field	Function
31-4 —	Reserved
3-0 MC_PRG_HS_TRAIL	DPHY mc_PRG_HS_TRAIL input

49.7.4.9 Powerdown PLL (PD_PLL)

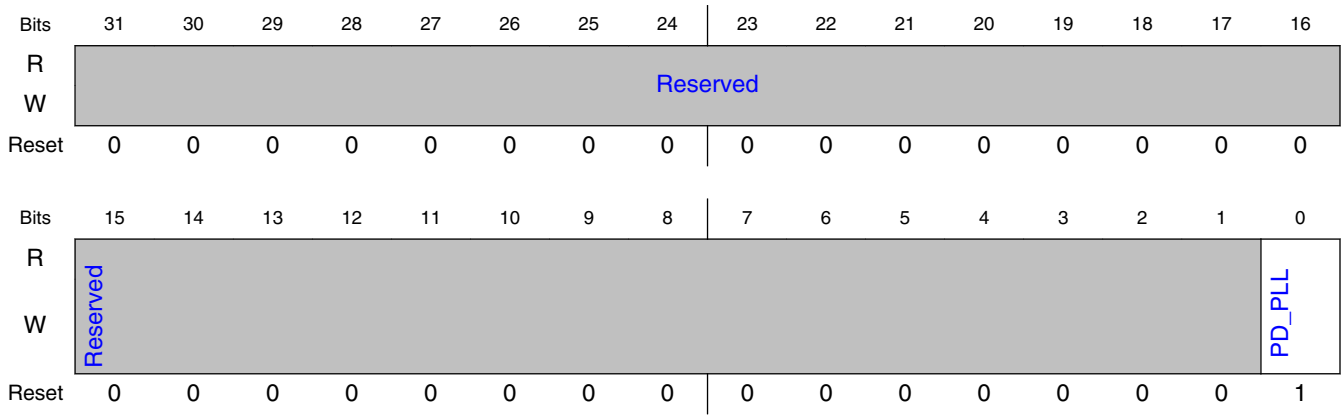
49.7.4.9.1 Offset

Register	Offset
PD_PLL	1Ch

49.7.4.9.2 Function

When high, the PLL is powered down.

49.7.4.9.3 Diagram



49.7.4.9.4 Fields

Field	Function
31-1 —	Reserved
0 PD_PLL	Powerdown signal When high, the PLL is powered down 0b - Power up PLL 1b - Power down PLL

49.7.4.10 Test (TST)

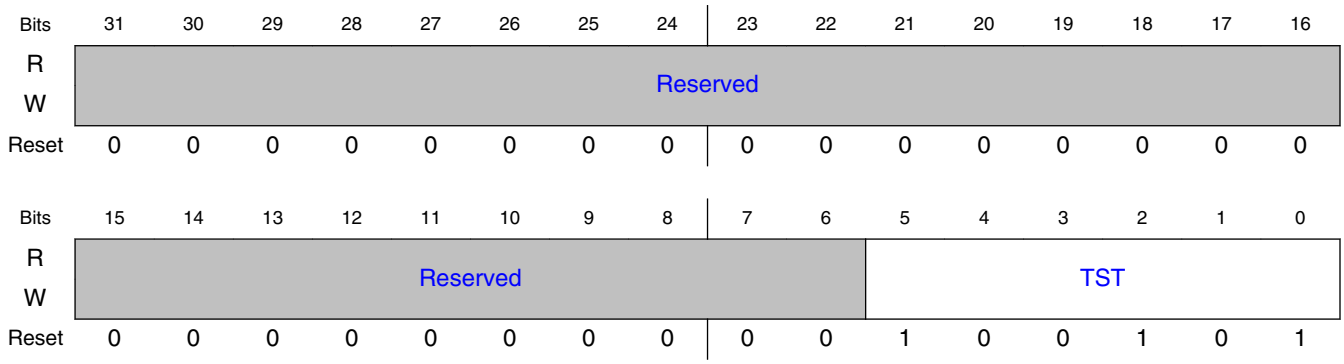
49.7.4.10.1 Offset

Register	Offset
TST	20h

49.7.4.10.2 Function

Set TST[5:0]=[100101] at normal operation

49.7.4.10.3 Diagram



49.7.4.10.4 Fields

Field	Function
31-6 —	Reserved
5-0 TST	Test

49.7.4.11 Control N Divider (CN)

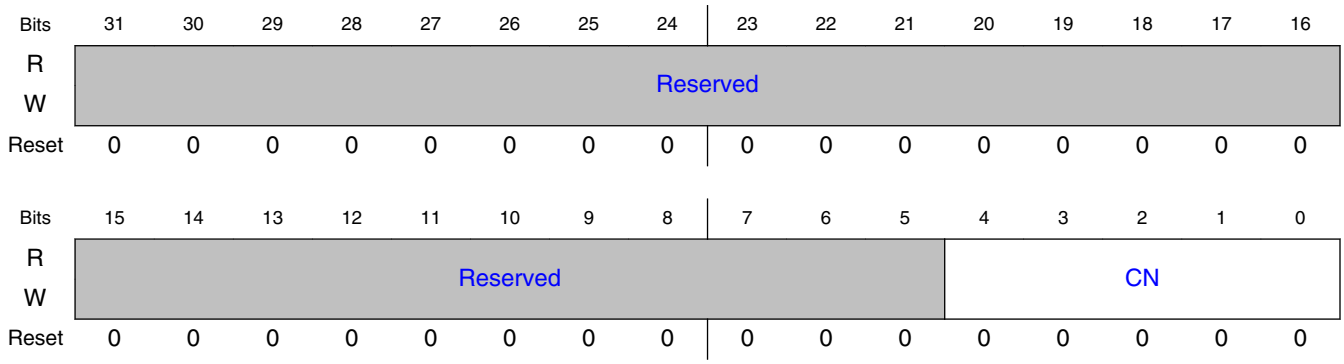
49.7.4.11.1 Offset

Register	Offset
CN	24h

49.7.4.11.2 Function

Sets Control N divider.

49.7.4.11.3 Diagram



49.7.4.11.4 Fields

Field	Function
31-5 —	Reserved
4-0 CN	Control N Divider

CN[4:0]	DVR
11111	**1**
00000	**2**
10000	**3**
11000	**4**
11100	**5**
01110	**6**
00111	**7**
10011	**8**
01001	**9**
00100	**10**
00010	**11**
10001	**12**
01000	**13**
10100	**14**
01010	**15**
10101	**16**
11010	**17**
11101	**18**
11110	**19**
01111	**20**
10111	**21**

Field	Function	
	CN[4:0]	DVR
	11011	**22**
	01101	**23**
	10110	**24**
	01011	**25**
	00101	**26**
	10010	**27**
	11001	**28**
	01100	**29**
	00110	**30**
	00011	**31**
	00001	**32**

49.7.4.12 Control M Divider (CM)

49.7.4.12.1 Offset

Register	Offset
CM	28h

49.7.4.12.2 Function

Sets Control M Divider

49.7.4.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49.7.4.12.4 Fields

Field	Function
31-8 —	Reserved
7-0 CM	Control M divider

CM[7:0]	DVR
111X0000	16
111X0001	17
111X0010	18
111X0011	19
111X0100	20
111X0101	21
111X0110	22
111X0111	23
111X1000	24
111X1001	25
111X1010	26
111X1011	27
111X1100	28
111X1101	29
111X1110	30
111X1111	31
11000000	32
11000001	33
11000010	34
11000011	35
11000100	36
11000101	37
11000110	38
11000111	39
11001000	40
11001001	41
11001010	42
11001011	43
11001100	44
11001101	45
11001110	46
11001111	47
11010000	48

Field	Function	
	CM[7:0]	DVR
	11010001	49
	11010010	50
	11010011	51
	11010100	52
	11010101	53
	11010110	54
	11010111	55
	11011000	56
	11011001	57
	11011010	58
	11011011	59
	11011100	60
	11011101	61
	11011110	62
	11011111	63
	10000000	64
	10000001	65
	10000010	66
	10000011	67
	10000100	68
	10000101	69
	10000110	70
	10000111	71
	10001000	72
	10001001	73
	10001010	74
	10001011	75
	10001100	76
	10001101	77
	10001110	78
	10001111	79
	10010000	80
	10010001	81
	10010010	82
	10010011	83
	10010100	84
	10010101	85
	10010110	86
	10010111	87
	10011000	88

Field	Function	
	CM[7:0]	DVR
	10011001	89
	10011010	90
	10011011	91
	10011100	92
	10011101	93
	10011110	94
	10011111	95
	10100000	96
	10100001	97
	10100010	98
	10100011	99
	10100100	100
	10100101	101
	10100110	102
	10100111	103
	10101000	104
	10101001	105
	10101010	106
	10101011	107
	10101100	108
	10101101	109
	10101110	110
	10101111	111
	10110000	112
	10110001	113
	10110010	114
	10110011	115
	10110100	116
	10110101	117
	10110110	118
	10110111	119
	10111000	120
	10111001	121
	10111010	122
	10111011	123
	10111100	124
	10111101	125
	10111110	126
	10111111	127
	00000000	128

Field	Function	
	CM[7:0]	DVR
	00000001	129
	00000010	130
	00000011	131
	00000100	132
	00000101	133
	00000110	134
	00000111	135
	00001000	136
	00001001	137
	00001010	138
	00001011	139
	00001100	140
	00001101	141
	00001110	142
	00001111	143
	00010000	144
	00010001	145
	00010010	146
	00010011	147
	00010100	148
	00010101	149
	00010110	150
	00010111	151
	00011000	152
	00011001	153
	00011010	154
	00011011	155
	00011100	156
	00011101	157
	00011110	158
	00011111	159
	00100000	160
	00100001	161
	00100010	162
	00100011	163
	00100100	164
	00100101	165
	00100110	166
	00100111	167
	00101000	168

Field	Function	
	CM[7:0]	DVR
	00101001	169
	00101010	170
	00101011	171
	00101100	172
	00101101	173
	00101110	174
	00101111	175
	00110000	176
	00110001	177
	00110010	178
	00110011	179
	00110100	180
	00110101	181
	00110110	182
	00110111	183
	00111000	184
	00111001	185
	00111010	186
	00111011	187
	00111100	188
	00111101	189
	00111110	190
	00111111	191
	01000000	192
	01000001	193
	01000010	194
	01000011	195
	01000100	196
	01000101	197
	01000110	198
	01000111	199
	01001000	200
	01001001	201
	01001010	202
	01001011	203
	01001100	204
	01001101	205
	01001110	206
	01001111	207
	01010000	208

Field	Function	
	CM[7:0]	DVR
	01010001	209
	01010010	210
	01010011	211
	01010100	212
	01010101	213
	01010110	214
	01010111	215
	01011000	216
	01011001	217
	01011010	218
	01011011	219
	01011100	220
	01011101	221
	01011110	222
	01011111	223
	01100000	224
	01100001	225
	01100010	226
	01100011	227
	01100100	228
	01100101	229
	01100110	230
	01100111	231
	01101000	232
	01101001	233
	01101010	234
	01101011	235
	01101100	236
	01101101	237
	01101110	238
	01101111	239
	01110000	240
	01110001	241
	01110010	242
	01110011	243
	01110100	244
	01110101	245
	01110110	246
	01110111	247
	01111000	248

Field	Function	
	CM[7:0]	DVR
	01111001	249
	01111010	250
	01111011	251
	01111100	252
	01111101	253
	01111110	254
	01111111	255

49.7.4.13 Control Divider (CO)

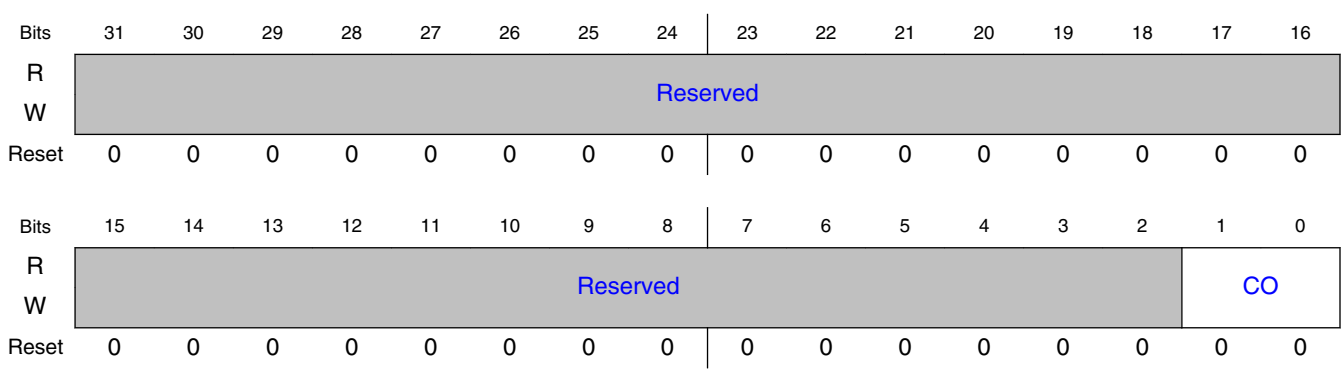
49.7.4.13.1 Offset

Register	Offset
CO	2Ch

49.7.4.13.2 Function

Sets the control divider.

49.7.4.13.3 Diagram



49.7.4.13.4 Fields

Field	Function
31-2 —	Reserved
1-0 CO	Control O divider 00b - Divide by 1 01b - Divide by 2 10b - Divide by 4 11b - Divide by 8

49.7.4.14 Lock (LOCK)

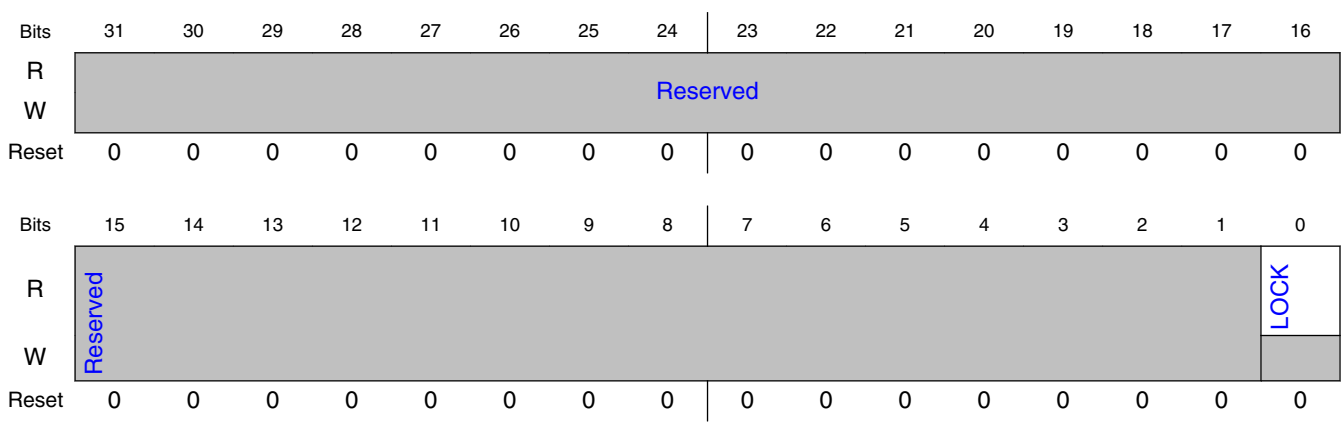
49.7.4.14.1 Offset

Register	Offset
LOCK	30h

49.7.4.14.2 Function

Asserted when the PLL has achieved frequency lock

49.7.4.14.3 Diagram



49.7.4.14.4 Fields

Field	Function
31-1 —	Reserved
0 LOCK	Lock Detect Output 0b - PLL not locked 1b - PLL has achieved frequency lock

49.7.4.15 Lock BYP (LOCK_BYP)

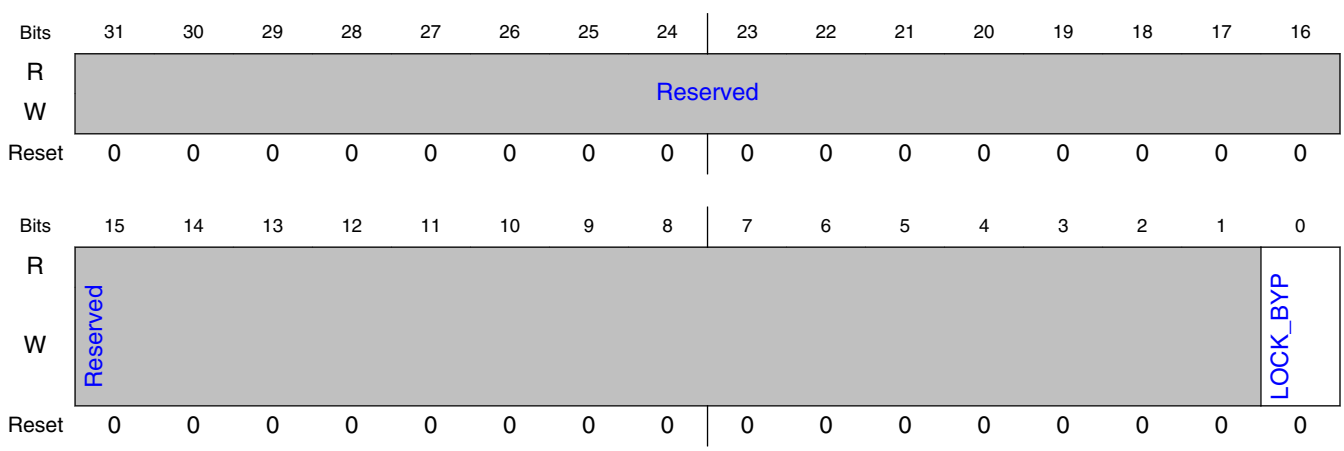
49.7.4.15.1 Offset

Register	Offset
LOCK_BYP	34h

49.7.4.15.2 Function

When clock lane exits from ULPS, this input determines if the PLL LOCK signal will be used to gate the TxByteClkHS

49.7.4.15.3 Diagram



49.7.4.15.4 Fields

Field	Function
31-1 —	Reserved
0 LOCK_BYP	DPHY LOCK_BYP input 0b - PLL LOCK signal will gate TxByteClkHS clock 1b - PLL LOCK signal will not gate TxByteClkHS clock, CIL based counter will be used to gate the TxByteClkHS

49.7.4.16 TX_RCAL (TX_RCAL)

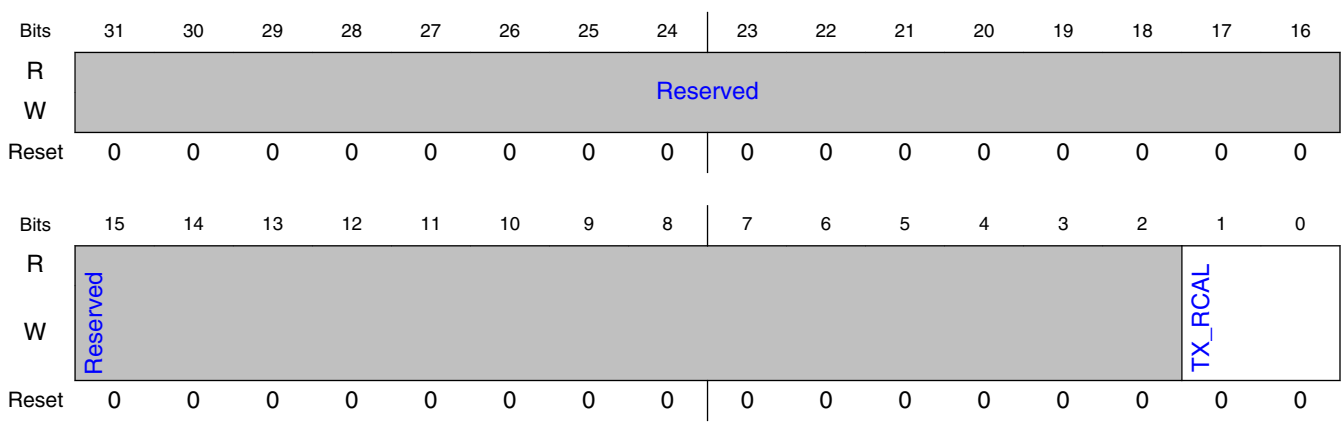
49.7.4.16.1 Offset

Register	Offset
TX_RCAL	38h

49.7.4.16.2 Function

On-chip termination control bits for manual calibration of HS-TX.

49.7.4.16.3 Diagram



49.7.4.16.4 Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Field	Function
—	
1-0 TX_RCAL	TX_RCAL 00b - 20% higher than mid-range. Highest impedance setting 01b - Mid-range impedance setting (default) 10b - 15% lower than mid-range 11b - 25% lower than mid-range. Lowest impedance setting

49.7.4.17 AUTO_PD_EN (AUTO_PD_EN)

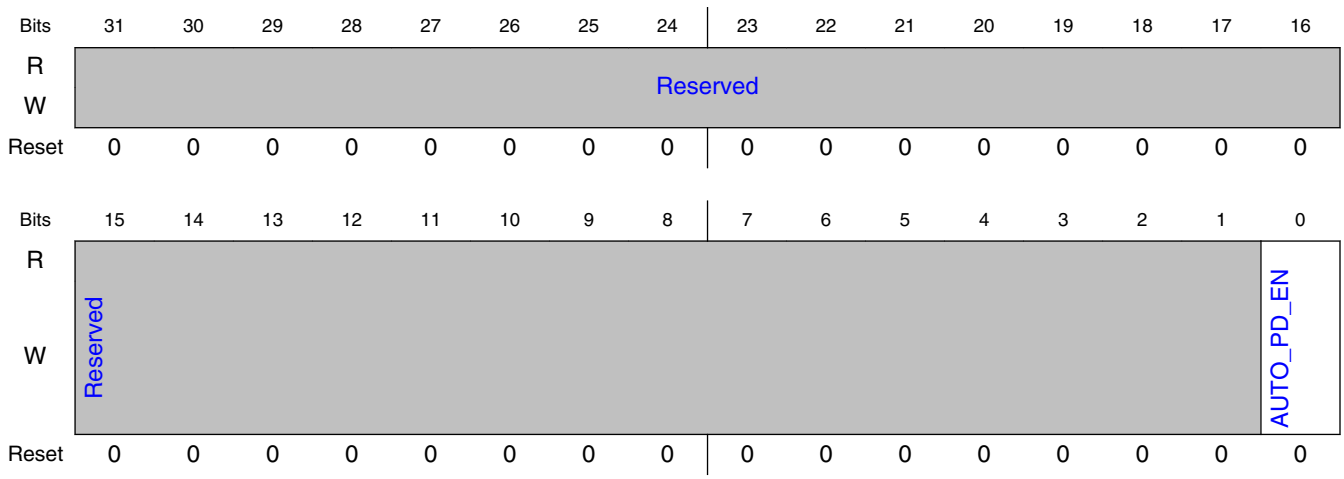
49.7.4.17.1 Offset

Register	Offset
AUTO_PD_EN	3Ch

49.7.4.17.2 Function

Powers down inactive lanes reported by CFG_NUM_LANES input bus.

49.7.4.17.3 Diagram



49.7.4.17.4 Fields

Field	Function
31-1	Reserved

Table continues on the next page...

Field	Function
—	
0 AUTO_PD_EN	DPHY AUTO_PD_EN input 0b - Inactive lanes are powered up and driving LP11 1b - inactive lanes are powered down

49.7.4.18 RXLPRP (RXLPRP)

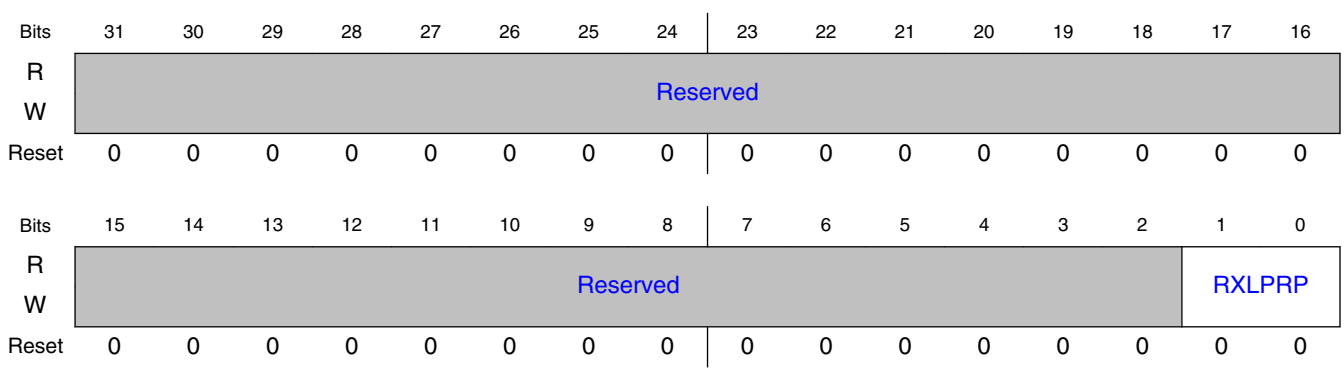
49.7.4.18.1 Offset

Register	Offset
RXLPRP	40h

49.7.4.18.2 Function

Inputs adjust the threshold voltage of LP-RX.

49.7.4.18.3 Diagram



49.7.4.18.4 Fields

Field	Function
31-2 —	Reserved
1-0 RXLPRP	DPHY RXLPRP input

Field	Function		
	Bitfield Value	**High Threshold Voltage**	**Low Threshold Voltage**
	2'b00	782mV	730mV
	2'b01 (Default)	745mV	692mV
	2'b10	708mV	655mV
	2'b11	Invalid	Invalid

49.7.4.19 RXCDRP (RXCDRP)

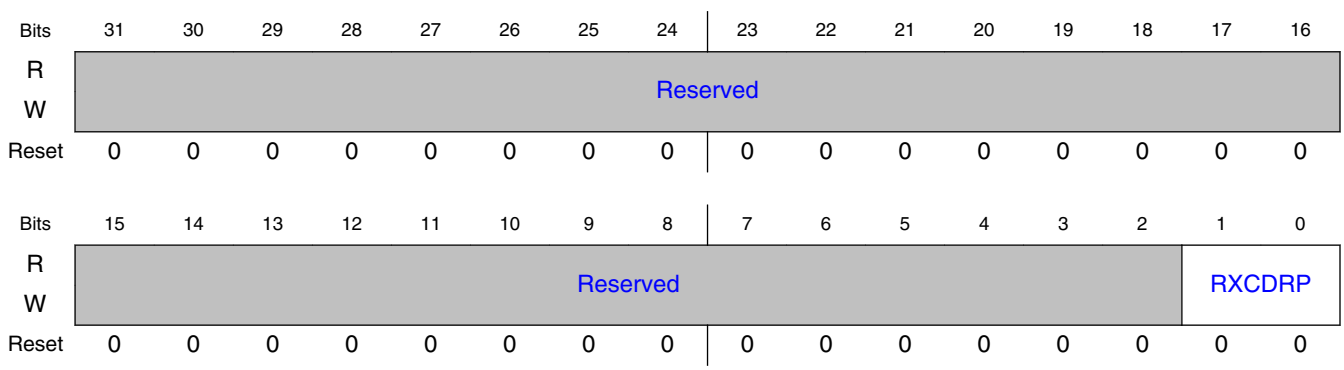
49.7.4.19.1 Offset

Register	Offset
RXCDRP	44h

49.7.4.19.2 Function

Inputs adjust the threshold voltage of LP-CD.

49.7.4.19.3 Diagram



49.7.4.19.4 Fields

Field	Function
31-2 —	Reserved

Table continues on the next page...

Field	Function
1-0 RXCDRP	DPHY RXCDRP input 00b - 344mV 01b - 325mV (Default) 10b - 307mV 11b - Reserved

Chapter 50
Video Mux Controller (VIDEO_MUX)

50.1 Chip-specific VIDEO MUX information

Table 50-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

50.2 Overview

The Video Multiplexer (VIDEO_MUX) contains mux controls between the following camera and display blocks:

- Parallel CSI
- MIPI CSI-2
- CSI
- MIPI DSI
- Parallel LCDIF
- LCDIFv2
- eLCDIF
- DCIC

50.2.1 Block diagram

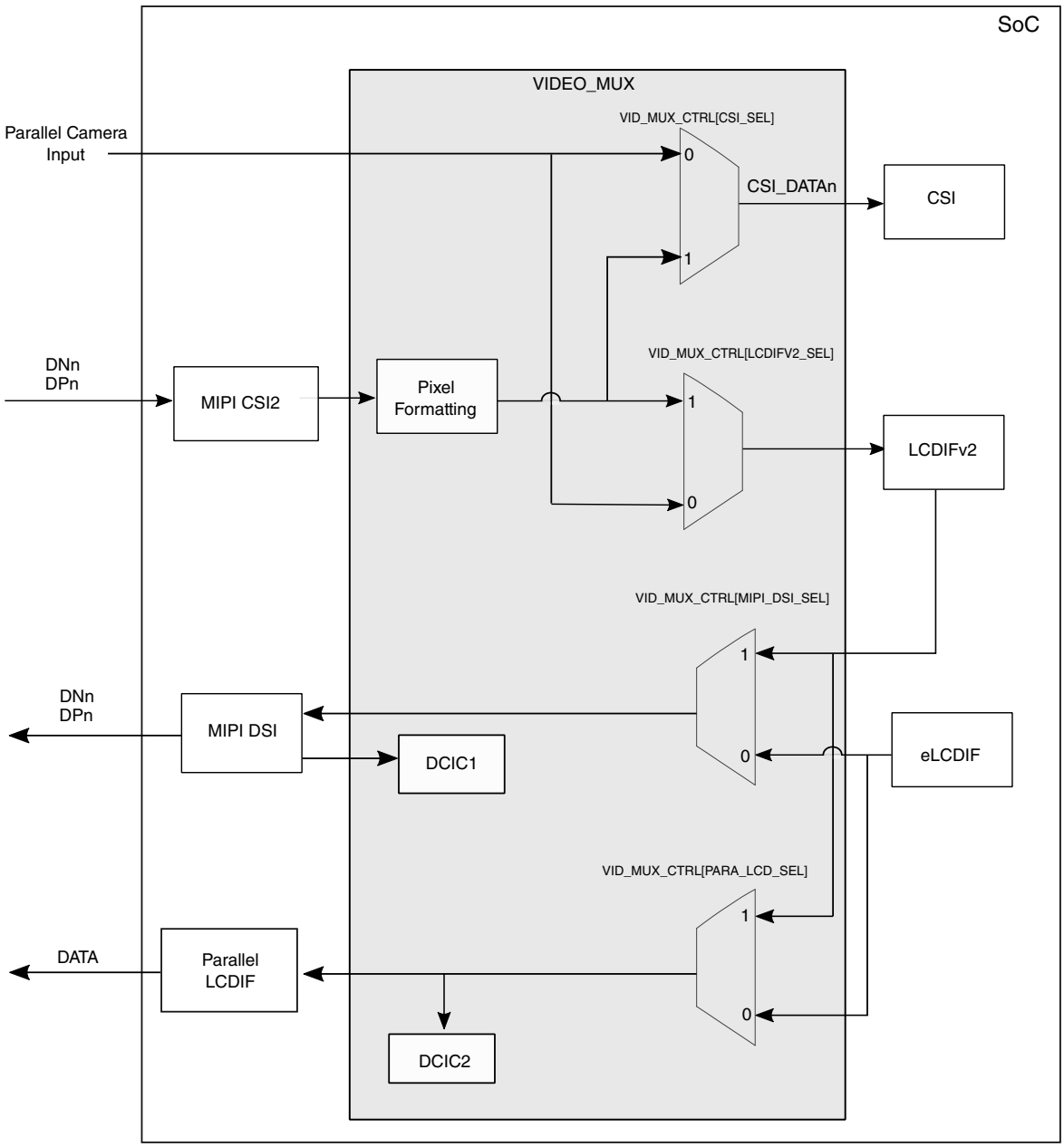


Figure 50-1. VIDEO_MUX block diagram

The pixel reformatting block converts the data format of MIPI CSI2 to meet the requirements of Pixel Link.

50.2.2 Features

This block includes the following features:

- Display Content Integrity Checker (DCIC) for Parallel LCDIF and MIPI DSI
- Support for two displays at one time

50.3 Functional description

The following sections describe functional details of this module.

50.3.1 Operation

The VIDEO_MUX block is a multiplexer which selects between various camera and display inputs and provide a desired output. The following selects are provided in this block:

- Input is selected between Parallel CSI and MIPI CSI2. The output can be either to CSI or LCDIFv2.
- Input is selected between eLCDIF and LCDIFv2. The output can be either to MIPI DSI or Parallel LCD display.

The DCIC function is provided to ensure the video data to the displays is not corrupted.

50.3.2 Clocks

The table below describes clocks of this module.

Table 50-2. Clocks

Clock	Description
ipg_clk_s	Peripheral access clock for register accesses

50.3.3 Reset

There are no special considerations.

50.4 External Signals

Table 50-3. External Signals Table

Module	External Signal	Description	I/O
CSI	CSI_DATAn	Data Sensor Signals, part of 16-bit Sensor Data Bus (Bayer, YUV,YCrCb, RGB)	I
	CSI_FIELD	CSI Field Signal	I
	CSI_HSYNC	Horizontal Sync (Blank Signal)	I
	CSI_MCLK	CMOS Sensor Master Clock Note: MCLK is provided by the CCM module directly, not from the CSI module itself	O
	CSI_PIXCLK	Pixel Clock Note: csi_hclk needs to be 10% faster (or more) than the Pixel Clock	I
	CSI_VSYNC	Vertical Sync (Start Of Frame)	I
DCIC	EXT_DCIC	-	-
eLCDIF	LCDIF_CLK	Clock Signal	O
	LCDIF_DATAn	Data Signals	IO
	LCDIF_ENABLE	Enable Signal	IO
	LCDIF_HSYNC	Horizontal Sync Signal	O
	LCDIF_VSYNC	Vertical Sync Signal	O
MIPI DSI	MIPI_DSI_DNn	Differential Data Signals (Negative)	O
	MIPI_DSI_DPn	Differential Data Signals (Positive)	O
	MIPI_DSI_CKN	Differential Clock Signal (Negative)	O
	MIPI_DSI_CKP	Differential Clock Signal (Positive)	O
MIPI CSI2	MIPI_CSI_DNn	Differential Data Signals (Negative)	I
	MIPI_CSI_DPn	Differential Data Signals (Positive)	I
	MIPI_CSI_CKN	Differential Clock Signal (Negative)	I
	MIPI_CSI_CKP	Differential Clock Signal (Positive)	I

50.5 Initialization

There are no special considerations.

50.6 Memory Map and register definition

This section includes the VIDEO_MUX module memory map and detailed descriptions of all registers.

50.6.1 VIDEO_MUX register descriptions

50.6.1.1 VIDEO_MUX memory map

VIDEO_MUX base address: 4081_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Video mux Control Register (VID_MUX_CTRL)	32	RW	0000_0000
4	Video mux Control Register (VID_MUX_CTRL_SET)	32	RW	0000_0000
8	Video mux Control Register (VID_MUX_CTRL_CLR)	32	RW	0000_0000
C	Video mux Control Register (VID_MUX_CTRL_TOG)	32	RW	0000_0000
20	Pixel Link Master(PLM) Control Register (PLM_CTRL)	32	RW	0000_0000
24	Pixel Link Master(PLM) Control Register (PLM_CTRL_SET)	32	RW	0000_0000
28	Pixel Link Master(PLM) Control Register (PLM_CTRL_CLR)	32	RW	0000_0000
2C	Pixel Link Master(PLM) Control Register (PLM_CTRL_TOG)	32	RW	0000_0000
30	YUV420 Control Register (YUV420_CTRL)	32	RW	0000_0000
34	YUV420 Control Register (YUV420_CTRL_SET)	32	RW	0000_0000
38	YUV420 Control Register (YUV420_CTRL_CLR)	32	RW	0000_0000
3C	YUV420 Control Register (YUV420_CTRL_TOG)	32	RW	0000_0000
50	Data Disable Register (CFG_DT_DISABLE)	32	RW	0000_0000
54	Data Disable Register (CFG_DT_DISABLE_SET)	32	RW	0000_0000
58	Data Disable Register (CFG_DT_DISABLE_CLR)	32	RW	0000_0000
5C	Data Disable Register (CFG_DT_DISABLE_TOG)	32	RW	0000_0000
70	MIPI DSI Control Register (MIPI_DSI_CTRL)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
74	MIPI DSI Control Register (MIPI_DSI_CTRL_SET)	32	RW	0000_0000
78	MIPI DSI Control Register (MIPI_DSI_CTRL_CLR)	32	RW	0000_0000
7C	MIPI DSI Control Register (MIPI_DSI_CTRL_TOG)	32	RW	0000_0000

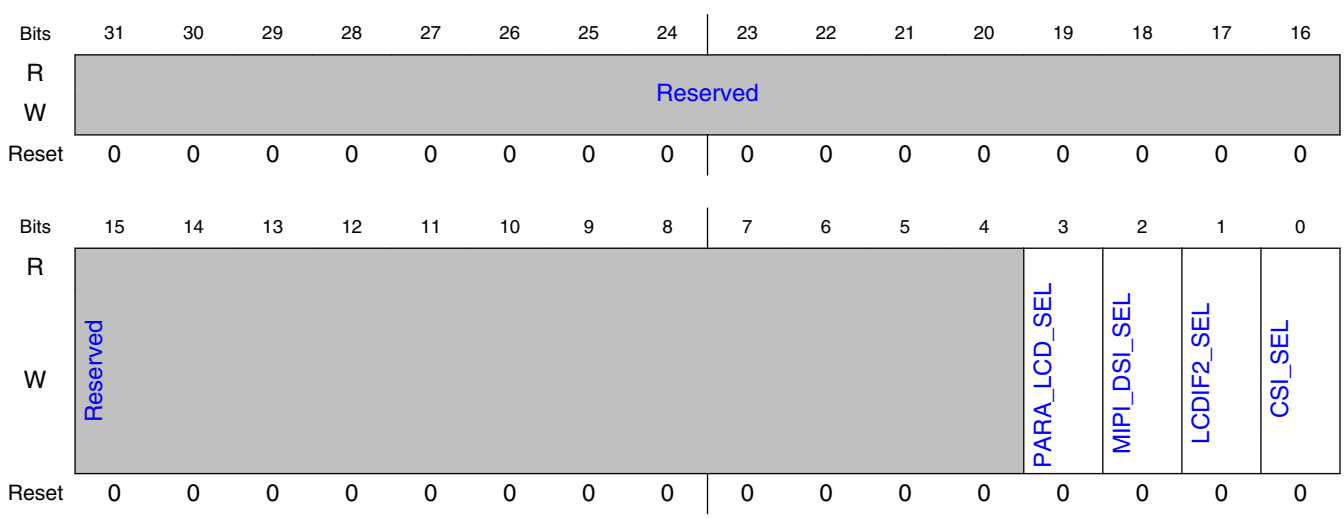
50.6.1.2 Video mux Control Register (VID_MUX_CTRL)

This register contains mux controls

50.6.1.2.1 Offset

Register	Offset	Description
VID_MUX_CTRL	0h	Video mux Control Register
VID_MUX_CTRL_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in VID_MUX_CTRL
VID_MUX_CTRL_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in VID_MUX_CTRL
VID_MUX_CTRL_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in VID_MUX_CTRL

50.6.1.2.2 Diagram



50.6.1.2.3 Fields

Field	Description
31-4 —	Reserved
3 PARA_LCD_SEL	Parallel LCDIF video data input mux selector 0 - Parallel LCDIF video data is from eLCDIF 1 - Parallel LCDIF video data is from LCDIFv2
2 MIPI_DSI_SEL	MIPI DSI video data input mux selector 0 - MIPI DSI video data is from eLCDIF 1 - MIPI DSI video data is from LCDIFv2
1 LCDIF2_SEL	LCDIF2 sensor data input mux selector 0 - LCDIFv2 sensor data is from Parallel CSI 1 - LCDIFv2 sensor data is from MIPI CSI
0 CSI_SEL	CSI sensor data input mux selector 0 - CSI sensor data is from Parallel CSI 1 - CSI sensor data is from MIPI CSI

50.6.1.3 Pixel Link Master(PLM) Control Register (PLM_CTRL)

These inputs will override the transmitted data on Pixel Link

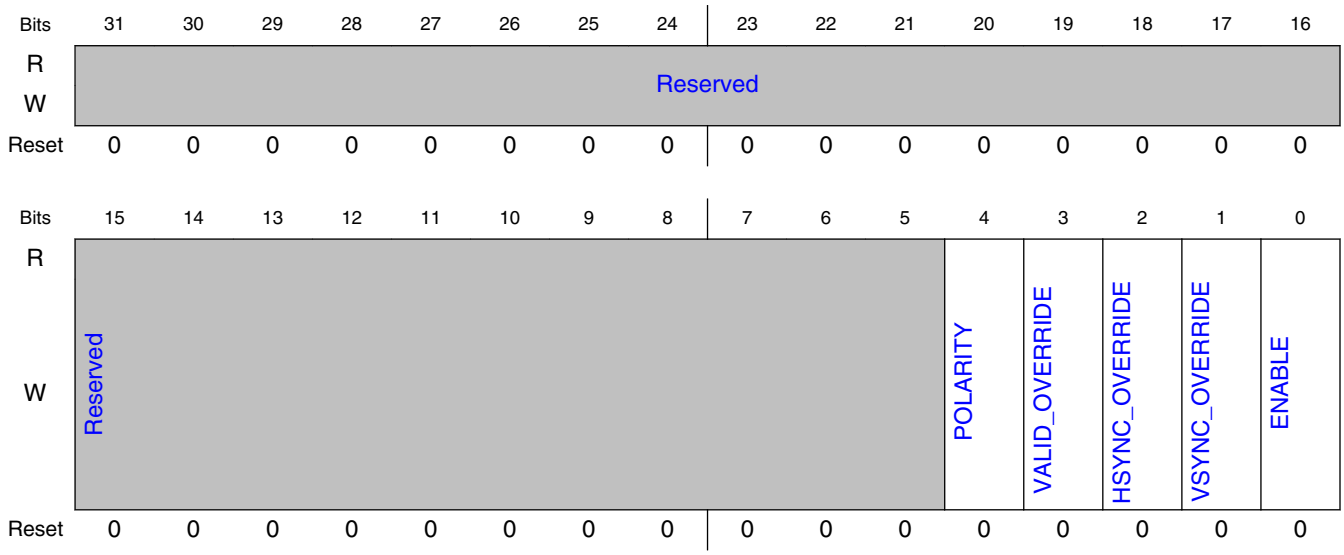
NOTE

If PLM_CTRL[VSYNC_OVERRIDE]= 0 and PLM_CTRL[HSYNC_OVERRIDE] = 0, PLM_CTRL[VALID_OVERRIDE] must be 1.

50.6.1.3.1 Offset

Register	Offset	Description
PLM_CTRL	20h	Pixel Link Master(PLM) Control Register
PLM_CTRL_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in PLM_CTRL
PLM_CTRL_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in PLM_CTRL
PLM_CTRL_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in PLM_CTRL

50.6.1.3.2 Diagram



50.6.1.3.3 Fields

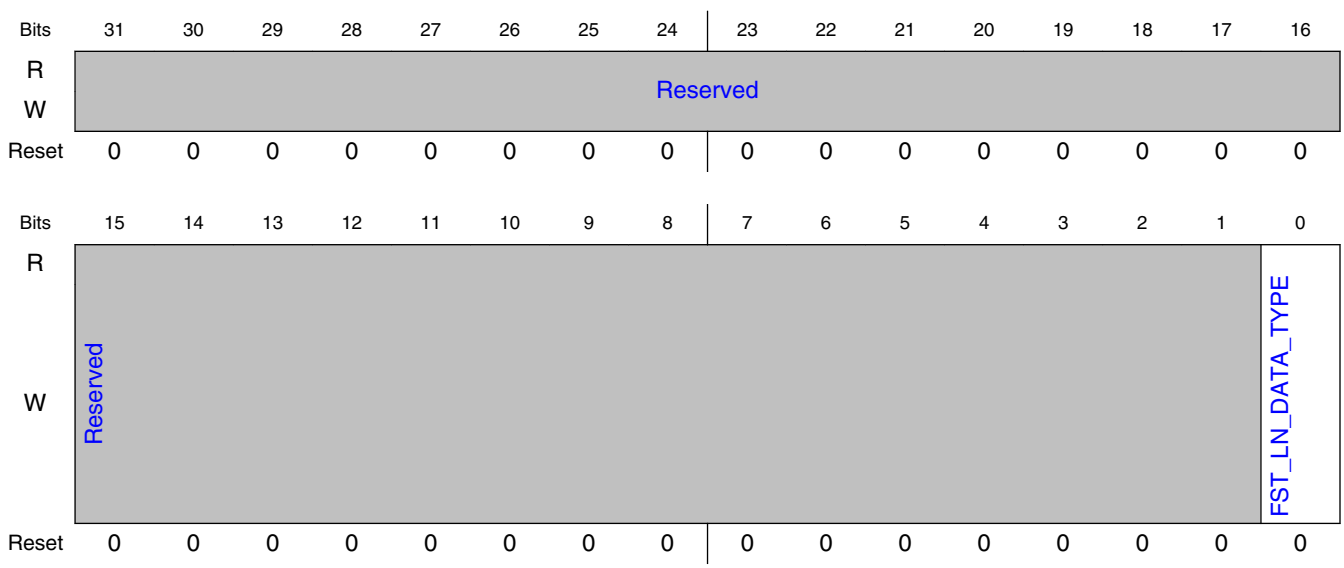
Field	Description
31-5 —	Reserved
4 POLARITY	Polarity of HYSNC/VSYNC 0 - Keep the current polarity of HSYNC and VSYNC 1 - Invert the polarity of HSYNC and VSYNC
3 VALID_OVERRIDE	Valid override This override is used to drive valid on the Pixel Link 0 - HSYNC and VSYNC is asserted 1 - HSYNC and VSYNC is not asserted
2 HSYNC_OVERRIDE	HSYNC override This override is used to force the PLM HSYNC input to be 1 0 - HSYNC is not asserted 1 - HSYNC is asserted
1 VSYNC_OVERRIDE	VSYNC override This override is used to force the PLM VSYNC input to be 1 0 - VSYNC is not asserted 1 - VSYNC is asserted
0 ENABLE	Enable the output of HYSNC and VSYNC 0 - No active HSYNC and VSYNC output 1 - Active HSYNC and VSYNC output

50.6.1.4 YUV420 Control Register (YUV420_CTRL)

50.6.1.4.1 Offset

Register	Offset	Description
YUV420_CTRL	30h	YUV420 Control Register
YUV420_CTRL_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in YUV420_CTRL
YUV420_CTRL_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in YUV420_CTRL
YUV420_CTRL_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in YUV420_CTRL

50.6.1.4.2 Diagram



50.6.1.4.3 Fields

Field	Description
31-1 —	Reserved
0	Data type of First Line

Field	Description
FST_LN_DATA_TYPE	0 - Odd (default) 1 - Even

50.6.1.5 Data Disable Register (CFG_DT_DISABLE)

50.6.1.5.1 Offset

Register	Offset	Description
CFG_DT_DISABLE	50h	Data Disable Register
CFG_DT_DISABLE_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in CFG_DT_DISABLE
CFG_DT_DISABLE_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in CFG_DT_DISABLE
CFG_DT_DISABLE_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CFG_DT_DISABLE

50.6.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								CFG_DT_DISABLE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_DT_DISABLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

50.6.1.5.3 Fields

Field	Description
31-24 —	Reserved
23-0	Data Type Disable Disable the data type. Each bit represents one data type.

Field	Description
CFG_DT_DISABLE	1'b0 - Enable data type 1'b1 - Disable data type **YUV Image Data Types:** [0]: YUV420_8BIT [1]: YUV420_10BIT [2]: LEGACY_YUV420_8BIT [3]: Reserved [4]: YUV420_8BIT_CS [5]: YUV420_10BIT_CS [6]: YUV422_8BIT [7] YUV422_10BIT **RGB Image Data Types:** [8]: RGB444 [9]: RGB555 [10] RGB565 [11]: RGB666 [12]: RGB888 [13-15]: Reserved **Raw Image Data Types** [16]: RAW6 [17]: RAW7 [18]: RAW8 [19]: RAW10 [20]: RAW12 [21]: RAW14 [22-23]: Reserved

50.6.1.6 MIPI DSI Control Register (MIPI_DSI_CTRL)

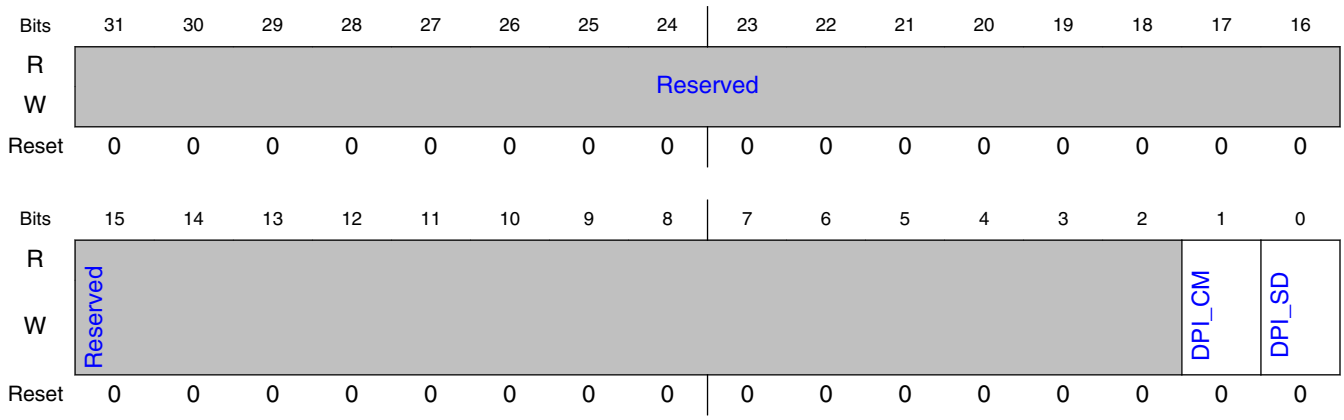
50.6.1.6.1 Offset

Register	Offset	Description
MIPI_DSI_CTRL	70h	MIPI DSI Control Register
MIPI_DSI_CTRL_SET	74h	Writing a 1 to a bit in this register sets the corresponding bit in MIPI_DSI_CTRL

Table continues on the next page...

Register	Offset	Description
MIPI_DSI_CTRL_CLR	78h	Writing a 1 to a bit in this register clears the corresponding bit in MIPI_DSI_CTRL
MIPI_DSI_CTRL_TOG	7Ch	Writing a 1 to a bit in this register toggles the corresponding bit in MIPI_DSI_CTRL

50.6.1.6.2 Diagram



50.6.1.6.3 Fields

Field	Description
31-2 —	Reserved
1 DPI_CM	Color Mode control This field is used to change the number of displayed colors 0 - Normal Mode 1 - 8-color mode (compared to Full-color mode)
0 DPI_SD	Shut Down – Control to shutdown display (type 4 only) 0 - No effect 1 - Send shutdown command

Chapter 51
Display Content Integrity Checker (DCIC)

51.1 Chip-specific DCIC information

Table 51-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

51.2 Overview

The Display Content Integrity Checker (DCIC) verifies whether safety-critical information sent to a display has been corrupted. This type of verification is required in some safety-sensitive systems, like the warning icons in an vehicle cluster, and includes compliance with the ASIL B (Automotive Safety Integrity Level B) specification.

51.2.1 Block Diagram

The top level block diagram of the DCIC module is provided below.

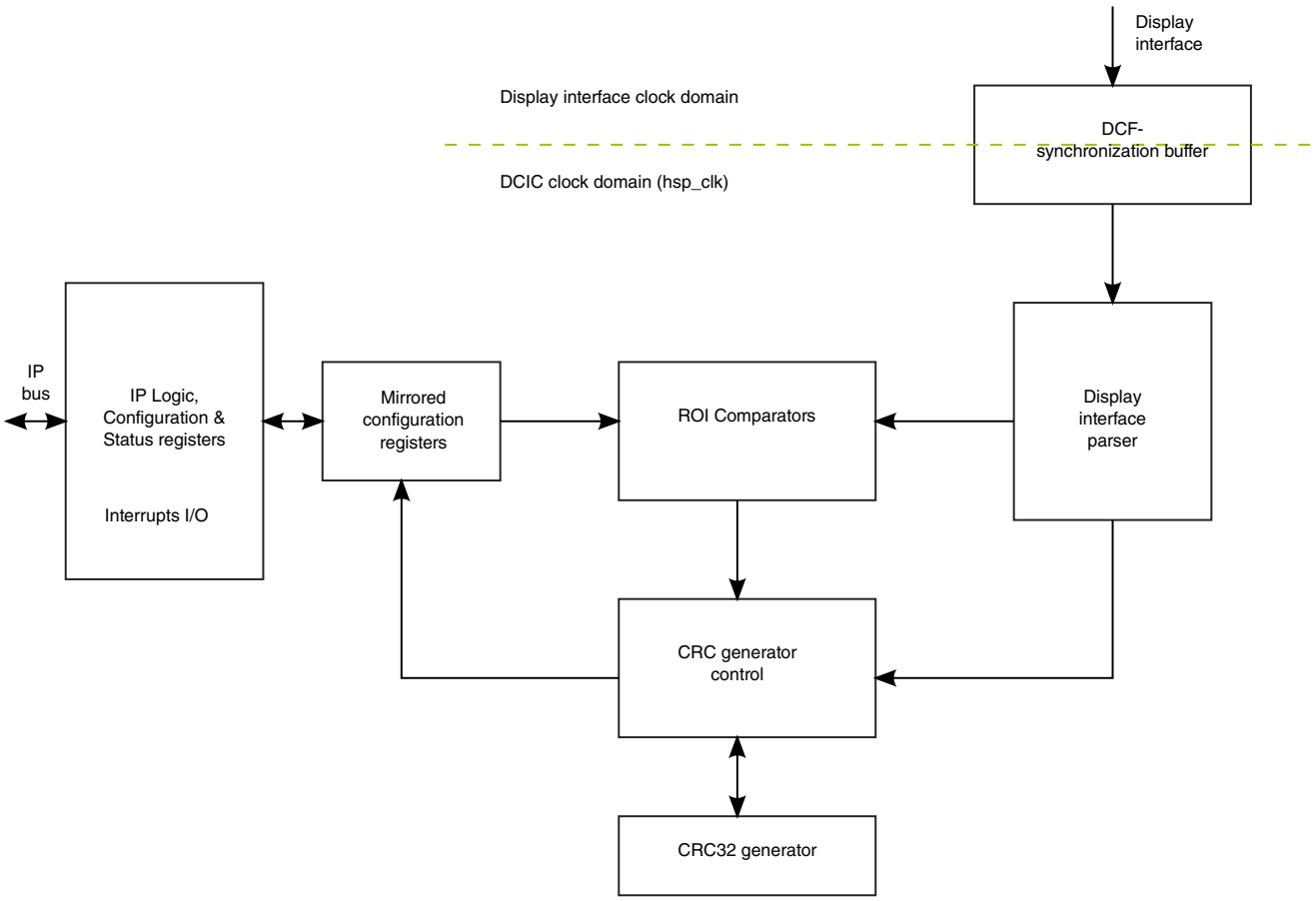


Figure 51-1. DCIC block diagram

51.2.2 Features

The DCIC module includes the following features:

- Pixel clock up to 75 MHz.
- Configurable polarity of the Display Interface control signals.
- 24-bit pixel data bus.
- Support for up to 16 rectangular ROIs (Regions of Interest) with configurable location and size.
- Independent CRC32 signature calculation for each ROI.
- External controller mismatch indication signal.

51.3 Functional Description

The following sections describe the functional details of this module.

51.3.1 Generic synchronous parallel display interface

The following figure depicts the LCD interface timing for a generic active matrix color TFT panel. In the figure, the signals are shown with negative polarity, but both polarity modes are supported by the DCIC module through the configuration of the appropriate bits in the DCIC Control Register.

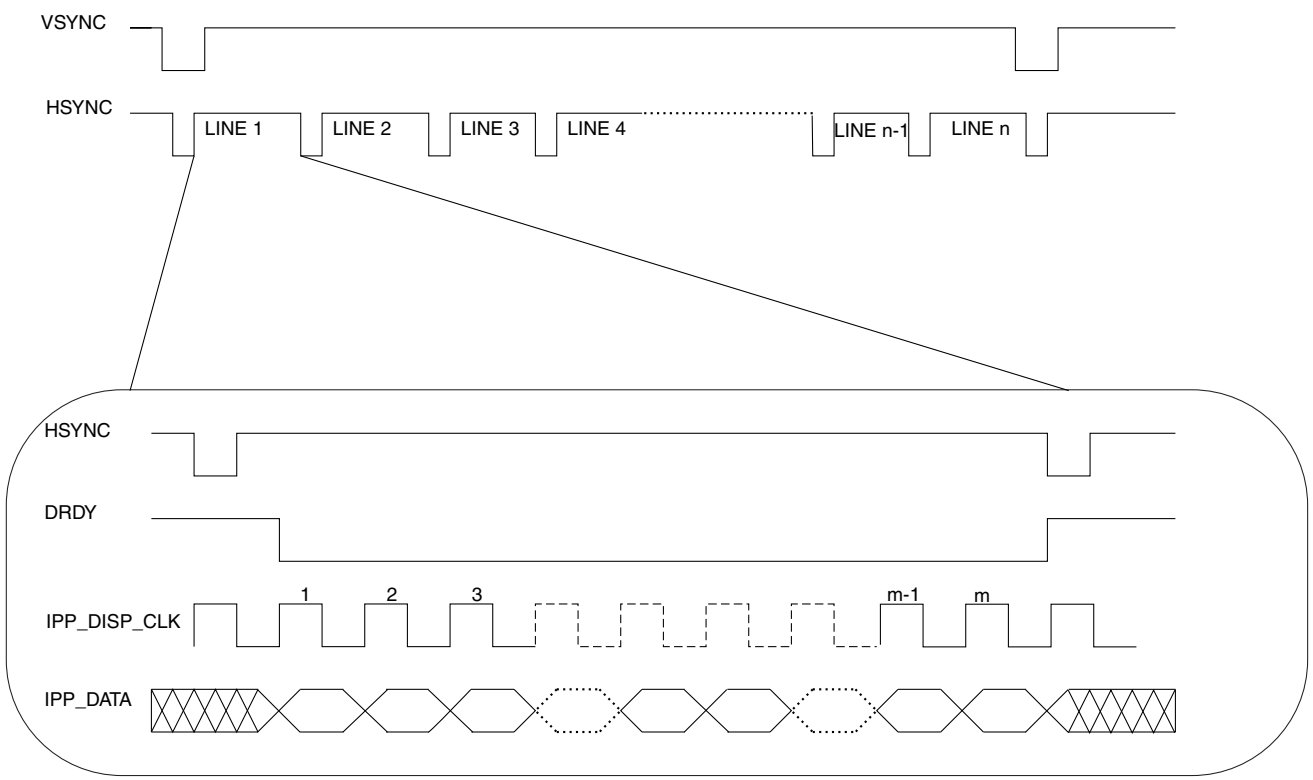


Figure 51-2. Interface Timing Diagram for TFT (Active Matrix) Panels

The sequence of events for the active matrix interface timing is as follows:

- IPP_DISP_CLK latches data into the panel on it's negative edge, when positive polarity is selected. In active mode, IPP_DISP_CLK runs continuously.
- HSYNC causes the panel to start a new line.
- VSYNC causes the panel to start a new frame. It always encompasses at least one HSYNC pulse.
- DRDY acts like an output enable signal. This output enables the data to be shifted onto the display. When disabled, the data is invalid and the trace is off.

51.3.2 CRC Polynomial

The DCIC uses the CRC32 polynomial to calculate the data signature. The initial value for the CRC calculation is 0x00000000.

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Figure 51-3. Polynomial function for CRC32

The CRC generator uses the XOR-ed combination of the input data bits and the previous result bits.

51.3.3 External controller mismatch indication signal

Apart from the interrupts to the SoC core, the DCIC provides an additional mismatch indication for the external controller. The EXT_DCIC signal, if enabled (DCIC_DCICIC[EXT_SIG_EN]), continuously oscillates at a rate which is an integer division of the main clock (hsp_clk).

- When the status bit of the mismatch interrupt is set, i.e., from the mismatch detection until the CPU clears the bit: division x16
- Otherwise: division x4

The indication signal is idle while the integrity check is disabled.

51.3.4 Mode of operation

After the DCIC is configured and enabled, provided that the eLCDIF/LCDIFv2 is already enabled, the first VSYNC is being watched. The VSYNC assertion zeroizes the main position counters and initializes the CRC storage registers.

Each HSYNC assertion after the first line in the frame that holds valid pixel data (non blanking) will signal a new line, i.e., increment Y-counter and zeroize X-counter. Each valid data cycle (DATA_EN asserted) will increment the X-counter. Thus, the X & Y values used to configure the module should be the coordinates on the actual visual part of the screen without blanking intervals. The X & Y position counters are constantly compared to the ROIs configuration. When the current position matches one of the enabled ROIs, it triggers the enablement of the CRC generator, which will use the current pixel data and the previous pixel CRC result for this ROI. The result of the new calculation is stored separately for each ROI. The next VSYNC signal, apart from

signalling the start of a new frame, will load the calculated CRC value into the DCIC_DCICRCSn register. These values will then be compared to the expected signatures and the appropriate status bits/interrupts will be asserted.

51.3.5 Interrupts

There are two maskable interrupts:

- Functional - Asserted when the signature match results are ready.
- Error - Asserted when there is a signature mismatch.

Both interrupts are generated immediately after completing the signature match check. Software should clear the interrupts by writing "1" to the DCIC_DCICS[FI_STAT] and DCIC_DCICS[ROI_MATCH_STAT] bits. Since the DCIC_DCICS[EI_STAT] bit is a result of the OR between all the DCIC_DCICS[ROI_MATCH_STAT] bits, it will be cleared automatically when these bits are cleared.

Interrupt masks can be set/reset only while the DCIC_DCICIC[FREEZE_MASK] bit is not set. This bit cannot be set back to zero.

Please see the IRQ chapter for the interrupt assignments.

51.3.6 Resets

Disabling and enabling the module operation (DCIC_DCICC[IC_EN]) will reset all the logic except for the configuration registers domain. The reset will be active immediately. The status bits of the DCIC Status Register will also be cleared.

51.3.7 Clocks

Table 51-2. Clocks

Clock	Description
hsp_clk	Main module clock driving the DCIC logic and IP registers.

An Asynchronous FIFO is used to sample raw display interface signals and transfer them into the DCIC fast clock domain (hsp_clk).

The IP registers are driven by the same fast clock (hsp_clk) and are synchronized to the system IP Bus clock by the external IP_SYNC module.

The display interface clock is always slower than the DCIC fast clock.

51.3.8 Power saving

Disabling the module by clearing the DCIC_DCICC[IC_EN] bit, will stop all module activities, including the sampling of the input signals into the FIFO.

51.4 External Signals

Table 51-3. External Signals

Signal	Description	Direction
EXT_DCIC	Variable frequency, oscillating signal indicating the comparison status for the external controller.	O

51.5 System Considerations

- The DCIC always assumes a 24-bit pixel. For proper functionality with lower color depth, one must ensure that:
  - When the CPU calculates the reference signature, it applies the same mapping of a pixel to a 24-bit field as that performed by the eLCDIF/LCDIFv2.
  - The display is connected to the appropriate pins.
  - The simplest mapping would be:
    - Map the pixel to the data bus in the same way as for 24 bpp.
    - Set the values at the extra LSBs to zero.
- Parameter updates
  - The reference signature can be updated from frame to frame, in coordination with the changing content since it is used only once per frame, just before the interrupt.
  - Each ROI can be independently enabled/disabled between frames since the enable bit is double buffered.
  - The size and location of a ROI can be modified while it is disabled.
  - Display interface signals polarity can be changed only when the module is disabled.
- ROIs do not overlap, i.e., each pixel belongs to a single ROI at most.

- The DCIC operation is intended for monitoring relatively static portions of the graphic interface, otherwise it will be a complicated software task to keep up with frame synchronization and update of expected signature for each frame.
- There should be no processing done at the eLCDIF/LCDIFv2 on monitored ROIs, which cannot be taken in consideration by software when calculating the expected signature.

51.6 DCIC Memory Map/Register Definition

NOTE

All write accesses have to be full word (32-bit) accesses. No error/abort will be responded in case of different access size, but no data will be written. Similarly, there will be no error response in case of access to undefined memory space.

Read access will return 32-bit data, which may be truncated on the system level in case it was not a full word access.

DCIC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_9000	DCIC Control Register (DCIC1_DCICC)	32	R/W	0000_0070h	51.6.1/3250
4081_9004	DCIC Interrupt Control Register (DCIC1_DCICIC)	32	R/W	0000_0003h	51.6.2/3251
4081_9008	DCIC Status Register (DCIC1_DCICS)	32	w1c	0000_0000h	51.6.3/3253
4081_9010	DCIC ROI Config Register (DCIC1_DCICRC1)	32	R/W	0000_0000h	51.6.4/3254
4081_9014	DCIC ROI Size Register (DCIC1_DCICRS1)	32	R/W	0000_0000h	51.6.5/3255
4081_9018	DCIC ROI Reference Signature Register (DCIC1_DCICRRS1)	32	R/W	0000_0000h	51.6.6/3255
4081_901C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS1)	32	R	0000_0000h	51.6.7/3256
4081_9020	DCIC ROI Config Register (DCIC1_DCICRC2)	32	R/W	0000_0000h	51.6.4/3254
4081_9024	DCIC ROI Size Register (DCIC1_DCICRS2)	32	R/W	0000_0000h	51.6.5/3255
4081_9028	DCIC ROI Reference Signature Register (DCIC1_DCICRRS2)	32	R/W	0000_0000h	51.6.6/3255
4081_902C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS2)	32	R	0000_0000h	51.6.7/3256
4081_9030	DCIC ROI Config Register (DCIC1_DCICRC3)	32	R/W	0000_0000h	51.6.4/3254
4081_9034	DCIC ROI Size Register (DCIC1_DCICRS3)	32	R/W	0000_0000h	51.6.5/3255
4081_9038	DCIC ROI Reference Signature Register (DCIC1_DCICRRS3)	32	R/W	0000_0000h	51.6.6/3255

Table continues on the next page...

DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_903C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS3)	32	R	0000_0000h	51.6.7/3256
4081_9040	DCIC ROI Config Register (DCIC1_DCICRC4)	32	R/W	0000_0000h	51.6.4/3254
4081_9044	DCIC ROI Size Register (DCIC1_DCICRS4)	32	R/W	0000_0000h	51.6.5/3255
4081_9048	DCIC ROI Reference Signature Register (DCIC1_DCICRRS4)	32	R/W	0000_0000h	51.6.6/3255
4081_904C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS4)	32	R	0000_0000h	51.6.7/3256
4081_9050	DCIC ROI Config Register (DCIC1_DCICRC5)	32	R/W	0000_0000h	51.6.4/3254
4081_9054	DCIC ROI Size Register (DCIC1_DCICRS5)	32	R/W	0000_0000h	51.6.5/3255
4081_9058	DCIC ROI Reference Signature Register (DCIC1_DCICRRS5)	32	R/W	0000_0000h	51.6.6/3255
4081_905C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS5)	32	R	0000_0000h	51.6.7/3256
4081_9060	DCIC ROI Config Register (DCIC1_DCICRC6)	32	R/W	0000_0000h	51.6.4/3254
4081_9064	DCIC ROI Size Register (DCIC1_DCICRS6)	32	R/W	0000_0000h	51.6.5/3255
4081_9068	DCIC ROI Reference Signature Register (DCIC1_DCICRRS6)	32	R/W	0000_0000h	51.6.6/3255
4081_906C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS6)	32	R	0000_0000h	51.6.7/3256
4081_9070	DCIC ROI Config Register (DCIC1_DCICRC7)	32	R/W	0000_0000h	51.6.4/3254
4081_9074	DCIC ROI Size Register (DCIC1_DCICRS7)	32	R/W	0000_0000h	51.6.5/3255
4081_9078	DCIC ROI Reference Signature Register (DCIC1_DCICRRS7)	32	R/W	0000_0000h	51.6.6/3255
4081_907C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS7)	32	R	0000_0000h	51.6.7/3256
4081_9080	DCIC ROI Config Register (DCIC1_DCICRC8)	32	R/W	0000_0000h	51.6.4/3254
4081_9084	DCIC ROI Size Register (DCIC1_DCICRS8)	32	R/W	0000_0000h	51.6.5/3255
4081_9088	DCIC ROI Reference Signature Register (DCIC1_DCICRRS8)	32	R/W	0000_0000h	51.6.6/3255
4081_908C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS8)	32	R	0000_0000h	51.6.7/3256
4081_9090	DCIC ROI Config Register (DCIC1_DCICRC9)	32	R/W	0000_0000h	51.6.4/3254
4081_9094	DCIC ROI Size Register (DCIC1_DCICRS9)	32	R/W	0000_0000h	51.6.5/3255
4081_9098	DCIC ROI Reference Signature Register (DCIC1_DCICRRS9)	32	R/W	0000_0000h	51.6.6/3255
4081_909C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS9)	32	R	0000_0000h	51.6.7/3256
4081_90A0	DCIC ROI Config Register (DCIC1_DCICRC10)	32	R/W	0000_0000h	51.6.4/3254
4081_90A4	DCIC ROI Size Register (DCIC1_DCICRS10)	32	R/W	0000_0000h	51.6.5/3255
4081_90A8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS10)	32	R/W	0000_0000h	51.6.6/3255

Table continues on the next page...

DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_90AC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS10)	32	R	0000_0000h	51.6.7/3256
4081_90B0	DCIC ROI Config Register (DCIC1_DCICRC11)	32	R/W	0000_0000h	51.6.4/3254
4081_90B4	DCIC ROI Size Register (DCIC1_DCICRS11)	32	R/W	0000_0000h	51.6.5/3255
4081_90B8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS11)	32	R/W	0000_0000h	51.6.6/3255
4081_90BC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS11)	32	R	0000_0000h	51.6.7/3256
4081_90C0	DCIC ROI Config Register (DCIC1_DCICRC12)	32	R/W	0000_0000h	51.6.4/3254
4081_90C4	DCIC ROI Size Register (DCIC1_DCICRS12)	32	R/W	0000_0000h	51.6.5/3255
4081_90C8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS12)	32	R/W	0000_0000h	51.6.6/3255
4081_90CC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS12)	32	R	0000_0000h	51.6.7/3256
4081_90D0	DCIC ROI Config Register (DCIC1_DCICRC13)	32	R/W	0000_0000h	51.6.4/3254
4081_90D4	DCIC ROI Size Register (DCIC1_DCICRS13)	32	R/W	0000_0000h	51.6.5/3255
4081_90D8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS13)	32	R/W	0000_0000h	51.6.6/3255
4081_90DC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS13)	32	R	0000_0000h	51.6.7/3256
4081_90E0	DCIC ROI Config Register (DCIC1_DCICRC14)	32	R/W	0000_0000h	51.6.4/3254
4081_90E4	DCIC ROI Size Register (DCIC1_DCICRS14)	32	R/W	0000_0000h	51.6.5/3255
4081_90E8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS14)	32	R/W	0000_0000h	51.6.6/3255
4081_90EC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS14)	32	R	0000_0000h	51.6.7/3256
4081_90F0	DCIC ROI Config Register (DCIC1_DCICRC15)	32	R/W	0000_0000h	51.6.4/3254
4081_90F4	DCIC ROI Size Register (DCIC1_DCICRS15)	32	R/W	0000_0000h	51.6.5/3255
4081_90F8	DCIC ROI Reference Signature Register (DCIC1_DCICRRS15)	32	R/W	0000_0000h	51.6.6/3255
4081_90FC	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS15)	32	R	0000_0000h	51.6.7/3256
4081_9100	DCIC ROI Config Register (DCIC1_DCICRC16)	32	R/W	0000_0000h	51.6.4/3254
4081_9104	DCIC ROI Size Register (DCIC1_DCICRS16)	32	R/W	0000_0000h	51.6.5/3255
4081_9108	DCIC ROI Reference Signature Register (DCIC1_DCICRRS16)	32	R/W	0000_0000h	51.6.6/3255
4081_910C	DCIC ROI Calculated Signature Register (DCIC1_DCICRCS16)	32	R	0000_0000h	51.6.7/3256
4081_A000	DCIC Control Register (DCIC2_DCICC)	32	R/W	0000_0070h	51.6.1/3250
4081_A004	DCIC Interrupt Control Register (DCIC2_DCICIC)	32	R/W	0000_0003h	51.6.2/3251
4081_A008	DCIC Status Register (DCIC2_DCICS)	32	w1c	0000_0000h	51.6.3/3253
4081_A010	DCIC ROI Config Register (DCIC2_DCICRC1)	32	R/W	0000_0000h	51.6.4/3254

Table continues on the next page...

DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_A014	DCIC ROI Size Register (DCIC2_DCICRS1)	32	R/W	0000_0000h	51.6.5/3255
4081_A018	DCIC ROI Reference Signature Register (DCIC2_DCICRRS1)	32	R/W	0000_0000h	51.6.6/3255
4081_A01C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS1)	32	R	0000_0000h	51.6.7/3256
4081_A020	DCIC ROI Config Register (DCIC2_DCICRC2)	32	R/W	0000_0000h	51.6.4/3254
4081_A024	DCIC ROI Size Register (DCIC2_DCICRS2)	32	R/W	0000_0000h	51.6.5/3255
4081_A028	DCIC ROI Reference Signature Register (DCIC2_DCICRRS2)	32	R/W	0000_0000h	51.6.6/3255
4081_A02C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS2)	32	R	0000_0000h	51.6.7/3256
4081_A030	DCIC ROI Config Register (DCIC2_DCICRC3)	32	R/W	0000_0000h	51.6.4/3254
4081_A034	DCIC ROI Size Register (DCIC2_DCICRS3)	32	R/W	0000_0000h	51.6.5/3255
4081_A038	DCIC ROI Reference Signature Register (DCIC2_DCICRRS3)	32	R/W	0000_0000h	51.6.6/3255
4081_A03C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS3)	32	R	0000_0000h	51.6.7/3256
4081_A040	DCIC ROI Config Register (DCIC2_DCICRC4)	32	R/W	0000_0000h	51.6.4/3254
4081_A044	DCIC ROI Size Register (DCIC2_DCICRS4)	32	R/W	0000_0000h	51.6.5/3255
4081_A048	DCIC ROI Reference Signature Register (DCIC2_DCICRRS4)	32	R/W	0000_0000h	51.6.6/3255
4081_A04C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS4)	32	R	0000_0000h	51.6.7/3256
4081_A050	DCIC ROI Config Register (DCIC2_DCICRC5)	32	R/W	0000_0000h	51.6.4/3254
4081_A054	DCIC ROI Size Register (DCIC2_DCICRS5)	32	R/W	0000_0000h	51.6.5/3255
4081_A058	DCIC ROI Reference Signature Register (DCIC2_DCICRRS5)	32	R/W	0000_0000h	51.6.6/3255
4081_A05C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS5)	32	R	0000_0000h	51.6.7/3256
4081_A060	DCIC ROI Config Register (DCIC2_DCICRC6)	32	R/W	0000_0000h	51.6.4/3254
4081_A064	DCIC ROI Size Register (DCIC2_DCICRS6)	32	R/W	0000_0000h	51.6.5/3255
4081_A068	DCIC ROI Reference Signature Register (DCIC2_DCICRRS6)	32	R/W	0000_0000h	51.6.6/3255
4081_A06C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS6)	32	R	0000_0000h	51.6.7/3256
4081_A070	DCIC ROI Config Register (DCIC2_DCICRC7)	32	R/W	0000_0000h	51.6.4/3254
4081_A074	DCIC ROI Size Register (DCIC2_DCICRS7)	32	R/W	0000_0000h	51.6.5/3255
4081_A078	DCIC ROI Reference Signature Register (DCIC2_DCICRRS7)	32	R/W	0000_0000h	51.6.6/3255
4081_A07C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS7)	32	R	0000_0000h	51.6.7/3256
4081_A080	DCIC ROI Config Register (DCIC2_DCICRC8)	32	R/W	0000_0000h	51.6.4/3254

Table continues on the next page...

DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_A084	DCIC ROI Size Register (DCIC2_DCICRS8)	32	R/W	0000_0000h	51.6.5/3255
4081_A088	DCIC ROI Reference Signature Register (DCIC2_DCICRRS8)	32	R/W	0000_0000h	51.6.6/3255
4081_A08C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS8)	32	R	0000_0000h	51.6.7/3256
4081_A090	DCIC ROI Config Register (DCIC2_DCICRC9)	32	R/W	0000_0000h	51.6.4/3254
4081_A094	DCIC ROI Size Register (DCIC2_DCICRS9)	32	R/W	0000_0000h	51.6.5/3255
4081_A098	DCIC ROI Reference Signature Register (DCIC2_DCICRRS9)	32	R/W	0000_0000h	51.6.6/3255
4081_A09C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS9)	32	R	0000_0000h	51.6.7/3256
4081_A0A0	DCIC ROI Config Register (DCIC2_DCICRC10)	32	R/W	0000_0000h	51.6.4/3254
4081_A0A4	DCIC ROI Size Register (DCIC2_DCICRS10)	32	R/W	0000_0000h	51.6.5/3255
4081_A0A8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS10)	32	R/W	0000_0000h	51.6.6/3255
4081_A0AC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS10)	32	R	0000_0000h	51.6.7/3256
4081_A0B0	DCIC ROI Config Register (DCIC2_DCICRC11)	32	R/W	0000_0000h	51.6.4/3254
4081_A0B4	DCIC ROI Size Register (DCIC2_DCICRS11)	32	R/W	0000_0000h	51.6.5/3255
4081_A0B8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS11)	32	R/W	0000_0000h	51.6.6/3255
4081_A0BC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS11)	32	R	0000_0000h	51.6.7/3256
4081_A0C0	DCIC ROI Config Register (DCIC2_DCICRC12)	32	R/W	0000_0000h	51.6.4/3254
4081_A0C4	DCIC ROI Size Register (DCIC2_DCICRS12)	32	R/W	0000_0000h	51.6.5/3255
4081_A0C8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS12)	32	R/W	0000_0000h	51.6.6/3255
4081_A0CC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS12)	32	R	0000_0000h	51.6.7/3256
4081_A0D0	DCIC ROI Config Register (DCIC2_DCICRC13)	32	R/W	0000_0000h	51.6.4/3254
4081_A0D4	DCIC ROI Size Register (DCIC2_DCICRS13)	32	R/W	0000_0000h	51.6.5/3255
4081_A0D8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS13)	32	R/W	0000_0000h	51.6.6/3255
4081_A0DC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS13)	32	R	0000_0000h	51.6.7/3256
4081_A0E0	DCIC ROI Config Register (DCIC2_DCICRC14)	32	R/W	0000_0000h	51.6.4/3254
4081_A0E4	DCIC ROI Size Register (DCIC2_DCICRS14)	32	R/W	0000_0000h	51.6.5/3255
4081_A0E8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS14)	32	R/W	0000_0000h	51.6.6/3255
4081_A0EC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS14)	32	R	0000_0000h	51.6.7/3256
4081_A0F0	DCIC ROI Config Register (DCIC2_DCICRC15)	32	R/W	0000_0000h	51.6.4/3254

Table continues on the next page...

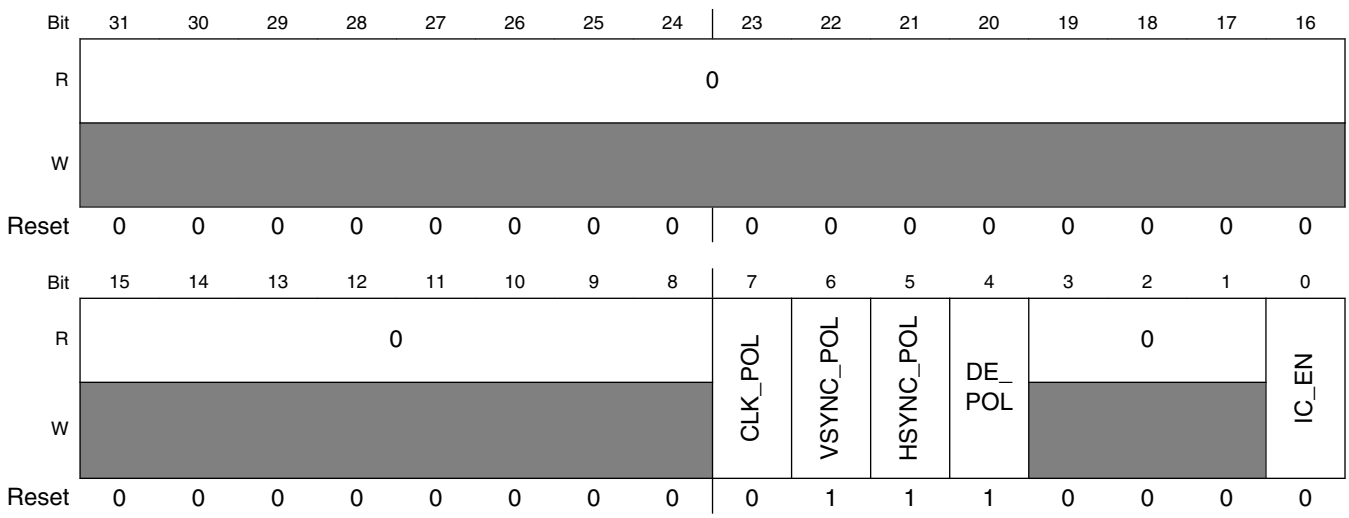
DCIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_A0F4	DCIC ROI Size Register (DCIC2_DCICRS15)	32	R/W	0000_0000h	51.6.5/3255
4081_A0F8	DCIC ROI Reference Signature Register (DCIC2_DCICRRS15)	32	R/W	0000_0000h	51.6.6/3255
4081_A0FC	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS15)	32	R	0000_0000h	51.6.7/3256
4081_A100	DCIC ROI Config Register (DCIC2_DCICRC16)	32	R/W	0000_0000h	51.6.4/3254
4081_A104	DCIC ROI Size Register (DCIC2_DCICRS16)	32	R/W	0000_0000h	51.6.5/3255
4081_A108	DCIC ROI Reference Signature Register (DCIC2_DCICRRS16)	32	R/W	0000_0000h	51.6.6/3255
4081_A10C	DCIC ROI Calculated Signature Register (DCIC2_DCICRCS16)	32	R	0000_0000h	51.6.7/3256

51.6.1 DCIC Control Register (DCICx_DCICC)

This register is used to configure the polarity of the signals and to enable/disable the DCIC module operation.

Address: Base address + 0h offset



DCICx_DCICC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
7 CLK_POL	DISP_CLK signal polarity. 0 Not inverted (default). 1 Inverted.

Table continues on the next page...

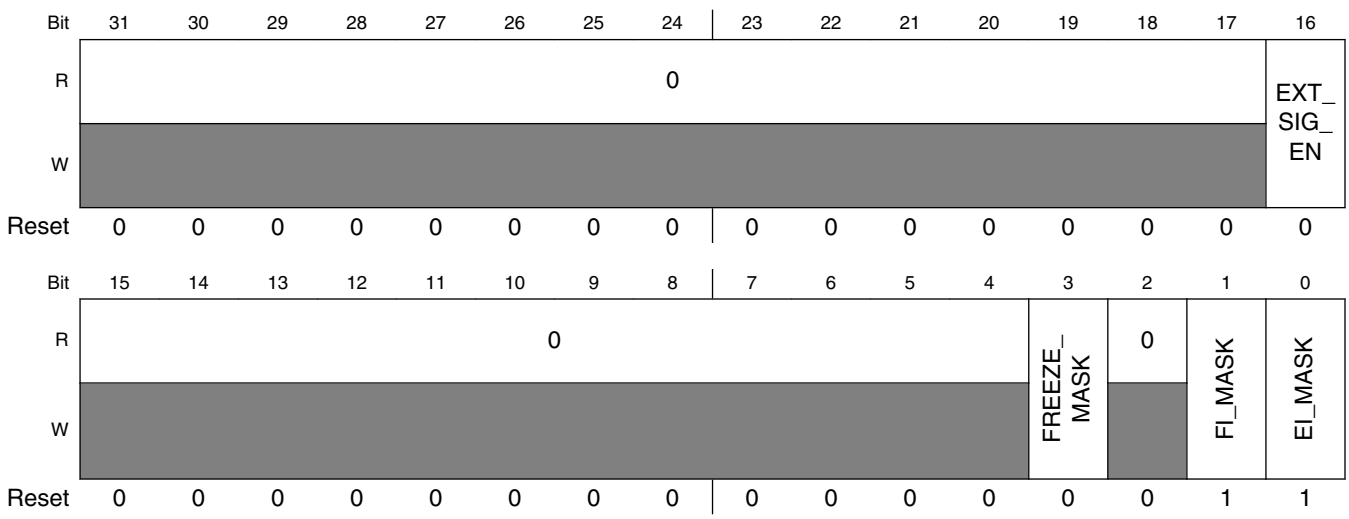
DCICx_DCICC field descriptions (continued)

Field	Description
6 VSYNC_POL	VSYNC_IN signal polarity. 0 Active High. 1 Active Low.
5 HSYNC_POL	HSYNC_IN signal polarity. 0 Active High. 1 Active Low.
4 DE_POL	DATA_EN_IN signal polarity. 0 Active High. 1 Active Low.
3–1 Reserved	This read-only field is reserved and always has the value 0.
0 IC_EN	Integrity Check enable. Main enable switch. 0 Disabled 1 Enabled

51.6.2 DCIC Interrupt Control Register (DCICx_DCICIC)

DCIC Interrupt Control Register

Address: Base address + 4h offset



DCICx_DCICIC field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

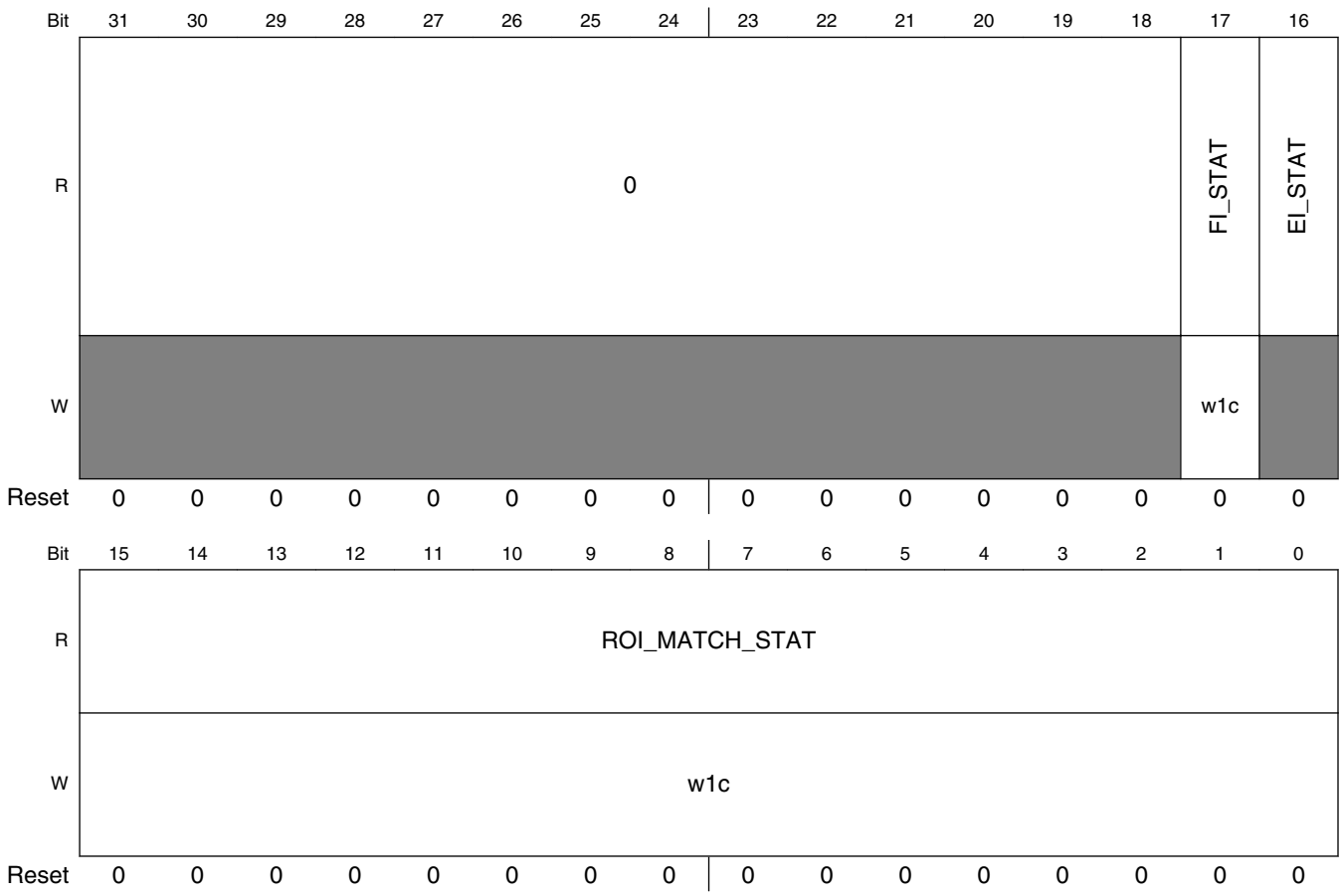
DCICx_DCICIC field descriptions (continued)

Field	Description
16 EXT_SIG_EN	External controller mismatch indication signal. 0 Disabled 1 Enabled
15–4 Reserved	This read-only field is reserved and always has the value 0.
3 FREEZE_MASK	Disable change of interrupt masks. "Sticky" bit which can be set once and cleared by reset only. 0 Masks change allowed 1 Masks are frozen
2 Reserved	This read-only field is reserved and always has the value 0.
1 FI_MASK	Functional Interrupt mask. Can be changed only while FREEZE_MASK = 0. 0 Mask disabled - Interrupt assertion enabled 1 Mask enabled - Interrupt assertion disabled
0 EI_MASK	Error Interrupt mask. Can be changed only while FREEZE_MASK = 0. 0 Mask disabled - Interrupt assertion enabled 1 Mask enabled - Interrupt assertion disabled

51.6.3 DCIC Status Register (DCICx_DCICS)

DCIC Status Register

Address: Base address + 8h offset



DCICx_DCICS field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 FI_STAT	Functional Interrupt status. Write "1" to clear. 0 No pending Interrupt 1 Pending Interrupt
16 EI_STAT	Error Interrupt status. Result of "OR" operation on the ROI_MATCH_STAT[15:0] bits. Cleared when these bits are clear. 0 No pending Interrupt 1 Pending Interrupt
ROI_MATCH_ STAT	Each set bit of this field indicates there was a mismatch at the appropriate ROIs signature during the last frame.

Table continues on the next page...

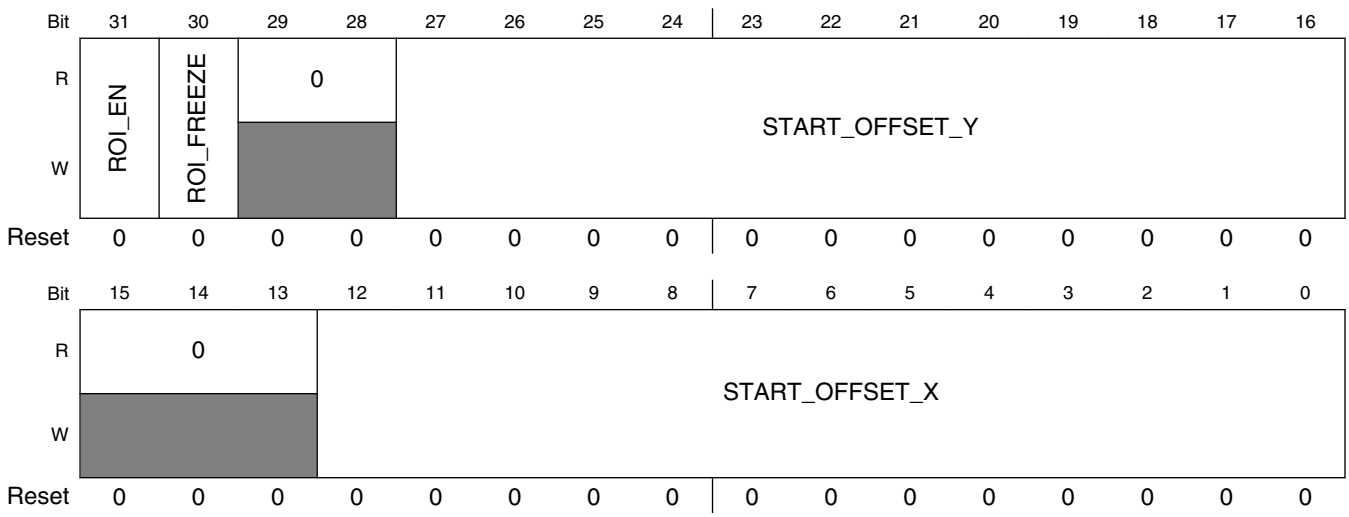
DCIC*x*_DCICS field descriptions (continued)

Field	Description
	Valid only for active ROIs. Write "1" to clear. 0 ROI calculated CRC matches expected signature 1 Mismatch at ROI calculated CRC

51.6.4 DCIC ROI Config Register (DCIC*x*_DCICRC*n*)

DCIC ROIn Configuration Register

Address: Base address + 10h offset + (16d × i), where i=0d to 15d



DCIC*x*_DCICRC*n* field descriptions

Field	Description
31 ROI_EN	ROI tracking enable 0 Disabled 1 Enabled
30 ROI_FREEZE	When set, the only parameter of the ROI that can be changed is the reference signature. "Sticky" bit - can be set once and cleared by reset only. 0 ROI configuration can be changed 1 ROI configuration is frozen
29–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 START_ OFFSET_Y	Row number of ROIs upper-left corner (Y coordinate) Range: 0 to 2^12-1

Table continues on the next page...

DCIC*x*_DCICRC*n* field descriptions (continued)

Field	Description
15–13 Reserved	This read-only field is reserved and always has the value 0.
START_ OFFSET_X	Column number of ROIs upper-left corner (X coordinate) Range: 0 to 2^13-1

51.6.5 DCIC ROI Size Register (DCIC*x*_DCICRS*n*)

DCIC ROIn Size Register

Address: Base address + 14h offset + (16d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				END_OFFSET_Y												0			END_OFFSET_X											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCIC*x*_DCICRS*n* field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–16 END_OFFSET_Y	Row number of ROIs lower-right corner (Y coordinate) Range: 1 to 2^12-1
15–13 Reserved	This read-only field is reserved and always has the value 0.
END_OFFSET_X	Column number of ROIs lower-right corner (X coordinate) Range: 1 to 2^13-1

51.6.6 DCIC ROI Reference Signature Register (DCIC*x*_DCICRRS*n*)

DCIC ROIn Reference Signature Register

Address: Base address + 18h offset + (16d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	REFERENCE_SIGNATURE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCIC*x*_DCICRRS*n* field descriptions

Field	Description
REFERENCE_ SIGNATURE	32-bit expected signature (CRC calculation result) for the ROI

51.6.7 DCIC ROI Calculated Signature Register (DCIC*x*_DCICRCS*n*)

DCIC ROIn Calculated Signature Register

Address: Base address + 1Ch offset + (16d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CALCULATED_SIGNATURE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCIC*x*_DCICRCS*n* field descriptions

Field	Description
CALCULATED_ SIGNATURE	32-bit actual signature (CRC calculation result) for the ROI during the last frame. Updated automatically at the beginning of a next frame.

Chapter 52
Pixel Pipeline (PXP)

52.1 Chip-specific PXP information

Table 52-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

52.2 Overview

The Pixel Processing Pipeline (PXP) is used to process graphics buffers or composite video and graphics data before sending to an LCD display or TV encoder. It is used to minimize the memory footprint required for the display pipeline and provide an area and performance optimized to both SDRAM-less and SRAM-based systems.

The PXP integrates of several independent processing stages into a cohesive strategy to create flexible pixel pipeline.

52.2.1 Block Diagram

The PXP block diagram is shown below:

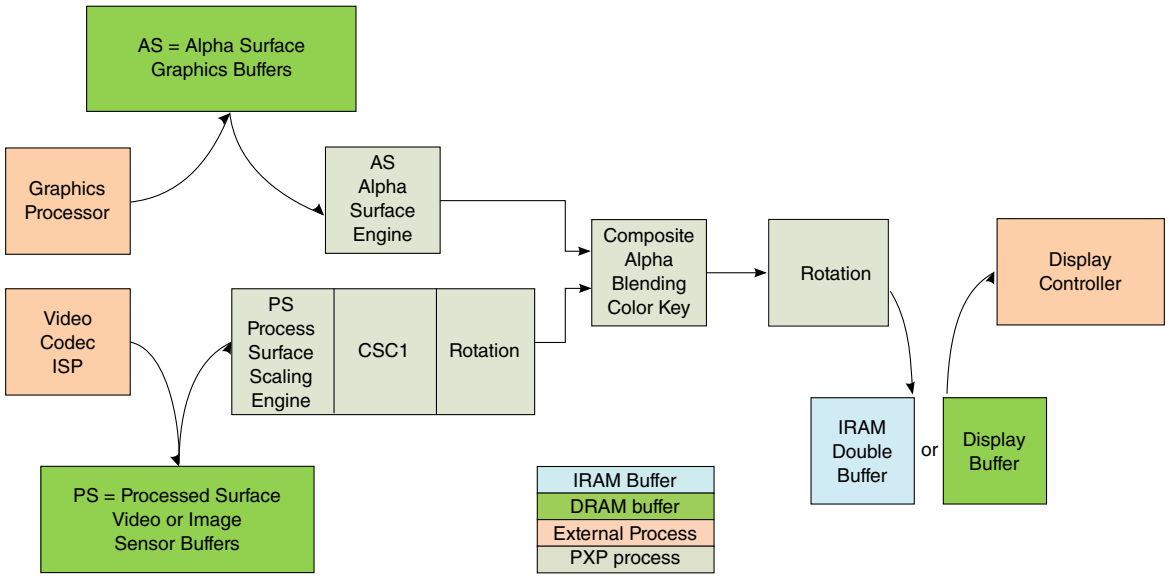


Figure 52-1. PXP Architecture

By integrating multiple blocks, intermediate buffer operations to external memory are removed, reducing external memory bandwidth, power, and software control complexity.

The PXP combines the following into a single processing engine:

- Scaling
- Color Space Conversion (CSC)
- Rotation

52.2.2 Features

The following features are supported in PXP:

- BitBlit
- Flexible image compositions
- Porter-Duff operation
- Image rotation (90°, 180°, 270°)
- Image resize
- Color space conversion
- Multiple pixel format support (RGB, YUV444, YUV422, YUV420, YUV400)
- Standard 2D-DMA operation

52.3 Functional Description

52.3.1 Top-level architecture

The PXP consist of several pipelined blocks that perform the video source frame scaling, color space conversion, alpha-blending/color key algorithm, secondary CSC, pixel correction.

The entire pipeline operate within the requirements of the PXP architecture, and perform operations on either 8x8 or 16x16 pixel blocks in the representative source buffers. The entire pipeline operate within the context of two iteration counters that iterate through the appropriate grid of input blocks to produce the rotated output grid blocks in scan-line order.

Figure 52-1 shows the high-level architecture of the scaling, color space conversion, blending, pixel correction,rotation engines.The Alpha Surface Engine fetches one RGB graphics plane alpha surface (AS). The scaling engine fetches a single processed surface (PS), which can be blended with the AS surface.Although the PXP processes NxN pixel macro blocks, each of the AS or PS surfaces can have any pixel alignment within the output buffer. There are no restrictions and any pixel coordinates within the output buffer are valid. The upper left origin of the output buffer is defined as pixel (0,0). The upper left and lower right coordinates for each of the AS and PS are inclusive within the output buffer.

Figure 52-2 represents a sample output buffer configuration with both an AS and PS included. The alignment of each AS and PS within the output buffer can be at any arbitrary pixel locations. For example, the PS has an upper left coordinate (ULC) of (2,2) and a lower right coordinate (LRC) at pixel (13,13). The maximum value for the ULC and LRC for each of the AS and PS is bounded by the LRC of the output buffer, (15,15) for this example.

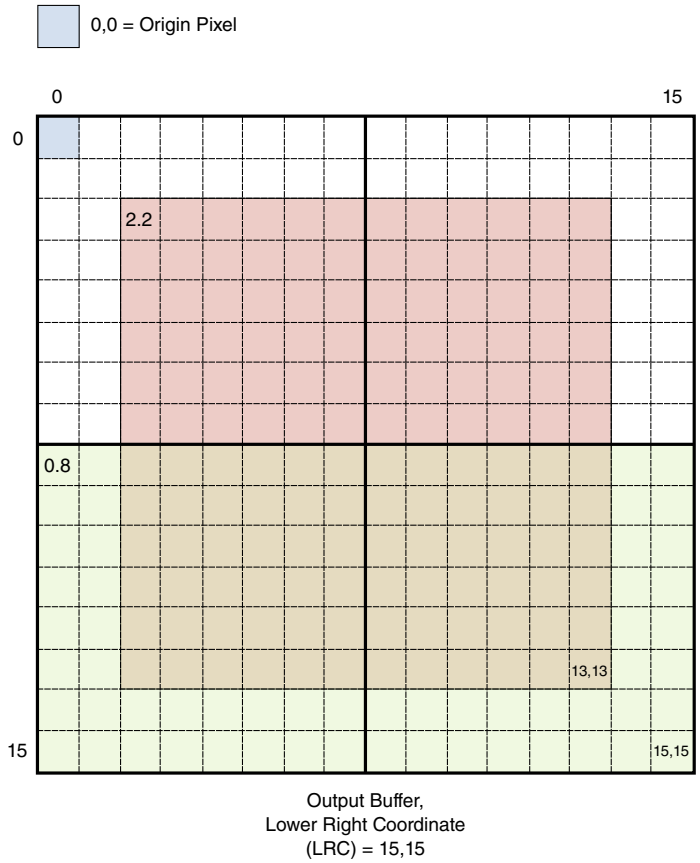


Figure 52-2. Sample output buffer configuration

The AS engine supports RGB pixel formats, and the PS engine supports RGB, YUV, and YCbCr pixel formats. The CSC1 can be used to convert to RGB pixel formats so that the PS surface can be blended with the AS surfaces in the compositing engine in the RGB color space. There is a single rotation engine in the PXP with a programmable location with two possible rotations, one after PS processing, other just before the transfer to display buffer. Rotation can occur after image composition or at the output of the PS engine. In the first scenario, all the data produced by the AS and PS engines is rotated. When the rotation module is programmed to rotate only PS images, the AS is not rotated, and AS pixels are combined with rotated PS surfaces.

52.3.1.1 Processing Details

The PXP architecture has been driven primarily by the requirement that the output buffer must be processed and rotated without intermediate frame buffer stored in external memory.

This reduces the use of external memory bandwidth requirements thus reducing overall system power consumed.

Since the output of the rotation block must be NxN pixel blocks in scan order, the entire pipeline will operate on NxN pixel blocks. In essence, the pipeline will be able to operate on blocks in a random access fashion, but the entire pipeline will operate within the context of two iteration counters that will iterate through the horizontal and vertical input blocks to generate the required output block.

Processing Pipeline

The control block will coordinate the processing of the pixel blocks within the source and destination image buffers. It begins by issuing a command to each stage of the pipeline requesting that operations be done for the block at offset x, y. When the block accepts the command, it asserts its acknowledge signal for a single cycle to indicate the acceptance and allow the control unit to move to the next block.

When the PS and AS fetch engines have received a command, they will fetch the required data and place it into their fetch buffers. If compositing the RGB AS surface with the PS surface, then the output of the PS engine needs to be converted to the RGB color space using CSC1, since all compositing occurs in the RGB color space. For YUV output pixel formats, the CSC1 unit can be enabled to convert pixels into the RGB space for subsequent compositing with AS pixels. If the final output color space is YUV and there is no compositing required (AS not present, for example), then the CSC unit can be bypassed and the pixel data path will pass the YUV pixels to the rotation engine. For YUV output formats, scaling operations, and rotation operations are still valid, but blending RGB AS surfaces with YUV PS surfaces is NOT supported.

NOTE

If input format and output format are both YUV, the CSC1_COFF0[YCBCR_MODE] need be set according to the input format even when CSC1 is bypassed.

The alpha blender/color key module will process a pixel any time that both inputs present valid data.

A handshake will be created between each stage and a pipeline controller to handle the advance of the pipeline and generation of the iteration counters. The pipeline controller will also maintain the interlocks with the LCD interface for the case where the LCD display and pixel processing pipeline use the SRAM to maintain the double buffer block intermediate buffer.

52.3.1.2 Scaling Operation

The scaling engine operates on YUV (or YCbCr) 422 or 420 and any RGB formatted pixels. Each color plane is sourced from color planes indicated by different base address registers.

The scaling source data can be stored as 3 individual planes for each Y, U, and V data, stored as two planes as a single Y and interleaved UV plane, or stored as a single plane with YUV/RGB interleaved on a per byte basis.

The scaled output image is presented to the CSC1 module as YUV444 or RGB888 pixels with a single byte for each color channel. The scaler can reduce an input image by a maximum factor of 16. In this case, the output image will be 1/16 the dimension of the input image in each of the X and Y axis. There are no limits, essentially, on increasing the source image size. The theoretical maximum increase is 4096 since a 12 bit fractional step function is used when scaling an input image. Scaling in either axis, X or Y is independent, so a source image can appear stretched in either direction.

All source images pass through the scale engine. The PXP alpha blend module and AS pixel streams are in the RGB888 format, so PS pixel buffers must be converted to the RGB888 format for alpha blending. The scaling engine works with the CSC1 module to translate YUV/YCbCr pixel formats to RGB888 for output frame buffer compositing using the alpha blender. The CSC1 unit can be bypassed so compositing can occur in the alpha engine.

The scaling operation is divided into two scaling steps. The first step is a decimation scaler, and the second step is a bilinear filter. The decimation filter provide a maximum down scaling factor of 8, and the subsequent bilinear filter provides a maximum scaling factor of 2. Combined, the maximum scaling factor can be up to 16. The decimation and bilinear scaling engines are independently programmable. There is also an initial offset that is programmable to allow more source data to be considered in the bilinear scaling engine.

52.3.1.3 Decimation Image Scaling

The first of two scaling engines is the decimation filter.

The intent of the decimation filter is to use as much source data as is possible to create the output image frame buffer. The decimation filter simply discards certain pixels from the source PS image depending on the reduction selected.

For RGB pixel formats, each color channel is treated equally since there is the same amount of pixel data within each color plane. For YUV422/420 formats, the chroma samples are already subsampled by 2. In these decimation scenarios, the chroma

decimation factor is adjusted to account for the pre-decimation of the chroma samples. For example, since YUV422 is already sub-sampled by 2 horizontally, an X decimation factor of 2 does not apply to the YUV422 pixels in the X direction. All the chroma samples are passed on to the bilinear filter in this case. As another example, an X decimation factor of 4 will decimate the chroma samples by 2, since this factor combined with the pre-decimation factor of 2 in the pixel source buffers totals an overall decimation factor of 4.

The following example will show which pixels (in green) in a source RGB buffer that are passed to the bilinear filter for an X decimation factor of 2 and a Y decimation factor of 4. All pixels coincident with dashed lines are discarded.

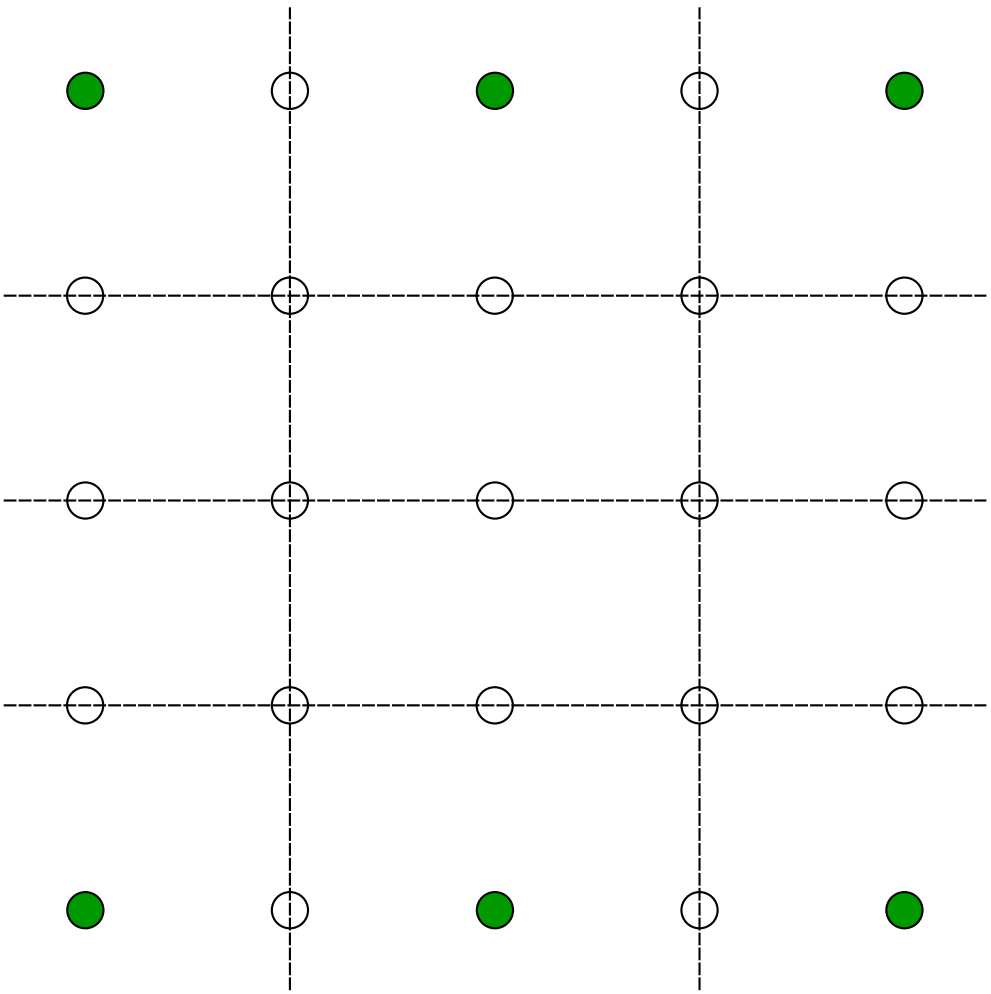


Figure 52-3. RGB decimation X /2, Y /4

Using the same decimation factor as the above scenario for RGB pixels, but using YUV420 source buffers, it can be shown that the decimation factor for the Y and UV components of data are decimated differently. This is due to the pre-decimation of the chroma samples in the source frame buffers. Figure 4: YUV420 decimation X /2, Y /4 indicates that the U/V samples in the X direction are not decimated, but the Y samples in the X direction are decimated by the factor of 2.

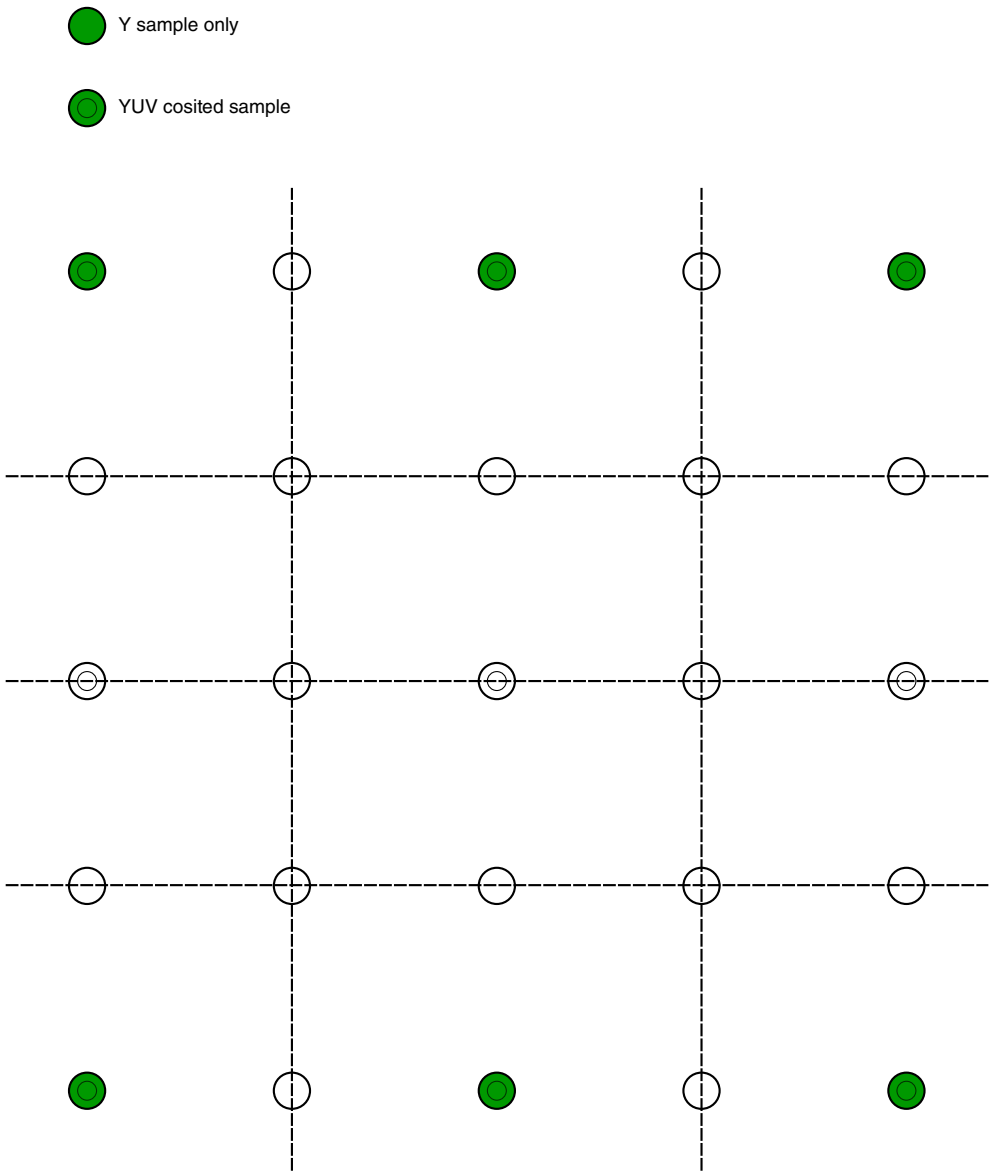


Figure 52-4. YUV420 decimation X /2, Y /4

52.3.1.4 Bilinear Image Scaling Filter

The PXP implements a bilinear scaling filter to resize an input image to a different resolution for display output.

The bilinear filter is a weighted average of the four nearest pixels that can be sourced to approximate the pixel in the output frame buffer.

When scaling YUV data, the UV values are offset by 0x80 (top bit inverted) to shift the signed UV bits into an unsigned equivalent with a range of 0 to 255. YCbCr data does not have to be shifted since it is defined as an unsigned byte. The REG_CSC1_COEF0[YCBCR_MODE] bit controls whether this operation is applied to the input UV bytes.

After scaling, the offset is removed so that the range for UV data is signed from -128 to 127.

The reason for this adjustment is based on the implementation of an unsigned scaling engine, and therefore, is to ensure that the scaled values are handled properly. Consider the following table:

Format	pixel0	pixel1	average	Result
decimal	-2	+2	0	Correct
CbCr	0x7E	0x82	0x80	Correct (0x80 is 0 in CbCr)
UV	0xFE	0x02	0x80	Incorrect (0x80 is -128 in UV)
decimal	-32	+16	-8	Correct
CbCr	0x60	0x90	0x78	Correct (0x78 is -8 in CbCr)
UV	0xE0	0x10	0x78	Incorrect (0x78 is +120 in UV)

To compute the output pixel value at position as indicated by P, consider the diagram below.

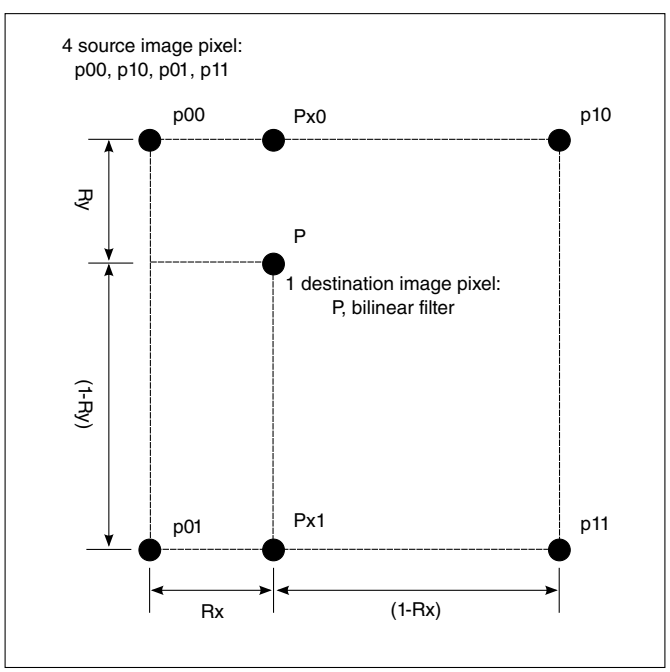


Figure 52-5. Output Pixel Value

A step function is used to indicate the position of the pixel "P" in the output frame. This position may not coincide with a single pixel position in the input frame buffer. In this case, the four closest pixels in the input frame are used to approximate the value of the pixel in the output frame.

The PXP scaler first computes a linear filter in the X axis to create the two intermediate pixel values Px0 and Px1. The step function's X fractional component is used to provide the weighting factor for blending p00 with p10 to provide Px0. Likewise, Px1 is also derived from a linear filter using p01 and p11.

The equations for Px0 and Px1 are as follows:

Px0 = p00*(1-Rx) + p10*Rx

Px1 = p01*(1-Rx) + p11*Rx

The PXP scaler uses the intermediate X pixels Px0 and Px1 and implements a bilinear filter on these two pixel values to produce the final pixel value at position P. The remainder of the step function for the Y axis is used to compute the weighted average pixel result. The equation for final filtered pixel is:

P = Px0*(1-Ry) + Px1*Ry

52.3.1.5 YUV 4:2:2 Image Scaling

The following figure illustrates the positioning of YUV samples for the 4:2:2 formats. There are twice as many Y luma samples then U and V chroma samples horizontally.

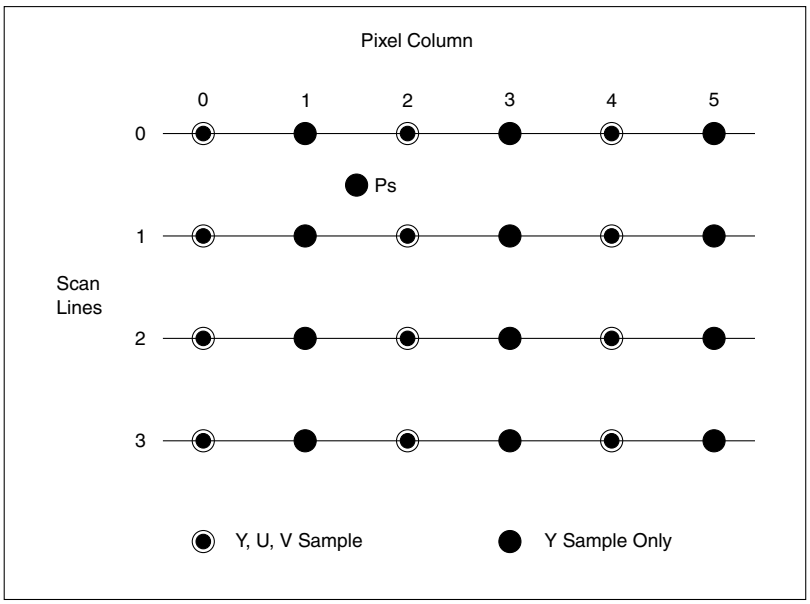


Figure 52-6. YUV Sample Positioning, 4:2:2

Consider the scaled output pixel Ps (pixel scaled) which has an accumulated step function of X=1.5 and Y=0.5. The remainder for the step function is Rx = 0.5 and Ry = 0.5. Or, the sub pixel position of output pixel Ps is half way between line 0 and 1 and half way between column 1 and 2.

The Y output component of Ps is simply the bilinear function of the four nearest Y samples from the input image. Specifically, the Y values at [1,0], [2,0], [1,1], and [2,1] are used to compute the Y for Ps.

For the U and V components of Ps, there are no samples present in the column position 1. The bilinear filter uses chroma components located at [0,0], [2,0], [0,1] and [2,1]. Since the chroma components are not sub sampled vertically, the remainder used to combine pixels vertically is Ry=0.5 (the same as for Y). However, horizontally, the scaling engine shifts the remainder by a factor of 2. So an X axis step function value of X=1.5 has a remainder Rx=0.75. Source chroma values are not replicated, they are completely interpolated using the four nearest chroma samples to approximate U and V at Ps.

52.3.1.6 YUV 4:2:0 Image Scaling

The following figure illustrates the positioning of YUV samples for the 4:2:0 formats. Chroma is sub sampled both horizontally and vertically. In this format, the chroma frame buffers contain ¼ the data that the luma frame buffers store.

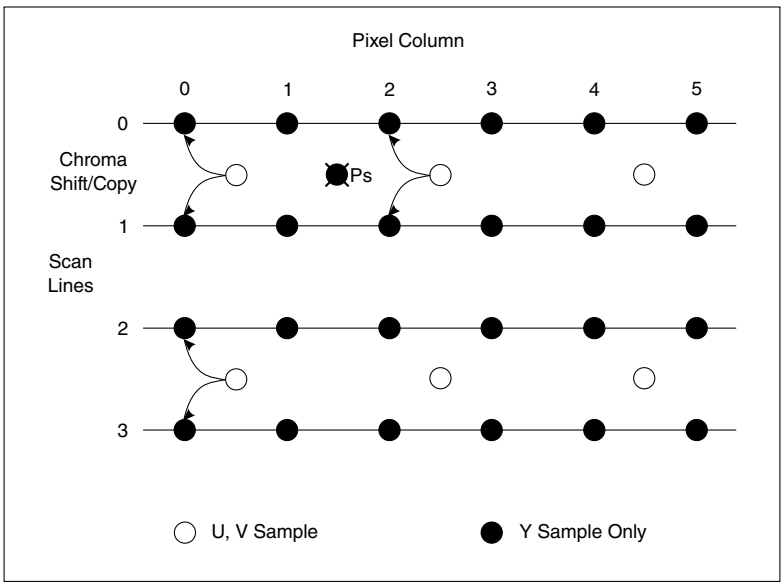


Figure 52-7. YUV Sample Positioning, 4:2:0

The Y output component for all scaled pixels in 4:2:0 formats are the same as for the 4:2:2 pixel formats.

The U and V output components have two considerations when computing the output pixel Ps.

1. All chroma samples from the input source image are shifted left and up by ½ a sample position of the input pixel matrix.
2. Odd scan lines are replicated using the previous even chroma scan line values. So, output image chroma values that map between even to odd scan lines are replicated in the vertical axis. In contrast, output image chroma values between odd to even scan lines are interpolated vertically.

The chroma values are interpolated horizontally as in the 4:2:2 pixel format.

As an example, consider the interpolated pixel Ps in the 4:2:0 diagram above. For the Y component, the interpolated output luma is a function of the Y values in the source frame buffer at position [1,0], [2,0], [1,1], [2,1].

For the U and V interpolated samples, the chroma values on scan line position 0.5 are shifted so that they coincide with the even luma sample points. They are also replicated so that a single chroma scan line is used twice. The chroma scan line at 0.5 is replicated to represent the 4:2:2 sample points for scan line 0 and 1. The chroma scan line at 2.5 is replicated to represent the 4:2:2 sample points for scan line 2 and 3. This pattern of chroma replication occurs for the entire source frame buffer during the scaling operation.

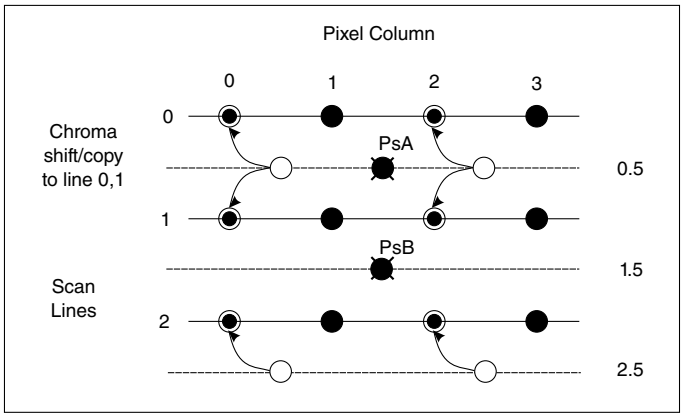


Figure 52-8. Scaled Chroma Computation Examples

The preceding diagram has two examples for the computation of the scaled chroma output pixel. For chroma at output position PsA (vertical position 0.5), interpolation occurs in the X axis using chroma values at column 0 and column 2. However, since line 0 and line 1 have equal chroma values due to chroma line replication, scaling in the Y axis results in replication of chroma values.

For chroma at output position PsB (vertical position 1.5), interpolation occurs in both the X and Y axis. The Y axis is an interpolation since the chroma values copied to scan line 1 and 2 and not the same.

In summary, any output image pixels that map to an odd scan line above and an even scan line below are interpolated vertically. Output image pixels that map to an even scan line above and an odd scan line below are replicated vertically.

52.3.1.7 RGB/YUV444 Image Scaling

For all RGB formats, the RGB pixels are converted up to RGB888 with 8 bits per each color component.

Then each color component is passed to the scaling engine and each component is treated in the same manor. The RGB scaling operation is the same as for the Y scaling operation described in the preceding sections. Also, YUV444 contains a byte for each color plane at each pixel location, so all three color components are scaled in the same manner.

52.3.1.8 Color Space Conversion (CSC)

The CSC unit converts pixels between color spaces. The CSC1 unit is right after the scaling unit and is dedicated to converting from YUV to RGB. All coefficients are programmed as two's compliment numbers and CSC unit can be bypassed if conversion is not desired in the pixel data path.

52.3.1.9 CSC1 Operation

The CSC1 module receives scaled YUV/YCbCr444 pixels from the scale engine and converts the pixels to the RGB888 color space only if CSC1 is enabled.

The CSC1 module will convert only to the RGB color space and it can be bypassed to allow YUV pixels through the data path. These pixels are loaded into the pixel FIFO for processing by subsequent modules in the pixel data path.

The following equations are used to perform YUV/YCbCr -> RGB conversion. The constants will be stored in the PXP control registers as two's compliment values to allow flexibility in the implementation and to allow for differences in the video encode and decode operations. In addition, this provides a software mechanism to manipulate brightness or contrast.

R = C0(Y+Yoffset) + C1(V+UVoffset)

G = C0(Y+Yoffset) + C3(U+UVoffset) + C2(V+UVoffset)

B = C0(Y+Yoffset) + C4(U+UVoffset)

Note: In the equations above, U and V are synonymous with Cb and Cr in regards to the color space format of the source frame buffer.

Saturation of each color channel is checked and corrected for excursions outside the nominal YUV/YCbCr color spaces. Overflow for the three channels are saturated at 0x255 and underflow is saturated at 0x00.

The table below indicates the expected coefficients for YUV and YCbCr modes of operation:

Coefficient	YUV	YCbCr
Yoffset	0x000	0x1F0 (-16)
UVoffset	0x000	0x180 (-128)
C0	0x100 (1.00)	0x12A (1.164)
C1	0x123 (1.140)	0x198 (1.596)
C2	0x76B (-0.581)	0x730 (-0.813)
C3	0x79B (-0.394)	0x79C (-0.392)

Table continues on the next page...

C4	0x208 (2.032)	0x204 (2.017)

52.3.1.10 YUV versus YCbCr Support

By default, the PXP color space coefficients are set to support the conversion of YUV data to RGB data.

If YCbCr input is present, software must change the coefficient registers appropriately (see the register definitions for values). Software must also set the YCBCR_MODE bit in the COEFF0 register to ensure proper conversion of YUV versus YCBCR data.

52.3.1.11 Alpha Blending/Color Key

Regardless of pixel input format, the PS and AS pixels are normalized to 32-bits, organized as one alpha and three data bytes. Alpha blending occurs in the RGB space, if blending is required, PS pixels should be converted to RGB space. If no alpha blending is required, then YUV pixels can bypass the alpha blending ALU without color space conversion. All pixels are processed by the pixel ALU, but the ALU operations can be disabled to achieve pixel pass through for either PS or AS source pixels.

52.3.1.12 Alpha Blend

The alpha value for an individual pixel represents a mathematical weighting factor applied to the AS pixel. An alpha value of 0x00 corresponds to a transparent pixel and a value of 0xFF corresponds to an opaque pixel.

The effective alpha value for an AS pixel is determined by the AS_CTRL[ALPHA] and AS_CTRL[ALPHA_CTRL] register fields. If AS_CTRL[ALPHA_CTRL] = ALPHA_OVERRIDE , the alpha value for the pixel is taken from the AS_CTRL[ALPHA] . This can be useful for applying a constant alpha to an entire image or for image formats that don't include an alpha value. If AS_CTRL[ALPHA_CTRL] = ALPHA_MULTIPLY, the pixel's alpha value will be multiplied by the pixel's ALPHA value in order to allow scaling of the pixel's alpha or to provide better control for pixel formats such as ARGB1555, which only contains a single bit of alpha.

For each color channel, the equation used to blend two source pixels is defined below:

Gá = PIO programmed global alpha (8-bit value).
Eá = Embedded alpha associated with AS pixel.
á = (Gá * Eá + 0x80)/128, and á doesn't exceed a value of 128

The result for the red channel as an example:

R[7:0] = (á * PS.r) + ((1 - á) * AS.r)

When á is 0xff, the PS pixel will not be blended with the AS pixel, but PS will be passed as the output pixel and will not be blended with AS. In this case, AS will be discarded. Likewise, if á is 0x00 for a given pixel, AS will be loaded as the output pixel.

AS_CTRL[ALPHA_INVERT] provides the option to invert the final alpha value. This essentially inverts the effect the alpha value has on the AS and PS blending operation.

52.3.1.12.1 Porter-Duff Alpha Blend

Porter-Duff blend includes 12 blending modes to describe digital image composite. These processes include:

- Clear
- Source Only
- Destination Only
- Source Over
- Source In
- Source Out
- Source Atop
- Destination Over
- Destination In
- Destination Out
- Destination Atop
- XOR

Through these processes it can achieve any 2D image composite.

To control the blending modes, please see the following picture. Please see PORTER_DUFF_CTRL register for more details.

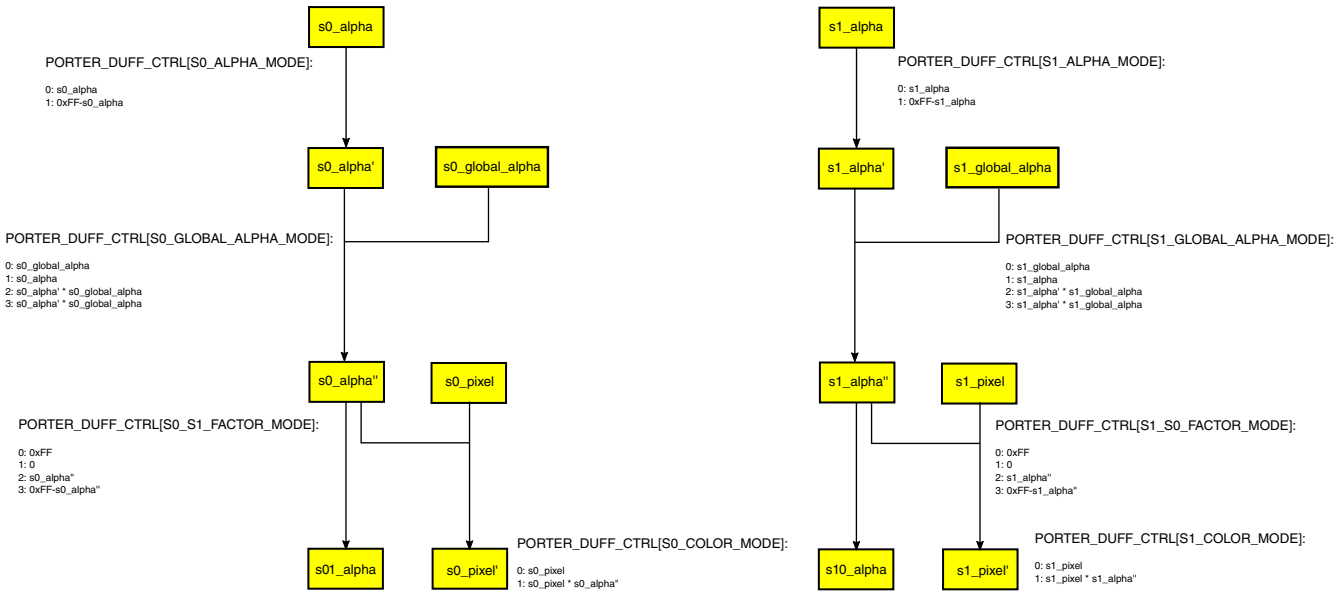


Figure 52-9. Porter-Duff Alpha Blend

image output = s01_alpha × s1_pixel' + s10_alpha × s0_pixel'

alpha output = s10_alpha × s0_alpha'' + s01_alpha × s1_alpha''

NOTE

s0 refer to PS and s1 refer to AS.

52.3.1.13 Color Key

The color key function is provided to create transparent effects on the output pixel.

Color keying is applied on the input pixels after they are converted to 8-bits for each red, green, and blue color channels (color keys are not applied directly to 16-bit pixel formats but to their corresponding 24-bit representation). A color key range is programmable for both PS and AS pixels. If the PS 24-bit pixel is within the PS color key range, then AS is passed through the pixel pipeline. In this case, alpha blending does NOT occur. Conversely, if AS is within AS color key range, then AS is passed via the PXP data pipeline. If both PS and AS color key tests pass, then the back ground color register is passed onto following PXP processing components in the pipeline.

The condition for color keying to be satisfied is:

CK0.r.low <= PS.r <= CK0.r.high

CK0.g.low <= PS.g <= CK0.g.high

CK0.b.low <= PS.b <= CK0.b.high

For example, if the "red" 8-bit value for the PS pixel (or PS.r) is between the color key low and high values (CK0.r.l and CK0.r.h), the condition is true for the red color plane. When All three color planes meet this condition, PS pixel is invisible in output register.

To disable color keying, program the low color key register value to 0xff and the high value to 0x00. This will guarantee that the color key range test will never be true.

52.3.1.14 Rotation

There is a single rotation resource integrated into the PXP. The location of this resource within the PXP data path is programmable. Rotation can occur after compositing the AS and PS buffers in the output stage.

As an alternative configuration, the PS buffer can be rotated and later composited with the AS surface that is not rotated. There is a single configuration bit that provides the configuration of where rotation is implemented within the PXP.

To rotate graphics, the hardware must read pixels in one direction across a frame buffer and write them in a alternate orientation. For the 90 and 270 degree cases, this means that lines of pixels must either be read or written vertically in a frame buffer.

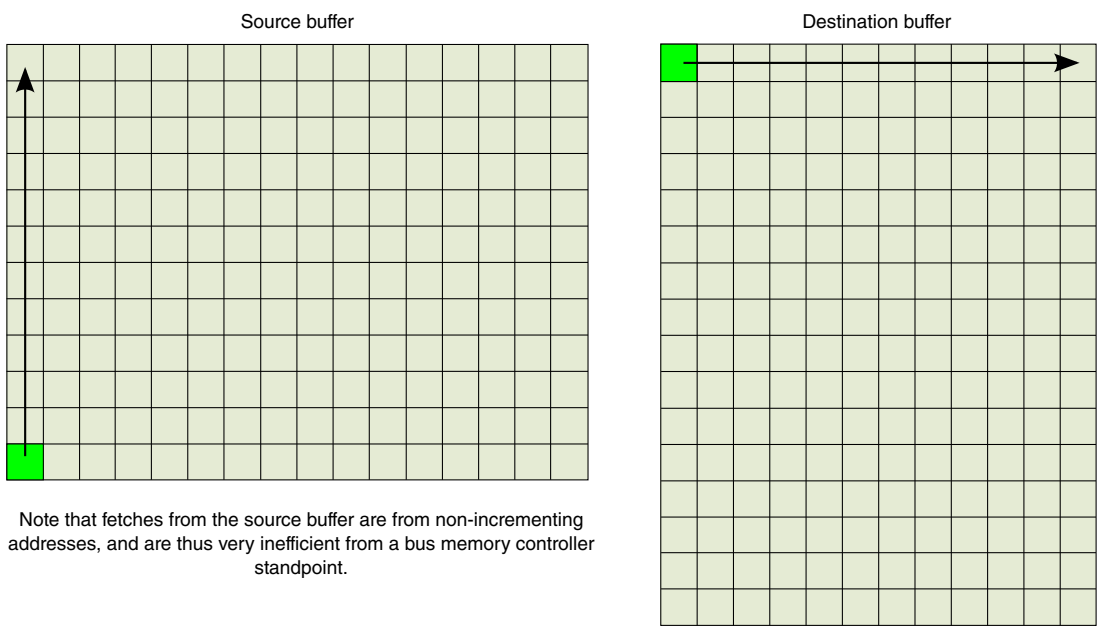


Figure 52-10. Rotation Read and Write

In order to rotate efficiently, multiple columns must be rotated to enable the engine to both fetch and store bursts of pixels, thus improving memory performance. The simplest method of doing this is to operate on square blocks of pixels. To rotate the image, each sub-block of pixels must be rotated by the required rotation angle.

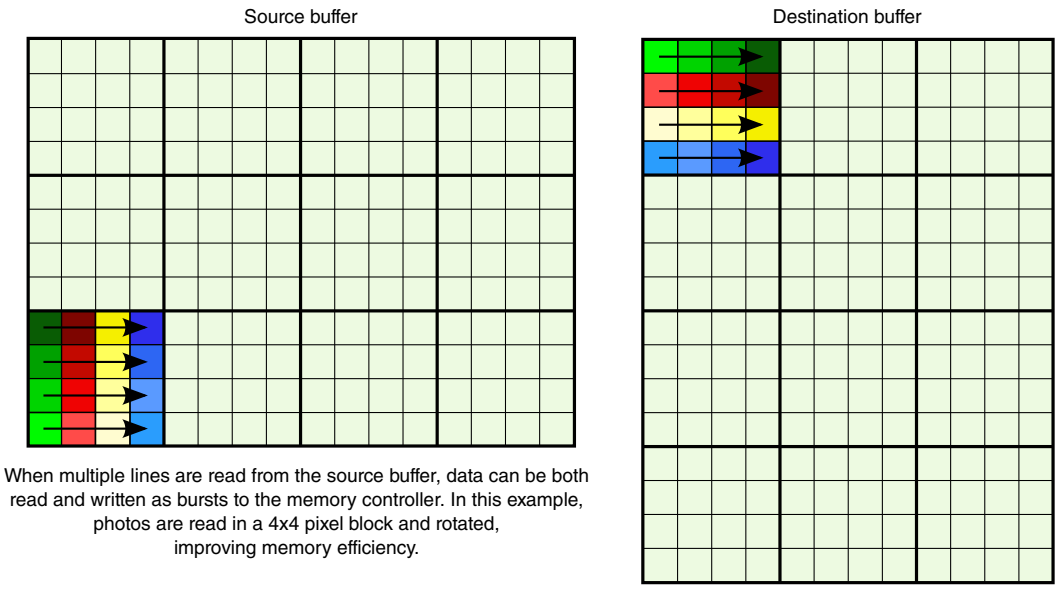


Figure 52-11. Rotated Sub-blocks

To manage the rotation process, the source image can be broken into a grid of sub-blocks that have coordinates as shown in the diagram below. In addition to rotating the sub-block, each block must be translated to a new coordinate location. For each of the rotation angles (0, 90, 180, 270), it is possible to define a simple algorithm for computing the new translated grid address. The hardware must then simply compute the memory address from the base grid address for both load and store operations.

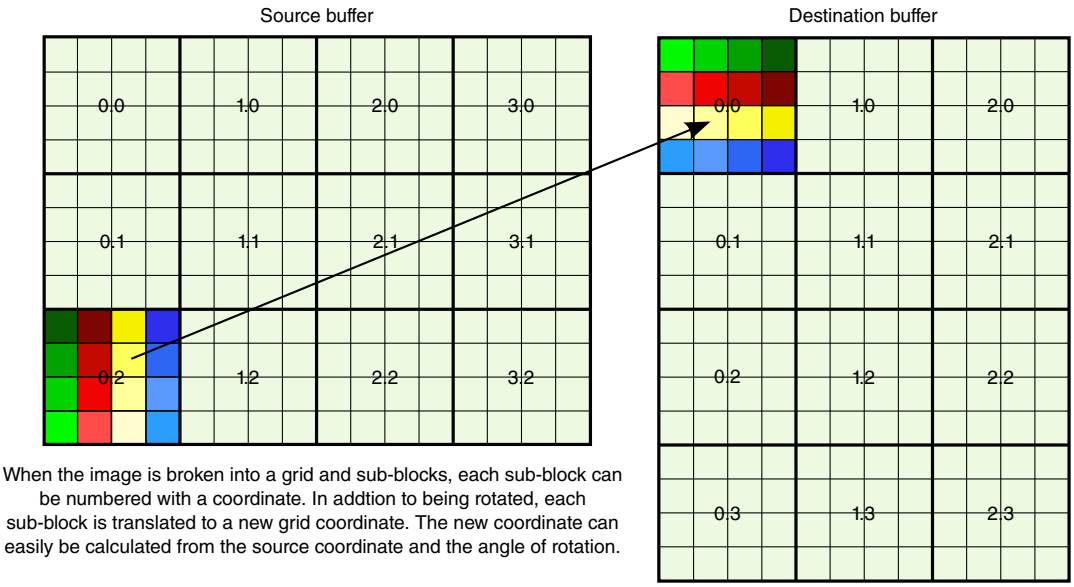


Figure 52-12. Grid of Sub-Blocks with Coordinates

In order to balance the requirements of reasonable burst sizes to the memory controller as well as keep the hardware storage requirements to a minimum, the blending/rotation engine will operate on either 8x8 or 16x16 pixel blocks. When using the Rotate engine with the input fetch engine, you need to program the input fetch engine to work in 8x8 block mode.

NOTE

An important artifact of the PXP is when rotating a source image and the output is NOT divisible by the block size selected. The output engine essentially truncates any output pixels after the desired number of pixels has been written. Since the output buffer is written as a horizontal row of blocks, the incorrect pixels could be truncated and the final output image can look shifted. In the case where the block size is programmed to 8x8, and the output size that is programmed is 12x12, then there is a remainder of 4 pixels that will be truncated in either the X and/or Y axis when the PXP operation is complete. The output will be shifted by 4 pixels in this example. To compensate for this, the source base address needs to be adjusted so the correct pixels get truncated and the image does not look shifted. In this example, with 90 degrees of rotation, the PS base address should be adjusted by 4 times the actual PS base address -(4*pitch).

52.3.1.15 Output Buffer

The output buffer engine accepts data from the PXP pixel pipeline and issues requests to transfer the output pixels to external DRAM or the internal SRAM double buffer row of blocks.

52.3.1.16 Address calculator

Each of the blocks will manage its own fetch address using a common address calculator block that computes real addresses from a base address and relative block offset from the base.

Each block will then perform the multiple line fetches (or stores) required to perform the operation. This hides all the address buffer computations from the processing blocks and allows each block to simply track the coordinate of the block it is working on.

52.3.1.17 Block size selection

The PXP can be configured to process blocks that are either 8x8 pixels or 16x16 pixels with the REG_CTRL[BLOCK_SIZE] control bit.

When selecting a 16x16 pixel block size, the accesses to fetch AS and PS images and write the final frame buffer are more efficient since twice as much data is requested and processed per memory request.

When optimizing the system for memory bandwidth and image processing time, configure the PXP to process 16x16 pixel blocks.

52.3.1.18 Interlaced Video Support

The PXP has some minimal ability to generate interlaced video content from a progressive source. There two available options, based on the bandwidth requirements and how software is managing video frames.

The PXP can either interlace on the input side (by reading every other line of input data) or on the output side (by writing the individual lines of video into two separate fields). Generally, output interleaving should be used since it is the most flexible mode (it allows scaling and full overlay support) and it only requires a single pass of the PXP to generate two separate output fields.

Input interleaving can be beneficial in cases where the PXP is running at 60fps, since it requires fewer fetches to produce the output data. There is no direct hardware support for input interleaving, in that, there is no configuration bit that can be set to alter how the PXP processes a frame for input interleaving. Input interleaving is achieved by simply setting the source frame buffer pitch value to twice the value it would normally be set to for the equivalent progressive frame. The output parameters also need to be consistent with the desired processing effect. For example, the vertical resolution would be set to account for the reduced resolution to process the interlaced input buffers.

52.3.1.19 LCDIF Handshake

The PXP and LCDIF support a mode where the internal SRAM can be used for the frame buffer to minimize external memory bandwidth required.

This is accomplished by creating two buffers in SRAM, a double buffer row of blocks, where each correspond to 8/16-lines of the frame buffer. The buffers must be consecutive and allocated as a single block of data.

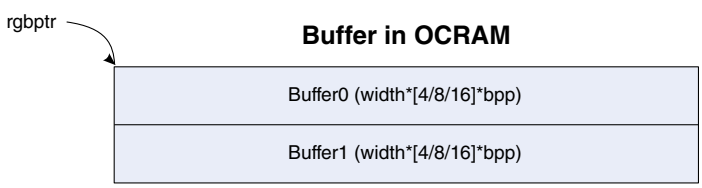


Figure 52-13. Buffer in OCRAM

The storage required can be calculated for an 8x8 block size as

- storage = 16 (lines) * rotated_row_length * pixel_size

and for 16x16 block size as

- storage = 32 (lines) * rotated_row_length * pixel_size

where pixel_size = 4 for 32bpp or 2 for 16bpp modes. The following table lists the storage requirements for common image sizes using 8x8 block size:

Image Size	Storage (16bpp)	Storage (24bpp)	Storage (32bpp)
320x240 (QVGA) - 0/180 rotation	10KB	15KB	20KB
320x240 (QVGA) - 90/270 rotation	7.5KB	11.5KB	15KB
640x480 (VGA) - 0/180 rotation	20KB	30KB	40KB

Table continues on the next page...

640x480 (VGA) - 90/270 rotation	15KB	22.5KB	30KB

The following diagram shows how the minimal rotation buffer would be organized. As the engine and LCD progress down the image, they continually swap roles of filling and emptying each eight-line buffer.

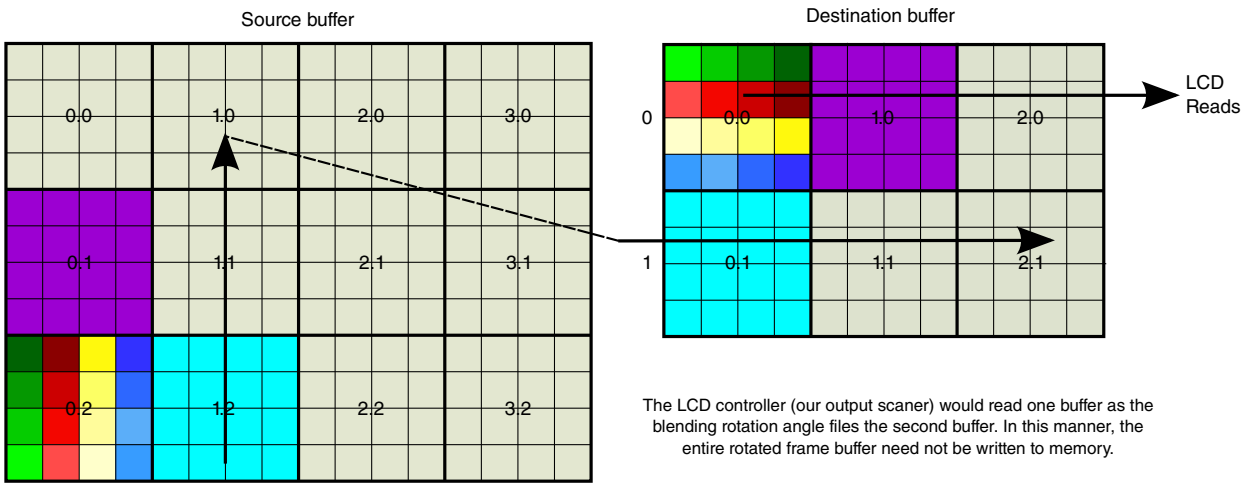


Figure 52-14. Minimal Rotation Buffer Organization

When this mode is enabled, the PXP will process one row of pixel blocks and write the results to the first SRAM buffer (buffer 0). The PXP will then alternate between writing subsequent rows to buffer 0 and buffer 1. After the PXP generates the data for one buffer, the LCDIF will begin reading that buffer and send the contents to the display device. After the LCDIF finishes reading a buffer, it will start displaying from the other buffer while the PXP continues filling the previously processed buffer.

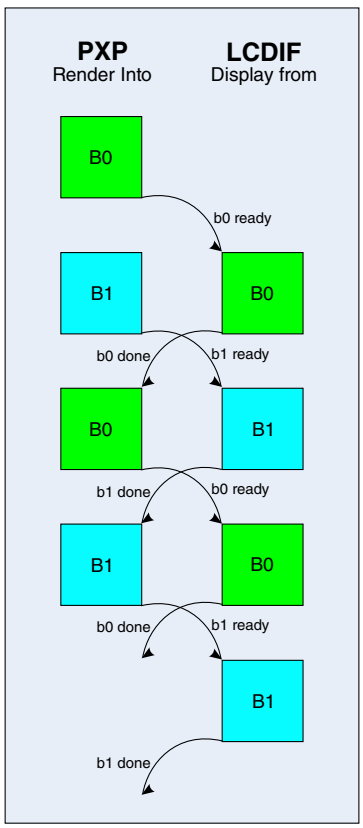


Figure 52-15. PXP and LCDIF Buffer Sharing

To accomplish the buffer sharing, the PXP and LCDIF will maintain buffer status using a pair of handshake signals. When a buffer is filled by the PXP, it will assert the pxp_lcdif_bx_ready (where x is 0 or 1) signal to indicate to the LCDIF that the buffer has valid data. The LCDIF will then release the buffer by asserting the lcdif_pxp_bx_done signal.

The basic protocol is shown in the diagram below:

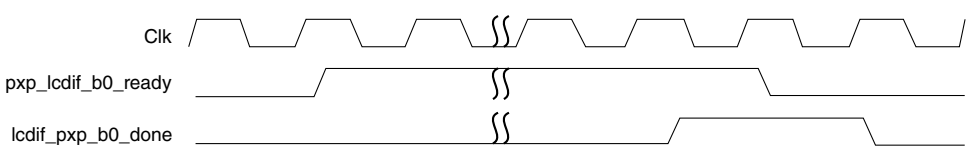


Figure 52-16. Buffer Sharing Protocol

The PXP will continue to assert the bx_ready signal until the corresponding bx_done signal is sampled high for one clock cycle. It will then deassert the bn_ready until the next time the buffer has been filled. After the PXP samples the bx_done signal asserted, it is free to begin filling the buffer with the next block size lines of display data. If a buffer has not been released when the PXP is ready to process data for that buffer, it will suspend rendering operations until the buffer has been released by the LCDIF.

52.3.1.20 LCDIF Abort

When the memory subsystem is not loaded, the PXP should be able to render the buffers faster than the LCDIF can drain the buffers.

It is possible under some scenarios (high LCDIF output rates with high memory latency) that the PXP may not be able to keep up with the LCDIF, even in the SRAM mode of operation. When this happens, the LCDIF will signal that it has completed one of the buffers before the other has been rendered by the PXP. This condition will be detected by the PXP's control logic as an "LCDIF Abort", which will cause the PXP to abort processing in the current row and proceed to the following row. It will acknowledge the abort to the LCDIF by raising the buffer_ready signal for the current buffer to enable the LCDIF to begin displaying the partially-filled buffer. While an abort will create artifacts in the video display, it does minimize the artifacts by limiting them to the remaining pixels blocks in the current row versus ruining the entire frame buffer.

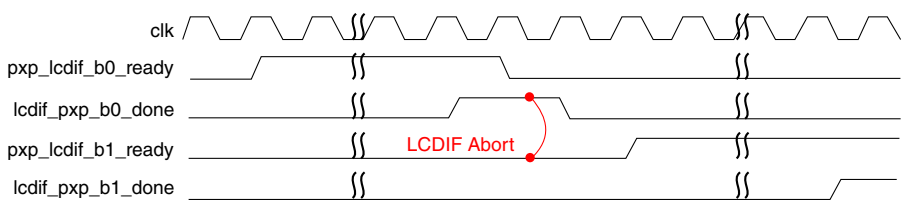


Figure 52-17. LCDIF Abort

52.3.1.21 Theory of Operation

The PXP can be used to accelerate graphics operations by offloading graphics processing from the processor. The block can perform alpha blending and color key substitution on two RGB graphics buffers.

The PXP is organized as having a processed surface (PS) and an alpha surface (AS) that can be blended with the processed surface. There are no restrictions on the location of the AS or PS within the output surface (OS). As the PXP processes NxN blocks, operations

are performed on a pixel by pixel basis. The AS and PS pixels are alpha blended, color keyed, process by CSC resources as individual pixel components. This allows efficient block processing with supporting arbitrary alignment for both the AS and PS surfaces. The resulting pixel block is then written to the corresponding block in the output buffer.

52.3.1.22 Pixel Handling

All pixels are internally represented as 32-bit values regardless of input or output pixel formats. The pixels get converted in the AS and PS buffer engines to 24-bit pixel values. There is also an 8-bit alpha value at stages up to the alpha blender within the PXP for blending within the RGB color space. Compositing of AS and PS images can only occur in the RGB color space. If compositing is not required, then YUV pixels can be transferred and processed at all PXP pixel resource components.. The color orientation of pixels within the PXP can be controlled by the CSC1 resource.

For RGB, input pixels are converted into 32-bit pixel values using the following rules for both AS and PS:

1. 32-bit ARGB8888 pixels are read directly with no conversion.
2. 32-bit RGB888 pixels are assumed to have an alpha value of 0xFF (full opaque).
3. 6-bit RGB565 and RGB555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of 0xFF (opaque). The expansion process replicates the upper pixel bits into the lower pixel bits (for instance a 16-bit RGB555 triplet of 0x1F/0x10/0x07 would be expanded to 0xFF/0x84/0x39).
4. 16-bit RGB1555 values are expanded into the corresponding 24-bit color space and assigned an alpha value of either 0x00 or 0xFF, based on the 1-bit alpha value in the pixel. The ALPHA_MULTIPLY function is useful in this scenario to allow scaling of the opaque pixels to a semi-transparent value.

Alpha values can be passed through the entire PXP data path and output in ARGB8888 or ARGB1555 pixel modes. Also, output pixels can be assigned an alpha value using the OUT_CTRL[ALPHA] register. 16-bit pixels values are formed from the most significant bits of the 24-bit pixel values.

When YUV/YCbCr output formats are selected, all pixels are internally represented as either RGB or YUV pixels values.

52.3.1.23 Output Buffer Composition

The output buffer will be rendered by composing each pixel block from the associated PS and AS buffers.

The AS pixel buffer can be blended or color-keyed with the associated data from the PS buffer (either the PS image pixels or PS_BACKGROUND register based on PS programmed coordinates).

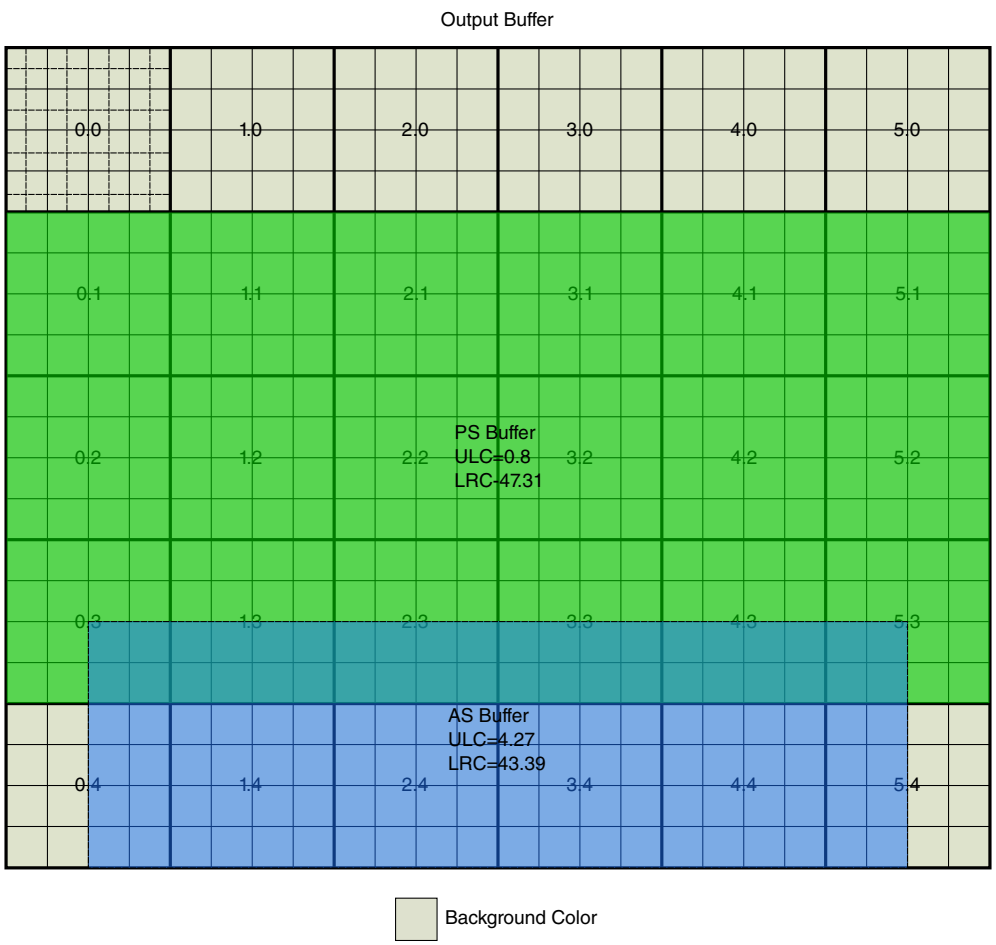


Figure 52-18. Output Buffer Composition

52.3.1.24 PS Image Processing

As the PXP processes image buffers, it iterates over the output buffer by fetching the corresponding input buffer blocks and processing the pixels embedded in these.

52.3.1.25 Letterboxing

At each pixel coordinate, the control logic determines if the PS pixel (argument also applies to AS pixels) will be used in rendering the output pixel.

This is determined by checking the output pixel's coordinates against the REG_OUT_PS_ULC and REG_OUT_PS_LRC (ULC and LRC in short) register contents. For pixels outside this region, the PS pixel will be loaded with the pixel value from REG_PS_BACKGROUND, which can be used to effectively control the letterboxing color. There are no block size or block boundary restrictions when setting the ULC or LRC for either the AS or PS. The only restriction is that the ULC and LRC are within the OUT LRC extents.

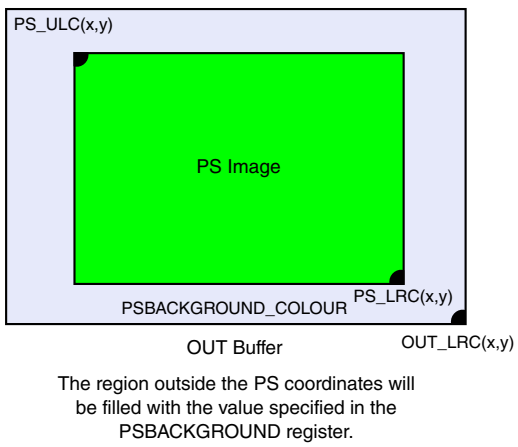


Figure 52-19. OUT Buffer

52.3.1.26 Clipping source images

A subset of the PS buffer can be used in rendering the output buffer. The PXP_PS_BUF register can indicate an offset into the PS buffer that will be used for display within the OUTPUT buffer.

The pixel at the address defined in the PXP_PS_BUF register will be the pixel that is displayed at the pixel coordinate indicated by PXP_OUT_PS_ULC within the output buffer. Essentially, the PXP_PS_BUF register can be used to establish an offset into the PS buffer thus clipping all PS buffer pixels that are at a lower address. The PXP_PS_PITCH will always indicate the number of bytes that are vertically adjacent in the PS buffer. The settings in the PXP_PS_BUF, PXP_OUT_PS_ULC, and PXP_OUT_PS_LRC will determine the subset of the PS buffer, or clipped PS source buffer, that will be used in the output buffer.

It is important to note that when scaling the PS buffer, the coordinates of the PS buffer within the output buffer need to be consistent with the scaling factors and original PS buffer size.

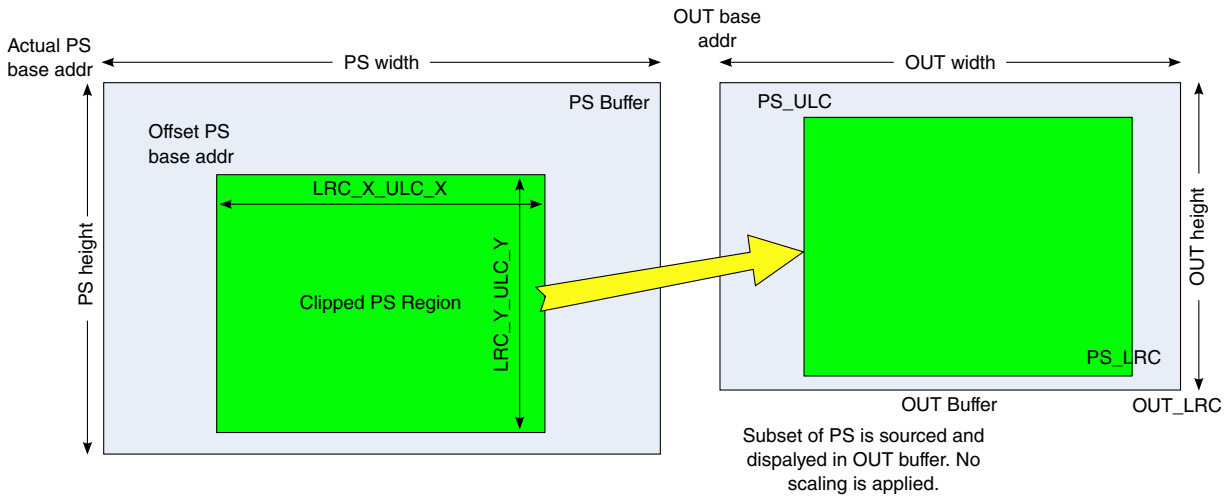


Figure 52-20. PS Buffer Scaling

When sourcing a subset of the PS image, it should fall completely within the PS buffer to avoid displaying incorrect data. The following conditions should be met:

x_base_addr_offset + x_scale * (LRC_X-ULC_X) <= PS_pitch

y_base_addr_offset + y_scale * (LRC_Y-ULC_Y) <= PS_size

The PXP hardware does not check for these conditions and will render the image as programmed. The following case could indicate invalid programming parameters for the PXP:

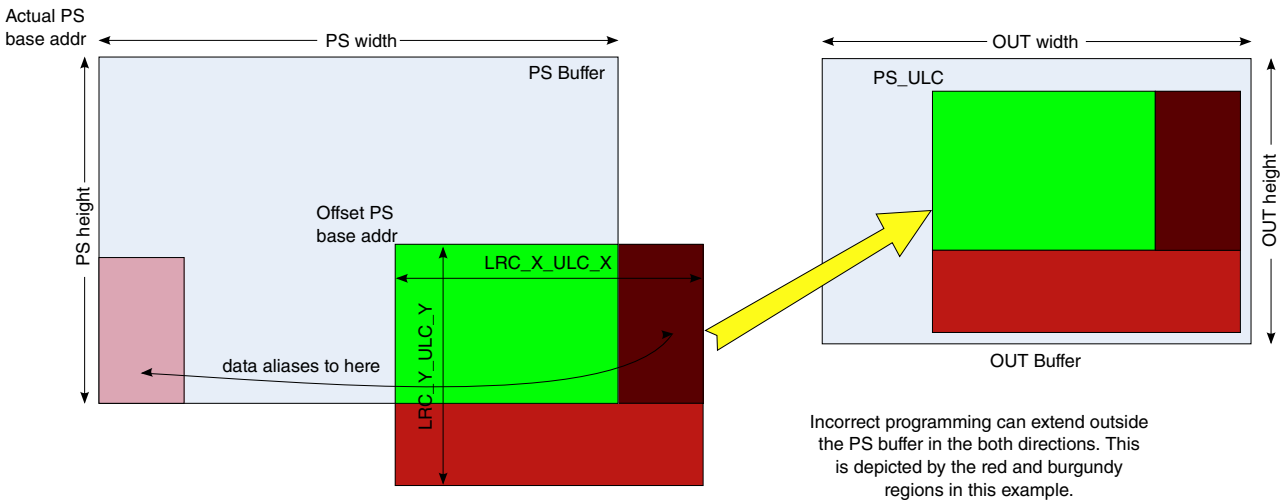


Figure 52-21. Example with Invalid Parameters

52.3.1.27 Color Key Processing

Pixels may be made transparent to the corresponding AS by using the PS color key registers.

If a PS pixel matches the range specified by the REG_PS_COLORKEYLOW and REG_PS_COLORKEYHIGH registers, the pixel from the associated AS will be displayed. If no AS is present for the pixel, a black pixel will be generated since the default AS pixel is 0x00000000 (transparent black pixel).

The most common use for this is when a bitmap does not support an alpha-field or for applications such as "green screen" where an image is substituted for a solid background color .

Figure 52-22. The PS image (player) and AS image (stadium)

The green portion of the background image can be color keyed to display the contents of the AS buffer for locations that match the color range. For this example, the color range is:

PS Colorkey: 00<R<80 70<G<ff 00<B<80

The resulting image becomes:

Figure 52-23. Resulting Image

52.3.1.28 In Place Processing (PS buffer is destination buffer)

The PXP also has the ability to process an image and write the resulting buffer back to the original PS buffer. This is referred to as "in place" rendering.

This could be useful for basic blit operations into the PS buffer. IN_PLACE operations are achieved by programming the OUT base address to the pixel location in the PS buffer that marks the upper left pixel of the update region. The actual region that is updated should be indicated by programming the ULC = (0,0) and the LRC = (X,Y). The region bounded by the coordinates will be updated, and the rest of the PS buffer will not be modified.

52.3.1.29 Alpha Surface (AS) Processing

The AS surface has a complete set of registers that determines how the AS effects the final OUT surface.

Most of the registers that exist for the PS surface also are defined for the AS surface where applicable. This is provided to replicate the SW interface for each PS and AS processes.

52.3.1.30 Alpha Handling

Alpha values in the AS are embedded in the source image pixels. For AS pixel formats that do not support an alpha value, the pixel is assigned an alpha value of 0xFF (opaque).

This can be modified by the AS control by setting either the ALPHA_MULTIPLY or ALPHA_OVERRIDE bit in the associated AS_CTRL register. If ALPHA_MULTIPLY is enabled, the 8-bit ALPHA value from the AS_CTRL register is multiplied by the source alpha before blending with the PS image. If the ALPHA_OVERRIDE bit is set, the 8-bit ALPHA value is simply substituted for the pixel.

52.3.1.31 Color Key Processing (AS_CTRL)

The AS_CTRL[ENABLE_COLORKEY] bit that can be used to enable or disable color key substitution for the AS.

When enabled, the pixel values are compared to the AS_COLORKEYLOW and AS_COLORKEYHIGH registers to determine if a match has occurred. When an AS pixel matches the color key range, the pixel from the AS image is considered transparent and the corresponding PS pixel is rendered.

AS color keys are handled in a manner similar to PS color keys. The same images used in the PS color key example could be used with the images swapped. In this case, matches on the AS image to the AS_COLORKEY register would display the PS pixels.

52.3.2 Output Image Processing

Several PXP options affect the resulting output image.

52.3.2.1 Output Image Size

The PXP generates an output image in the resolution programmed by the OUT_LRC register. As the PXP processes pixels, it iterates over the NxN blocks (in output scan-block order) based on the final image resolution.

52.3.2.2 Output Format

The result of PXP operations are written to the buffer pointed to by the OUT_BUF/ OUT_BUF2 registers. The pixel format is controlled by the OUT_CTRL[FORMAT] bit-field.

32-bit pixels are formed directly from the internal 24-bit representations and 16-bit pixel formats are generated by truncating the internal 24-bit values to the appropriate number of bits. For formats supporting an alpha value, the PXP assigns the alpha using the 8-bit value in the OUT_CTRL[ALPHA] field. For ARGB1555, the most significant alpha bit is appended to the output pixel. Also, for ARGB4444, the most significant nibble is appended to the output pixel. Single and dual buffer YUV output formats are also available. Since each pixel in the data path is represented by a full YUV444 24bpp value, decimation reduces the output in cases of YUV422/420 output formats.

52.3.2.3 Rotation/Flip operations

The PXP supports four rotation angles in conjunction with vertical and horizontal flip options. The flip operations effectively take place before the rotation.

Rotations of 0, 90, 180, and 270 degrees are supported and any combination of rotation and flip are supported. There is no performance difference between any of these modes of operation.

52.3.3 Queuing PXP transactions

The PXP supports a primitive ability to queue up one operation while the current operation is running. This is enabled through the use of the NEXT register.

When this register is written, it enables the PXP to reload its current register contents with the data found at the location pointed to by this address when it completes processing of the current frame. This feature may be useful in helping to reduce the interrupt latency in servicing the PXP, especially in cases where the PXP and eLCDIF are using the on-chip SRAM buffer handshake (since the PXP must begin generating next frame data immediately).

If the PXP is idle when the NEXT register is written, the PXP treats this as an indication that it should immediately load the values at the pointer and begin processing the frame. This ability should allow software to use the same routines when programming the PXP (so that the first frame doesn't differ from subsequent frames).

When loading values from the NEXT register, all registers in the PXP are reloaded. Some register loads have no effect

After writing the NEXT register, the PXP will set the NEXT[ENABLED] bit of the NEXT register to indicate that the next command has been queued. Software should first check the status of this bit to ensure that a previous command has not been enabled. Likewise, after programming the first frame in a sequence of frames, software should poll this bit until it is sampled logic 1'b0 before queuing the next operation.

The PXP will issue interrupts from frames as they complete, regardless of whether they were started by writing the control registers directly or using the NEXT register. When software receives an interrupt, it should check/clear the PXP's status register as normal, poll the NEXT[ENABLED] bit, and then issue the next operation. A queued operation may be cancelled by issuing a CLEAR operation to the NEXT[ENABLED] register bit. The SET and TOGGLE operations should never be used with this register.

52.3.4 Error Handling

The PXP does minimal checking on the control registers, so it is important that these are correctly specified. The PXP does monitor the bus transactions for errors and will report errors in the status register.

Upon receipt of a bus error, the PXP will set the ERROR interrupt and abort any further operations. Bus errors can be generated from any system access that results in an error response returned from the internal SIM Bus errors in the PXP are signaled as either a read or a write error, but do not indicate the failing address. Software may deduce the failing address from the current block status indicators.

52.3.4.1 Known PXP Limitations/Issues

The PXP has the following known limitations:

1. When using the NEXT register, the interrupt enable setting should remain the same for all frames. If not, the PXP will change the interrupt enable register value and possible cause the loss of an interrupt.
2. Rotations of 180/270 are not supported when performing LCD handshakes
3. PS buffer rotation and decimation does not function at the same time
4. No combination of rotate with flip, scaling or decimation is possible if buffer is unaligned.

52.3.5 Clocks

The following table describes the clock sources for PXP. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 52-2. PXP Clocks

Clock name	Clock Root	Description
clk	ipg_clk_root	PXP clock

52.3.6 Interrupts

The PXP consists of the following interrupts. Refer to the STAT register for details.

- Current PXP status interrupt (STAT[IRQ])
- LUT DMA transfer interrupt (STAT[LUT_DMA_LOAD_DONE_IRQ]
- Next Command issued interrupt (STAT[NEXT_IRQ])
- AXI ID error interrupt (STAT[AXI_ERROR_ID])
- AXI read error interrupt (STAT[AXI_READ_ERROR])
- AXI write error interrupt (STAT[AXI_WRITE_ERROR])

52.4 External Signals

PXP does not contain external signals.

52.5 Initialization

To initialize PXP, follow the steps:

1. Set CTRL[SFTRST] initially. Subsequently clear CTRL[SFTRST] and CTRL[CLKGATE] to reset PXP.
2. Configure the PXP registers
3. Set CTRL[ENABLE] = 1 to start PXP processing
4. Monitor STAT register to check if the processing is complete

Repeat steps 2-4 for the next fetch.

52.6 PXP Memory Map/Register Definition

PXP Hardware Register Format Summary

PXP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_4000	Control Register 0 (PXP_CTRL)	32	R/W	C000_0000h	52.6.1/3295
4081_4004	Control Register 0 (PXP_CTRL_SET)	32	R/W	C000_0000h	52.6.1/3295
4081_4008	Control Register 0 (PXP_CTRL_CLR)	32	R/W	C000_0000h	52.6.1/3295
4081_400C	Control Register 0 (PXP_CTRL_TOG)	32	R/W	C000_0000h	52.6.1/3295
4081_4010	Status Register (PXP_STAT)	32	R/W	0000_0000h	52.6.2/3298
4081_4014	Status Register (PXP_STAT_SET)	32	R/W	0000_0000h	52.6.2/3298
4081_4018	Status Register (PXP_STAT_CLR)	32	R/W	0000_0000h	52.6.2/3298
4081_401C	Status Register (PXP_STAT_TOG)	32	R/W	0000_0000h	52.6.2/3298
4081_4020	Output Buffer Control Register (PXP_OUT_CTRL)	32	R/W	0000_0000h	52.6.3/3300
4081_4024	Output Buffer Control Register (PXP_OUT_CTRL_SET)	32	R/W	0000_0000h	52.6.3/3300
4081_4028	Output Buffer Control Register (PXP_OUT_CTRL_CLR)	32	R/W	0000_0000h	52.6.3/3300
4081_402C	Output Buffer Control Register (PXP_OUT_CTRL_TOG)	32	R/W	0000_0000h	52.6.3/3300
4081_4030	Output Frame Buffer Pointer (PXP_OUT_BUF)	32	R/W	0000_0000h	52.6.4/3302

Table continues on the next page...

PXP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_4040	Output Frame Buffer Pointer #2 (PXP_OUT_BUF2)	32	R/W	0000_0000h	52.6.5/3303
4081_4050	Output Buffer Pitch (PXP_OUT_PITCH)	32	R/W	0000_0000h	52.6.6/3303
4081_4060	Output Surface Lower Right Coordinate (PXP_OUT_LRC)	32	R/W	0000_0000h	52.6.7/3304
4081_4070	Processed Surface Upper Left Coordinate (PXP_OUT_PS_ULC)	32	R/W	0000_0000h	52.6.8/3305
4081_4080	Processed Surface Lower Right Coordinate (PXP_OUT_PS_LRC)	32	R/W	0000_0000h	52.6.9/3306
4081_4090	Alpha Surface Upper Left Coordinate (PXP_OUT_AS_ULC)	32	R/W	0000_0000h	52.6.10/ 3307
4081_40A0	Alpha Surface Lower Right Coordinate (PXP_OUT_AS_LRC)	32	R/W	0000_0000h	52.6.11/ 3308
4081_40B0	Processed Surface (PS) Control Register (PXP_PS_CTRL)	32	R/W	0000_0000h	52.6.12/ 3309
4081_40B4	Processed Surface (PS) Control Register (PXP_PS_CTRL_SET)	32	R/W	0000_0000h	52.6.12/ 3309
4081_40B8	Processed Surface (PS) Control Register (PXP_PS_CTRL_CLR)	32	R/W	0000_0000h	52.6.12/ 3309
4081_40BC	Processed Surface (PS) Control Register (PXP_PS_CTRL_TOG)	32	R/W	0000_0000h	52.6.12/ 3309
4081_40C0	PS Input Buffer Address (PXP_PS_BUF)	32	R/W	0000_0000h	52.6.13/ 3311
4081_40D0	PS U/Cb or 2 Plane UV Input Buffer Address (PXP_PS_UBUF)	32	R/W	0000_0000h	52.6.14/ 3312
4081_40E0	PS V/Cr Input Buffer Address (PXP_PS_VBUF)	32	R/W	0000_0000h	52.6.15/ 3312
4081_40F0	Processed Surface Pitch (PXP_PS_PITCH)	32	R/W	0000_0000h	52.6.16/ 3313
4081_4100	PS Background Color (PXP_PS_BACKGROUND)	32	R/W	0000_0000h	52.6.17/ 3314
4081_4110	PS Scale Factor Register (PXP_PS_SCALE)	32	R/W	1000_1000h	52.6.18/ 3314
4081_4120	PS Scale Offset Register (PXP_PS_OFFSET)	32	R/W	0000_0000h	52.6.19/ 3316
4081_4130	PS Color Key Low (PXP_PS_CLRKEYLOW)	32	R/W	00FF_FFFFh	52.6.20/ 3317
4081_4140	PS Color Key High (PXP_PS_CLRKEYHIGH)	32	R/W	0000_0000h	52.6.21/ 3318
4081_4150	Alpha Surface Control (PXP_AS_CTRL)	32	R/W	0000_0000h	52.6.22/ 3318
4081_4160	Alpha Surface Buffer Pointer (PXP_AS_BUF)	32	R/W	0000_0000h	52.6.23/ 3321
4081_4170	Alpha Surface Pitch (PXP_AS_PITCH)	32	R/W	0000_0000h	52.6.24/ 3321

Table continues on the next page...

PXP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4081_4180	Overlay Color Key Low (PXP_AS_CLRKEYLOW)	32	R/W	00FF_FFFFh	52.6.25/ 3322
4081_4190	Overlay Color Key High (PXP_AS_CLRKEYHIGH)	32	R/W	0000_0000h	52.6.26/ 3322
4081_41A0	Color Space Conversion Coefficient Register 0 (PXP_CSC1_COEF0)	32	R/W	0400_0000h	52.6.27/ 3323
4081_41B0	Color Space Conversion Coefficient Register 1 (PXP_CSC1_COEF1)	32	R/W	0123_0208h	52.6.28/ 3325
4081_41C0	Color Space Conversion Coefficient Register 2 (PXP_CSC1_COEF2)	32	R/W	079B_076Ch	52.6.29/ 3325
4081_4320	PXP Power Control Register (PXP_POWER)	32	R/W	0000_0000h	52.6.30/ 3326
4081_4400	Next Frame Pointer (PXP_NEXT)	32	R/W	0000_0000h	52.6.31/ 3327
4081_4440	PXP Alpha Engine A Control Register. (PXP_PORTER_DUFF_CTRL)	32	R/W	0000_0000h	52.6.32/ 3329

52.6.1 Control Register 0 (PXP_CTRL*n*)

The CTRL register contains controls for the PXP module.

PXP_CTRL: 0x000

PXP_CTRL_SET: 0x004

PXP_CTRL_CLR: 0x008

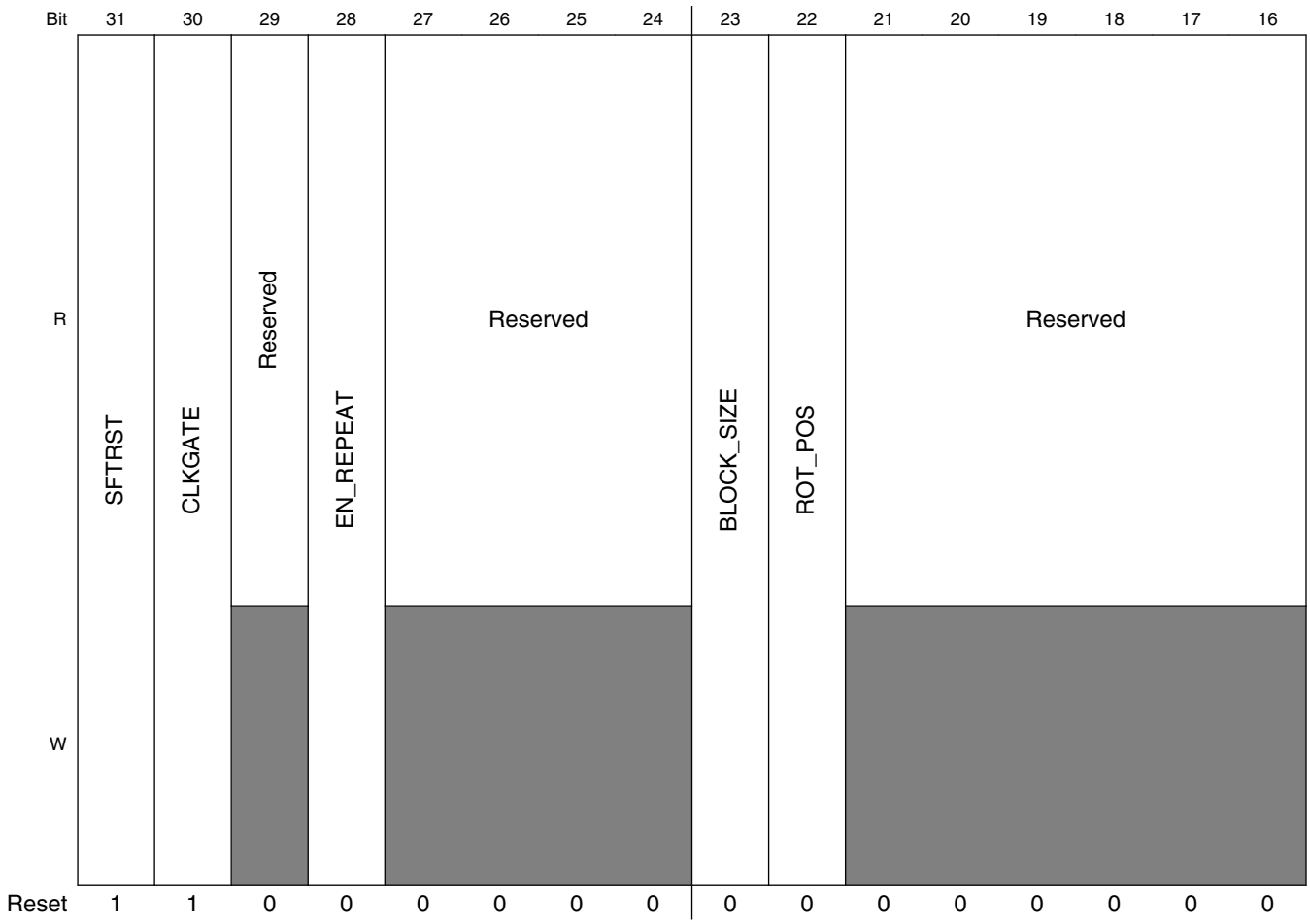
PXP_CTRL_TOG: 0x00C

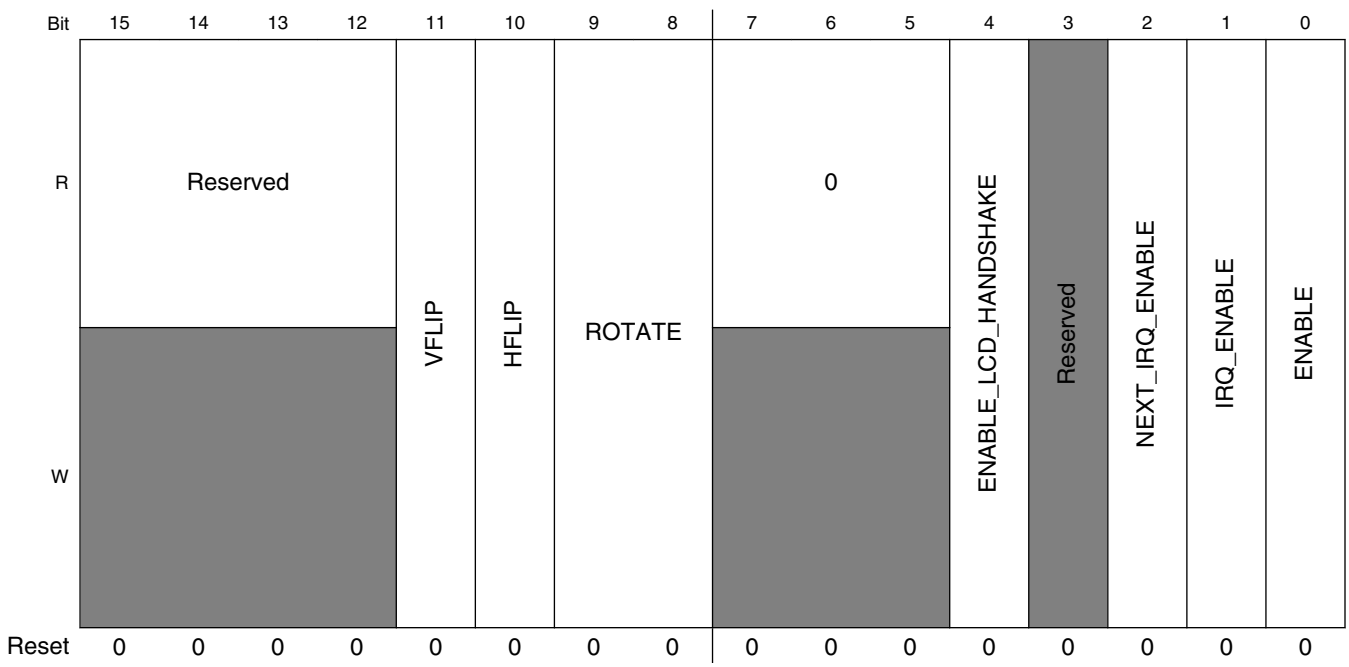
The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 4081_4000h base + 0h offset + (4d × i), where i=0d to 3d





PXP_CTRL*n* field descriptions

Field	Description
31 SFTRST	This bit can be turned on and then off to reset the PXP block to its default state. 0x0 **Enabled** — Normal PXP operation is enabled 0x1 **Disabled** — Clocking with PXP is disabled and held in its reset (lowest power) state. This is the default value.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block. 0x0 **Normal** — Normal operation 0x1 **Gated** — All clocks to PXP is gated-off
29 Reserved	This field is reserved. Always set to zero.
28 EN_REPEAT	Enable the PXP to run continuously. This bit should be set when the LCDIF handshake mode is enabled so that the next frame is automatically generated for the next screen refresh cycle. If it not set and the handshake mode is enabled, the CPU will have to initiate the PXP for the next refresh cycle. When the PXP NEXT feature is used, it has priority over the REPEAT mode, in that the new register settings are fetched first, and then the next PXP operation will continue. 0x1 **Repeat** — PXP will repeat based on the current configuration register settings 0x0 **Complete** — PXP will complete the process and enter the idle state ready to accept the next frame to be processed
27–24 Reserved	This field is reserved. Always set to zero.
23 BLOCK_SIZE	Select the block size to process. 0x0 **8X8** — Process 8x8 pixel blocks. 0x1 **16X16** — Process 16x16 pixel blocks.
22 ROT_POS	This bit controls where rotation will occur in the PXP datapath. Setting this bit to 1'b0 will place the rotation resources at the output stage of the PXP data path. Image compositing will occur before pixels are processed for rotation. Setting this bit to a 1'b1 will place the rotation resources before image composition. Only the PS can be rotated in this configuration and AS will not be rotated.
21–12 Reserved	This field is reserved. Always set to zero.
11 VFLIP	Indicates that the output buffer should be flipped vertically (effect applied before rotation). 0x0 **Disabled** — Vertical Flip is disabled 0x1 **Enabled** — Vertical Flip is enabled
10 HFLIP	Indicates that the output buffer should be flipped horizontally (effect applied before rotation). 0x0 **Disabled** — Horizontal Flip is disabled 0x1 **Enabled** — Horizontal Flip is enabled
9–8 ROTATE	Indicates the clockwise rotation to be applied at the output buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. 0x0 **ROT_0** — 0x1 **ROT_90** — 0x2 **ROT_180** — 0x3 **ROT_270** —
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

PXP_CTRL*n* field descriptions (continued)

Field	Description
4 ENABLE_LCD_ HANDSHAKE	Enable handshake with LCD controller. When this is set, the PXP will not process an entire framebuffer, but will instead process rows of NxN blocks in a double-buffer handshake with the LCDIF. This enables the use of the onboard SRAM for a partial frame buffer.
3 Reserved	This field is reserved.
2 NEXT_IRQ_ ENABLE	Next command interrupt enable. When set, the PXP will issue an interrupt when a queued command initiated by a write to the PXP_NEXT register has been loaded into the PXP's registers. This interrupt also indicates that a new command may now be queued. 0x0 **Disabled** — 0x1 **Enabled** —
1 IRQ_ENABLE	Interrupt enable **NOTE:** When using the PXP_NEXT functionality to reprogram the PXP, the new value of this bit will be used and may therefore enable or disable an interrupt unintentionally. 0x1 **Enabled** — PXP interrupt is enabled 0x0 **Disabled** — PXP interrupt is disabled
0 ENABLE	Enables PXP operation with specified parameters. The ENABLE bit will remain set while the PXP is active and will be cleared after the current operation completes. Software should use the IRQ bit in the PXP_STAT when polling for PXP completion. 0x1 **Enabled** — PXP is enabled 0x0 **Disabled** — PXP is disabled

52.6.2 Status Register (PXP_STAT*n*)

The PXP Interrupt Status register provides interrupt status information.

PXP_STAT: 0x010

PXP_STAT_SET: 0x014

PXP_STAT_CLR: 0x018

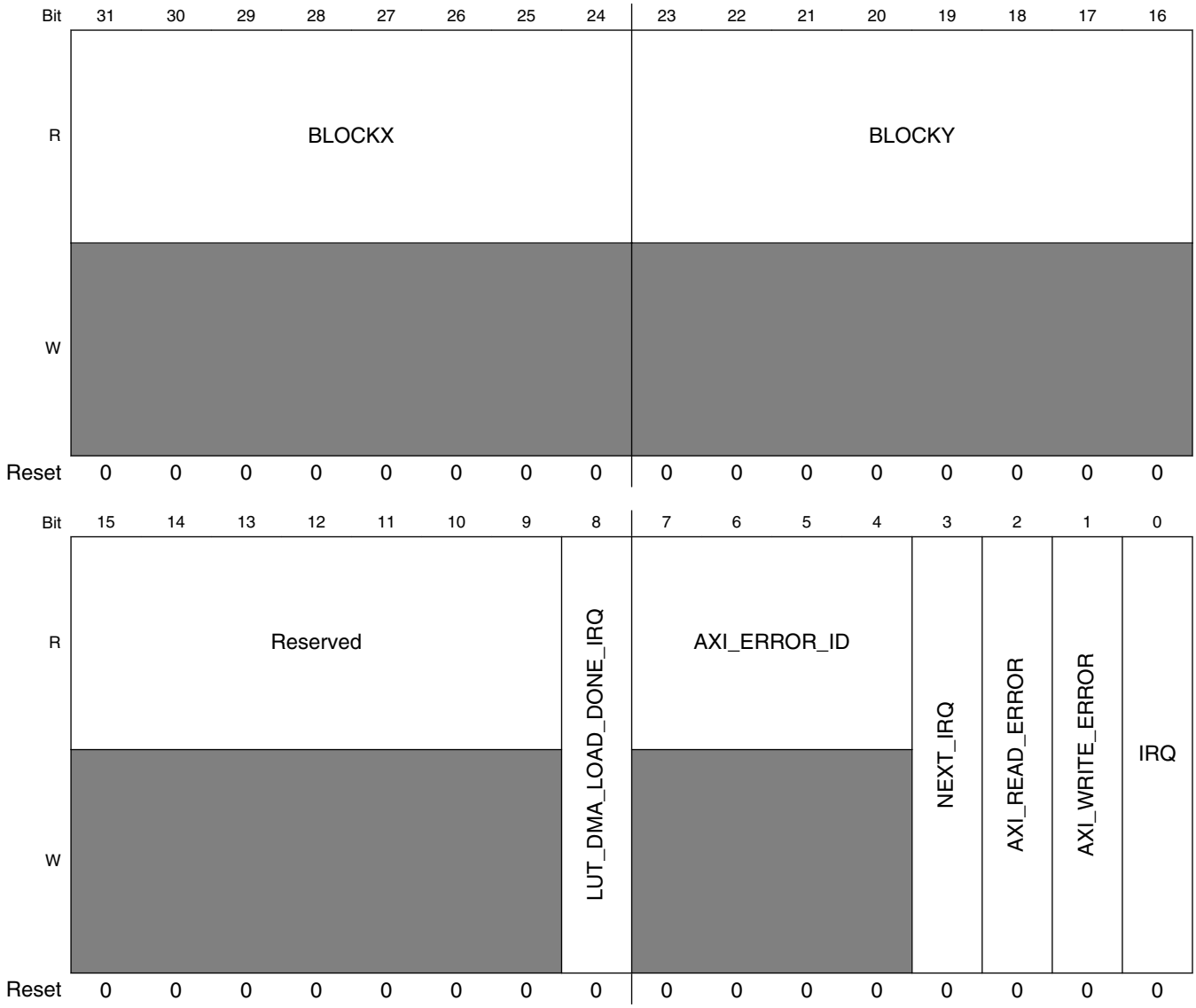
PXP_STAT_TOG: 0x01C

This register provides PXP interrupt status and the current X/Y block coordinate that is being processed.

EXAMPLE

```
PXP_STAT_CLR(BM_PXP_STAT_IRQ);  // clear CSC interrupt
```

Address: 4081_4000h base + 10h offset + (4d × i), where i=0d to 3d



PXP_STAT*n* field descriptions

Field	Description
31–24 BLOCKX	Indicates the X coordinate of the block currently being rendered.
23–16 BLOCKY	Indicates the X coordinate of the block currently being rendered.
15–9 Reserved	This field is reserved. Always set to zero.
8 LUT_DMA_ LOAD_DONE_ IRQ	Indicates that the LUT DMA transfer has completed. 0x0 **Active** — LUT DMA LOAD transfer is active 0x1 **Complete** — LUT DMA LOAD transfer is complete
7–4 AXI_ERROR_ID	Indicates the AXI ID of the failing bus operation.

Table continues on the next page...

PXP_STAT*n* field descriptions (continued)

Field	Description
3 NEXT_IRQ	Indicates that a command issued with the "Next Command" functionality has been issued and that a new command may be initiated with a write to the PXP_NEXT register.
2 AXI_READ_ ERROR	Indicates PXP encountered an AXI read error and processing has been terminated. 0x0 **Normal** — AXI read is normal 0x1 **Error** — AXI read error has occurred
1 AXI_WRITE_ ERROR	Indicates PXP encountered an AXI write error and processing has been terminated. 0x0 **Normal** — AXI write is normal 0x1 **Error** — AXI write error has occurred
0 IRQ	Indicates current PXP interrupt status. The IRQ is routed through the pxp_irq when the IRQ_ENABLE bit in the control register is set. 0x0 **No interrupt** — PXP interrupt not generated 0x1 **Interrupt generated** — PXP interrupt is generated

52.6.3 Output Buffer Control Register (PXP_OUT_CTRL*n*)

The OUT_CTRL register contains controls for the Output Buffer.

PXP_OUT_CTRL: 0x020

PXP_OUT_CTRL_SET: 0x024

PXP_OUT_CTRL_CLR: 0x028

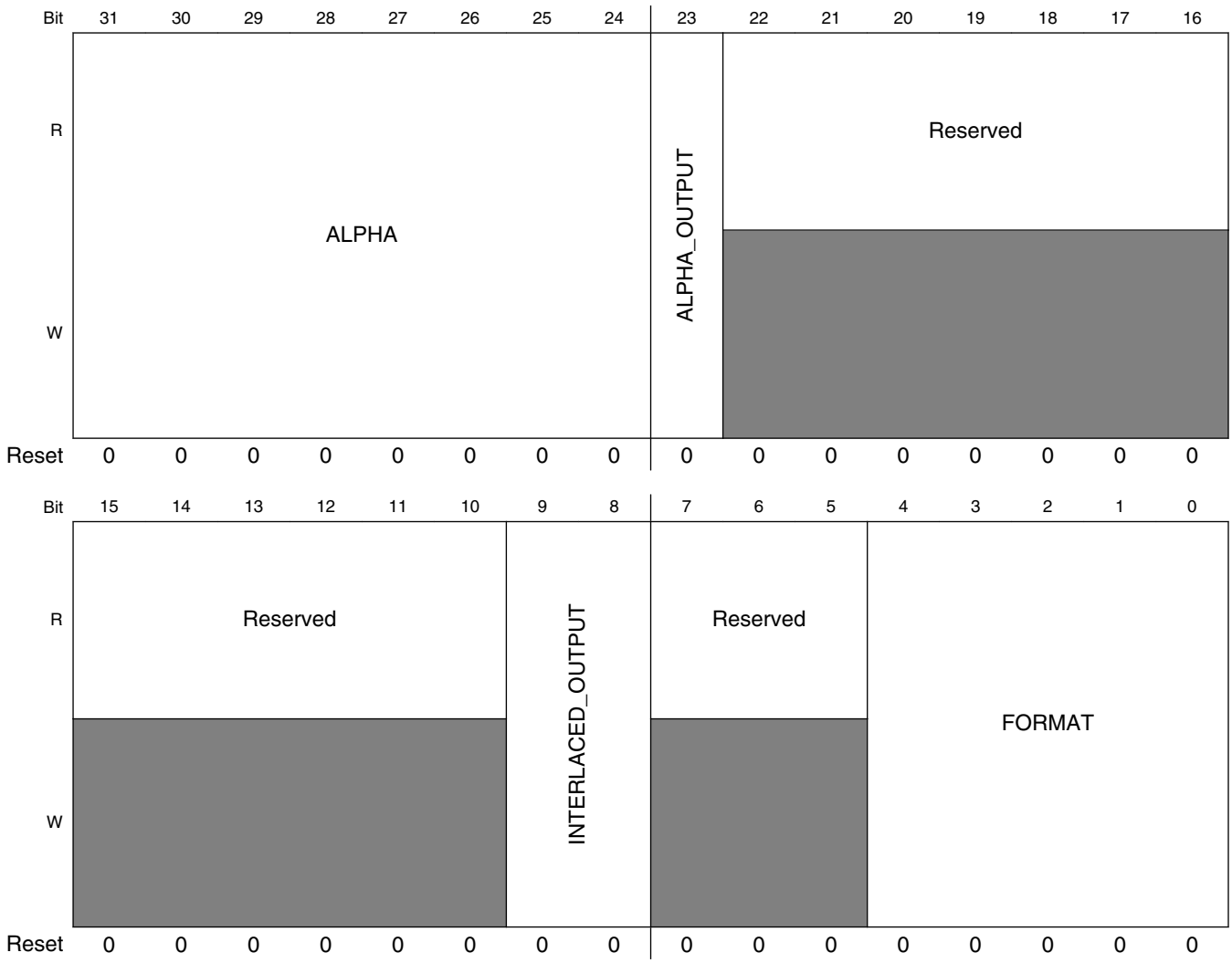
PXP_OUT_CTRL_TOG: 0x02C

The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 4081_4000h base + 20h offset + (4d × i), where i=0d to 3d



PXP_OUT_CTRL*n* field descriptions

Field	Description
31–24 ALPHA	When generating an output buffer with an alpha component, the value in this field will be used when enabled to override the alpha passed through the pixel data pipeline.
23 ALPHA_ OUTPUT	Indicates that alpha component in output buffer pixels should be overwritten by PXP_OUT_CTRL[ALPHA]. If 0, retain their alpha value from the computed alpha for that pixel. 0x0 **Retain** — 0x1 **Overwritten** —
22–10 Reserved	This field is reserved. Always set to zero.
9–8 INTERLACED_ OUTPUT	Determines how the PXP writes it's output data. Output interlacing should not be used in conjunction with input interlacing. Splitting frames into fields is most efficient using output interlacing. 2-plane output formats AND interlaced output is NOT supported. 0x0 **PROGRESSIVE** — All data written in progressive format to the OUTBUF Pointer. 0x1 **FIELD0** — Interlaced output: only data for field 0 is written to the OUTBUF Pointer.

Table continues on the next page...

PXP_OUT_CTRL*n* field descriptions (continued)

Field	Description
	0x2 **FIELD1** — Interlaced output: only data for field 1 is written to the OUTBUF2 Pointer. 0x3 **INTERLACED** — Interlaced output: data for field 0 is written to OUTBUF and data for field 1 is written to OUTBUF2.
7–5 Reserved	This field is reserved. Always set to zero.
FORMAT	Output framebuffer format. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, the YUV2P420 format should be selected when the output is YCbCr 2-plane 420 output format. 0x0 **ARGB8888** — 32-bit pixels 0x4 **RGB888** — 32-bit pixels (unpacked 24-bit pixel in 32 bit DWORD.) 0x5 **RGB888P** — 24-bit pixels (packed 24-bit format) 0x8 **ARGB1555** — 16-bit pixels 0x9 **ARGB4444** — 16-bit pixels 0xC **RGB555** — 16-bit pixels 0xD **RGB444** — 16-bit pixels 0xE **RGB565** — 16-bit pixels 0x10 **YUV1P444** — 32-bit pixels (1-plane XYUV unpacked) 0x12 **UYVY1P422** — 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 0x13 **VYUY1P422** — 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) 0x14 **Y8** — 8-bit monochrome pixels (1-plane Y luma output) 0x15 **Y4** — 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 0x18 **YUV2P422** — 16-bit pixels (2-plane UV interleaved bytes) 0x19 **YUV2P420** — 16-bit pixels (2-plane UV) 0x1A **YVU2P422** — 16-bit pixels (2-plane VU interleaved bytes) 0x1B **YVU2P420** — 16-bit pixels (2-plane VU)

52.6.4 Output Frame Buffer Pointer (PXP_OUT_BUF)

Output Framebuffer Pointer. This register points to the beginning of the output frame buffer. This pointer is used for progressive format and field 0 when generating interlaced output.

This register is used by the logic to point to the current output location for the output frame buffer.

EXAMPLE

```
PXP_OUT_BUF_WR( buffer );
```

Address: 4081_4000h base + 30h offset = 4081_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_BUF field descriptions

Field	Description
ADDR	Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

52.6.5 Output Frame Buffer Pointer #2 (PXP_OUT_BUF2)

Output Framebuffer Pointer #2. This register points to the beginning of the output frame buffer for either field 1 when generating interlaced output or for the UV buffer when in YUV 2-plane output modes. Both interlaced output AND 2-plane output modes are not supported in a single PXP operation. This register is not used as the pointer to the second buffer when in LCDIF_HANDSHAKE mode.

This register is used by the logic to point to the current output location for the field 1 or UV output frame buffer.

EXAMPLE

```
PXP_OUT_BUF_WR( field0 );  // buffer for interlaced field 0
PXP_OUT_BUF2_WR( field1 ); // buffer for interlaced field 1
```

Address: 4081_4000h base + 40h offset = 4081_4040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_BUF2 field descriptions

Field	Description
ADDR	Current address pointer for the output frame buffer. The address can have any byte alignment. 64B alignment is recommended for optimal performance.

52.6.6 Output Buffer Pitch (PXP_OUT_PITCH)

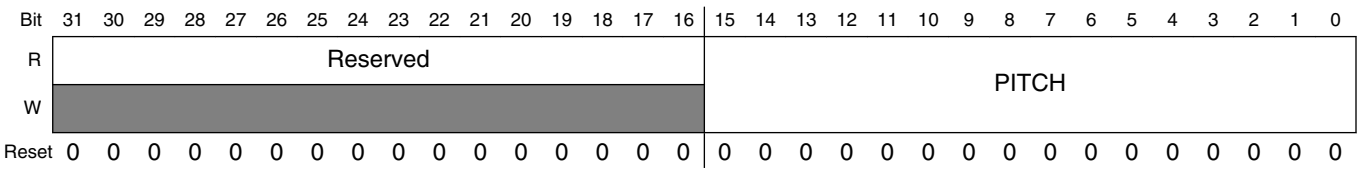
This register contains the output buffer pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in output pixel address calculations.

EXAMPLE

```
PXP_OUT_PITCH_WR( 68 * 4 );  // The output buffer pitch is 68 pixels times 32 bits per pixel
```

Address: 4081_4000h base + 50h offset = 4081_4050h



PXP_OUT_PITCH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

52.6.7 Output Surface Lower Right Coordinate (PXP_OUT_LRC)

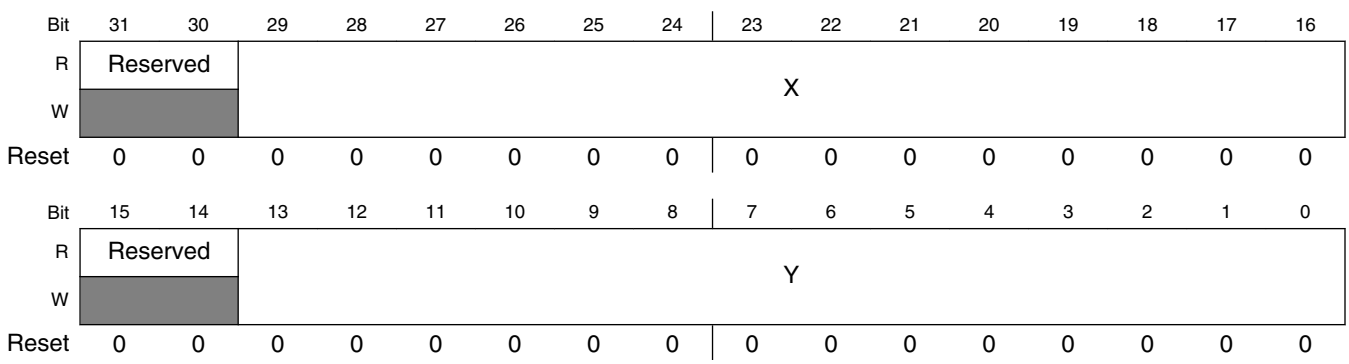
This register contains the size, or lower right coordinate, of the output buffer NOT rotated. It is implied that the upper left coordinate of the output surface is always [0,0]. When rotating the framebuffer, the PXP will automatically swap the X/Y, or WIDTH/HEIGHT, to accomodate the rotated size.

This register sets the size of the output frame buffer in pixels, not blocks. The frame buffer need not be a multiple of NxN pixels. Partial blocks will be written for output frame buffer sizes that are not divisable by N pixels in either dimension.

EXAMPLE

```
PXP_OUT_LRC[X]=319;   // set width of output frame buffer to 320 pixels
PXP_OUT_LRC[Y]=243;  // set height of output frame buffer to 244 pixels which is not
divisible by block size N

PXP_OUT_LRC_WR( BF_PXP_OUT_LRC_X(319)   |   BF_PXP_OUT_LRC_Y(243) );
```

Address: 4081_4000h base + 60h offset = 4081_4060h



PXP_OUT_LRC field descriptions

Field	Description
31–30 Reserved	This field is reserved. Always set to zero.
29–16 X	Indicates number of horizontal PIXELS in the output surface (non-rotated). The output buffer pixel width minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. The PXP will clip the pixel output at this boundary.
15–14 Reserved	This field is reserved. Always set to zero.
Y	Indicates the number of vertical PIXELS in the output surface (non-rotated). The output buffer pixel height minus 1 should be programmed. The image size is not required to be a multiple of 8 pixels. The PXP will clip the pixel output at this boundary.

52.6.8 Processed Surface Upper Left Coordinate (PXP_OUT_PS_ULC)

This register contains the upper left pixel coordinate for the Processed Surface in the OUTPUT buffer.

This register contains the upper left coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. If the value of the PXP_OUT_PS_ULC is greater than the PXP_OUT_LRC, then no PS pixels will be fetched from memory, but only PXP_PS_BACKGROUND pixels will be processed by the PS engine. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the PXP_OUT_LRC extents will use the PS to render pixels into the output buffer.

EXAMPLE

```
PXP_OUT_PS_ULC_WR(0,0x0002_0002); // Processed Surface upper left coordinate at (X,Y) =
2,2.  The PS surface will not effect pixels in the first and second row and column of the
output buffer.
```

Address: 4081_4000h base + 70h offset = 4081_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		X													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Y													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_PS_ULC field descriptions

Field	Description
31–30 Reserved	This field is reserved. Always set to zero.
29–16 X	This field indicates the upper left X-coordinate (in pixels) of the processed surface (PS) in the output buffer.
15–14 Reserved	This field is reserved. Always set to zero.
Y	This field indicates the upper left Y-coordinate (in pixels) of the processed surface in the output buffer.

52.6.9 Processed Surface Lower Right Coordinate (PXP_OUT_PS_LRC)

This register contains the lower right extent for the Processed Surface in the OUTPUT buffer.

This register contains the lower right coordinate of the Processed Surface in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the PS upper left coordinates, less than or equal to the PS lower right coordinates, and within the PXP_OUT_LRC extents will use the PS to render pixels into the output buffer.

EXAMPLE

```
PXP_OUT_PS_ULC_WR(0,0x03FF_03FF); // With this UL/LR pair of pixel coordinates, only one
pixel at OUT[X,Y]=1023,1023 will use the PS to contribute to its value.
PXP_OUT_PS_LRC_WR(0,0x03FF_03FF);
```

Address: 4081_4000h base + 80h offset = 4081_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		X													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Y													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_PS_LRC field descriptions

Field	Description
31–30 Reserved	This field is reserved. Always set to zero.
29–16 X	This field indicates the lower right X-coordinate (in pixels) of the processed surface (PS) in the output frame buffer.
15–14 Reserved	This field is reserved. Always set to zero.
Y	This field indicates the lower right Y-coordinate (in pixels) of the processed surface in the output frame buffer.

52.6.10 Alpha Surface Upper Left Coordinate (PXP_OUT_AS_ULC)

This register contains the upper left location for the Alpha Surface in the output buffer.

This register contains the upper left coordinate of AS in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are greater than or equal to the upper left coordinates will use the AS to render pixels in the output buffer.

EXAMPLE

```
PXP_OUT_AS_ULC_WR(0,0x0001_0001); // Alpha Surface upper left coordinate at (X,Y) = 1,1.
The AS surface will not effect pixels in the first row or first column of the output buffer.
```

Address: 4081_4000h base + 90h offset = 4081_4090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		X													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Y													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_AS_ULC field descriptions

Field	Description
31–30 Reserved	This field is reserved. Always set to zero.
29–16 X	This field indicates the upper left X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.

Table continues on the next page...

PXP_OUT_AS_ULC field descriptions (continued)

Field	Description
15–14 Reserved	This field is reserved. Always set to zero.
Y	This field indicates the upper left Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

52.6.11 Alpha Surface Lower Right Coordinate (PXP_OUT_AS_LRC)

This register contains the lower right extent for Alpha Surface in the output buffer.

This register contains the lower right coordinate of AS in the output frame buffer (in pixels). Values that are within the PXP_OUT_LRC X,Y extents are valid. The lowest valid value for these fields is 0,0. Pixel locations that are less than or equal to the lower right coordinates will use the AS to render pixels in the output buffer.

EXAMPLE

```
PXP_AS_LRC_WR(0,0x03FF_03FF); // Alpha Surface lower right coordinate at (X,Y) = 1023,1023.
```

Address: 4081_4000h base + A0h offset = 4081_40A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		X													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Y													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_OUT_AS_LRC field descriptions

Field	Description
31–30 Reserved	This field is reserved. Always set to zero.
29–16 X	This field indicates the lower right X-coordinate (in pixels) of the alpha surface (AS) in the output frame buffer.
15–14 Reserved	This field is reserved. Always set to zero.
Y	This field indicates the lower right Y-coordinate (in pixels) of the alpha surface in the output frame buffer.

52.6.12 Processed Surface (PS) Control Register (PXP_PS_CTRL*n*)

The PS_CTRL register contains controls for the Processed Surface Buffer.

PXP_PS_CTRL: 0x0B0

PXP_PS_CTRL_SET: 0x0b4

PXP_PS_CTRL_CLR: 0x0B8

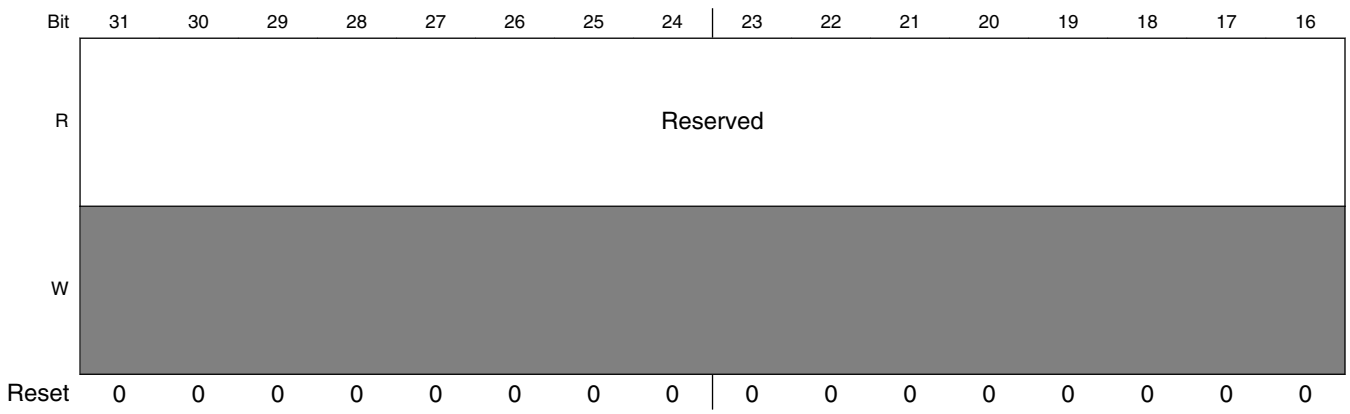
PXP_PS_CTRL_TOG: 0x0BC

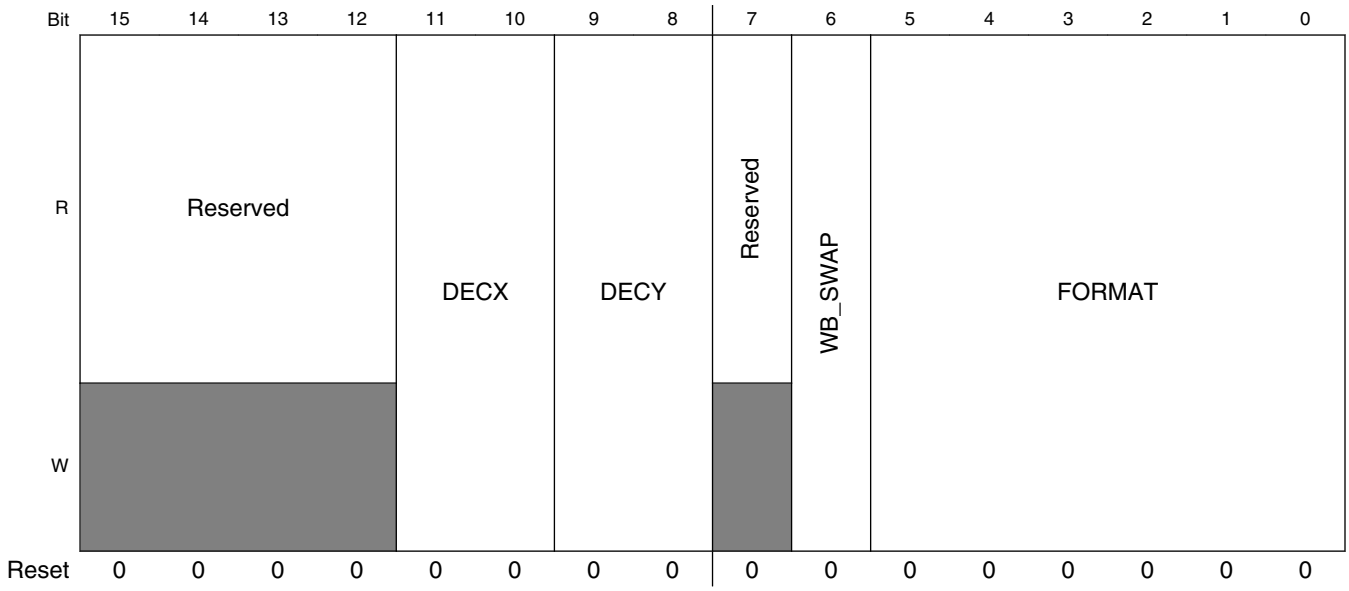
The Control register contains the primary controls for the PXP block. The present bits indicate which of the sub-features of the block are present in the hardware.

EXAMPLE

```
PXP_CTRL_SET(BM_PXP_CTRL_SFTRST);
PXP_CTRL_CLR(BM_PXP_CTRL_SFTRST | BM_PXP_CTRL_CLKGATE);
```

Address: 4081_4000h base + B0h offset + (4d × i), where i=0d to 3d





PXP_PS_CTRL*n* field descriptions

Field	Description
31–12 Reserved	This field is reserved. Always set to zero.
11–10 DECX	Horizontal pre decimation filter control. 0x0 **DISABLE** — Disable pre-decimation filter. 0x1 **DECX2** — Decimate PS by 2. 0x2 **DECX4** — Decimate PS by 4. 0x3 **DECX8** — Decimate PS by 8.
9–8 DECY	Verticle pre decimation filter control. 0x0 **DISABLE** — Disable pre-decimation filter. 0x1 **DECY2** — Decimate PS by 2. 0x2 **DECY4** — Decimate PS by 4. 0x3 **DECY8** — Decimate PS by 8.
7 Reserved	This field is reserved. Always set to zero.
6 WB_SWAP	Swap bytes in words. For each 16 bit word, the two bytes will be swapped. 0x0 **Disabled** — Byte swap is disabled 0x1 **Enabled** — Byte swap is enabled
FORMAT	PS buffer format. To select between YUV and YCbCr formats, see bit 31 of the CSC1_COEF0 register. 0x4 **RGB888_ARGB8888** — 32-bit pixels (unpacked 24-bit format with/without alpha at high 8bits) 0xC **RGB555_ARGB1555** — 16-bit pixels with/without alpha at high 1bit 0xD **RGB444_ARGB4444** — 16-bit pixels with/without alpha at high 4 bits 0xE **RGB565** — 16-bit pixels 0x10 **YUV1P444** — 32-bit pixels (1-plane XYUV unpacked) 0x12 **UYVY1P422** — 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) 0x13 **VYUY1P422** — 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes)

Table continues on the next page...

PXP_PS_CTRL*n* field descriptions (continued)

Field	Description
	0x14 **Y8** — 8-bit monochrome pixels (1-plane Y luma output) 0x15 **Y4** — 4-bit monochrome pixels (1-plane Y luma, 4 bit truncation) 0x18 **YUV2P422** — 16-bit pixels (2-plane UV interleaved bytes) 0x19 **YUV2P420** — 16-bit pixels (2-plane UV) 0x1A **YVU2P422** — 16-bit pixels (2-plane VU interleaved bytes) 0x1B **YVU2P420** — 16-bit pixels (2-plane VU) 0x1E **YUV422** — 16-bit pixels (3-plane format) 0x1F **YUV420** — 16-bit pixels (3-plane format) 0x24 **RGBA8888** — 2-bit pixels with alpha at the low 8 bits 0x2C **RGBA5551** — 16-bit pixels with alpha at the low 1bits 0x2D **RGBA4444** — 16-bit pixels with alpha at the low 4 bits

52.6.13 PS Input Buffer Address (PXP_PS_BUF)

PS Input Buffer Address. This should be programmed to the starting address of the RGB data or Y (luma) data for the PS plane.

This register contains the pointer to the Luma/RGB buffer. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_rgb); // RGB image
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 4081_4000h base + C0h offset = 4081_40C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_BUF field descriptions

Field	Description
ADDR	Address pointer for the PS RGB or Y (luma) input buffer.

52.6.14 PS U/Cb or 2 Plane UV Input Buffer Address (PXP_PS_UBUF)

PS Chroma (U/Cb/UV) Input Buffer Address. This register points to the beginning of the PS U/Cb input buffer. In two plane operation, this register points to the beginning of the PS UV chroma input buffer.

This register contains the pointer to the Chroma U/Cb or 2 plane UV buffer. This register is unused when processing 1-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 4081_4000h base + D0h offset = 4081_40D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_UBUF field descriptions

Field	Description
ADDR	Address pointer for the PS U/Cb or 2 plane UV Chroma input buffer.

52.6.15 PS V/Cr Input Buffer Address (PXP_PS_VBUF)

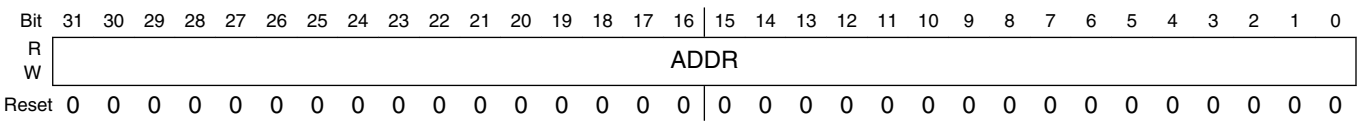
PS Chroma (V/Cr) Input Buffer Address. This register points to the beginning of the PS V/Cr input buffer. In one or two plane operation, this register is not used. In monochrome modes Y8 and Y4, the low 16 bits are used as the U/V data in the datapath instead of sourcing U/V data from external buffers. In this case, it represents a fixed value for U/V data.

This register contains the pointer to the Chroma V/Cr buffer. For Y8/Y4 modes, the low 16 bits are used as the monochrome U and V values in the data path. Bits [15:8] represent the U data byte, and bits]7:0] represent the V data byte. Other than with Y8/Y4 input buffer formats, this register is unused when processing 1 or 2-plane buffer formats. If the application requires an offset into the PS buffer, then this address can be set so that the desired offset is achieved. Any byte address is valid. For best performance, 64B alignment is recommended.

EXAMPLE

```
PXP_PS_BUF_WR(image_y);   // Y (luma) image data
PXP_PS_UBUF_WR(image_u);  // U (Cb) image data
PXP_PS_VBUF_WR(image_v);  // V (Cr) image data
```

Address: 4081_4000h base + E0h offset = 4081_40E0h



PXP_PS_VBUF field descriptions

Field	Description
ADDR	Address pointer for the PS V/Cr Chroma input buffer.

52.6.16 Processed Surface Pitch (PXP_PS_PITCH)

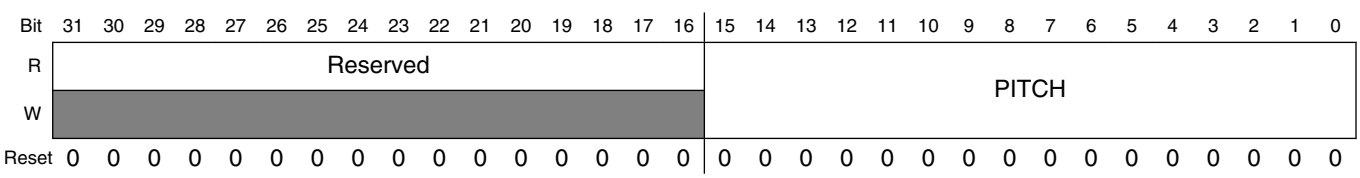
This register contains the processed surface pitch in bytes.

Any byte value will indicate the vertical pitch of the PS source frame buffer. This value will be used in PS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input PS surface. For multi-plane formats, the Y buffer pitch should be programmed. For 2-plane YUV422, the UV pitch is the same as the Y pitch. For 3-plane YUV422, the U and V pitch is 1/2 the Y pitch. For 2-plane YUV420, the UV pitch is 1/2 the Y pitch. For 3-plane YUV420, the U and V pitch is 1/4 the Y pitch. All source buffers should comply with these U and V resolution reductions with respect to their Y source buffers.

EXAMPLE

```
PXP_PS_PITCH_WR( 64 * 4 );  // The output buffer pitch is 64 pixels times 32 bits per pixel
```

Address: 4081_4000h base + F0h offset = 4081_40F0h



PXP_PS_PITCH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

52.6.17 PS Background Color (PXP_PS_BACKGROUND)

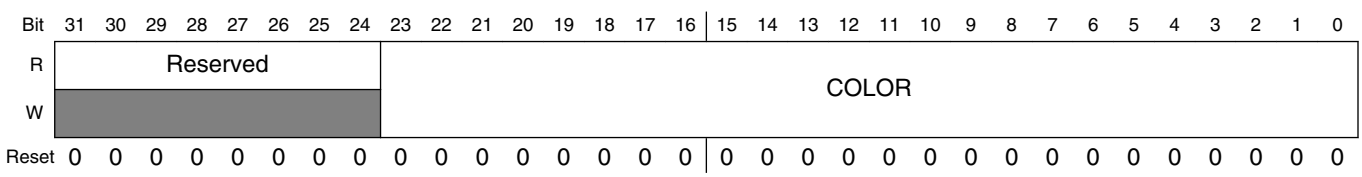
PS Background Pixel Color. This register provides a pixel value used when processing pixels outside of the region specified by the PS Coordinate registers. This value can effectively be used to set the color of the letterboxing region around the PS image.

This register contains a pixel value to be used for any PS pixels that fall outside the PS extents. This is effectively a background or letterbox color. The CSC1 control and datapath pixel format should be considered when selecting the background color.

EXAMPLE

```
PXP_PS_BACKGROUND_WR(0x00000000);  // letterbox is black
PXP_PS_BACKGROUND_WR(0x00800000);  // letterbox is dark red
PXP_PS_BACKGROUND_WR(0x00008000);  // letterbox is dark green
PXP_PS_BACKGROUND_WR(0x00000080);  // letterbox is dark blue
```

Address: 4081_4000h base + 100h offset = 4081_4100h



PXP_PS_BACKGROUND field descriptions

Field	Description
31–24 Reserved	This field is reserved. Always set to zero.
COLOR	Background color (in 24bpp format) for any pixels not within the buffer range specified by the PS ULC/ LRC.

52.6.18 PS Scale Factor Register (PXP_PS_SCALE)

PS Scale Factor. This register provides the scale factor for the PS buffer.

The maximum down scaling factor is 1/2 such that the output image in either axis is 1/2 the size of the source. The maximum up scaling factor is 2^12 for either axis. The reciprocal of the scale factor should be loaded into this register. To reduce the PS buffer by a factor of two in the output frame buffer, a value of 10.0000_0000_0000 should be loaded into this register. To scale up by a factor of 4, the value of 1/4, or 00.0100_0000_0000, should be loaded into this register. To scale up by 8/5, the value of 00.1010_0000_0000 should be loaded.

EXAMPLE

```
PXP_PS_SCALE_WR(0x10001000);  // 1:1  scaling (0x1.000)
PXP_PS_SCALE_WR(0x08000800);  // 2x   scaling (0x0.800)
PXP_PS_SCALE_WR(0x20002000);  // 1/2x scaling (0x2.000)
```

Address: 4081_4000h base + 110h offset = 4081_4110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								YSCALE							
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								XSCALE							
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

PXP_PS_SCALE field descriptions

Field	Description
31 Reserved	This field is reserved. Always set to zero.
30–16 YSCALE	This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the Y scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.
15 Reserved	This field is reserved. Always set to zero.
XSCALE	This is a two bit integer and 12 bit fractional representation (##.####_####_####) of the X scaling factor for the PS source buffer. The maximum value programmed should be 2 since scaling down by a factor greater than 2 is not supported with the bilinear filter. Decimation and the bilinear filter should be used together to achieve scaling by more than a factor of 2.

52.6.19 PS Scale Offset Register (PXP_PS_OFFSET)

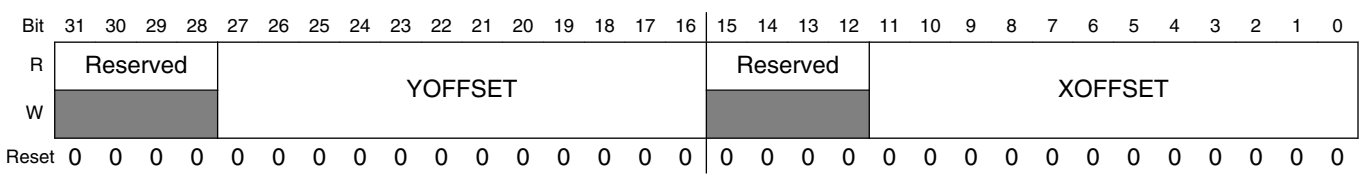
PS Scale Offset. This register provides the initial scale offset for the PS buffer.

The X and Y offset provides the ability to access the source image with a per sub-pixel granularity. This provides the capability to use all source pixels to effect the output PS image. The fixed offset values can be used for sub-pixel adjustments in the bilinear scaling filter. For example, when scaling an image down by a factor of 2, an initial offset of 0x0 would result in sub-sampling every other pixel. If a fixed offset of 0x800 (1/2), all pixels are used in scaling the final output pixel value. In this case, the first output pixel would be the sum of (1/2*P0) + (1/2*P1). This fixed offset is applied after the decimation filter stage, and before the bilinear filter stage.

EXAMPLE

```
PXP_PS_SCALE_WR(0x2000_2000);   // 1/2x scaling (0x2.000)
PXP_PS_OFFSET_WR(0x0800_0800);  // half-pixel offset in both X and Y to ensure averaging
versus pixel decimation
```

Address: 4081_4000h base + 120h offset = 4081_4120h



PXP_PS_OFFSET field descriptions

Field	Description
31–28 Reserved	This field is reserved. Always set to zero.
27–16 YOFFSET	This is a 12 bit fractional representation (0.####_####_####) of the Y scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.
15–12 Reserved	This field is reserved. Always set to zero.
XOFFSET	This is a 12 bit fractional representation (0.####_####_####) of the X scaling offset. This represents a fixed pixel offset which gets added to the scaled address to determine source data for the scaling engine.

52.6.20 PS Color Key Low (PXP_PS_CLRKEYLOW)

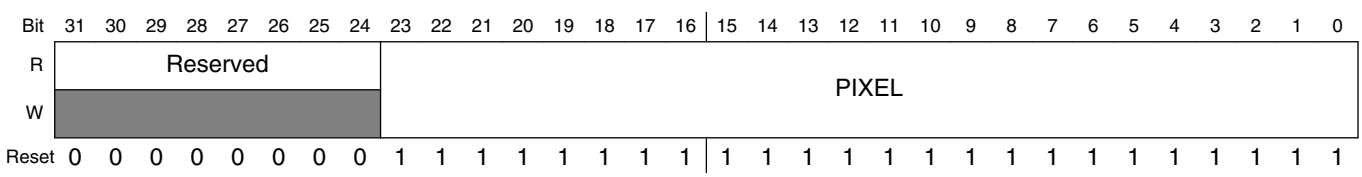
This register contains the color key low value for the PS buffer.

When processing an image, if the PXP finds a pixel in the PS buffer with a color that falls in the range between PXP_PS_CLRKEYLOW and PXP_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the PXP_PS_BACKGROUND color is passed down the pixel pipeline.

EXAMPLE

```
                                    // colorkey values between
PXP_PS_CLRKEYLOW_WR (0x008000); // medium green and
PXP_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Address: 4081_4000h base + 130h offset = 4081_4130h



PXP_PS_CLRKEYLOW field descriptions

Field	Description
31–24 Reserved	This field is reserved. Always set to zero.
PIXEL	Low range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

52.6.21 PS Color Key High (PXP_PS_CLRKEYHIGH)

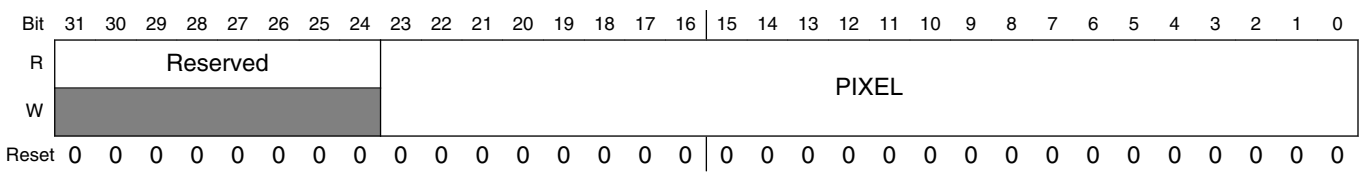
This register contains the color key high value for the PS buffer.

When processing an image, if the PXP finds a pixel in the PS buffer with a color that falls in the range between PXP_PS_CLRKEYLOW and PXP_PS_CLRKEYHIGH, it will insert the pixel from the AS channel. If the current AS pixel is letterboxed or if the AS also matches its colorkey range, the PXP_PS_BACKGROUND color is passed down the pixel pipeline.

EXAMPLE

```
                                    // colorkey values between
PXP_PS_CLRKEYLOW_WR (0x008000); // medium green and
PXP_PS_CLRKEYHIGH_WR(0x00FF00); // light green
```

Address: 4081_4000h base + 140h offset = 4081_4140h



PXP_PS_CLRKEYHIGH field descriptions

Field	Description
31–24 Reserved	This field is reserved. Always set to zero.
PIXEL	High range of color key applied to PS buffer. To disable PS colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

52.6.22 Alpha Surface Control (PXP_AS_CTRL)

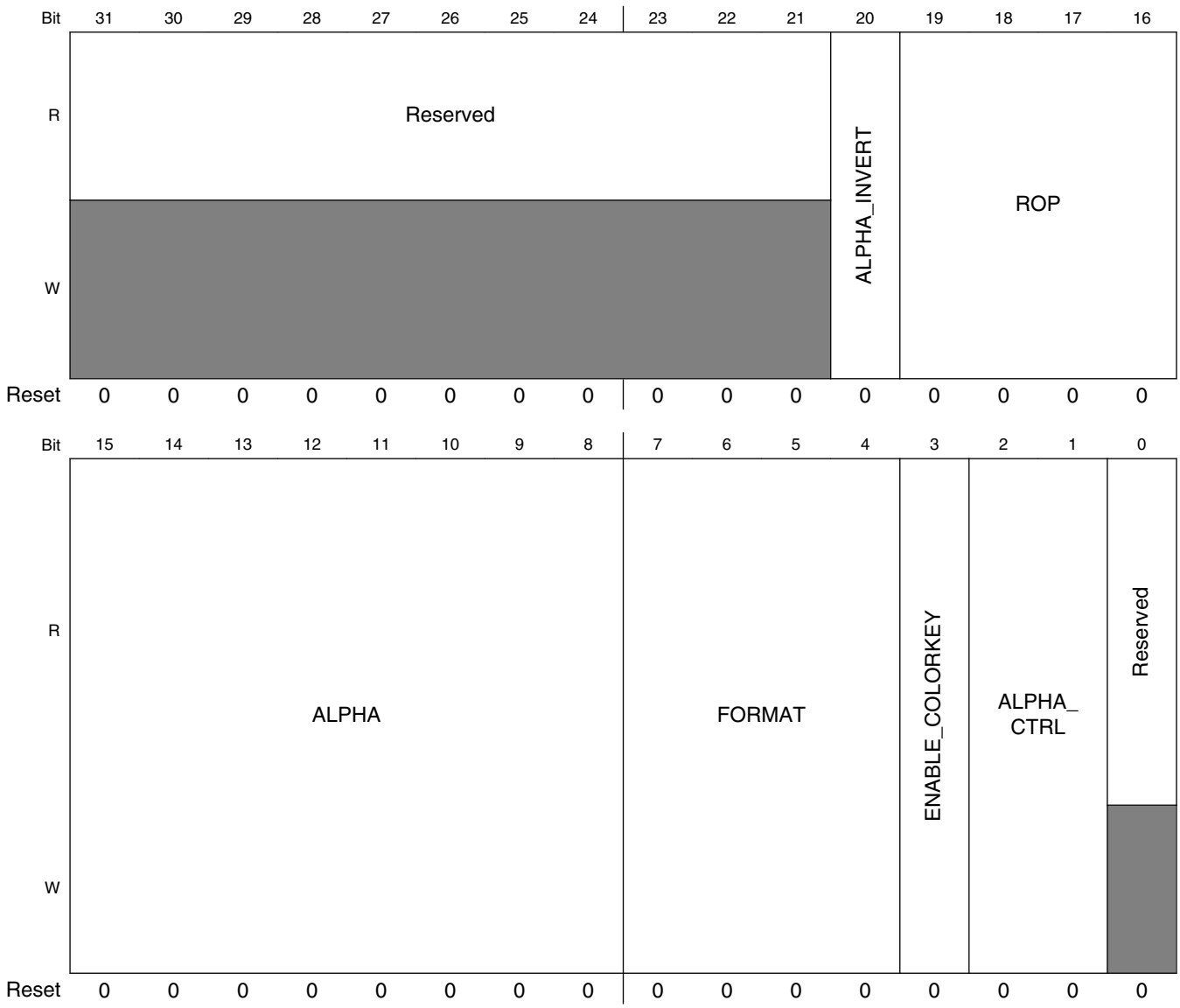
This register contains buffer control for the Alpha Surface 0 input buffer.

The Alpha Surface Parameter register provides additional controls for AS.

EXAMPLE

```
u32 asparam;
      asparam  = BF_PXP_ASPARAM_ENABLE    (1);
      asparam |= BF_PXP_ASPARAM_ALPHA_CTRL(BV_PXP_ASPARAM_ALPHA_CTRL__ROPs);
      asparam |= BF_PXP_ASPARAM_FORMAT    (BV_PXP_ASPARAM_FORMAT__ARGB8888);
      asparam |= BF_PXP_ASPARAM_ROP       (BV_PXP_ASPARAM_ROP__XORAS);
PXP_ASPARAM_WR(0,asparam);  // enable alpha surface to perform XOR ROP using RGB8888 AS
pixel format
```

Address: 4081_4000h base + 150h offset = 4081_4150h



PXP_AS_CTRL field descriptions

Field	Description
31–21 Reserved	This field is reserved. Always set to zero.
20 ALPHA_INVERT	Setting this bit to logic 0 will not alter the alpha value. A logic 1 will invert the alpha value and apply (1-alpha) for image composition. 0x0 **Not inverted** — 0x1 **Inverted** —
19–16 ROP	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CTRL field. 0x0 **MASKAS** — AS AND PS 0x1 **MASKNOTAS** — nAS AND PS

Table continues on the next page...

PXP_AS_CTRL field descriptions (continued)

Field	Description
	0x2 **MASKASNOT** — AS AND nPS 0x3 **MERGEAS** — AS OR PS 0x4 **MERGENOTAS** — nAS OR PS 0x5 **MERGEASNOT** — AS OR nPS 0x6 **NOTCOPYAS** — nAS 0x7 **NOT** — nPS 0x8 **NOTMASKAS** — AS NAND PS 0x9 **NOTMERGEAS** — AS NOR PS 0xA **XORAS** — AS XOR PS 0xB **NOTXORAS** — AS XNOR PS
15–8 ALPHA	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE values are programmed in PXP_AS_CTRL[ALPHA_CTRL]. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when selected.
7–4 FORMAT	Indicates the input buffer format for AS. 0x0 **ARGB8888** — 32-bit pixels with alpha 0x1 **RGBA888** — 2-bit pixel with alpha at low 8 bits 0x4 **RGB888** — 32-bit pixels without alpha (unpacked 24-bit format) 0x8 **ARGB1555** — 16-bit pixels with alpha 0x9 **ARGB4444** — 16-bit pixels with alpha 0xA **RGBA5551** — 16-bit pixel with alpha at low 1 bit 0xB **RGBA4444** — 16-bit pixel with alpha at low 4 bits 0xC **RGB555** — 16-bit pixels without alpha 0xD **RGB444** — 16-bit pixels without alpha 0xE **RGB565** — 16-bit pixels without alpha
3 ENABLE_ COLORKEY	Indicates that colorkey functionality is enabled for this alpha surface. Pixels found in the alpha surface colorkey range will be displayed as transparent (the PS pixel will be used). 0x0 **Disabled** — 0x1 **Enabled** —
2–1 ALPHA_CTRL	Determines how the alpha value is constructed for this alpha surface. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 0x0 **Embedded** — Indicates that the AS pixel alpha value will be used to blend the AS with PS. The ALPHA field is ignored. 0x1 **Override** — Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. 0x2 **Multiply** — Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. 0x3 **ROPs** — Enable ROPs. The ROP field indicates an operation to be performed on the alpha surface and PS pixels.
0 Reserved	This field is reserved. Always set to zero.

52.6.23 Alpha Surface Buffer Pointer (PXP_AS_BUF)

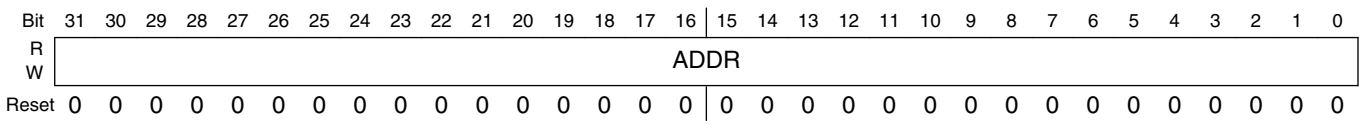
Alpha Surface 0 Buffer Address Pointer. This register points to the beginning of the Alpha Surface 0 input buffer.

This register is used to indicate the base address of the AS buffer.

EXAMPLE

```
u32* alpha_ptr;
PXP_ASn_WR(0,alpha_ptr);
```

Address: 4081_4000h base + 160h offset = 4081_4160h



PXP_AS_BUF field descriptions

Field	Description
ADDR	Address pointer for the alpha surface 0 buffer.

52.6.24 Alpha Surface Pitch (PXP_AS_PITCH)

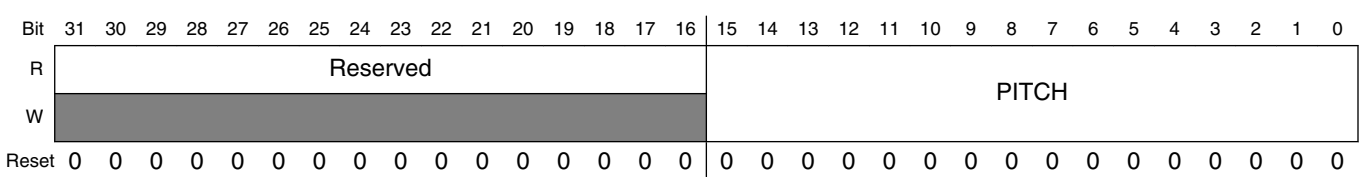
This register contains the alpha surface pitch in bytes.

Any byte value will indicate the vertical pitch. This value will be used in AS pixel address calculations. This value has no relation to the UL and LR registers. It specifies how many bytes are between two vertically adjacent pixels in the input AS surface.

EXAMPLE

```
PXP_AS_PITCH_WR( 1920 * 4 );  // The output buffer pitch is HD resolution at 32 bits per
pixel
```

Address: 4081_4000h base + 170h offset = 4081_4170h



PXP_AS_PITCH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Always set to zero.
PITCH	Indicates the number of bytes in memory between two vertically adjacent pixels.

52.6.25 Overlay Color Key Low (PXP_AS_CLRKEYLOW)

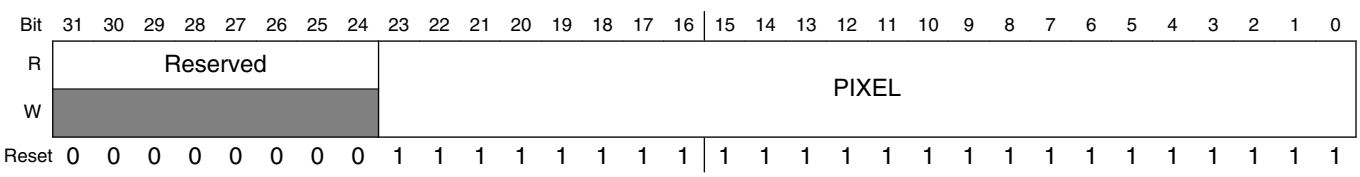
This register contains the color key low value for the AS buffer.

When processing an image, the if the PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

EXAMPLE

```
                                    // colorkey values between
PXP_AS_CLRKEYLOW_WR (0x000000); // black and
PXP_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Address: 4081_4000h base + 180h offset = 4081_4180h



PXP_AS_CLRKEYLOW field descriptions

Field	Description
31–24 Reserved	This field is reserved. Always set to zero.
PIXEL	Low range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

52.6.26 Overlay Color Key High (PXP_AS_CLRKEYHIGH)

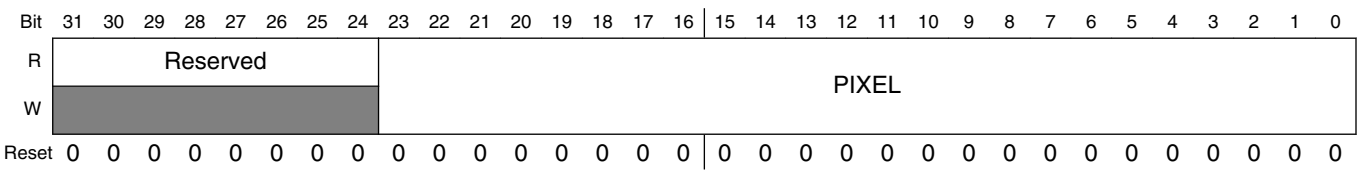
This register contains the color key high value for the AS buffer.

When processing an image, the if the PXP finds a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. If no PS image is present or if the PS image also matches its colorkey range, the PS background color is used. Colorkey operations are higher priority than alpha or ROP operations.

EXAMPLE

```
                                    // colorkey values between
PXP_AS_CLRKEYLOW_WR (0x000000); // black and
PXP_AS_CLRKEYHIGH_WR(0x800000); // medium red
```

Address: 4081_4000h base + 190h offset = 4081_4190h



PXP_AS_CLRKEYHIGH field descriptions

Field	Description
31–24 Reserved	This field is reserved. Always set to zero.
PIXEL	High range of RGB color key applied to AS buffer. Each overlay has an independent colorkey enable.

52.6.27 Color Space Conversion Coefficient Register 0 (PXP_CSC1_COEF0)

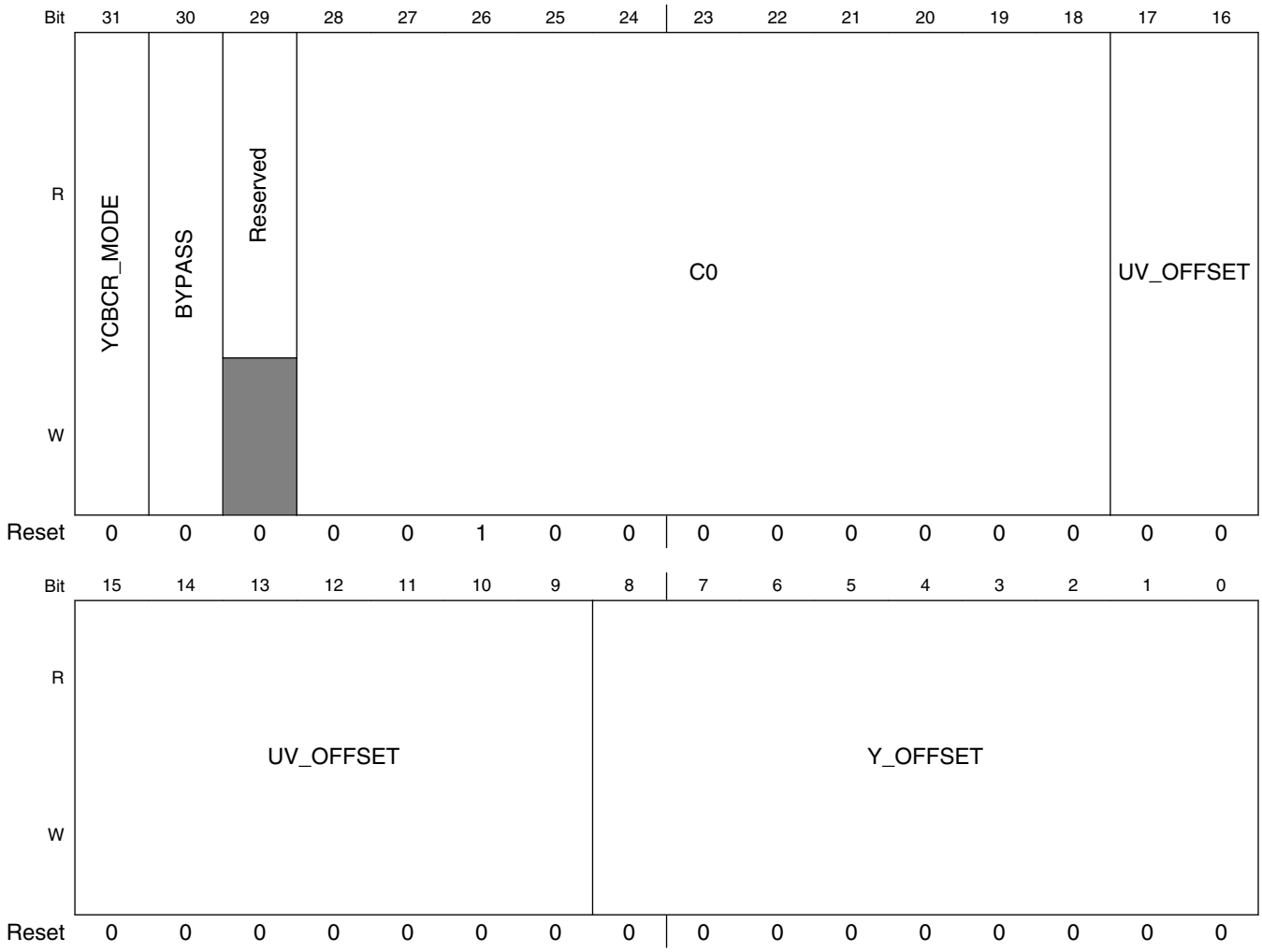
This register contains color space conversion coefficients in two's compliment notation.

The coefficient 0 register contains coefficients used in the color space conversion algorithm. The Y and UV offsets are added to the source buffer to normalize them before the conversion. C0 is the coefficient that is used to multiply the luma component of the data for all three RGB components.

EXAMPLE

```
//  The equations used for Colorspace conversion are:
//     R = C0*(Y+YOFFSET)                    + C1(V+UV_OFFSET)
//     G = C0*(Y+YOFFSET) + C3(U+UV_OFFSET) + C2(V+UV_OFFSET)
//     R = C0*(Y+YOFFSET) + C4(U+UV_OFFSET)

   PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
   PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
   PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 4081_4000h base + 1A0h offset = 4081_41A0h



PXP_CSC1_COEF0 field descriptions

Field	Description
31 YCBCR_MODE	Set to 1 when performing YCbCr conversion to RGB. This bit changes the behavior of the scaler when performing U/V scaling. 0 YUV to RGB 1 YCbCr to RGB
30 BYPASS	Bypass the CSC unit in the scaling engine. When set to logic 1, bypass is enabled and the output pixels will be in the YUV/YCbCr color space. When set to logic 0, the CSC unit is enabled and the pixels will be converted based on the programmed coefficients.
29 Reserved	This field is reserved. Always set to zero.
28–18 C0	Two's compliment Y multiplier coefficient. YUV=0x100 (1.000) YCbCr=0x12A (1.164)
17–9 UV_OFFSET	Two's compliment phase offset implicit for CbCr data. Generally used for YCbCr to RGB conversion. YCbCr=0x180, YUV=0x000 (typically -128 or 0x180 to indicate normalized -0.5 to 0.5 range)
Y_OFFSET	Two's compliment amplitude offset implicit in the Y data. For YUV, this is typically 0 and for YCbCr, this is typically -16 (0x1F0)

52.6.28 Color Space Conversion Coefficient Register 1 (PXP_CSC1_COEF1)

This register contains color space conversion coefficients in two's compliment notation.

The Coeffient 1 register contains coefficients used in the color space conversion algorithm. C1 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the red component. C4 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the blue component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

EXAMPLE

```
PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 4081_4000h base + 1B0h offset = 4081_41B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					C1											Reserved					C4											
W																																
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

PXP_CSC1_COEF1 field descriptions

Field	Description
31–27 Reserved	This field is reserved. Always set to zero.
26–16 C1	Two's compliment Red V/Cr multiplier coefficient. YUV=0x123 (1.140) YCbCr=0x198 (1.596)
15–11 Reserved	This field is reserved. Always set to zero.
C4	Two's compliment Blue U/Cb multiplier coefficient. YUV=0x208 (2.032) YCbCr=0x204 (2.017)

52.6.29 Color Space Conversion Coefficient Register 2 (PXP_CSC1_COEF2)

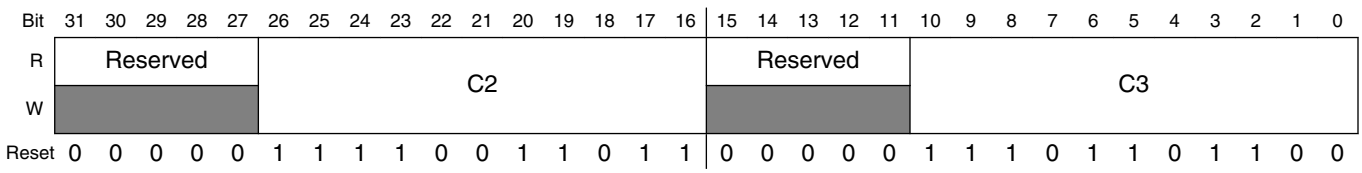
This register contains color space conversion coefficients in two's compliment notation.

The Coefficient 2 register contains coefficients used in the color space conversion algorithm. C2 is the coefficient that is used to multiply the chroma (Cr/V) component of the data for the green component. C3 is the coefficient that is used to multiply the chroma (Cb/U) component of the data for the green component. Both values should be coded as a two's compliment fixed point number with 8 bits right of the decimal.

EXAMPLE

```
// NOTE: The default values for the CSCCOEF2 register are incorrect.  C2 should be 0x76B and
C3 should be 0x79C for proper operation.

   PXP_CSCCOEF0_WR(0x04030000); // YUV coefficients: C0, Yoffset, UVoffset
   PXP_CSCCOEF1_WR(0x01230208); // YUV coefficients: C1, C4
   PXP_CSCCOEF2_WR(0x076B079b); // YUV coefficients: C2, C3
```

Address: 4081_4000h base + 1C0h offset = 4081_41C0h



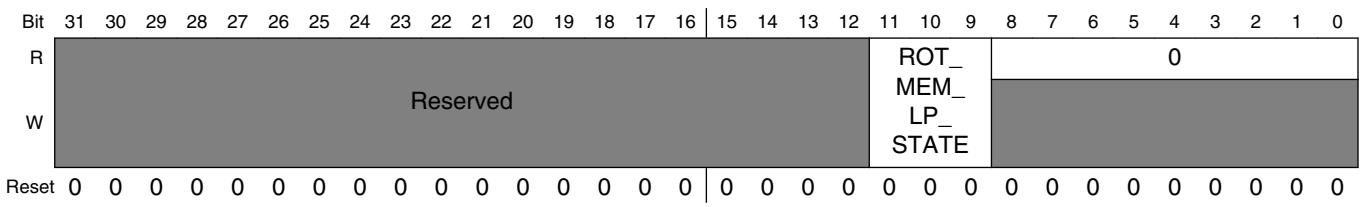
PXP_CSC1_COEF2 field descriptions

Field	Description
31–27 Reserved	This field is reserved. Always set to zero.
26–16 C2	Two's complement Green V/Cr multiplier coefficient. YUV=0x76B (-0.581) YCbCr=0x730 (-0.813)
15–11 Reserved	This field is reserved. Always set to zero.
C3	Two's complement Green U/Cb multiplier coefficient. YUV=0x79C (-0.394) YCbCr=0x79C (-0.392)

52.6.30 PXP Power Control Register (PXP_POWER)

This register controls power states for PXP memories.

Address: 4081_4000h base + 320h offset = 4081_4320h



PXP_POWER field descriptions

Field	Description
31–12 Reserved	This field is reserved.
11–9 ROT_MEM_LP_STATE	Select the low power state of the Rotation (ROT) memory. 0x0 **NONE** — Memory is not in low power state. 0x1 **LS** — Light Sleep Mode. Low leakage mode, maintain memory contents. 0x2 **DS** — Deep Sleep Mode. Low leakage mode, maintain memory contents. 0x4 **SD** — Shut Down Mode. Shut Down periphery and core, no memory retention.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.6.31 Next Frame Pointer (PXP_NEXT)

This register contains a pointer to a data structure used to reload the PXP registers at the end of the current frame.

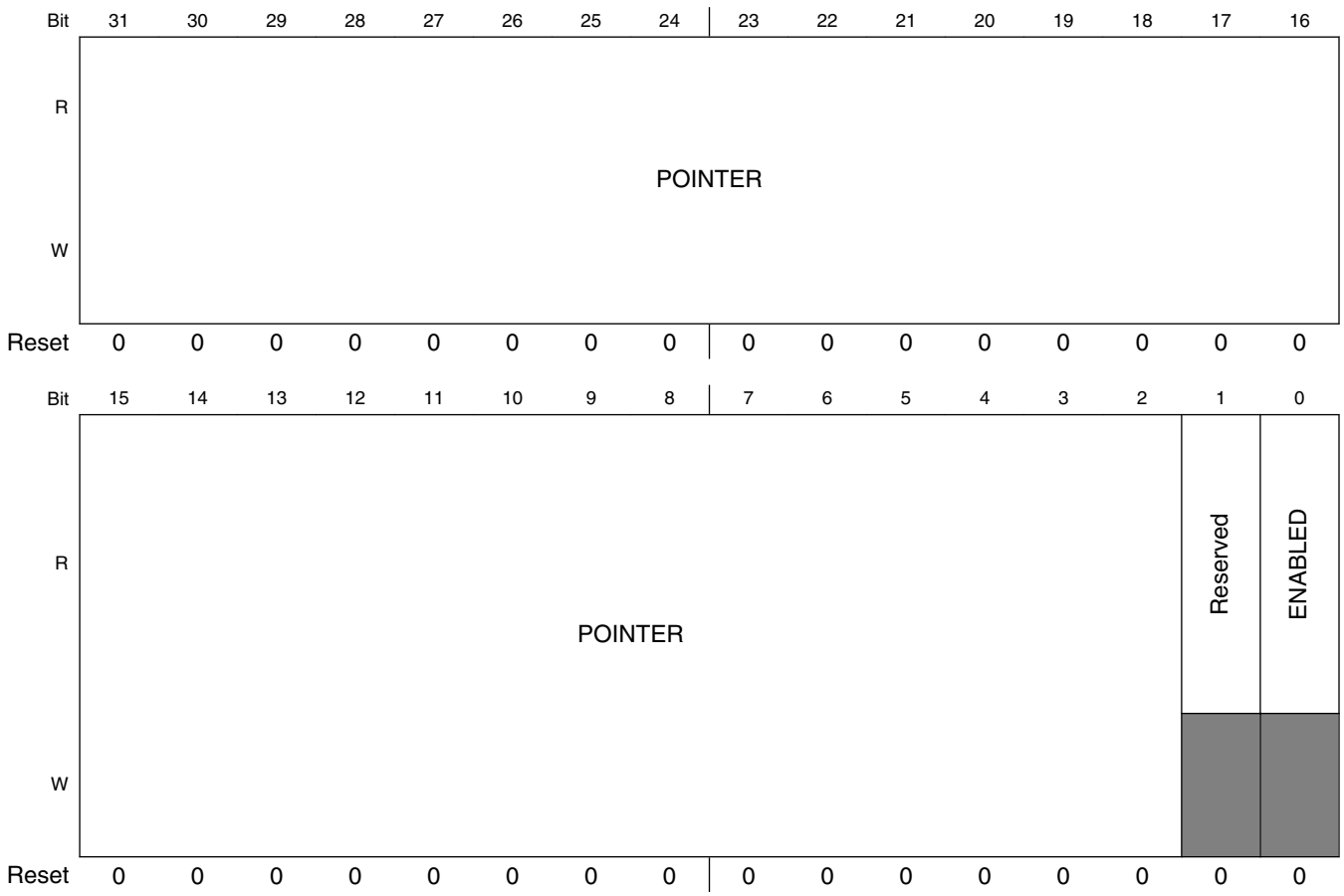
To enable this functionality, software must write this register while the PXP is processing the current data frame (if the PXP is currently idle, this will also initiate an immediate load of registers from the pointer). The process of writing this register (WRITE operation) will set a semaphore in hardware to notify the control logic that a register reload operation must be performed when the current frame processing is complete. At the end of a frame, the PXP will fetch the register settings from this location, signal an interrupt to software, then proceed with rendering the next frame of data. Software may cancel the reload operation by issuing a CLEAR operation to this register. SET and TOGGLE operations should not be used when addressing this register. All registers will be reloaded with the exception of the following: STAT, CSCCOEFn, NEXT. All other registers will be loaded in the order they appear in the register map. After the pointer's contents have been loaded into the PXP's registers, the NEXT_IRQ interrupt will be issued (see the PXP_STATUS register).

EXAMPLE

```
// create register command structure in memory
  u32* pxp_commands0[48], pxp_commands1;
  u32  rc;

// initialize control structure for frame 0
  pxp_commands0[0] = ...; // CTRL
  pxp_commands0[1] = ...; // OUT Buffer
  ...
  pxp_commands0[47] = ..; // Overlay7 param2

// initialize control structure for frame 1
  pxp_commands1[0] = ...; // CTRL
  pxp_commands1[1] = ...; // OUT Buffer
  ...
```

```
  pxp_commands1[47] = ..; // Overlay7 param2

// poll until a command isn't queued
  while (rc=PXP_NEXT_RD() & BM_PXP_NEXT_ENABLED );
  PXP_NEXT_WR(pxp_commands0);  // enable PXP operation 0 via command pointer

// poll until first command clears
  while (rc=PXP_NEXT_RD() & BM_PXP_NEXT_ENABLED );
  PXP_NEXT_WR(pxp_commands1);  // enable PXP operation 1 via command pointer
```

Address: 4081_4000h base + 400h offset = 4081_4400h

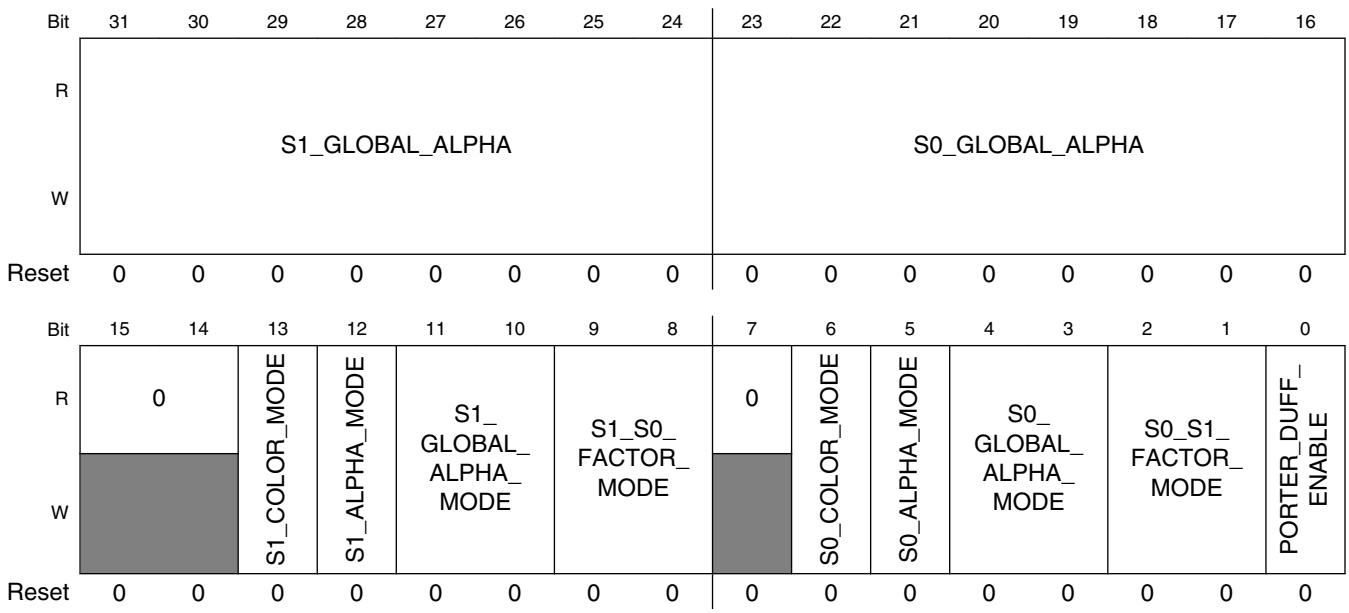


PXP_NEXT field descriptions

Field	Description
31–2 POINTER	A pointer to a data structure containing register values to be used when processing the next frame. The pointer must be 32-bit aligned and should reside in on-chip or off-chip memory.
1 Reserved	This field is reserved. Always set to zero.
0 ENABLED	Indicates that the "next frame" functionality has been enabled. This bit reflects the status of the hardware semaphore indicating that a reload operation is pending at the end of the current frame.

52.6.32 PXP Alpha Engine A Control Register. (PXP_PORTER_DUFF_CTRL)

Address: 4081_4000h base + 440h offset = 4081_4440h



PXP_PORTER_DUFF_CTRL field descriptions

Field	Description
31–24 S1_GLOBAL_ALPHA	s1 global alpha
23–16 S0_GLOBAL_ALPHA	s0 global alpha
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 S1_COLOR_MODE	s1 color mode 0x0 **Original pixel** — 0x1 **Scaled pixel** —
12 S1_ALPHA_MODE	s1 alpha mode (Porter-Duff Alpha mode) 0x0 **Straight mode** — 0x1 **Inverted mode** —
11–10 S1_GLOBAL_ALPHA_MODE	s1 global alpha mode (Porter-Duff Global Alpha mode) 0x0 **Global alpha** — 0x1 **Local alpha** —

Table continues on the next page...

PXP_PORTER_DUFF_CTRL field descriptions (continued)

Field	Description
	0x2 **Scaled alpha** — 0x3 **Scaled alpha** —
9–8 S1_S0_ FACTOR_MODE	s1 to s0 factor mode (Porter-Duff factor mode) 0x0 **1** — 0x1 **0** — 0x2 **Straight alpha** — 0x3 **Inverse alpha** —
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 S0_COLOR_ MODE	s0 color mode (Porter-Duff color mode) 0x0 **Original pixel** — 0x1 **Scaled pixel** —
5 S0_ALPHA_ MODE	s0 alpha mode (Porter-Duff alpha mode) 0x0 **Straight mode** — 0x1 **Inverted mode** —
4–3 S0_GLOBAL_ ALPHA_MODE	s0 global alpha mode 0x0 **Global alpha** — 0x1 **Local alpha** — 0x2 **Scaled alpha** — 0x3 **Scaled alpha** —
2–1 S0_S1_ FACTOR_MODE	s0 to s1 factor mode 0x0 **1** — 0x1 **0** — 0x2 **Straight alpha** — 0x3 **Inverse alpha** —
0 PORTER_ DUFF_ENABLE	Porter-Duff Enable 0x0 **Disabled** — 0x1 **Enabled** —

Chapter 53
Graphics Processing Unit (GPU2D)

53.1 Chip-specific GPU information

Table 53-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

53.2 Overview

The Graphics Processing Unit (GPU) provides high performance, low power consumption, and high quality graphics. It supports raster, vector graphics encompassing the majority of the embedded graphics use-case scenarios.

53.2.1 Block diagram

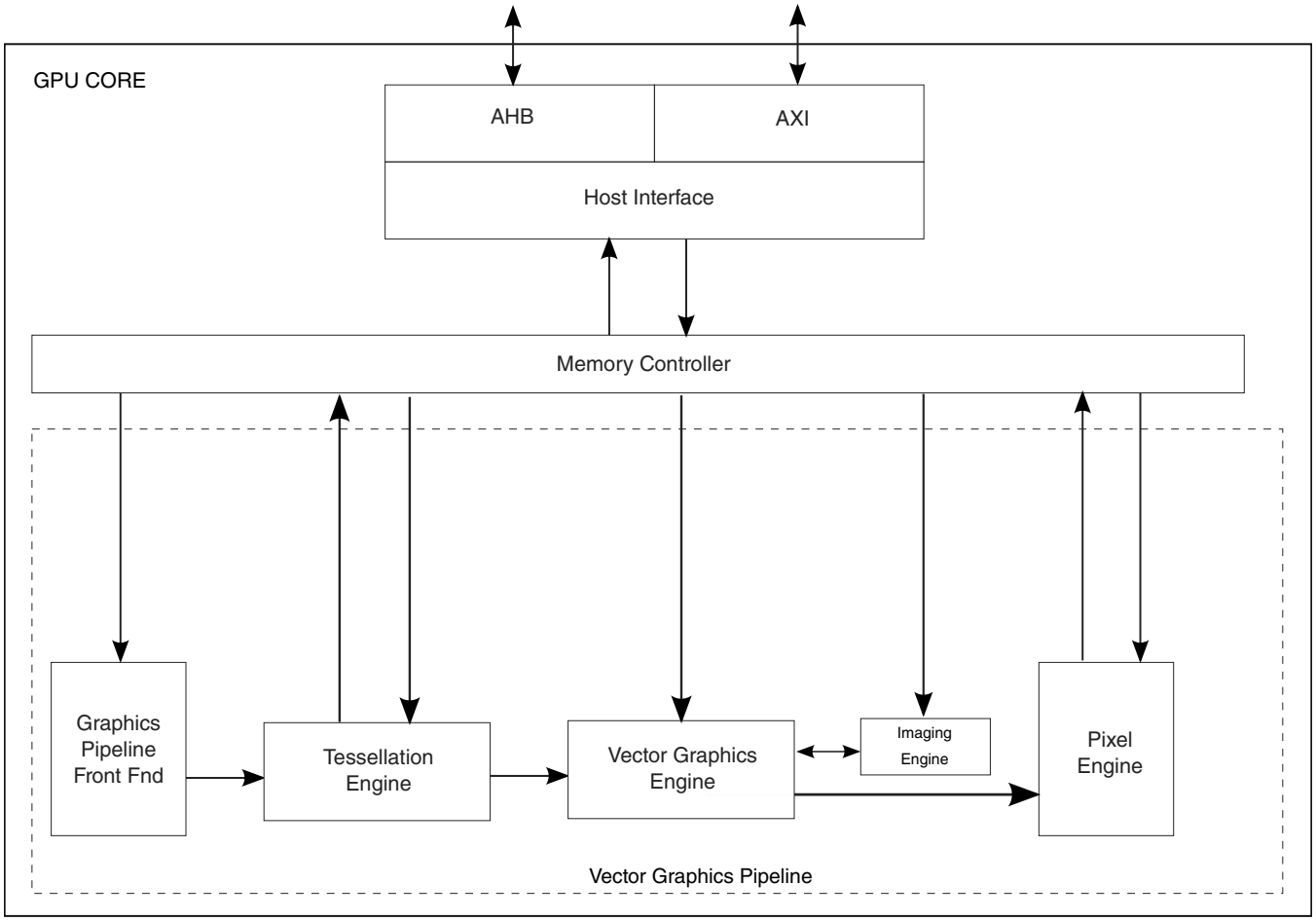


Figure 53-1. GPU block diagram

- Host Interface: Allows the GPU core to communicate with external memory and the CPU through the AXI or AHB bus. In this block, data crosses clock domain boundaries.
- Memory Controller: Internal memory management unit that controls the block-to-host memory request interface.
- Graphics Pipeline Front End: Inserts high level primitives and commands into the graphics pipeline.
- Tessellation Engine: Transforms vertices, and control points and tessellates lines, quadratic, and cubic Bezier curves.
- Vector Graphics Engine: Rasterizer that converts primitives to pixels.
- Imaging Engine: Paint and Image generator that colors each pixel.
- Pixel Engine: Renderer that combines different sources into the final pixel value.

53.2.2 Features

The following features are supported by the GPU 2D unit:

- OpenVG 1.1
- Hardware Tessellation
- High Quality 16x FSAA rendering
- Configurable Co-ordinate systems and transformations (3x3 matrix)
- Blending including 8 Porter Duff blending modes
- Rotation to any angle
- Odd/even and non-zero fill rules
- Linear and curved paint paths (linear and gradient)
- Solid and linear pattern for paints
- Supports 14 pixel formats for image interpolation
- Support for various VG input formats including A4,A8, L8, YV, NV, YUV2 Etc.

The table below describes the Graphics Hardware features:

Table 53-2. Graphics Hardware Features

Feature	Description
Coordinate systems and transformations	Image drawing uses a 3x3 perspective transformation matrix
Scissoring and viewport clipping	-
Porter-Duff blending	-
Blending	Surface (alpha) masking for variable transparency Porter Duff blending plus additional blend modes
Rotation	To any angle
Fill rules	Odd/even and non-zero
Paint paths	Linear and radial gradient paint paths
Paints	Solid, linear, radial gradient and pattern
Pixel formats for image interpolation	14 pixel formats for image interpolation, including A4/A8
Image filters and interpolations	Point, Linear and Bi-linear blends
Transformations	sRGBA color transformation

53.3 Functional description

The following sections describe functional details of the GPU2D module.

53.3.1 VG Format Support

The following tables describes the available graphic formats:

Table 53-3. VG Format Support

<table>
<tr><th>Feature</th><th colspan="4">GPU Support</th></tr>
<tr><td>Image formats

The graphics engine supports these formats for source image and destination render targets.</td><td colspan="4"><table>
<tr><th>Bit Depth</th><th>Format</th><th>Supported for Source Image</th><th>Supported for Destination</th></tr>
<tr><td>4</td><td>A4</td><td>Yes</td><td></td></tr>
<tr><td>8</td><td>A8</td><td>Yes</td><td>Yes</td></tr>
<tr><td>8</td><td>L8</td><td>Yes</td><td></td></tr>
<tr><td>16</td><td>R5G6B5</td><td>Yes</td><td>Yes</td></tr>
<tr><td>16</td><td>A1R5G5B5</td><td>Yes</td><td>Yes</td></tr>
<tr><td>16</td><td>A4R4G4B4</td><td>Yes</td><td>Yes</td></tr>
<tr><td>32</td><td>A8R8G8B8</td><td>Yes</td><td>Yes</td></tr>
<tr><td>32</td><td>X8R8G8B8</td><td>Yes</td><td>Yes</td></tr>
<tr><td></td><td>NV12</td><td>Yes</td><td></td></tr>
<tr><td></td><td>NV16</td><td>Yes</td><td></td></tr>
<tr><td></td><td>YUY2</td><td>Yes</td><td></td></tr>
<tr><td></td><td>YV12</td><td>Yes</td><td></td></tr>
<tr><td></td><td>YV16</td><td>Yes</td><td></td></tr>
<tr><td></td><td>YV24</td><td>Yes</td><td></td></tr>
</table></td></tr>
<tr><td>Color component swizzle support</td><td colspan="4">Supported color component swizzles are:

SWIZZLE_ABGR, SWIZZLE_ARGB, SWIZZLE_BGRA, SWIZZLE_RGBA</td></tr>
</table>

53.3.2 Clocks

Table 53-4. Clocks

Clock name	Description
aclk	AXI interface clock
hclk	AHB interface clock
clk2x	Functional 2X GPU core clock for the core logic. Also used internally, divided by 2.

53.4 Software Application Programming Interface (API)

There following software APIs are supported by the GPU:

- OpenVG 1.1 API Standard for Vector Graphics Acceleration
- VGLite Graphics API

53.4.1 OpenVG 1.1 API Standard for Vector Graphics Acceleration

OpenVG provides a low-level hardware acceleration interface for vector graphics libraries such as Flash and SVG. It is used for acceleration of high-quality vector graphics for user interfaces and text on small screen devices.

53.4.1.1 Core API features of OpenVG

Vector GPU supports the following key features of OpenVG:

- Coordinate Systems and Transformations (Image drawing uses a 3x3 perspective transformation matrix)
- Viewport Clipping, Scissoring and Alpha Masking
- Paths
- Images
- Image Filters
- Paint (gradient and pattern)
- Blending
- Dithering

53.4.1.2 OpenVG Rendering Pipeline

- Stage 1: Path, Transformation, Stroke, and Paint
- Stage 2: Stroked Path Generation
- Stage 3: Transformation
- Stage 4: Rasterization
- Stage 5: Clipping and Masking
- Stage 6: Paint Generation
- Stage 7: Image Interpolation
- Stage 8: Blending and Anti-aliasing

53.4.2 VGLite Graphics API

The GPU can also be used with the VGLite Graphics API. It is designed to support menu-driven user interfaces optimized for a system's overall resource requirements. The goal is to provide maximum performance, while keeping the memory footprint to a minimum.

The VGLite API is partitioned to provide controls for three main areas of functionality:

- Initialization: For Hardware and Software initialization
- Blit: For Raster rendering
- Draw: For Draw operations

The table below describes the Hardware supported Blit and/or Draw capabilities available for software control through the API:

Table 53-5. Blit and Draw Controls in VGLite API

Capability	GPU with VGLite API Support
Fill	Non-zero or Even-Odd
Path coordinate formats	S8, S16, S32, FP32
HW Anti-alias levels	Low, Medium, High
Blending	SrcOver, DstOver, SrcIn, DstIn, Screen, Multiply, Additive, Subtractive
Data Formats	ARGB4/8, RGB565, A4/8, L8 and BRGA variants
Path definitions	bounding box, path quality, coordinate format, buffer pointer, path length, path data

NOTE

The VGLite Graphics API document will be available as a part of the Software Development Kit (SDK).

Chapter 54
Audio Overview

54.1 Audio Overview

The audio subsystem consists of the following modules: SAI-1, SAI-2, SAI-3,SAI-4, PDM, ASRC, SPDIF and MQS. In addition, the IOMUX must be appropriately configured to get signals in and out of the chip.

Audio Module Overview provides an overview of each of the audio modules, followed by a module-specific section.

54.1.1 Audio Module Overview

The following figure shows a high level block diagram of the audio subsystem.

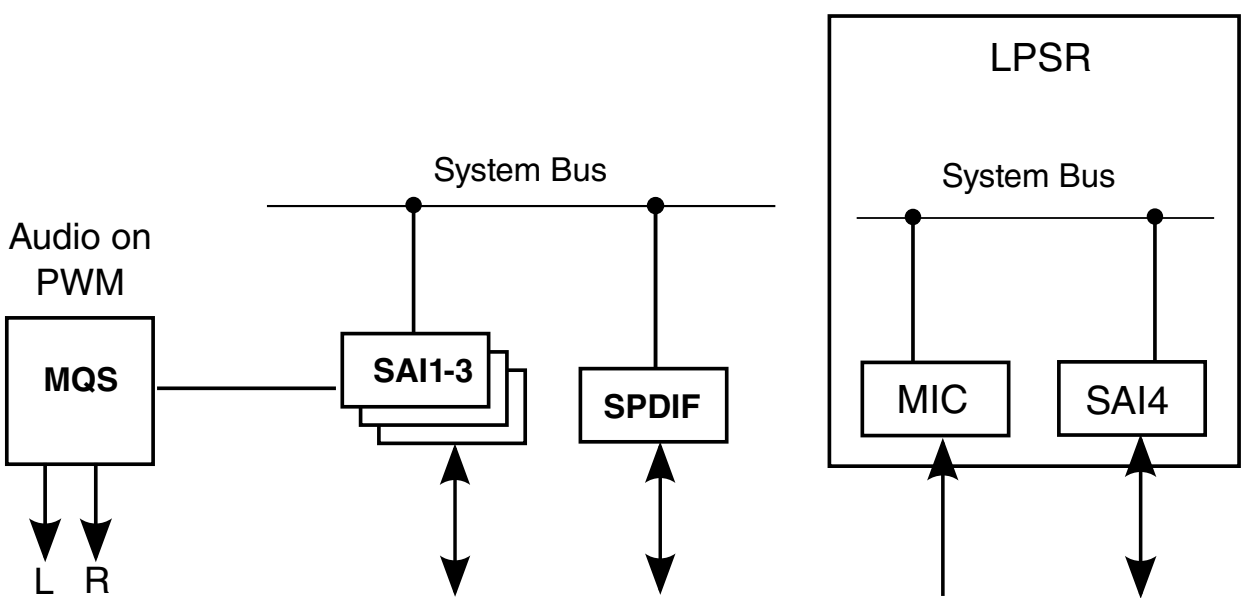


Figure 54-1. Audio peripherals block diagram

SAI*n* are synchronous serial interfaces used to transfer audio data. They can be accessed by both the eDMA and Arm CPUs. Their input/output are connected to the pads through IOMUX.

The PDM Microphone Interface (PDM) is used to receive audio from microphone. It supports 8-channel input with individual enable control for full or partial set of channels.

Asynchronous Sample Rate Converter (ASRC) is used for PCM sample rate conversion from one audio sampling clock domain to another.

MQS (medium quality speaker) is used to convert the I2S audio data from SAI to PWM signals that can drive external speaker directly. Its audio source comes from SAI-3 and its output is connected to pads through IOMUX.

The Sony/Philips digital interface (SPDIF) audio module is a stereo transceiver that allows the processor to receive and transmit digital audio over it. The SPDIF receiver section includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency. A recovered clock is provided by the SPDIF receiver section and may be used to drive both internal and external components in the system.

The audio interfaces are summarized as the table below.

Table 54-1. Audio Interface Summary

Interface	Function	Configuration	
		RX Data Line	TX Data Line
SAI-1	External audio	4	4
SAI-2	External audio	1	1
SAI-3	External audio	1	1
SAI-4	External audio	1	1
MQS	External audio	0	2
SPDIF	External audio	1	1
PDM	PDM microphone interface	Four-pair inputs	
ASRC	Asynchronous Sample Rate Converter	-	

SAI-1 is used for multi-channel audio interface, which supports up to 8-channels audio input or 8-channels audio output at 384 kHz/32-bit. SAI-2, SAI-3 and SAI-4 can be used for stereo audio input and output up to 384 kHz/32-bit. Also SAI-3 is able to drive MQS directly as a low-cost audio output.

To reduce IO count and keep the flexibilities of supporting multiple RX and TX data lines application, SAI-1 has following options on data pin multiplexing. The multiplexing is done in the IOMUX. The following table is not an exhastive list of options, but does show that there can be up to 5 SAI-1 RX/TX data pins.

Table 54-2. SAI-1 TX/RX Data Pin Multiplexing

Options	Number of RX Data Pins	Number of TX Data Pins
0	1	4
1	2	3
2	3	2
3	4	1

SAI-1 to IOMUXC connection is implemented as shown in the diagram below.

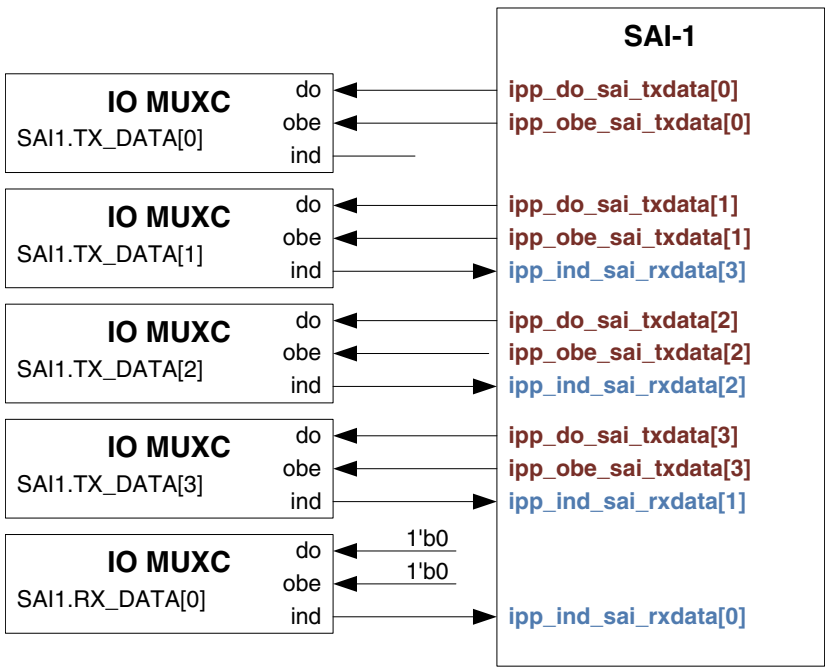


Figure 54-2. SAI-1 TX/RX Data Multiplexing

In the SAI-1 TX/RX Data Multiplexing figure:

- 'do' in the signal names indicate data transmitted to the PAD from SAI
- 'ind' in the signal name indicate data sensed from PAD and received by SAI
- 'obe' in the signal name indicate transmit enable

For example, if a pin is configured to use SAI-1 TXD3, then the PAD level is driven from SAI-1 TXD3 when transmit is enabled (obe=1). When the signal works as a receive function, the PAD level is sensed and sent to SAI-1 RXD1.

Detailed clock multiplexing scheme is shown in the following figure.

Please note that SAI-4 is a standalone SAI module and is not shown in the Audio subsystem clocking diagram below.

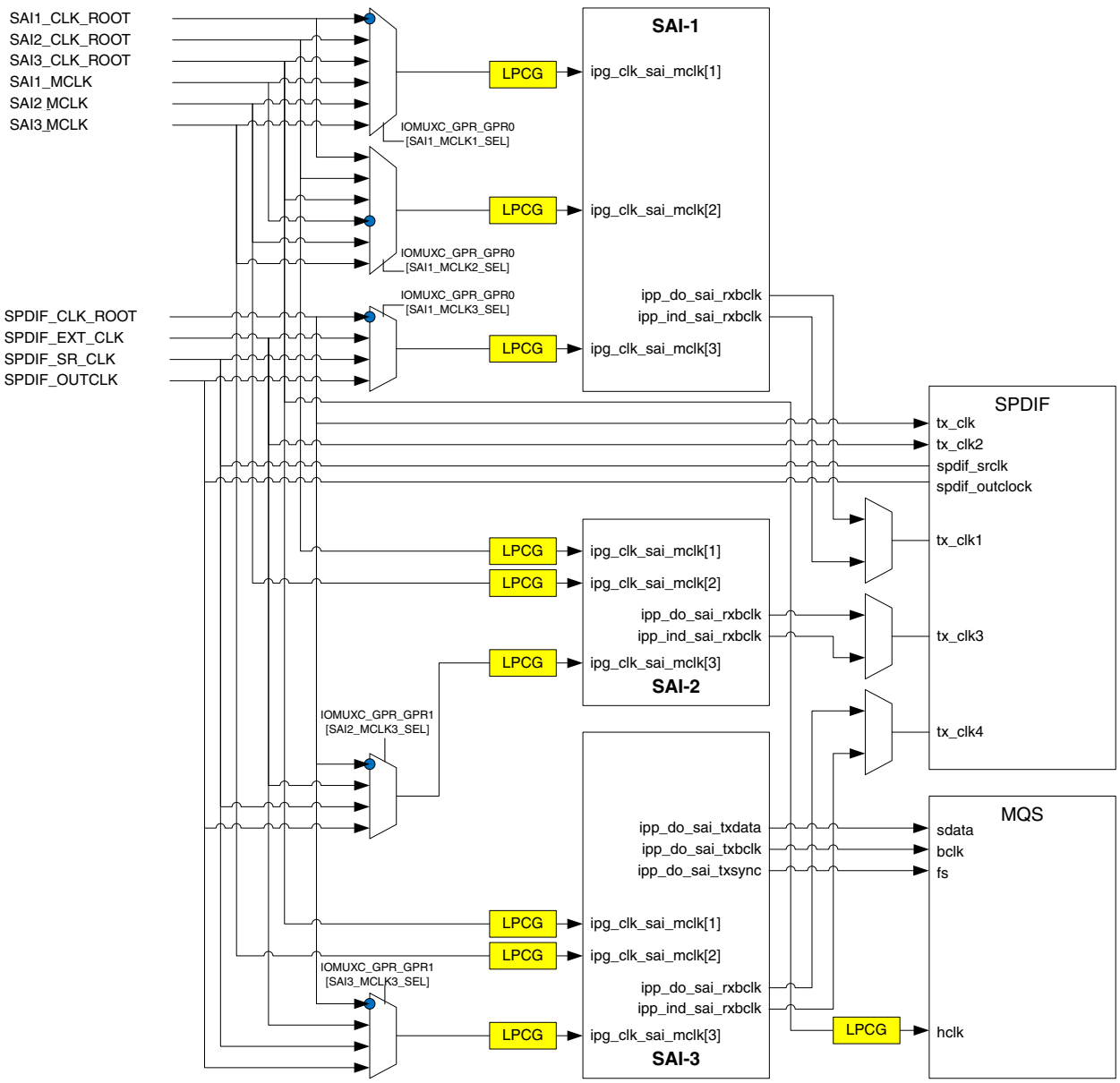


Figure 54-3. Audio subsystem clocking diagram

NOTE

SPDIF tx_clk1, tx_clk3, and tx_clk4 inputs are selected inside the SAI-1, SAI-2, and SAI-3, respectively.

54.1.2 Medium Quality Sound (MQS)

MQS is used to generate medium quality audio via a standard GPIO in the pinmux. The user can connect stereo speakers or headphones to a power amplifier without an additional DAC chip.

- 2-channel, LSB-valid 16 bit, MSB shift-out first serial data (sdata)

- Frame sync aligned with the left channel data
- 44.1 kHz or 48 kHz I2S signals from SAI3
- SNR target as no more than 20 dB for the signals below 10 kHz
- Signals over 10 kHz have worse THD+N values

54.1.3 Synchronous Audio Interface (SAI)

- Transmitter with independent Bit Clock and Frame Sync supporting up to 4 data lines
- Receiver with independent Bit Clock and Frame Sync supporting up to 4 data lines
- Maximum Frame Size of 32 Words
- Word size programmable from 8-bits to 32-bits
- Word size configured separately for first word and remaining words in frame
- Asynchronous FIFO for each Transmit and Receive data line
- Graceful restart after FIFO Error

54.1.4 Sony/Philips Digital Interface (SPDIF)

The Sony/Philips Digital Interface (SPDIF) module is a stereo transceiver that allows the processor to receive and transmit digital audio over it using the IEC60958 standard, consumer format. The chip provides a single SPDIF receiver with one input, and one SPDIF transmitter with one output.

The SPDIF transceiver allows the handling of both SPDIF channel status (CS) and User (U) data and features a frequency measurement block that allows the precise measurement of the incoming sampling frequency.

The clock recovered by the SPDIF receiver is provided to drive both internal and external components in the system such as the SPDIF transmitter, SAI ports, as well as external A/Ds or D/As, with clocking control provided via related registers.

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter. The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in two 16-word-deep FIFOs, one FIFO for the left channel, the other FIFO for the right channel. The FIFOs support programmable watermark levels so that FIFO Full service request can be triggered when the combined number of data words stored in both FIFOs is 2, 8, 16 or 32 words. It is recommended to program the watermark level to trigger a FIFO Full service request when 16 word locations are filled. For optimal performance when servicing the FIFO Full service request, the FIFOs should be read alternately, starting

with the left channel FIFO. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers, and the data is stored in two 16-word-deep FIFOs, one for the right channel, the other for the left channel. The FIFOs support programmable watermark levels so that FIFO Empty service request can be triggered when the combined number of empty data words locations in both FIFOs is 8, 16, 24 or 32 words. It is recommended to program the watermark level to trigger a FIFO Empty service request when 16 word locations are empty. For optimal performance when servicing the FIFO Empty service request, the FIFOs should be written alternately, starting with the left channel FIFO. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates an SPDIF output bitstream in the biphase mark format (IEC 60958), which consists of audio data, channel status and user bits.

The data handled by the SPDIF module is 24-bit wide. The 24-bit SPDIF data is aligned in the 24 least significant bits of the 32-bit shared peripheral bus data word. The 8 most significant bits of the 32-bit word are ignored by the SPDIF Transmitter when data is being stored in the Transmit FIFOs from the peripheral bus. The 8 most significant bits of the 32-bit word are zeroed by the SPDIF Receiver module when the data is being read from the Receiver FIFOs to the peripheral bus.

Note that 16-bit data is left-aligned in the 24-bit word format of the SPDIF. This means that when receiving 16-bit data, it will be located in the middle two bytes of the 32-bit peripheral bus data word, while the 8 bits of the MSB and the 8 bits of the LSB will be zero. When 16-bit data is to be transmitted, the 32-bit word to be written to the SPDIF Transmit FIFOs should be created as follows: the 16-bit data should be located in the middle two bytes of the 32-bit data word and the 8 bits of the LSB must be set to zero, while the 8 bits of the MSB will be ignored.

The SPDIF Transmit clock is generated by the SPDIF internal clock generator module and the clock sources are from outside of the SPDIF block. The clock sources should provide a clock that is at least 64 x Fs, where Fs is the sampling frequency. The external clock source should provide at least 128 x Fs. Clocks of higher frequency may be provided as long as the multiplication factor is a power of 2 (for example, 128x, 256x or 512x). Also, clock frequency precision of 100ppm or better should be provided.

54.1.5 PDM Microphone Interface (PDM)

The PDM module receives audio from a microphone. It includes the following features:

- Fixed filtering characteristics for audio application
- Full or partial set of channels operation with individual enable control
- Programmable PDM clock generator
- Programmable decimation rate
- 16-bit signed output result
- Overall stopband attenuation more than 80dB
- Overall passband ripple less than 0.2dB
- CIC filter
- DC remover
- Half Band filter
- FIR filter
- FIFOs with DMA capability
- Hardware Voice Activity Detector (HWVAD)
- 4 lines, 8 channels

Chapter 55
Asynchronous Sample Rate Converter (ASRC)

55.1 Chip-specific Asynchronous Sample Rate Converter (ASRC) information

Table 55-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

The below figure shows the ASRC System Overview for this chip.

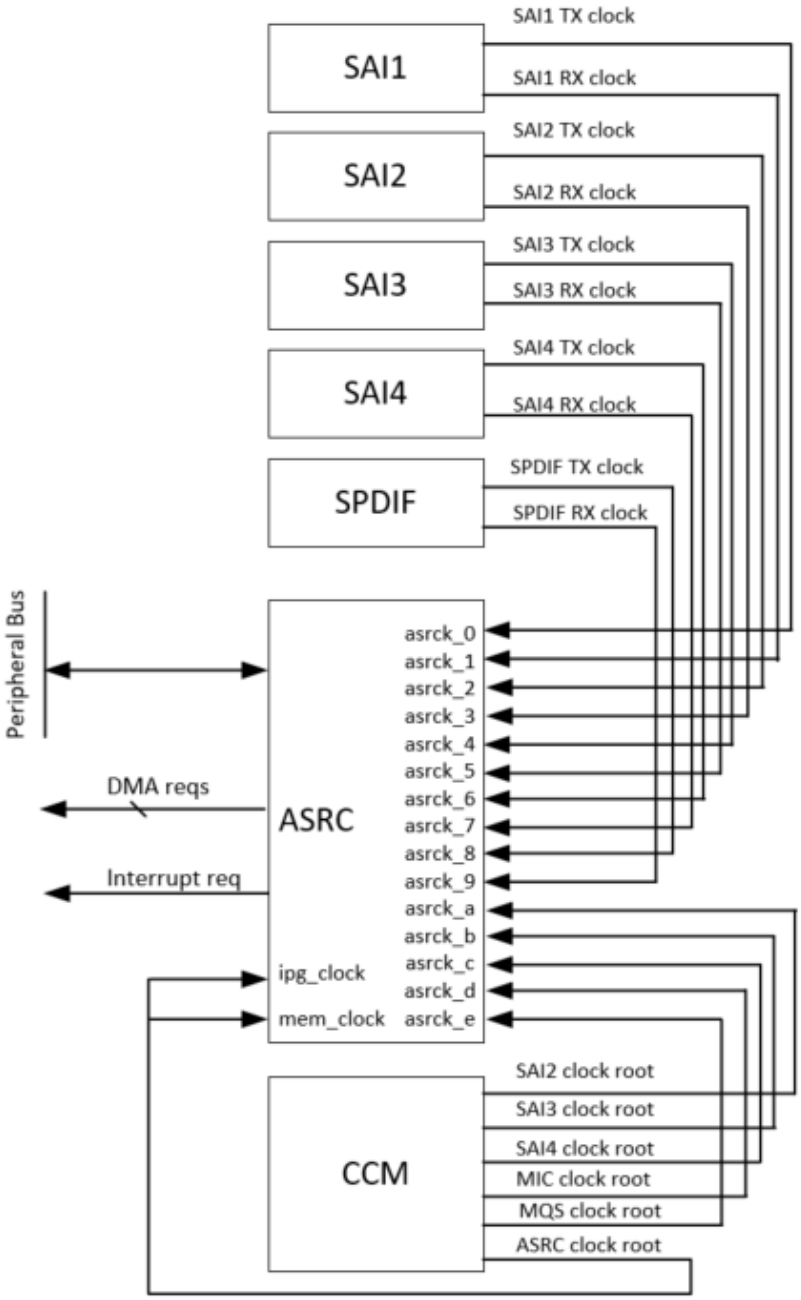


Figure 55-1. ASRC System Overview

55.2 Overview

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated with an input clock into a signal associated with a different output clock.

The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The ASRC supports up to three sampling rate pairs.

The incoming audio data to this chip may be received from various sources at different sampling rates. The outgoing audio data of this chip may have different sampling rates and it can also be associated with output clocks that are asynchronous to the input clocks.

The ASRC is implemented as a co-processor in hardware, with minimal ARM Platform intervention required.

The ASRC Filter Processor (FP) handles the main signal processing algorithm. It has its own simplified instructions. The FP resources (MIPS,RAM,ROM ALU etc) are shared among the different conversation pairs.

55.2.1 Block Diagram

The following figure is the ASRC block diagram.

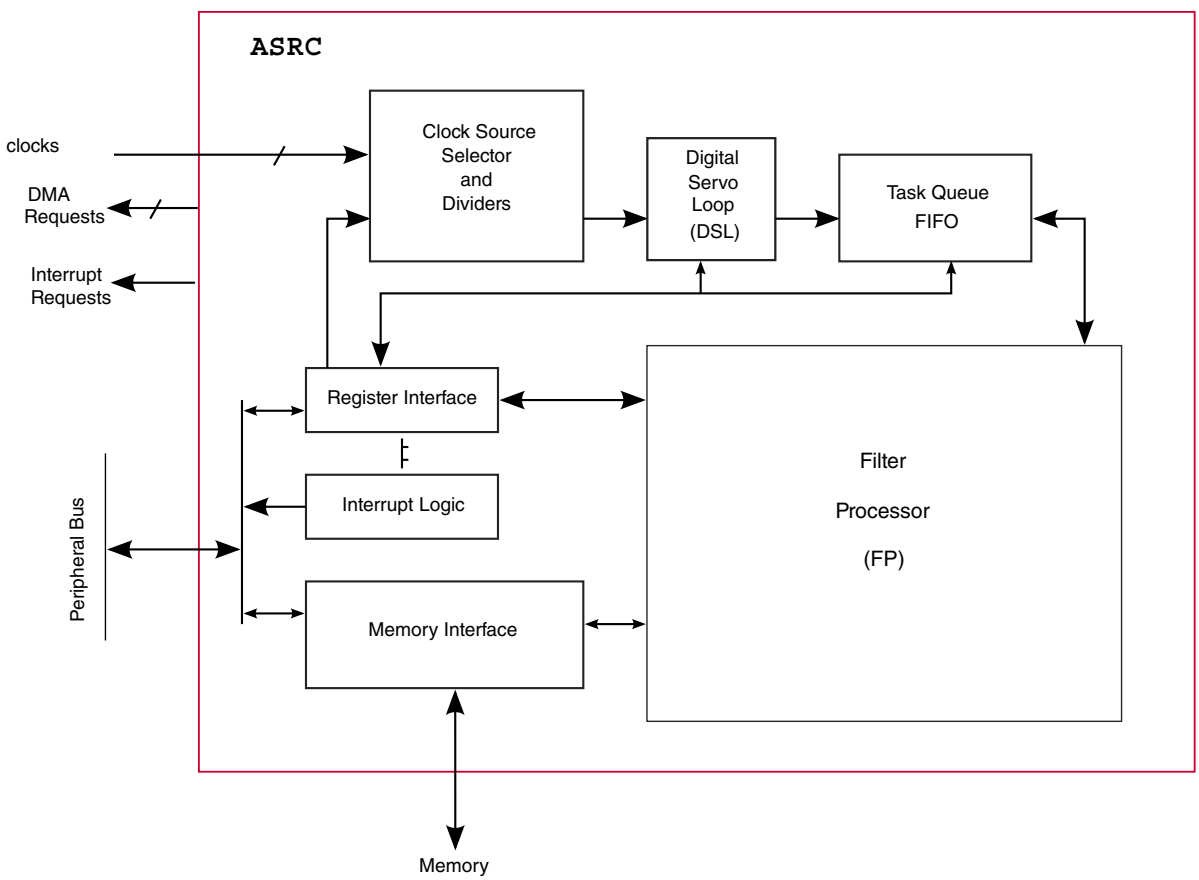


Figure 55-2. ASRC block diagram

55.2.2 Features

The ASRC features are listed below:

- Any number (0-10) of contiguous channels can be associated to one of the sampling rate pairs
- Support user-programmable threshold for the input/output FIFOs
- Support flexible 8/16/24-bit width of input data, and 16/24-bit width of output data
- Designed for rate conversion between 32 kHz, 44.1 kHz, 48 kHz, 96 kHz, and 192 kHz. The useful signal bandwidth is below 24 kHz
- Other input sampling rates in the range of 8 kHz to 200 kHz is also supported[1]
- Other output sampling rates in the range of 30 kHz to 200 kHz is also supported[1]
  - The limitation is the supported ratio (Fsin/Fsout) range (between 1/24 to 8)
- Automatic accommodation to slow variations in the incoming and outgoing sampling rates.
- Linear phase
- Tolerant to sample clock jitter
- Designed for real-time streaming audio usage. The output sampling clock must be always physically available in the system.

Clock/Data Connections

- The sampling rate clocks are directly connected to the ASRC block. The ratio estimation of the input clocks with output clocks are done in ASRC hardware when both input/output sampling clocks are physically available.
- When both the input sampling clock and the output sampling clock are physically available, the rate conversion can work by configuring the physical clocks.
- When the input sampling clock is not physically available, the rate conversion can still work by setting ideal-ratio values into ASRC interface registers.
- The clock signals come from the following blocks:
  - SAI, receiving bit clock and transmitting bit clock
  - SPDIF, receiving bit clock and transmitting bit clock
  - other audio peripherals etc.
- The Digital Servo Loop (DSL) is a digital PLL for tracking the relationship between input/output sampling clocks. It is shared among all pairs in a first-come-first-serve form
- The exchange of audio data is done by the processor accessing ASRC block through registers defined on shared peripheral bus

1. The ASRC designed input and output sampling rate are between 44.1 kHz, 32 kHz, 48 kHz, 96 kHz, and 192kHz. Useful signal bandwidth is below 24 kHz.

55.3 Functional Description

This section provides a complete functional description of the block.

55.3.1 Operating Modes

The ASRC supports modes described in the following sections:

- Data Transfer Schemes
- Word Alignment Supported

55.3.1.1 Data Transfer Schemes

55.3.1.1.1 Data Input Modes

The input mode for each of the three channel sets may be set independently. Three modes of supplying data to the ASRC input FIFOs are available:

- Polling
- Interrupt
- DMA

In all input-data transfer schemes, the ASRC fetches data from each enabled FIFO and processes the data sample-by-sample after each rising edge of the associated input sampling clock until the FIFO level reaches a threshold.

After the threshold is reached, the ASRC requests data. The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples.

The threshold can be defined by interface registers ASRMCRx, x = A, B or C.

If the ASRC attempts to fetch data from an empty FIFO, an error is generated and the ASRSTR[AOLE] bit is set. If the ASRC overload interrupt is enabled (ASRIER[AOLIE] bit is set), an interrupt is generated.

When writing data to an input FIFO, you must ensure that it is in a predefined sequence. For example, when writing to an input FIFO, the sequence should be: channel_0, channel_1, channel_2,..., channel_n, channel_0, channel_1, channel_2, etc. Here channel_n stands for the data intended for the n-th channel. The hardware re-allocates each data to its corresponding channel FIFO. The channel being re-allocated is shown by ASRCCR[ACIx], x=A,B or C.

55.3.1.1.1.1 Data Input Polling Mode

Polling mode is the default mode following power-on or individual reset, and is selected by clearing the associated channel set A, B, or C data-input interrupt enable bit (ASRIER[ADIEx], where x=A, B or C). In this mode, data-input interrupts are disabled. When the FIFO level is below the threshold, the associated status bit (ASRSTR[AIDEx], where x=A, B, or C) is set. To clear the status bit, the FIFO must be written with enough data to raise the level above the threshold.

55.3.1.1.1.2 Data Input Interrupt Mode

The ASRC input FIFOs can also be serviced by interrupts. To enable interrupts, the corresponding data-input interrupt enable bits (ASRIER[ADIEx], where x=A, B, or C) should be set. An interrupt is automatically generated any time the input FIFO level is below the threshold. The interrupt is cleared when enough data is written to the FIFO to raise the level above the threshold.

55.3.1.1.1.3 Data Input DMA Mode

The ASRC input FIFOs can also be filled using DMA. In this mode, the data-input interrupt-enable bits (ASRIER[ADIEx], where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

55.3.1.1.2 Data Output Modes

The output mode for each of the 3 channel sets (A, B, and C) may be set independently.

Three modes of retrieving data from the ASRC output FIFOs are available:

- Polling
- Interrupt
- DMA

In all output-data transfer schemes, the ASRC places a processed sample into the associated output FIFO. After a threshold is reached, the ASRC requests that data be transferred out of the FIFO.

The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x = A, B or C.

If the ASRC attempts to place data into a FIFO that is already full, an error is generated and the ASRSTR[AOLE] bit is set. If the ASRC overload interrupt is enabled (ASRIER[AOLIE] bit is set), an interrupt is generated.

Each output FIFO is organized in the same channel order in which the associated input FIFO was written.

Three transfer modes are supported by Interface Block.

55.3.1.1.2.1 Data Output Polling Mode

The ASRC output FIFOs can be serviced by polling. In this mode, ensure the associated output-data interrupt enable bit (ASRIER[ADOEx], where x=A, B, or C) is cleared. In this mode, all output-data interrupts are disabled. Any time the output FIFO exceeds the threshold the associated status bit (ASRSTR[AODFx], where x=A, B, or C) is set. To clear the status bit, enough data must be read from the associated output FIFO to lower the level below the threshold.

55.3.1.1.2.2 Data Output Interrupt Mode

The ASRC output FIFOs may also be serviced using interrupts. To enable this mode, the corresponding output-data interrupt-enable bits (ASRIER[ADOEx], where x=A, B, or C) should be set. Any time the output FIFO level exceeds the threshold, an interrupt is automatically generated. The interrupt is cleared when enough data is read from the FIFO to lower the level below the threshold.

55.3.1.1.2.3 Data Output DMA Mode

The ASRC output FIFOs can also be read using DMA. In this mode, the output-data interrupt-enable bits (ASRIER[ADOEx], where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

55.3.1.2 Word Alignment Supported

55.3.1.2.1 Input Data Alignment Modes

The position and length of input data word to the input data FIFOs ASRDIA, ASRDIB, ASRDIC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes:

Table 55-2. Input Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								

Table continues on the next page...

Table 55-2. Input Data Alignment (continued)

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

55.3.1.2.2 Output Data Alignment Modes

The position and length of output data word from the output data FIFOs ASRDOA, ASRDOB, ASRDOC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes:

Table 55-3. Output Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned with Sign Extension	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit LSB Aligned with Sign Extension	23	23	23	23	23	23	23	23	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

55.3.2 Algorithm Description

55.3.2.1 Signal processing flow

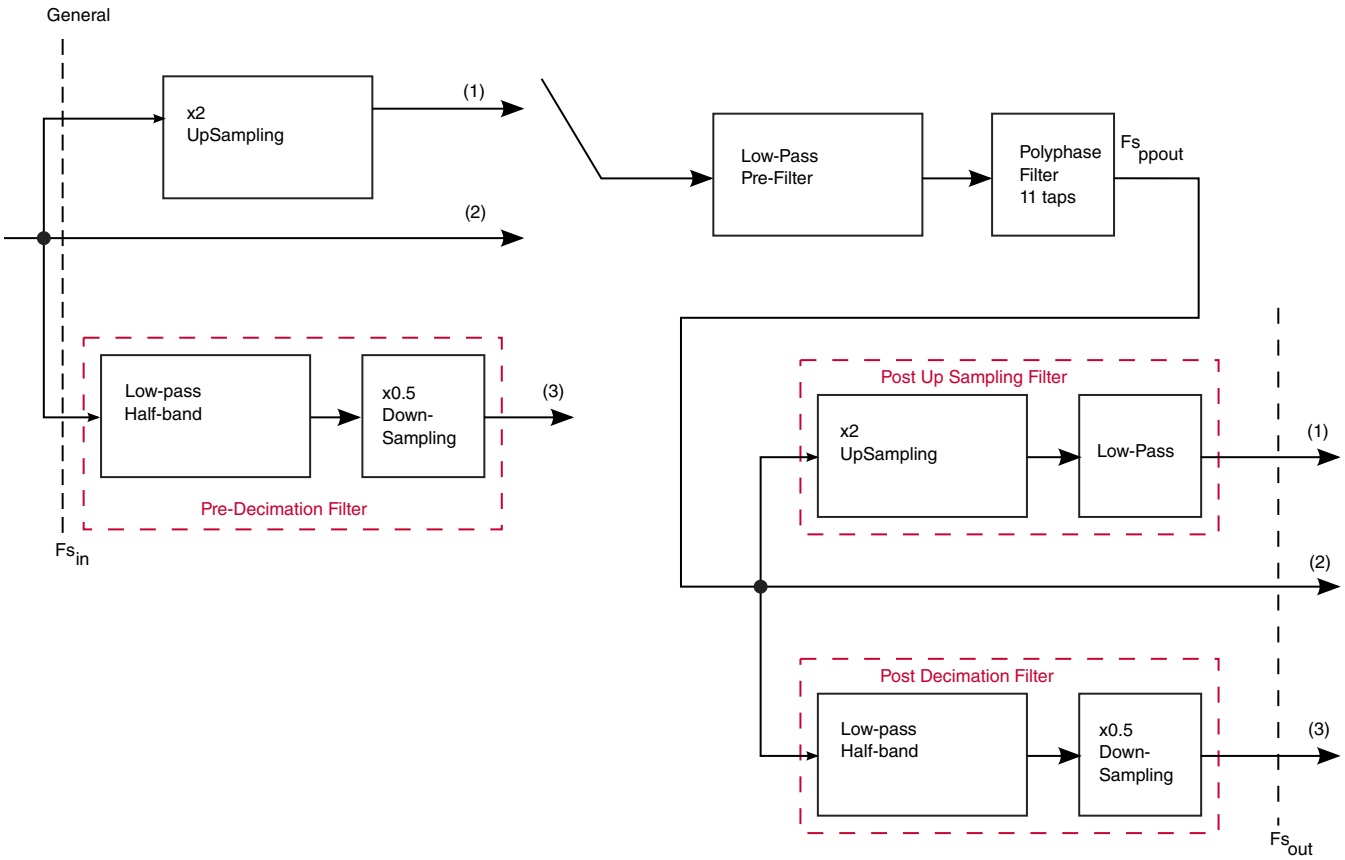


Figure 55-3. Signal processing configurations

The figure above shows the possible configurations of the ASRC. Each configuration consists of 2 to 4 stages.

- x2 up-sampling rate expander (zero insertion only) (input branch 1), direct connection (input branch 2), or low-pass pre decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (input branch 3),
- low-pass pre-filter, the low-pass bandwidth is at most 0.25 x Fs, where Fs is the sampling rate of the input signal to this low-pass pre-filter,
- polyphase filter,
- x2 post upsampling filter (consisting of a x2 up-sampling rate expander (zero insertion only) with low-pass half-band FIR filter) (output branch 1), direct connection (output branch 2), or low-pass post decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (output branch 3).

By flowing through different processing branches and different setups of the pre-filter, this ASRC scheme can be used to handle different rate conversion requirements.

Table 55-4. ASRC Rate Conversion Configurations

Configuration	Max Signal Bandwidth before polyphase filter	Signal Sampling Rate of polyphase filter output
a: Input Branch 1 + Output Branch 1	$BW_{in} = Fs_{in}/2$	$Fs_{ppout} = Fs_{out}/2$
b: Input Branch 1 + Output Branch 2	$BW_{in} = Fs_{in}/2$	$Fs_{ppout} = Fs_{out}$
c: Input Branch 1 + Output Branch 3	$BW_{in} = Fs_{in}/2$	$Fs_{ppout} = 2Fs_{out}$
d: Input Branch 2 + Output Branch 1	$BW_{in} = Fs_{in}/4$	$Fs_{ppout} = Fs_{out}/2$
e: Input Branch 2 + Output Branch 2	$BW_{in} = Fs_{in}/4$	$Fs_{ppout} = Fs_{out}$
f: Input Branch 2 + Output Branch 3	$BW_{in} = Fs_{in}/4$	$Fs_{ppout} = 2Fs_{out}$
g: Input Branch 3 + Output Branch 1	$BW_{in} = Fs_{in}/8$	$Fs_{ppout} = Fs_{out}/2$
h: Input Branch 3 + Output Branch 2	$BW_{in} = Fs_{in}/8$	$Fs_{ppout} = Fs_{out}$
i: Input Branch 3 + Output Branch 3	$BW_{in} = Fs_{in}/8$	$Fs_{ppout} = 2Fs_{out}$

Table 55-5 lists some typical settings for typical sample rate conversation pairs to achieve less MIPS. ASRC can be set to automatically fill these settings depending on input/ouptut sample rate or users can configure the fields manually to achieve better filter bandwidth performance.

Table 55-5. Pre-processing, post-processing options

{Pre_Proc, Post_Proc}		Fsout (KHz)											
		8	12	16	24	32	44.1	48	64	88.2	96	128	192
Fsin (KHz)	8	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	12	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	16	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	24	{1,2}	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}
	32	{1,2}	{1,2}	{1,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}
	44.1	{2,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}
	48	{2,2}	{1,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	64	{2,2}	{2,2}	{1,2}	{1,2}	{0,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	88.2	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	96	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	128	NA	NA	{2,2}	{2,2}	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	192	NA	NA	NA	{2,2}	{2,2}	{2,2}	{2,2}	{2,1}	{2,1}	{2,1}	{2,1}	{2,1}

NOTE: In the {Pre_Proc, Post_Proc} pair, the meaning of the values are:

Pre_Proc:

- 0 --- Pre-processing Branch 1 as shown in Figure 55-3
- 1 --- Pre-processing Branch 2 as shown in Figure 55-3
- 2 --- Pre-processing Branch 3 as shown in Figure 55-3, decimation-by-2

Post_Proc:

- 0 --- Post-processing Branch 1 as shown in Figure 55-3
- 1 --- Post-processing Branch 2 as shown in Figure 55-3
- 2 --- Post-processing Branch 3 as shown in Figure 55-3

The latencies of the different option can be roughly calculated as follows:

- For PreProc = 0, PostProc = 1 : min latency = constant_A / input-sample-rate + constant_B / output-sample-rate
- For PreProc = 0, PostProc = 0 : min latency = constant_A / input-sample-rate + constant_C / output-sample-rate
- For PreProc = 1, PostProc = 1 : min latency = constant_D / input-sample-rate + constant_B / output-sample-rate

The constants above (e.g., constant_A means the Constant for Preproc = 0, constant_B means the Constant for Postproc = 1, ...) are only influenced by the PreProc/PostProc and (input/output) sampling rate to which they are connected. Input latencies have no relationship with the output latencies, but both elements add together to form the total latencies.

For a rough estimation, the constants can be set as:

- Constant for Preproc = 0: 39
- Constant for Preproc = 1: 78.5
- Constant for Preproc = 2: 235
- Constant for Postproc = 0: 42.5
- Constant for Postproc = 1: 8.5
- Constant for Postproc = 2: 172

The max latency can be derived from this value by using the following formula (where 32 means the input/output FIFO depth that initiates data transfer):

- max latency = min latency + 32 / input-sample-rate + 32 / output-sample-rate

55.3.2.2 Operation of the Filter

55.3.2.2.1 Support of Physical Clocks

This design supports physical sampling clocks. The clocks can be provided by:

- Sony/Phillips digital interface (SPDIF)
- Serial Audio Interface (SAI)
- Core master clock derivative as ASRCK1

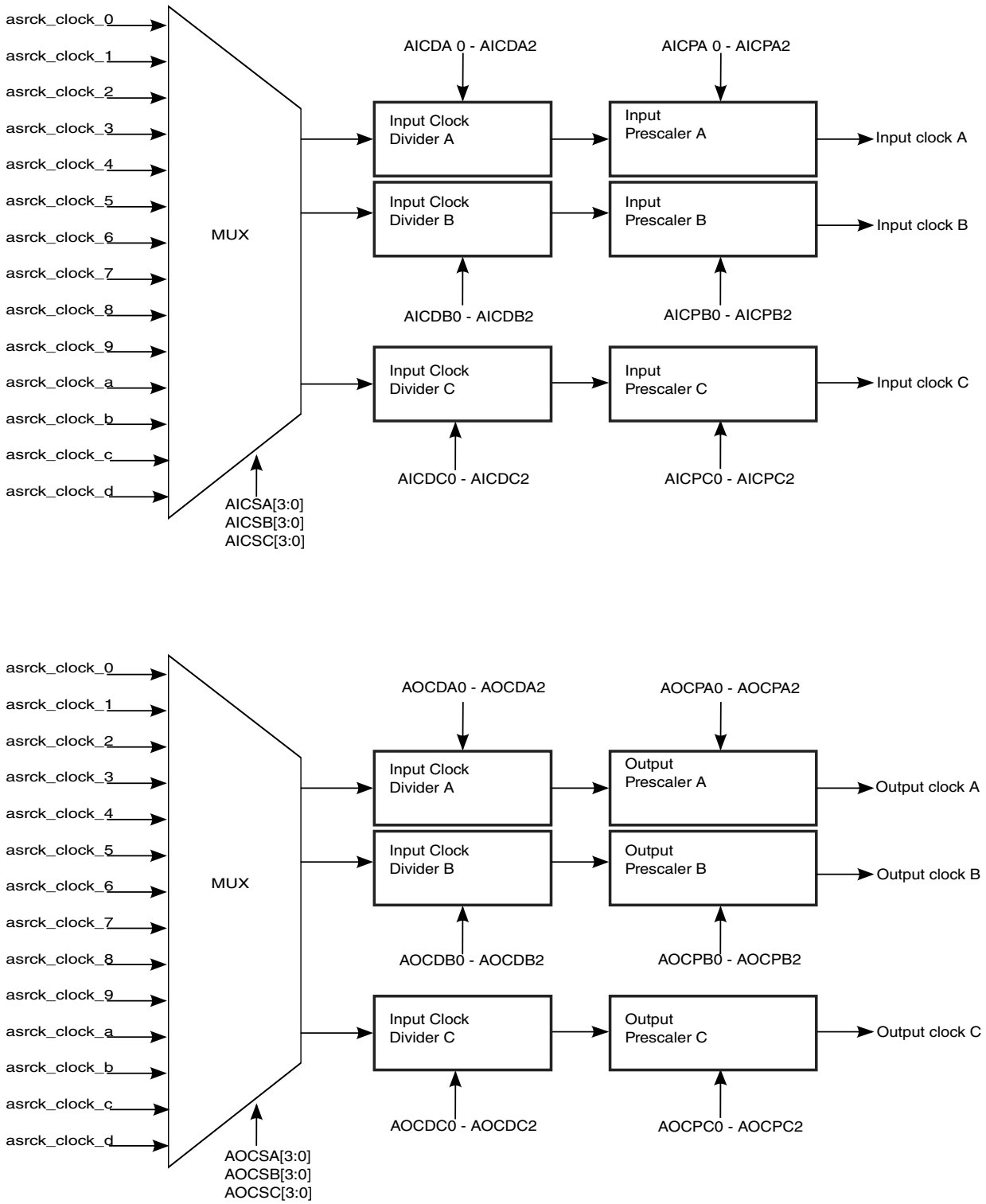


Figure 55-4. Clock Source Selector and Divider

Software can set the ASRC Clock Source Register (ASRCSR) and the Clock Divider Register to select the desired clock source and divide it to the needed sample rate clock for use by the ASRC.

The following is the description of the Clock Source Selector and Divider figure above (x refers to A, B or C):

- If the input clock is available, Pair x divider and prescaler depends on the input clock and input sample rate, which can be calculated by:
  - Input Clock x = (Input Sample Rate x) * AICDx * (2^(AICPx))
  - Where the Input Sample Clock Source x is determined by ASRCSR[AICSx]
- If the output clock is available, Pair x divider and prescaler depends on the output clock and output sample rate, which can be calculated by:
  - Output Clock x = (Output Sample Rate x) * AOCDx * (2^(AOCPx))
  - Where the Output Sample Clock Source x is determined by ASRCSR[AOCSx]

NOTE

The clocks have the following restriction: If the prescaler is set to 1, the clock divider can only be set to 1 and the clock source must have a 50% duty cycle.

55.3.3 Clocks

The table below describes the clock sources for ASRC. Please see clock control block for clock setting, configuration and gating information.

Table 55-6. ASRC Clocks

Clock name	Description
asrck_clock_d	ASRC module clock
ipg_clk	Peripheral clock
mem_clk	Peripheral access clock

55.3.4 Interrupts

ASRC has several interrupts events.

The priorities are shown in the following table.

Table 55-7. Interrupt Priorities/Vector

Priority	Offset	Description
lowest	0x0	ASRC Pair A input data needed
	0x2	ASRC Pair B input data needed
	0x4	ASRC Pair C input data needed
	0x6	ASRC Pair A output data ready

Table continues on the next page...

Table 55-7. Interrupt Priorities/Vector (continued)

Priority	Offset	Description
	0x8	ASRC Pair B output data ready
	0xA	ASRC Pair C output data ready
highest	0xC	ASRC Overload

55.3.5 DMA requests

ASRC has six DMA requests. They are directly connected to the lowest six status bits in the ASRSTR register. For some ARM platforms, the six status bits are directly connected to the DMA peripheral as DMA event signals.

Table 55-8. DMA requests

Type	Description
0	ASRC Pair A input data needed
1	ASRC Pair B input data needed
2	ASRC Pair C input data needed
3	ASRC Pair A output data ready
4	ASRC Pair B output data ready
5	ASRC Pair C output data ready

55.4 External Signals

The ASRC does not have external signals pinned out.

The following table shows the bit definitions that are used by the ASRCSR register to control the sources of the input and output clocks of the ASRC.

Table 55-9. ASRC Bit Clock Definitions

Bit Clk Name	Definition	Description
bit clock 0	SAI1_TX_BCLK	SAI1 Transmitter serial bit clock
bit clock 1	SAI1_RX_BCLK	SAI1 Receiver serial bit clock
bit clock 2	SAI2_TX_BCLK	SAI2 Transmitter serial bit clock
bit clock 3	SAI2_RX_BCLK	SAI2 Receiver serial bit clock
bit clock 4	SAI3_TX_BCLK	SAI3 Transmitter serial bit clock
bit clock 5	SAI3_RX_BCLK	SAI3 Receiver serial bit clock
bit clock 6	SAI4_TX_BCLK	SAI4 Transmitter serial bit clock

Table continues on the next page...

Table 55-9. ASRC Bit Clock Definitions (continued)

Bit Clk Name	Definition	Description
bit clock 7	SAI4_RX_BCLK	SAI4 Receiver serial bit clock
bit clock 8	SPDIF_TX_BCLK	SPDIF Transmitter serial bit clock
bit clock 9	SPDIF_RX_BCLK	SPDIF Receiver serial bit clock
bit clock A	SAI2_CLK_ROOT	SAI2 Clock Root
bit clock B	SAI3_CLK_ROOT	SAI3 Clock Root
bit clock C	SAI4_CLK_ROOT	SAI4 Clock Root
bit clock D	MIC_CLK_ROOT	MIC Clock Root
bit clock E	MQS_CLK_ROOT	MQS Clock Root

55.5 Application Information

The following outlines an **example** setup procedure for the ASRC block:

1. Configure and enable the ASRC clock source. See this device's clocking chapter for more information
2. Disable the ASRC (write 0x00000000 to ASRCTR)
3. Enable the peripheral that is connected to the ASRC. See this device's clocking chapter and the ASRC chip specific section for more information
4. Configure the following:
   a. Enable ASRC function(ASRCTR[ASRCEN]=1)
   b. Enable Use of Ideal Ratio for Pair A (ASRCTR[IDRA]=1)
   c. Enable Use of Ratio for Pair A (ASRCTR[URSA]=1)
   d. Assign 2 channels to Pair A (ASRCNCR[ANCA]=2)
   e. Configure Post-Processing Configuration and Pre-Processing Configuration for Conversion Pair A. See Signal processing flow
   f. Set ASRC Clock Divider 1 (ASRCDR1) and ASRC Clock Divider 2 (ASRCDR2. See Support of Physical Clocks for more information
   g. Set the Input Clock Source A (ASRCSR[AICSA]) and Output Clock Source A (ASRCSR[AOCSA]) both to Bit Clock D (0x0000d00d)
   h. Set the Ideal Ratio for Pair A. The ratio depends on input sample rate and output sample rate. See the ASRC Ideal Ratio for Pair A-High Part (ASRIDRHA) and ASRC Ideal Ratio for Pair A -Low Part (ASRIDRLA) register descriptions. If the input sample rate is equal to the output sample rate, Ideal Ratio is 1:
      - Ideal Ratio for Pair A-High Part (ASRIDRHA = 0x4)
      - Ideal Ratio for Pair A-Low Part (ASRIDRLA = 0x0)
5. For Pair A - Set input threshold to 32 and output threshold to 8 (ASRMCRA[INFIFO_THRESHOLDA], ASRMCRA[OUTFIFO_THRESHOLDA])
7. Zeroize Pair A buffers(ASRMCRA[ZEROBUFA]=1)

8. Do not bypass polyA filter (ASRMCRA[BYPASSPOLYA]=0)
9. Set Pair A data width (ASRMCR1A[IWD]) and data alignment(ASRMCR1A[IMSB] and ASRC_ASRMCR1A[OMSB]) depends on the audio data format
10. Disable interrupt handler (ASRIER = 0x00000000)
11. After configuration is done, enable the ASRC pair(ASRCTR[ASREA]=1)
12. Poll until the initialization is done (ASRCFG and ASRCFG[INIRQA]) == 0)
13. Feed Input Data (ASRDIA)
14. Get Output Data (ASRDOA)
15. Disable ASRC (ASRCTR[ASRCEN] = 0)

55.6 ASRC Memory Map/Register Definition

55.6.1 ASRC register descriptions

All useful registers are listed in the memory map below. The access of undefined registers behaves as normal registers.

All the interface registers are LSB aligned except the input FIFOs and the output FIFOs, and each register has only 24 effective bits.

The input FIFO and output FIFO word alignment can be defined using ASRMCR1{A,B,C} registers in 32-bit interface system.

55.6.1.1 ASRC memory map

ASRC0 base address: 4041_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	ASRC Control Register (ASRCTR)	32	RW	0000_0000
4	ASRC Interrupt Enable Register (ASRIER)	32	RW	0000_0000
C	ASRC Channel Number Configuration Register (ASRCNCR)	32	RW	0000_0000
10	ASRC Filter Configuration Status Register (ASRCFG)	32	RW	0000_0000
14	ASRC Clock Source Register (ASRCSR)	32	RW	0000_0000
18	ASRC Clock Divider Register 1 (ASRCDR1)	32	RW	0000_0000
1C	ASRC Clock Divider Register 2 (ASRCDR2)	32	RW	0000_0000
20	ASRC Status Register (ASRSTR)	32	RO	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
40 - 50	ASRC Parameter Register n (ASRPM1 - ASRPM5)	32	RW	0000_0000
54	ASRC Task Queue FIFO Register 1 (ASRTFR1)	32	RW	0000_0000
5C	ASRC Channel Counter Register (ASRCCR)	32	RW	0000_0000
60	ASRC Data Input Register for Pair x (ASRDIA)	32	WO	0000_0000
64	ASRC Data Output Register for Pair x (ASRDOA)	32	RO	0000_0000
68	ASRC Data Input Register for Pair x (ASRDIB)	32	WO	0000_0000
6C	ASRC Data Output Register for Pair x (ASRDOB)	32	RO	0000_0000
70	ASRC Data Input Register for Pair x (ASRDIC)	32	WO	0000_0000
74	ASRC Data Output Register for Pair x (ASRDOC)	32	RO	0000_0000
80	ASRC Ideal Ratio for Pair A-High Part (ASRIDRHA)	32	RW	0000_0000
84	ASRC Ideal Ratio for Pair A -Low Part (ASRIDRLA)	32	RW	0000_0000
88	ASRC Ideal Ratio for Pair B-High Part (ASRIDRHB)	32	RW	0000_0000
8C	ASRC Ideal Ratio for Pair B-Low Part (ASRIDRLB)	32	RW	0000_0000
90	ASRC Ideal Ratio for Pair C-High Part (ASRIDRHC)	32	RW	0000_0000
94	ASRC Ideal Ratio for Pair C-Low Part (ASRIDRLC)	32	RW	0000_0000
98	ASRC 76 kHz Period in terms of ASRC processing clock (ASR76K)	32	RW	0000_0A47
9C	ASRC 56 kHz Period in terms of ASRC processing clock (ASR56K)	32	RW	0000_0DF3
A0	ASRC Misc Control Register for Pair A (ASRMCRA)	32	RW	0000_0000
A4	ASRC FIFO Status Register for Pair A (ASRFSTA)	32	RO	0000_0000
A8	ASRC Misc Control Register for Pair B (ASRMCRB)	32	RW	0000_0000
AC	ASRC FIFO Status Register for Pair B (ASRFSTB)	32	RO	0000_0000
B0	ASRC Misc Control Register for Pair C (ASRMCRC)	32	RW	0000_0000
B4	ASRC FIFO Status Register for Pair C (ASRFSTC)	32	RO	0000_0000
C0 - C8	ASRC Misc Control Register 1 for Pair X (ASRMCR1A - ASRMCR1C)	32	RW	0000_0000

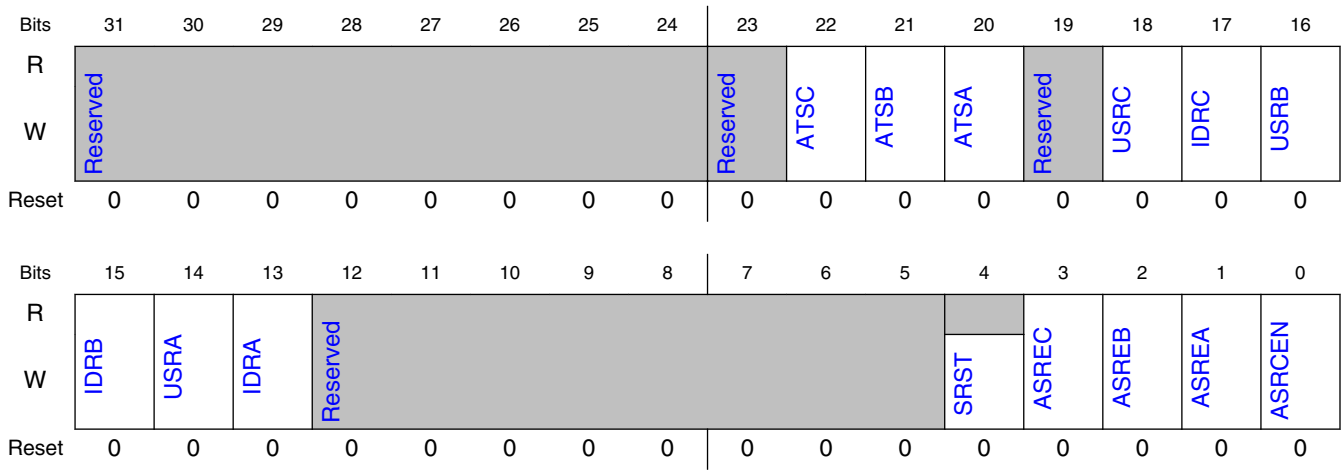
55.6.1.2 ASRC Control Register (ASRCTR)

The ASRC control register (ASRCTR) is a read/write register that controls the ASRC operations.

55.6.1.2.1 Offset

Register	Offset
ASRCTR	0h

55.6.1.2.2 Diagram



55.6.1.2.3 Fields

Field	Description
31-24 —	- Reserved
23 —	- Reserved. Should be written as zero for compatibility.
22 ATSC	ATSC ASRC Pair C Automatic Selection For Processing Options When this bit is 1, pair C automatically updates its pre-processing and post-processing options (ASRCFG[PREMODC], ASRCFG[POSTMODC] see ASRC Filter Configuration Status Register for Pair C) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC 76 kHz Period in terms of ASRC processing clock and ASRC 56 kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair C. This bit should be disabled when {USRC, IDRC}={1,1}. 0 - Pair C does not automatically update its pre-processing and post-processing options 1 - Pair C automatically updates its pre-processing and post-processing options
21 ATSB	ATSB ASRC Pair B Automatic Selection For Processing Options When this bit is 1, pair B automatically updates its pre-processing and post-processing options (ASRCFG[PREMODB], ASRCFG[POSTMODB] see ASRC Filter Configuration Status Register for Pair B) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC 76 kHz Period in terms of ASRC processing clock and ASRC 56 kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair B. This bit should be disabled when {USRB, IDRB}={1,1}.

Table continues on the next page...

Field	Description
	0 - Pair B does not automatically update its pre-processing and post-processing options 1 - Pair B automatically updates its pre-processing and post-processing options
20 ATSA	ATSA ASRC Pair A Automatic Selection For Processing Options When this bit is 1, pair A automatically updates its pre-processing and post-processing options (ASRCFG[PREMODA], ASRCFG[POSTMODA] see ASRC Filter Configuration Status Register for Pair A) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC 76 kHz Period in terms of ASRC processing clock and ASRC 56 kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair A. This bit should be disabled when {USRA, IDRA}={1,1}. 0 - Pair A does not automatically update its pre-processing and post-processing options 1 - Pair A automatically updates its pre-processing and post-processing options
19 —	- Reserved. Should be written as zero for compatibility.
18 USRC	USRC Use Ratio for Pair C Use ratio as the input to ASRC. This bit is used in conjunction with IDRC control bit. 0 - Do not use ratio as the input to ASRC for pair C 1 - Use ratio as the input to ASRC for pair C
17 IDRC	IDRC Use Ideal Ratio for Pair C When USRC=0, this bit has no usage. When USRC=1 and IDRC=0, ASRC internal measured ratio is used. When USRC=1 and IDRC=1, the idea ratio from the interface register ASRIDRHC, ASRIDRLC is used. It is suggested to manually set ASRCFG[POSTMODC], ASRCFG[PREMODC] according to Table 55-5 in this case. 0 - ASRC internal measured ratio is used 1 - Ideal ratio from the interface register ASRIDRHC, ASRIDRLC is used
16 USRB	USRB Use Ratio for Pair B Use ratio as the input to ASRC. This bit is used in conjunction with IDRB control bit. 0 - Do not use ratio as the input to ASRC for pair B 1 - Use ratio as the input to ASRC for pair B
15 IDRB	IDRB Use Ideal Ratio for Pair B When USRB=0, this bit has no usage. When USRB=1 and IDRB=0, ASRC internal measured ratio is used. When USRB=1 and IDRB=1, the idea ratio from the interface register ASRIDRHB, ASRIDRLB is used.It is suggested to manually set ASRCFG[POSTMODB], ASRCFG[PREMODB] according to Table 55-5 in this case.

Table continues on the next page...

Field	Description
	0 - ASRC internal measured ratio is used 1 - Ideal ratio from the interface register ASRIDRHB, ASRIDRLB is used
14 USRA	USRA Use Ratio for Pair A Use ratio as the input to ASRC. This bit is used in conjunction with IDRA control bit. 0 - Do not use ratio as the input to ASRC for pair A 1 - Use ratio as the input to ASRC for pair A
13 IDRA	IDRA Use Ideal Ratio for Pair A When USRA=0, this bit has no usage. When USRA=1 and IDRA=0, ASRC internal measured ratio is used. When USRA=1 and IDRA=1, the ideal ratio from the interface register ASRIDRHA, ASRIDRLA is used. It is suggested to manually set ASRCFG[POSTMODA], ASRCFG[PREMODA] according to Table 55-5 in this case. 0 - ASRC internal measured ratio is used 1 - Ideal ratio from the interface register ASRIDRHA, ASRIDRLA is used
12-5 —	- Reserved. Should be written as zero for compatibility.
4 SRST	SRST Software Reset This bit is self-clear bit. Once it is been written as 1, it generates a software reset signal inside ASRC. After 9 cycles of the ASRC processing clock, this reset process stops, and this bit is cleared automatically. 0 - ASRC Software reset cleared 1 - ASRC Software reset generated. NOTE: This is a self-clear bit
3 ASREC	ASREC ASRC Enable C Enable the operation of the conversion C of ASRC. 0 - operation of conversion C is disabled 1 - operation of conversion C is enabled
2 ASREB	ASREB ASRC Enable B Enable the operation of the conversion B of ASRC. 0 - operation of conversion B is disabled 1 - operation of conversion B is enabled
1 ASREA	ASREA ASRC Enable A Enable the operation of the conversion A of ASRC. 0 - operation of conversion A is disabled 1 - operation of conversion A is enabled

Table continues on the next page...

Field	Description
0 ASRCEN	ASRCEN ASRC Enable Enable the operation of ASRC. 0 - operation of ASRC disabled 1 - operation ASRC is enabled

55.6.1.3 ASRC Interrupt Enable Register (ASRIER)

The ASRC Interrupt Enable Register (ASRIER) is a read/write register that enables ASRC data input, data output, overload, and wait state interrupts.

55.6.1.3.1 Offset

Register	Offset
ASRIER	4h

55.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AFPWE	AOLIE	ADOEC	ADOEB	ADOEA	ADIEC	ADIEB	ADIEA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.3.3 Fields

Field	Description
31-24	-

Table continues on the next page...

Field	Description
—	Reserved
23-8 —	- Reserved. Should be written as zero for compatibility.
7 AFPWE	AFPWE FP in Wait State Interrupt Enable Enables the FP in wait state interrupt. 0 - interrupt disabled 1 - interrupt enabled
6 AOLIE	AOLIE Overload Interrupt Enable Enables the overload interrupt. 0 - interrupt disabled 1 - interrupt enabled
5 ADOEC	ADOEC Pair C Data Output Interrupt Enable Enables the data output C interrupt. 0 - interrupt disabled 1 - interrupt enabled
4 ADOEB	ADOEB Pair B Data Output Interrupt Enable Enables the data output B interrupt. 0 - interrupt disabled 1 - interrupt enabled
3 ADOEA	ADOEA Pair A Data Output Interrupt Enable Enables the data output A interrupt. 0 - interrupt disabled 1 - interrupt enabled
2 ADIEC	ADIEC Pair C Data Input Interrupt Enable Enables the data input C interrupt. 0 - interrupt disabled 1 - interrupt enabled
1 ADIEB	ADIEB Pair B Data Input Interrupt Enable Enables the data input B interrupt. 0 - interrupt disabled 1 - interrupt enabled
0	ADIEA

Field	Description
ADIEA	Pair A Data Input Interrupt Enable Enables the data input A Interrupt. 0 - interrupt disabled 1 - interrupt enabled

55.6.1.4 ASRC Channel Number Configuration Register (ASRCNCR)

The ASRC channel number configuration register (ASRCNCR) is a read/write register that sets the number of channels used by each ASRC conversion pair.

There are 10 channels available for distribution among 3 conversion pairs, they are ordered as 0,1,...,9. The bottom [0, ANCA-1] channels are used for pair A, the top [10-ANCC, 9] channels are used for pair C, and the [ANCA, ANCA+ANCB-1] channels are allocated for pair B. In case that ANCA=0, then the [0, ANCB-1] channels are assigned for pair B.

55.6.1.4.1 Offset

Register	Offset
ASRCNCR	Ch

55.6.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				ANCC				ANCB				ANCA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.4.3 Fields

Field	Description
31-24 —	- Reserved
23-12 —	- Reserved. Should be written as zero for compatibility.
11-8 ANCC	ANCC Number of C Channels[1] 0000 - 0 channels in C (Pair C is disabled) 0001 - 1 channel in C 0010 - 2 channels in C 0011 - 3 channels in C 0100 - 4 channels in C 0101 - 5 channels in C 0110 - 6 channels in C 0111 - 7 channels in C 1000 - 8 channels in C 1001 - 9 channels in C 1010 - 10 channels in C 1011-1111 - Should not be used.
7-4 ANCB	ANCB Number of B Channels 0000 - 0 channels in B (Pair B is disabled) 0001 - 1 channel in B 0010 - 2 channels in B 0011 - 3 channels in B 0100 - 4 channels in B 0101 - 5 channels in B 0110 - 6 channels in B 0111 - 7 channels in B 1000 - 8 channels in B 1001 - 9 channels in B 1010 - 10 channels in B 1011-1111 - Should not be used.
3-0 ANCA	ANCA Number of A Channels 0000 - 0 channels in A (Pair A is disabled) 0001 - 1 channel in A 0010 - 2 channels in A

Field	Description
	0011 - 3 channels in A 0100 - 4 channels in A 0101 - 5 channels in A 0110 - 6 channels in A 0111 - 7 channels in A 1000 - 8 channels in A 1001 - 9 channels in A 1010 - 10 channels in A 1011-1111 - Should not be used.

1. ANCC+ANCB+ANCA<=10. Hardware is not checking the constraint. Programmer should take the responsibility to ensure the constraint is satisfied.

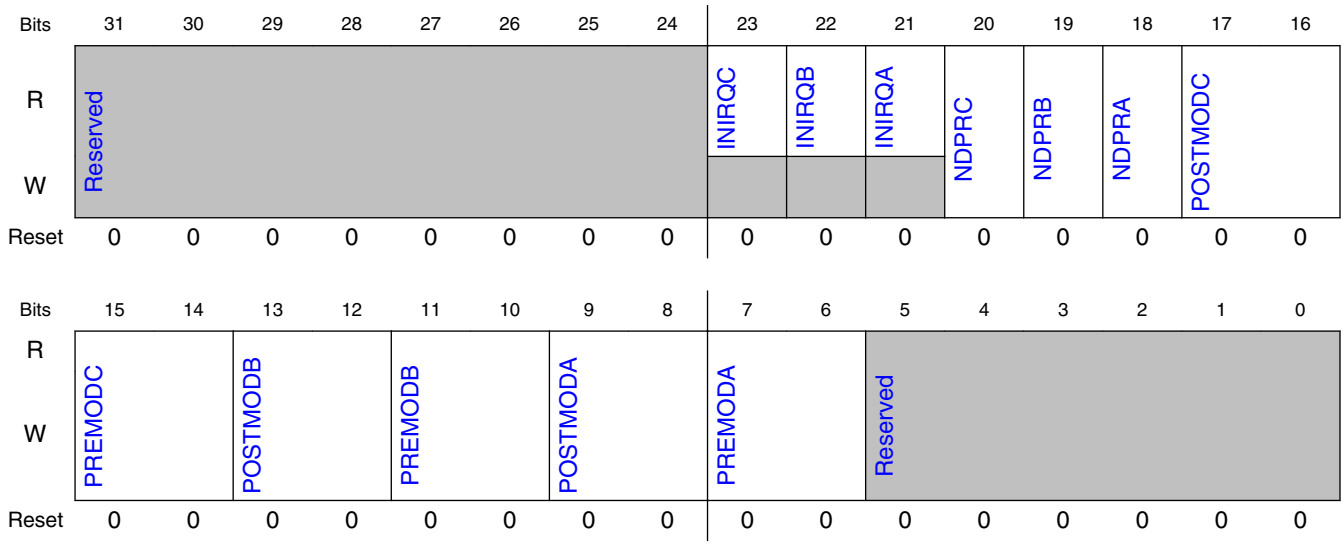
55.6.1.5 ASRC Filter Configuration Status Register (ASRCFG)

The ASRC configuration status register (ASRCFG) is a read/write register that sets and/or automatically senses the ASRC operations.

55.6.1.5.1 Offset

Register	Offset
ASRCFG	10h

55.6.1.5.2 Diagram



55.6.1.5.3 Fields

Field	Description
31-24 —	- Reserved
23 INIRQC	INIRQC Initialization for Conversion Pair C is served When this bit is 1, it means the initialization for conversion pair C is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR[ASREC]=0 or ASRCTR[ASRCEN]=0). 0 - Initialization for Conversion Pair C not served 1 - Initialization for Conversion Pair C served
22 INIRQB	INIRQB Initialization for Conversion Pair B is served When this bit is 1, it means the initialization for conversion pair B is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR[ASREB]=0 or ASRCTR[ASRCEN]=0). 0 - Initialization for Conversion Pair B not served 1 - Initialization for Conversion Pair B served
21 INIRQA	INIRQA Initialization for Conversion Pair A is served When this bit is 1, it means the initialization for conversion pair A is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR[ASREA]=0 or ASRCTR[ASRCEN]=0). 0 - Initialization for Conversion Pair A not served 1 - Initialization for Conversion Pair A served
20 NDPRC	NDPRC Not Use Default Parameters for RAM-stored Parameters For Conversion Pair C

Table continues on the next page...

Field	Description
	0 - Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 - Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
19 NDPRB	NDPRB Not Use Default Parameters for RAM-stored Parameters For Conversion Pair B 0 - Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 - Don't use default parameters for RAM-stored parameter. Use the parameters already stored in RAM.
18 NDPRA	NDPRA Not Use Default Parameters for RAM-stored Parameters For Conversion Pair A 0 - Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 - Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
17-16 POSTMODC	POSTMODC Post-Processing Configuration for Conversion Pair C These bits are read/written by the user if ASRCTR[ATSC]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSC]=1 (see ASRC Control Register for Pair C). These bits set the selection of the post-processing configuration. 00 - Select Upsampling-by-2 as defined in Signal Processing Flow. 01 - Select Direct-Connection as defined in Signal Processing Flow. 10 - Select Downsampling-by-2 as defined in Signal Processing Flow. 11 - Reserved.
15-14 PREMODC	PREMODC Pre-Processing Configuration for Conversion Pair C These bits are read/written by the user if ASRCTR[ATSC]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSC]=1 (see ASRC Control Register for Pair C). These bits set the selection of the pre-processing configuration. 00 - Select Upsampling-by-2. This is defined in Signal processing flow 01 - Select Direct-Connection. This is defined in Signal processing flow 10 - Select Downsampling-by-2. as defined in Signal processing flow 11 - Select passthrough mode. In this case, POSTMODC[1:0] have no use.
13-12 POSTMODB	POSTMODB Post-Processing Configuration for Conversion Pair B These bits are read/written by the user if ASRCTR[ATSB]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSB]=1 (see ASRC Control Register for Pair B). These bits set the selection of the post-processing configuration. 00 - Select Upsampling-by-2. This is defined in Signal processing flow 01 - Select Direct-Connection. This is defined in Signal processing flow 10 - Select Downsampling-by-2. This is defined in Signal processing flow 11 - Reserved.
11-10 PREMODB	PREMODB Pre-Processing Configuration for Conversion Pair B

Table continues on the next page...

Field	Description
	These bits are read/written by the user if ASRCTR[ATSB]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSB]=1 (see ASRC Control Register for Pair B). These bits set the selection of the pre-processing configuration. 00 - Select Upsampling-by-2. This is defined in Signal processing flow 01 - Select Direct-Connection. This is defined in Signal processing flow 10 - Select Downsampling-by-2. This is defined in Signal processing flow 11 - Select passthrough mode. In this case, POSTMODB[1:0] have no use.
9-8 POSTMODA	POSTMODA Post-Processing Configuration for Conversion Pair A These bits are read/written by the user if ASRCTR[ATSA]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSA]=1 (see ASRC Control Register for Pair A). These bits set the selection of the post-processing configuration. 00 - Select Upsampling-by-2. This is defined in Signal processing flow 01 - Select Direct-Connection. This is defined in Signal processing flow 10 - Select Downsampling-by-2. This is defined in Signal processing flow 11 - Reserved.
7-6 PREMODA	PREMODA Pre-Processing Configuration for Conversion Pair A These bits are read/written by the user if ASRCTR[ATSA]=0, and can also be automatically updated by the ASRC internal logic if ASRCTR[ATSA]=1 (see ASRC Control Register for Pair A). These bits set the selection of the pre-processing configuration. 00 - Select Upsampling-by-2. This is defined in Signal processing flow 01 - Select Direct-Connection. This is defined in Signal processing flow 10 - Select Downsampling-by-2. This is defined in Signal processing flow 11 - Select passthrough mode. In this case, POSTMODA[1:0] have no use.
5-0 —	- Reserved. Should be written as zero for compatibility.

55.6.1.6 ASRC Clock Source Register (ASRCSR)

The ASRC clock source register (ASRCSR) is a read/write register that controls the sources of the input and output clocks of the ASRC.

Please see External Signals

55.6.1.6.1 Offset

Register	Offset
ASRCSR	14h

55.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								AOCSC				AOCSB			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AOCSA				AICSC				AICSB				AICSA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.6.3 Fields

Field	Description
31-24 —	- Reserved
23-20 AOCSC	AOCSC **Output Clock Source C** 0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2 0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E

Table continues on the next page...

Field	Description
	1111 - clock disabled, connected to zero
19-16 AOCSB	AOCSB **Output Clock Source B** 0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2 0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E 1111 - clock disabled, connected to zero
15-12 AOCSA	AOCSA **Output Clock Source A** 0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2 0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E 1111 - clock disabled, connected to zero
11-8 AICSC	AICSC **Input Clock Source C**

Table continues on the next page...

Field	Description
	0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2 0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E 1111 - clock disabled, connected to zero
7-4 AICSB	AICSB **Input Clock Source B** 0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2 0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E 1111 - clock disabled, connected to zero
3-0 AICSA	AICSA **Input Clock Source A** 0000 - bit clock 0 0001 - bit clock 1 0010 - bit clock 2

Field	Description
	0011 - bit clock 3 0100 - bit clock 4 0101 - bit clock 5 0110 - bit clock 6 0111 - bit clock 7 1000 - bit clock 8 1001 - bit clock 9 1010 - bit clock A 1011 - bit clock B 1100 - bit clock C 1101 - bit clock D 1110 - bit clock E 1111 - clock disabled, connected to zero

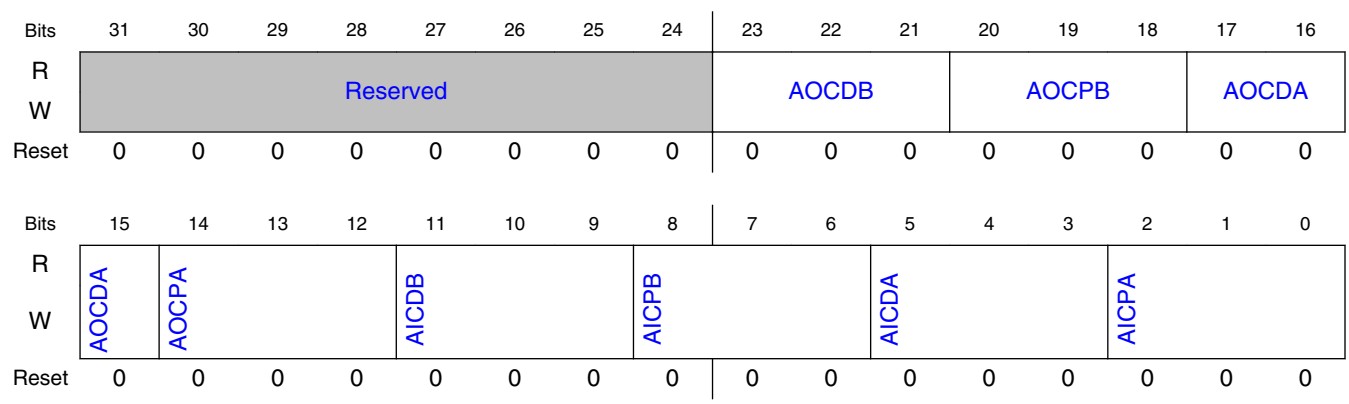
55.6.1.7 ASRC Clock Divider Register 1 (ASRCDR1)

The ASRC clock divider register (ASRCDR1) is a read/write register that controls the division factors of the ASRC input and output clock sources.

55.6.1.7.1 Offset

Register	Offset
ASRCDR1	18h

55.6.1.7.2 Diagram



55.6.1.7.3 Fields

Field	Description
31-24 —	- Reserved
23-21 AOCDB	AOCDB Output Clock Divider B Specify the divide ratio of the output clock divider B. The divide ratio may range from 1 to 8 (AOCDB[2:0] = 000 to 111).
20-18 AOCPB	AOCPB Output Clock Prescaler B Specify the prescaling factor of the output prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
17-15 AOCDA	AOCDA Output Clock Divider A Specify the divide ratio of the output clock divider A. The divide ratio may range from 1 to 8 (AOCDA[2:0] = 000 to 111).
14-12 AOCPA	AOCPA Output Clock Prescaler A Specify the prescaling factor of the output prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.
11-9 AICDB	AICDB Input Clock Divider B Specify the divide ratio of the input clock divider B. The divide ratio may range from 1 to 8 (AICDB[2:0] = 000 to 111).
8-6 AICPB	AICPB Input Clock Prescaler B Specify the prescaling factor of the input prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
5-3 AICDA	AICDA Input Clock Divider A Specify the divide ratio of the input clock divider A. The divide ratio may range from 1 to 8 (AICDA[2:0] = 000 to 111).
2-0 AICPA	AICPA Input Clock Prescaler A Specify the prescaling factor of the input prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.

55.6.1.8 ASRC Clock Divider Register 2 (ASRCDR2)

The ASRC clock divider register (ASRCDR2) is a read/write register that controls the division factors of the ASRC input and output clock sources.

55.6.1.8.1 Offset

Register	Offset
ASRCDR2	1Ch

55.6.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				AOCDC				AOCPC			AICDC			AICPC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.8.3 Fields

Field	Description
31-24 —	- Reserved
23-12 —	- Reserved. Should be written as zero for compatibility.
11-9 AOCDC	AOCDC Output Clock Divider C Specify the divide ratio of the output clock divider C. The divide ratio may range from 1 to 8 (AOCDC[2:0] = 000 to 111).
8-6 AOCPC	AOCPC Output Clock Prescaler C Specify the prescaling factor of the output prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.
5-3	AICDC

Table continues on the next page...

Field	Description
AICDC	Input Clock Divider C Specify the divide ratio of the input clock divider C. The divide ratio may range from 1 to 8 (AICDC[2:0] = 000 to 111).
2-0 AICPC	AICPC Input Clock Prescaler C Specify the prescaling factor of the input prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.

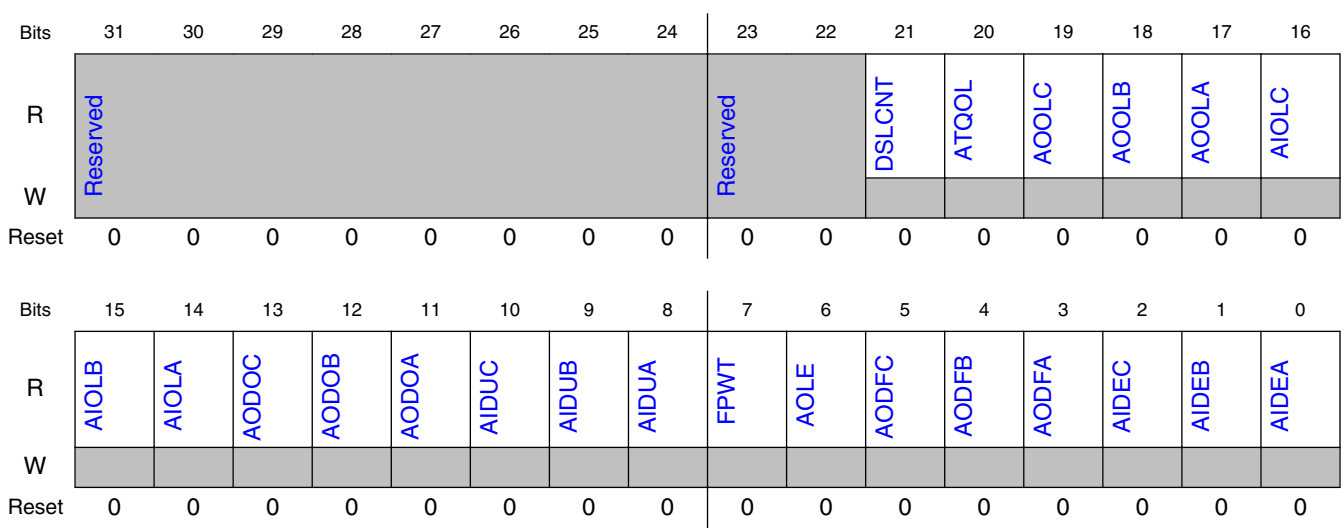
55.6.1.9 ASRC Status Register (ASRSTR)

The ASRC status register (ASRSTR) is a read/write register used by the processor core to examine the status of the ASRC block and clear the overload interrupt request and AOLE flag bit. Read of the status register returns the current state of ASRC.

55.6.1.9.1 Offset

Register	Offset
ASRSTR	20h

55.6.1.9.2 Diagram



55.6.1.9.3 Fields

Field	Description
31-24 —	- Reserved
23-22 —	- Reserved. Should be written as zero for compatibility.
21 DSLCNT	DSLCNT Digital Servo Loop (DSL) Counter Input to FIFO ready When set, this bit indicates that new DSL counter information is stored in the internal ASRC FIFO. When clear, this bit indicates that new DSL counter information is in the process of storage into the internal ASRC FIFO. When ASRIER[AFPWE]=1, the rising edge of this signal proposes an interrupt request. Writing any value with this bit set clears the interrupt request proposed by the rising edge of this bit. 0 - New DSL counter information is in the process of storage into the internal ASRC FIFO 1 - New DSL counter information is stored in the internal ASRC FIFO
20 ATQOL	ATQOL Task Queue FIFO overload When set, this bit indicates that task queue FIFO logic is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - Task queue FIFO logic is not oveloaded 1 - Task queue FIFO logic is oveloaded
19 AOOLC	AOOLC Pair C Output Task Overload When set, this bit indicates that pair C output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - Pair C output task is not oveloaded 1 - Pair C output task is oveloaded
18 AOOLB	AOOLB Pair B Output Task Overload When set, this bit indicates that pair B output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - Pair B output task is not oveloaded 1 - Pair B output task is oveloaded
17 AOOLA	AOOLA Pair A Output Task Overload When set, this bit indicates that pair A output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRSTR[AOLE] as 1.

Table continues on the next page...

Field	Description
	0 - Pair A output task is not oveloaded
	1 - Pair A output task is oveloaded
16	AIOLC
AIOLC	Pair C Input Task Overload
	When set, this bit indicates that pair C input task is oveloaded. This may help to check the reason why overload interrupt happens.
	The bit is cleared when writing ASRSTR[AOLE] as 1.
	0 - Pair C input task is not oveloaded
	1 - Pair C input task is oveloaded
15	AIOLB
AIOLB	Pair B Input Task Overload
	When set, this bit indicates that pair B input task is oveloaded. This may help to check the reason why overload interrupt happens.
	The bit is cleared when writing ASRSTR[AOLE] as 1.
	0 - Pair B input task is not oveloaded
	1 - Pair B input task is oveloaded
14	AIOLA
AIOLA	Pair A Input Task Overload
	When set, this bit indicates that pair A input task is oveloaded. This may help to check the reason why overload interrupt happens.
	The bit is cleared when writing ASRSTR[AOLE] as 1.
	0 - Pair A input task is not oveloaded
	1 - Pair A input task is oveloaded
13	AODOC
AODOC	Output Data Buffer C has overflowed
	When set, this bit indicates that output data buffer C has overflowed. When clear, this bit indicates that output data buffer C has not overflowed
	The bit is cleared when writing ASRSTR[AOLE] as 1.
	0 - No Overflow in Output data buffer C
	1 - Overflow in Output data buffer C
12	AODOB
AODOB	Output Data Buffer B has overflowed
	When set, this bit indicates that output data buffer B has overflowed. When clear, this bit indicates that output data buffer B has not overflowed
	The bit is cleared when writing ASRSTR[AOLE] as 1.
	0 - No Overflow in Output data buffer B
	1 - Overflow in Output data buffer B
11	AODOA
AODOA	Output Data Buffer A has overflowed
	When set, this bit indicates that output data buffer A has overflowed. When clear, this bit indicates that output data buffer A has not overflowed

Table continues on the next page...

Field	Description
	The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - No Overflow in Output data buffer A 1 - Overflow in Output data buffer A
10 AIDUC	AIDUC Input Data Buffer C has underflowed When set, this bit indicates that input data buffer C has underflowed. When clear, this bit indicates that input data buffer C has not underflowed. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - No Underflow in Input data buffer C 1 - Underflow in Input data buffer C
9 AIDUB	AIDUB Input Data Buffer B has underflowed When set, this bit indicates that input data buffer B has underflowed. When clear, this bit indicates that input data buffer B has not underflowed. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - No Underflow in Input data buffer B 1 - Underflow in Input data buffer B
8 AIDUA	AIDUA Input Data Buffer A has underflowed When set, this bit indicates that input data buffer A has underflowed. When clear, this bit indicates that input data buffer A has not underflowed. The bit is cleared when writing ASRSTR[AOLE] as 1. 0 - No Underflow in Input data buffer A 1 - Underflow in Input data buffer A
7 FPWT	FPWT FP is in wait states This bit is for debug only. When set, this bit indicates that ASRC is in wait states. When clear, this bit indicates that ASRC is not in wait states. 0 - ASRC is not in wait state 1 - ASRC is in wait state
6 AOLE	AOLE Overload Error Flag When set, this bit indicates that the task rate is too high for the ASRC to handle. The reasons for overload may be: - too high input clock frequency, - too high output clock frequency, - incorrect selection of the pre-filter, - low ASRC processing clock,

Table continues on the next page...

Field	Description
	- too many channels, - underrun, - or any combination of the reasons above. Since the ASRC uses the same hardware resources to perform various tasks, the real reason for the overload is not straight forward, and it should be carefully analyzed by the programmer. If ASRIER[AOLIE]=1, an interrupt is proposed when this bit is set. Write any value with this bit set as one into the status register clears this bit and the interrupt request proposed by this bit. 0 - No overload 1 - Task rate is too high
5 AODFC	AODFC Number of data in Output Data Buffer C is greater than threshold When set, this bit indicates that number of data already existing in ASRDOC is greater than threshold and the processor can read data from ASRDOC. A DMA request is always generated when the AODFC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has not yet been met and no data output C interrupt is generated 1 - When AODFC is set, the ASRC generates data output C interrupt request to the processor if ASRIER[ADOEC] = 1
4 AODFB	AODFB Number of data in Output Data Buffer B is greater than threshold When set, this bit indicates that number of data already existing in ASRDOB is greater than threshold and the processor can read data from ASRDOB. A DMA request is always generated when the AODFB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has not yet been met and no data output B interrupt is generated 1 - When AODFB is set, the ASRC generates data output B interrupt request to the processor if ASRIER[ADOEB] = 1
3 AODFA	AODFA Number of data in Output Data Buffer A is greater than threshold When set, this bit indicates that number of data already existing in ASRDOA is greater than threshold and the processor can read data from ASRDOA. A DMA request is always generated when the AODFA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has not yet been met and no data output A interrupt is generated 1 - When AODFA is set, the ASRC generates data output A interrupt request to the processor if ASRIER[ADOEA] = 1
2 AIDEC	AIDEC Number of data in Input Data Buffer C is less than threshold When set, this bit indicates that the number of data still available in ASRDIC is less than the threshold and the processor can write data to ASRDIC. A DMA request is always generated when the AIDEC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has been met and no data input C interrupt is generated 1 - When AIDEC is set, the ASRC generates data input C interrupt request to the processor if ASRIER[AIDEC] = 1
1	AIDEB

Table continues on the next page...

Field	Description
AIDEB	Number of data in Input Data Buffer B is less than threshold When set, this bit indicates that number of data still available in ASRDIB is less than threshold and the processor can write data to ASRDIB. A DMA request is always generated when the AIDEB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has been met and no data input B interrupt is generated 1 - When AIDEB is set, the ASRC generates data input B interrupt request to the processor if ASRIER[AIDEB] = 1
0 AIDEA	AIDEA Number of data in Input Data Buffer A is less than threshold When set, this bit indicates that number of data still available in ASRDIA is less than the threshold and the processor can write data to ASRDIA. A DMA request is always generated when the AIDEA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event. 0 - The threshold has been met and no data input A interrupt is generated 1 - When AIDEA is set, the ASRC generates data input A interrupt request to the processor if ASRIER[AIDEA] = 1

55.6.1.10 ASRC Parameter Register n (ASRPM1 - ASRPM5)

Parameter registers determine the performance of ASRC.

The parameter registers must be initialized by software before ASRC is enabled. Recommended values are given in the table below:

Table 55-10. ASRC Parameter Registers (ASRPM1~ASRPM5)

Register	Offset	Access	Reset Value	Recommend Value
ASRPM1	0x40	R/W	0x00_0000	0x7fffff
ASRPM2	0x44	R/W	0x00_0000	0x255555
ASRPM3	0x48	R/W	0x00_0000	0xff7280
ASRPM4	0x4C	R/W	0x00_0000	0xff7280
ASRPM5	0x50	R/W	0x00_0000	0xff7280

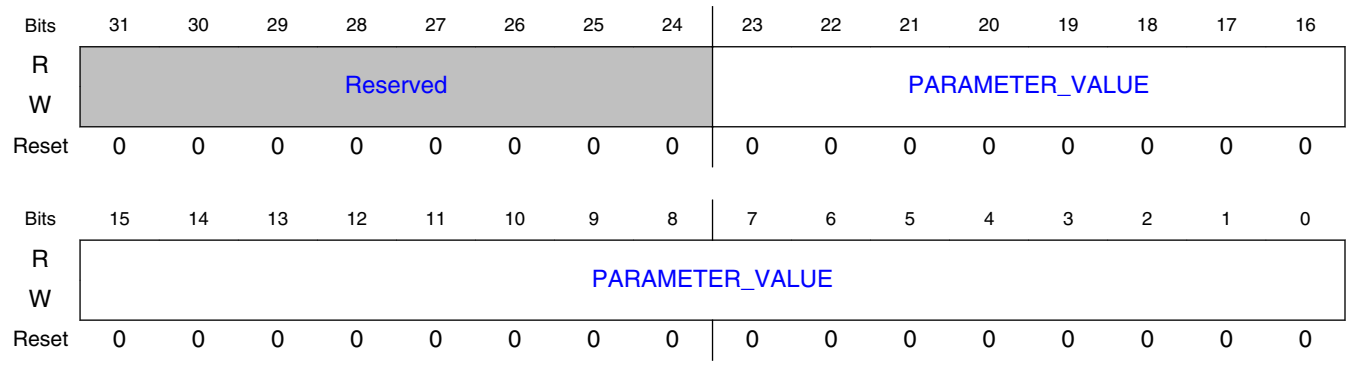
55.6.1.10.1 Offset

Register	Offset
ASRPM1	40h
ASRPM2	44h
ASRPM3	48h

Table continues on the next page...

Register	Offset
ASRPM4	4Ch
ASRPM5	50h

55.6.1.10.2 Diagram



55.6.1.10.3 Fields

Field	Description
31-24 —	- Reserved
23-0 PARAMETER_VALUE	PARAMETER_VALUE See recommended values table.

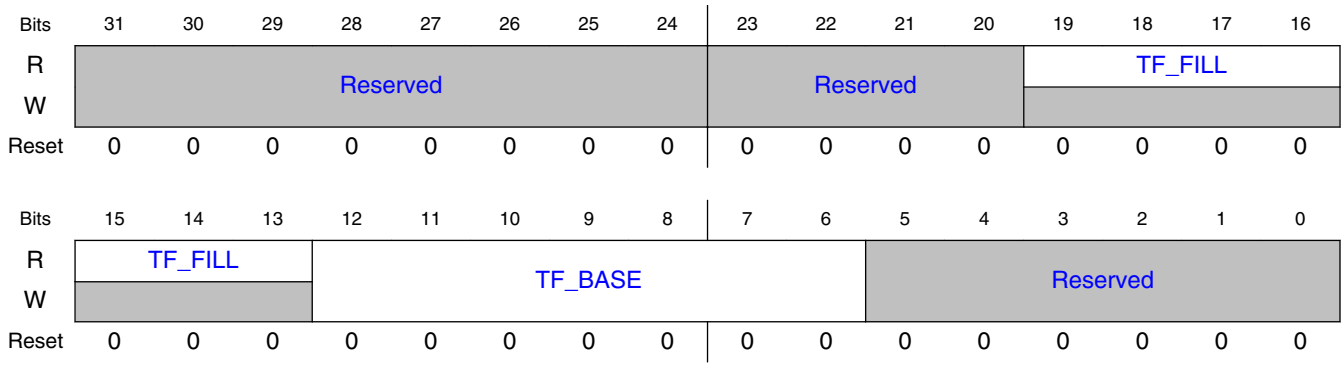
55.6.1.11 ASRC Task Queue FIFO Register 1 (ASRTFR1)

This register defines and shows the parameters for ASRC inner task queue FIFOs.

55.6.1.11.1 Offset

Register	Offset
ASRTFR1	54h

55.6.1.11.2 Diagram



55.6.1.11.3 Fields

Field	Description
31-24 —	- Reserved
23-20 —	- Reserved. Should be written as zero for compatibility.
19-13 TF_FILL	TF_FILL Current number of entries in task queue FIFO.
12-6 TF_BASE	TF_BASE Base address for task queue FIFO. Set to 0x7C.
5-0 —	- Reserved. Should be written as zero for compatibility.

55.6.1.12 ASRC Channel Counter Register (ASRCCR)

The ASRC channel counter register (ASRCCR) is a read/write register that sets and reflects the current specific input/output FIFO being accessed through shared peripheral bus for each ASRC conversion pair.

55.6.1.12.1 Offset

Register	Offset
ASRCCR	5Ch

55.6.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ACOC				ACOB			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ACOA				ACIC				ACIB				ACIA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.12.3 Fields

Field	Description
31-24 —	- Reserved
23-20 ACOC	ACOC The channel counter for Pair C's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's output FIFO's usage. The value can be any value between [0, ANCC-1]
19-16 ACOB	ACOB The channel counter for Pair B's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's output FIFO's usage. The value can be any value between [0, ANCB-1]
15-12 ACOA	ACOA The channel counter for Pair A's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's output FIFO's usage. The value can be any value between [0, ANCA-1]
11-8 ACIC	ACIC The channel counter for Pair C's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's input FIFO's usage. The value can be any value between [0, ANCC-1]
7-4 ACIB	ACIB The channel counter for Pair B's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's input FIFO's usage. The value can be any value between [0, ANCB-1]
3-0 ACIA	ACIA The channel counter for Pair A's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's input FIFO's usage. The value can be any value between [0, ANCA-1]

55.6.1.13 ASRC Data Input Register for Pair x (ASRDIA - ASRDIC)

These registers are the interface registers for the audio data input of pair A,B,C respectively. They are backed by FIFOs.

55.6.1.13.1 Offset

Register	Offset
ASRDIA	60h
ASRDIB	68h
ASRDIC	70h

55.6.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W									DATA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.13.3 Fields

Field	Description
31-24 —	- Reserved
23-0 DATA	DATA Audio data input

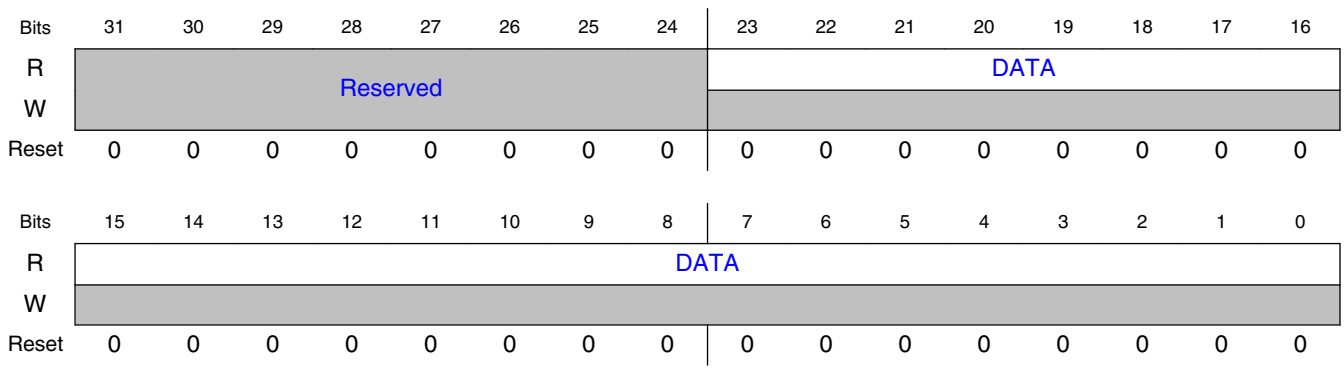
55.6.1.14 ASRC Data Output Register for Pair x (ASRDOA - ASRDOC)

These registers are the interface registers for the audio data output of pair A,B,C respectively. They are backed by FIFOs.

55.6.1.14.1 Offset

Register	Offset
ASRDOA	64h
ASRDOB	6Ch
ASRDOC	74h

55.6.1.14.2 Diagram



55.6.1.14.3 Fields

Field	Description
31-24 —	- Reserved
23-0 DATA	DATA Audio data output

55.6.1.15 ASRC Ideal Ratio for Pair A-High Part (ASRIDRHA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = $Fs_{inA}/Fs_{outA} = Ts_{outA}/Ts_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRA, IDRA]=2'b11.

55.6.1.15.1 Offset

Register	Offset
ASRIDRHA	80h

55.6.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOA_H							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.15.3 Fields

Field	Description
31-24 —	- Reserved
23-8 —	- Reserved
7-0 IDRATIOA_H	IDRATIOA_H IDRATIOA[31:24]. High part of ideal ratio value for pair A

55.6.1.16 ASRC Ideal Ratio for Pair A -Low Part (ASRIDRLA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = $Fs_{inA}/Fs_{outA} = Ts_{outA}/Ts_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRA, IDRA]=2'b11.

55.6.1.16.1 Offset

Register	Offset
ASRIDRLA	84h

55.6.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								IDRATIOA_L							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDRATIOA_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.16.3 Fields

Field	Description
31-24 —	- Reserved
23-0 IDRATIOA_L	IDRATIOA_L IDRATIOA[23:0]. Low part of ideal ratio value for pair A

55.6.1.17 ASRC Ideal Ratio for Pair B-High Part (ASRIDRHB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB}/Fs_{outB} = Ts_{outB}/Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRB, IDRB]=2'b11.

55.6.1.17.1 Offset

Register	Offset
ASRIDRHB	88h

55.6.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOB_H							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.17.3 Fields

Field	Description
31-24 —	- Reserved
23-8 —	- Reserved
7-0 IDRATIOB_H	IDRATIOB_H IDRATIOB[31:24]. High part of ideal ratio value for pair B.

55.6.1.18 ASRC Ideal Ratio for Pair B-Low Part (ASRIDRLB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB}/Fs_{outB} = Ts_{outB}/Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRB, IDRB]=2'b11.

55.6.1.18.1 Offset

Register	Offset
ASRIDRLB	8Ch

55.6.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								IDRATIOB_L							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDRATIOB_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.18.3 Fields

Field	Description
31-24 —	- Reserved
23-0 IDRATIOB_L	IDRATIOB_L IDRATIOB[23:0]. Low part of ideal ratio value for pair B.

55.6.1.19 ASRC Ideal Ratio for Pair C-High Part (ASRIDRHC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC}/Fs_{outC} = Ts_{outC}/Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRC, IDRC]=2'b11.

55.6.1.19.1 Offset

Register	Offset
ASRIDRHC	90h

55.6.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOC_H							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.19.3 Fields

Field	Description
31-24 —	- Reserved
23-8 —	- Reserved
7-0 IDRATIOC_H	IDRATIOC_H IDRATIOC[31:24]. High part of ideal ratio value for pair C.

55.6.1.20 ASRC Ideal Ratio for Pair C-Low Part (ASRIDRLC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC}/Fs_{outC} = Ts_{outC}/Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR[USRC, IDRC]=2'b11.

55.6.1.20.1 Offset

Register	Offset
ASRIDRLC	94h

55.6.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								IDRATIOC_L							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDRATIOC_L															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55.6.1.20.3 Fields

Field	Description
31-24 —	- Reserved
23-0 IDRATIOC_L	IDRATIOC_L IDRATIOC[23:0]. Low part of ideal ratio value for pair C.

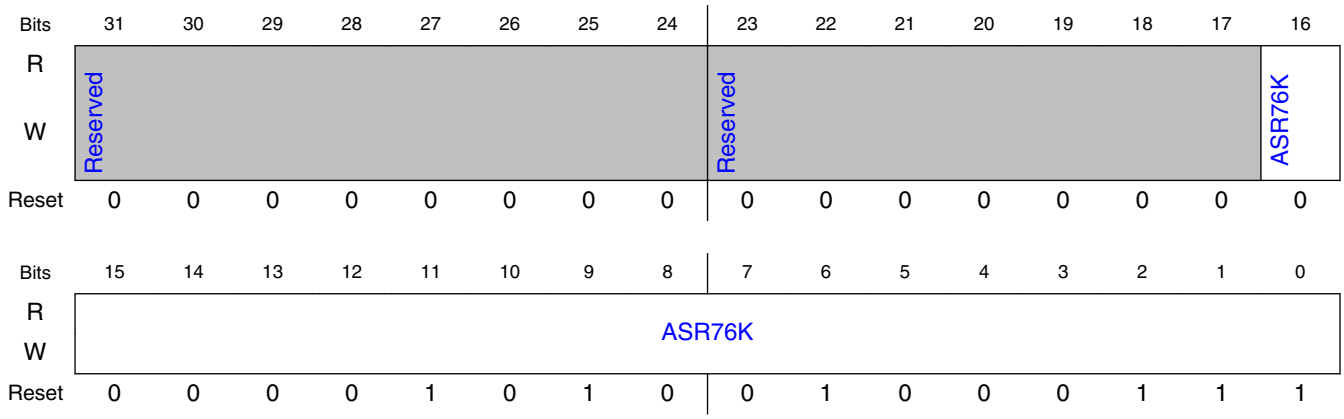
55.6.1.21 ASRC 76 kHz Period in terms of ASRC processing clock (ASR76K)

The register (ASR76K) holds the period of the 76 kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC}. ASR76K = Fs_{ASRC}/Fs_{76k}. Reset value is 0x0A47 which assumes that Fs_{ASRC}=200 MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Control Register and ASRC Filter Configuration Status Register). In a system when Fs_{ASRC} = 133 MHz, the value should be assigned explicitly as 0x06D6 in user application code.

55.6.1.21.1 Offset

Register	Offset
ASR76K	98h

55.6.1.21.2 Diagram



55.6.1.21.3 Fields

Field	Description
31-24 —	- Reserved
23-17 —	- Reserved
16-0 ASR76K	ASR76K Value for the period of the 76 kHz sampling clock.

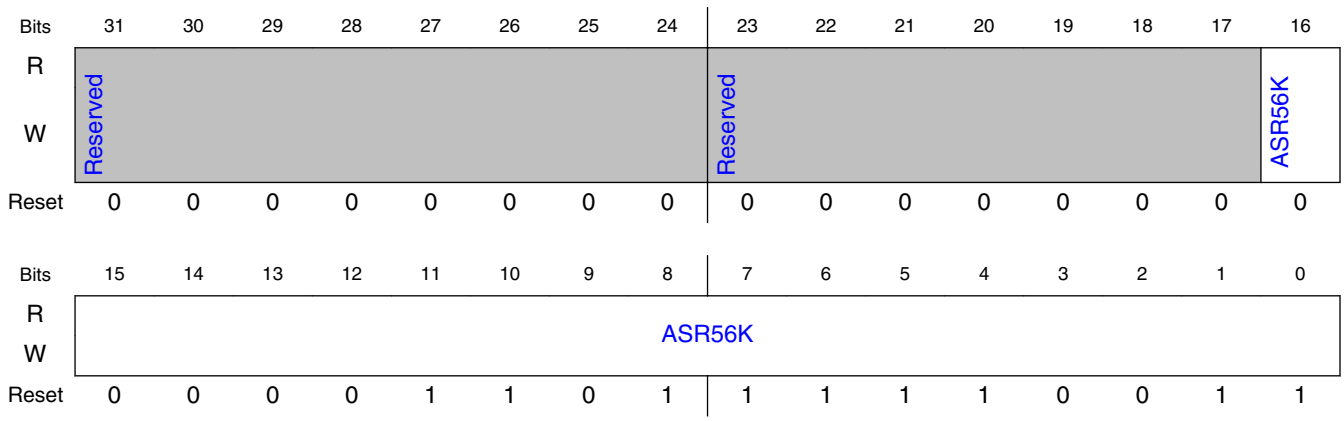
55.6.1.22 ASRC 56 kHz Period in terms of ASRC processing clock (ASR56K)

The register (ASR56K) holds the period of the 56 kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC}. ASR56K = Fs_{ASRC}/Fs_{56k}. Reset value is 0x0DF3 which assumes that Fs_{ASRC} = 200 MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Control Register and ASRC Misc Control Register 1 for Pair C). In a system when Fs_{ASRC} = 133 MHz, the value should be assigned explicitly as 0x0947 in user application code.

55.6.1.22.1 Offset

Register	Offset
ASR56K	9Ch

55.6.1.22.2 Diagram



55.6.1.22.3 Fields

Field	Description
31-24 —	- Reserved
23-17 —	- Reserved
16-0 ASR56K	ASR56K Value for the period of the 56 kHz sampling clock

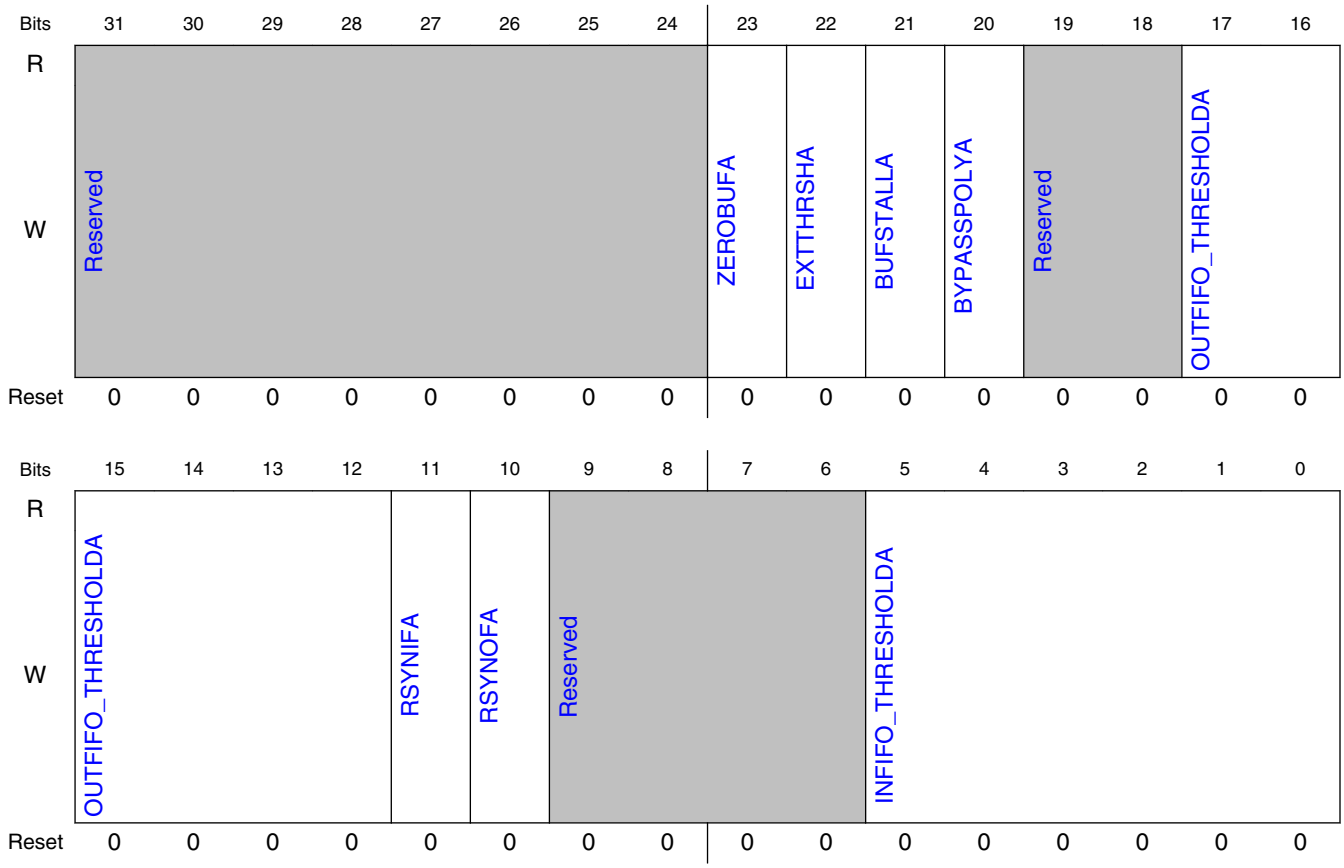
55.6.1.23 ASRC Misc Control Register for Pair A (ASRMCRA)

The register (ASRMCRA) is used to control Pair A internal logic.

55.6.1.23.1 Offset

Register	Offset
ASRMCRA	A0h

55.6.1.23.2 Diagram



55.6.1.23.3 Fields

Field	Description
31-24 —	- Reserved
23 ZEROBUFA	ZEROBUFA Initialize buffer of Pair A when pair A is enabled. Always clear option. This bit is used to control whether the buffer is to be zeroized when pair A is enabled. 0 - Zeroize the buffer 1 - Don't zeroize the buffer
22 EXTTHRSHA	EXTTHRSHA Use external thresholds for FIFO control of Pair A This bit determines whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair A. 0 - Use default thresholds.

Table continues on the next page...

Field	Description
	1 - Use external defined thresholds.
21 BUFSTALLA	BUFSTALLA Stall Pair A conversion in case of Buffer Near Empty/Full Condition This bit determines whether the near empty/full FIFO condition stalls the rate conversion for pair A. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 0 - Don't stall Pair A conversion even in case of near empty/full FIFO conditions. 1 - Stall Pair A conversion in case of near empty/full FIFO conditions.
20 BYPASSPOLYA	BYPASSPOLYA Bypass Polyphase Filtering for Pair A This bit determines whether the polyphase filtering part of Pair A conversion is bypassed. 0 - Don't bypass polyphase filtering. 1 - Bypass polyphase filtering.
19-18 —	- Reserved. Should be written as zero for future compatibility.
17-12 OUTFIFO_THRESHOLDA	OUTFIFO_THRESHOLDA The threshold for Pair A's output FIFO per channel These bits stand for the threshold for Pair A's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFA	RSYNIFA Re-sync Input FIFO Channel Counter If bit set, force ASRCCR[ACIA]=0. If bit clear, untouch ASRCCR[ACIA]. 0 - Do not touch ASRCCR[ACIA] 1 - Force ASRCCR[ACIA]=0
10 RSYNOFA	RSYNOFA Re-sync Output FIFO Channel Counter If bit set, force ASRCCR[ACOA]=0. If bit clear, untouch ASRCCR[ACOA]. 0 - Do not touch ASRCCR[ACOA] 1 - Force ASRCCR[ACOA]=0
9-6 —	- Reserved. Should be written as zero for future compatibility.
5-0 INFIFO_THRESHOLDA	INFIFO_THRESHOLDA The threshold for Pair A's input FIFO per channel These bits stand for the threshold for Pair A's input FIFO per channel. Possible range is [0,63].

Field	Description
	When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

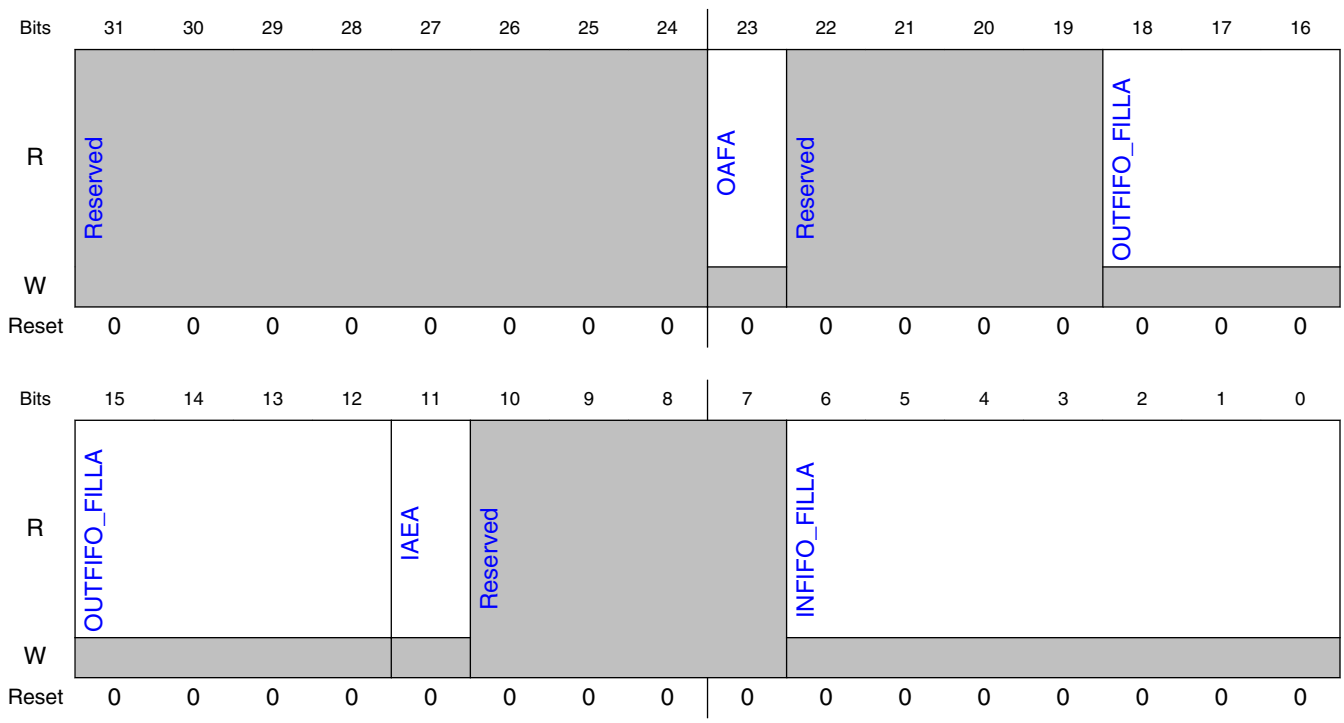
55.6.1.24 ASRC FIFO Status Register for Pair A (ASRFSTA)

The register (ASRFSTA) is used to show Pair A internal FIFO conditions.

55.6.1.24.1 Offset

Register	Offset
ASRFSTA	A4h

55.6.1.24.2 Diagram



55.6.1.24.3 Fields

Field	Description
31-24 —	- Reserved
23 OAFA	OAFA Output FIFO is near Full for Pair A This bit is to indicate whether the output FIFO of Pair A is near full. 0 - Output FIFO is not near full for Pair A 1 - Output FIFO is near full for Pair A
22-19 —	- Reserved. Should be written as zero for future compatibility.
18-12 OUTFIFO_FILLA	OUTFIFO_FILLA The fillings for Pair A's output FIFO per channel These bits stand for the fillings for Pair A's output FIFO per channel. Possible range is [0,64].
11 IAEA	IAEA Input FIFO is near Empty for Pair A This bit is to indicate whether the input FIFO of Pair A is near empty. 0 - Input FIFO is not near empty for Pair A 1 - Input FIFO is near empty for Pair A
10-7 —	- Reserved. Should be written as zero for future compatibility.
6-0 INFIFO_FILLA	INFIFO_FILLA The fillings for Pair A's input FIFO per channel These bits stand for the fillings for Pair A's input FIFO per channel. Possible range is [0,64].

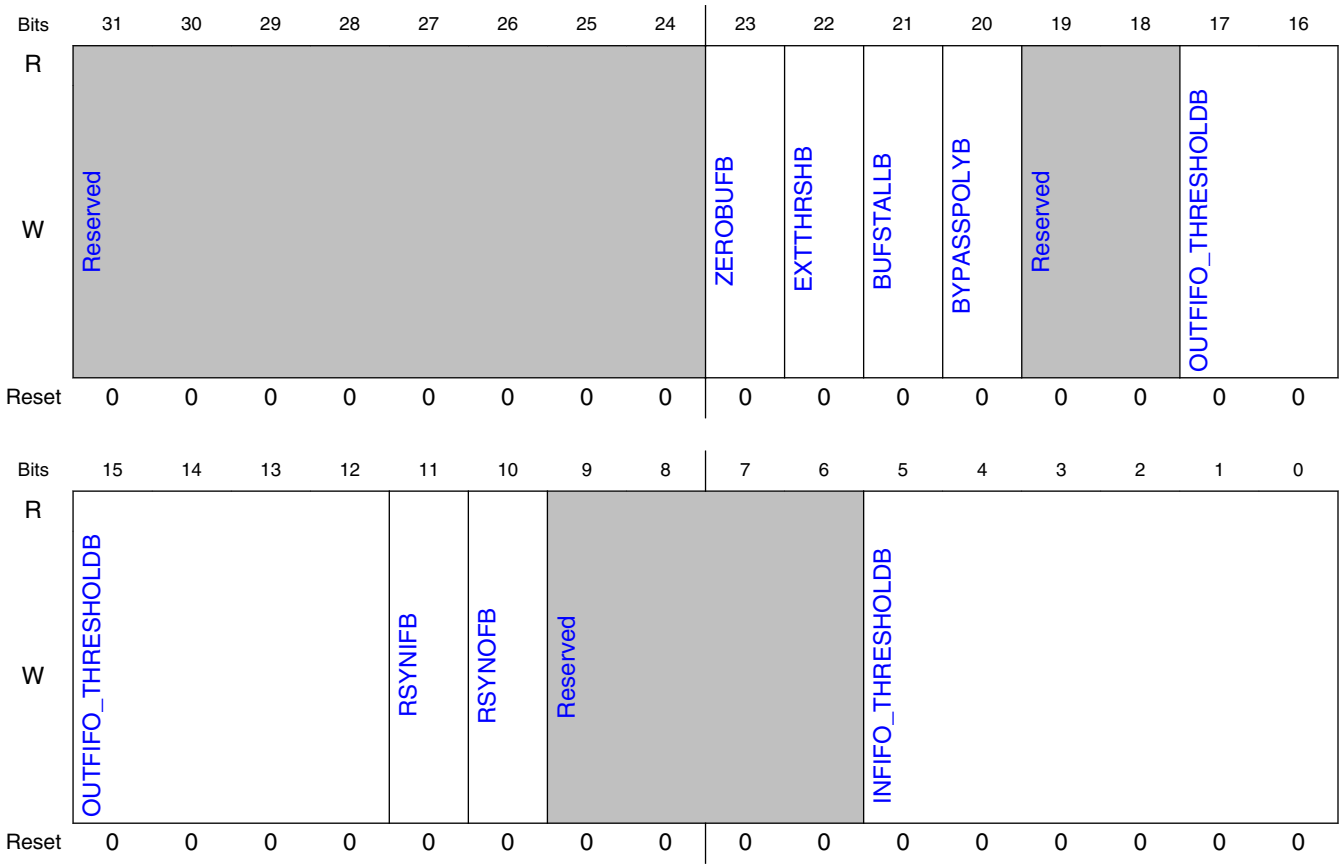
55.6.1.25 ASRC Misc Control Register for Pair B (ASRMCRB)

The register (ASRMCRB) is used to control Pair B internal logic.

55.6.1.25.1 Offset

Register	Offset
ASRMCRB	A8h

55.6.1.25.2 Diagram



55.6.1.25.3 Fields

Field	Description
31-24 —	- Reserved
23 ZEROBUFB	ZEROBUFB Initialize buffer of Pair B when pair B is enabled This bit is used to control whether the buffer is to be zeroized when pair B is enabled. 0 - Zeroize the buffer 1 - Don't zeroize the buffer
22 EXTTHRSHB	EXTTHRSHB Use external thresholds for FIFO control of Pair B This bit determines whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair B. 0 - Use default thresholds. 1 - Use external defined thresholds.
21	BUFSTALLB

Table continues on the next page...

Field	Description
BUFSTALLB	Stall Pair B conversion in case of Buffer Near Empty/Full Condition This bit determines whether the near empty/full FIFO condition stalls the rate conversion for pair B. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 0 - Don't stall Pair B conversion even in case of near empty/full FIFO conditions. 1 - Stall Pair B conversion in case of near empty/full FIFO conditions.
20 BYPASSPOLYB	BYPASSPOLYB Bypass Polyphase Filtering for Pair B This bit determines whether the polyphase filtering part of Pair B conversion is bypassed. 0 - Don't bypass polyphase filtering. 1 - Bypass polyphase filtering.
19-18 —	- Reserved. Should be written as zero for future compatibility.
17-12 OUTFIFO_THRESHOLDB	OUTFIFO_THRESHOLDB The threshold for Pair B's output FIFO per channel These bits stand for the threshold for Pair B's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFB	RSYNIFB Re-sync Input FIFO Channel Counter If bit set, force ASRCCR[ACIB]=0. If bit clear, untouch ASRCCR[ACIB]. 0 - Do not touch ASRCCR[ACIB] 1 - Force ASRCCR[ACIB]=0
10 RSYNOFB	RSYNOFB Re-sync Output FIFO Channel Counter If bit set, force ASRCCR[ACOB]=0. If bit clear, untouch ASRCCR[ACOB]. 0 - Do not touch ASRCCR[ACOB] 1 - Force ASRCCR[ACOB]=0
9-6 —	- Reserved. Should be written as zero for future compatibility.
5-0 INFIFO_THRESHOLDB	INFIFO_THRESHOLDB The threshold for Pair B's input FIFO per channel These bits stand for the threshold for Pair B's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set;

Field	Description
	when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

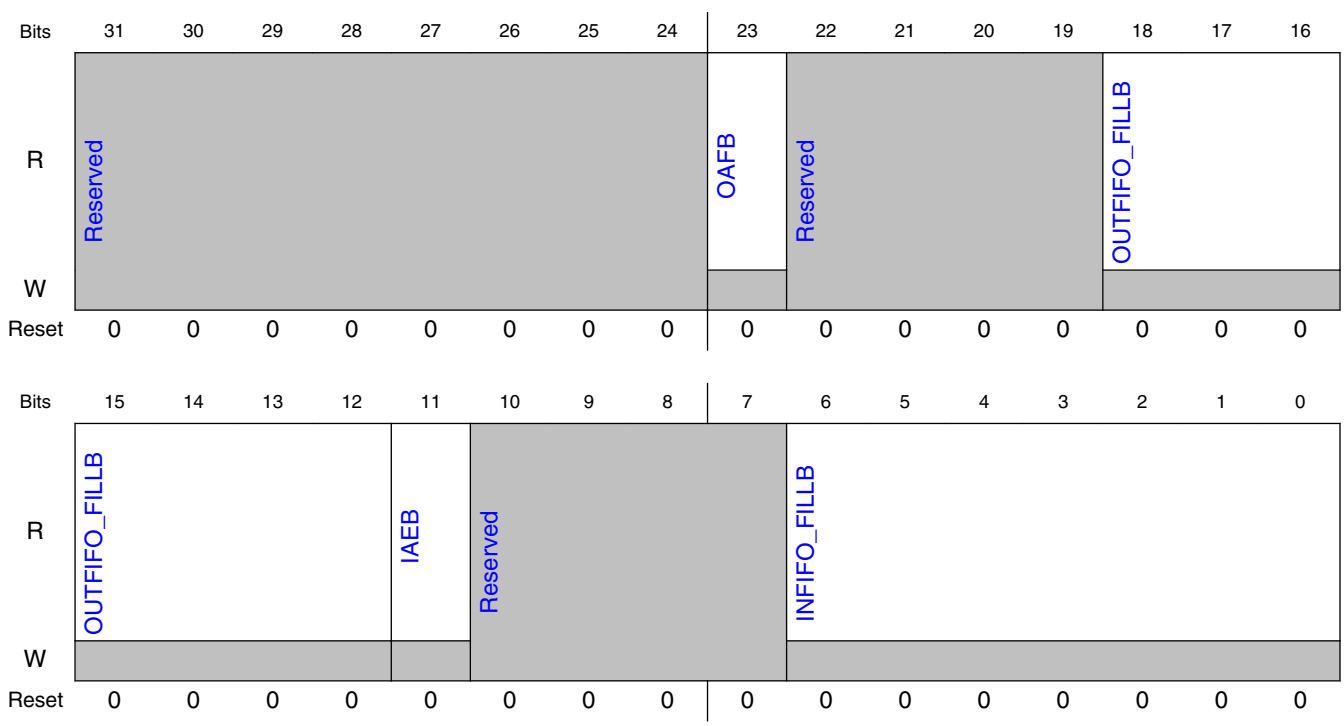
55.6.1.26 ASRC FIFO Status Register for Pair B (ASRFSTB)

The register (ASRFSTB) is used to show Pair B internal FIFO conditions.

55.6.1.26.1 Offset

Register	Offset
ASRFSTB	ACh

55.6.1.26.2 Diagram



55.6.1.26.3 Fields

Field	Description
31-24 —	- Reserved
23 OAFB	OAFB Output FIFO is near Full for Pair B This bit is to indicate whether the output FIFO of Pair B is near full. 0 - Output FIFO is not near full for Pair B 1 - Output FIFO is near full for Pair B
22-19 —	- Reserved. Should be written as zero for future compatibility.
18-12 OUTFIFO_FILLB	OUTFIFO_FILLB The fillings for Pair B's output FIFO per channel These bits stand for the fillings for Pair B's output FIFO per channel. Possible range is [0,64].
11 IAEB	IAEB Input FIFO is near Empty for Pair B This bit is to indicate whether the input FIFO of Pair B is near empty. 0 - Input FIFO is not near empty for Pair B 1 - Input FIFO is near empty for Pair B
10-7 —	- Reserved. Should be written as zero for future compatibility.
6-0 INFIFO_FILLB	INFIFO_FILLB The fillings for Pair B's input FIFO per channel These bits stand for the fillings for Pair B's input FIFO per channel. Possible range is [0,64].

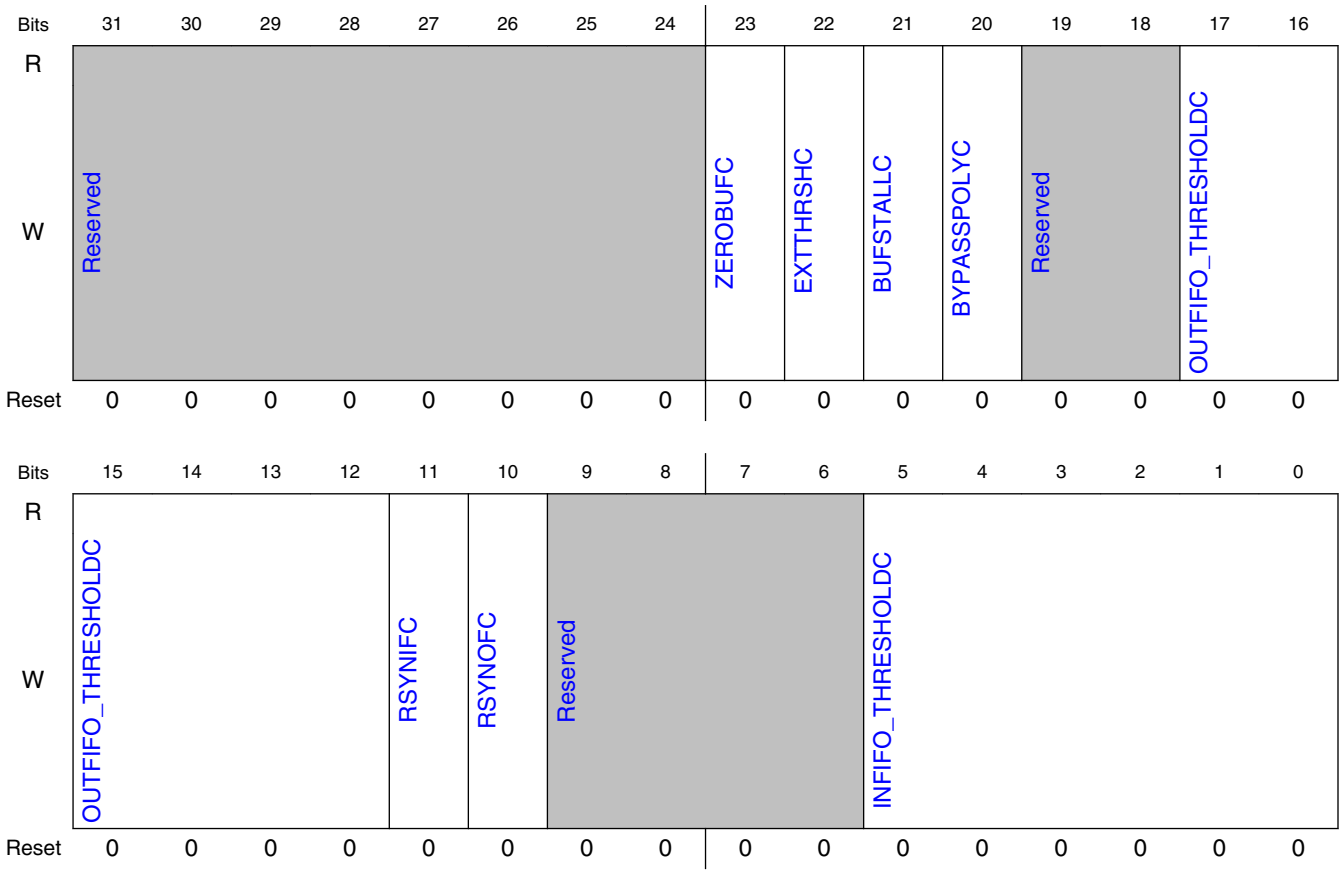
55.6.1.27 ASRC Misc Control Register for Pair C (ASRMCRC)

The register (ASRMCRC) is used to control Pair C internal logic.

55.6.1.27.1 Offset

Register	Offset
ASRMCRC	B0h

55.6.1.27.2 Diagram



55.6.1.27.3 Fields

Field	Description
31-24 —	- Reserved
23 ZEROBUFC	ZEROBUFC Initialize buffer of Pair C when pair C is enabled This bit is used to control whether the buffer is to be zeroized when pair C is enabled. 0 - Zeroize the buffer 1 - Don't zeroize the buffer
22 EXTTHRSHC	EXTTHRSHC Use external thresholds for FIFO control of Pair C This bit determines whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair C. 0 - Use default thresholds. 1 - Use external defined thresholds.
21	BUFSTALLC

Table continues on the next page...

Field	Description
BUFSTALLC	Stall Pair C conversion in case of Buffer Near Empty/Full Condition This bit determines whether the near empty/full FIFO condition stalls the rate conversion for pair C. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 0 - Don't stall Pair C conversion even in case of near empty/full FIFO conditions. 1 - Stall Pair C conversion in case of near empty/full FIFO conditions.
20 BYPASSPOLYC	BYPASSPOLYC Bypass Polyphase Filtering for Pair C This bit determines whether the polyphase filtering part of Pair C conversion is bypassed. 0 - Don't bypass polyphase filtering. 1 - Bypass polyphase filtering.
19-18 —	- Reserved. Should be written as zero for future compatibility.
17-12 OUTFIFO_THRESHOLDC	OUTFIFO_THRESHOLDC The threshold for Pair C's output FIFO per channel These bits stand for the threshold for Pair C's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFC	RSYNIFC Re-sync Input FIFO Channel Counter If bit set, force ASRCCR[ACIC]=0. If bit clear, untouch ASRCCR[ACIC]. 0 - Do not touch ASRCCR[ACIC] 1 - Force ASRCCR[ACIC]=0
10 RSYNOFC	RSYNOFC Re-sync Output FIFO Channel Counter If bit set, force ASRCCR[ACOC]=0. If bit clear, untouch ASRCCR[ACOC]. 0 - Do not touch ASRCCR[ACOC] 1 - Force ASRCCR[ACOC]=0
9-6 —	- Reserved. Should be written as zero for future compatibility.
5-0 INFIFO_THRESHOLDC	INFIFO_THRESHOLDC The threshold for Pair C's input FIFO per channel These bits stand for the threshold for Pair C's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set;

Field	Description
	when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

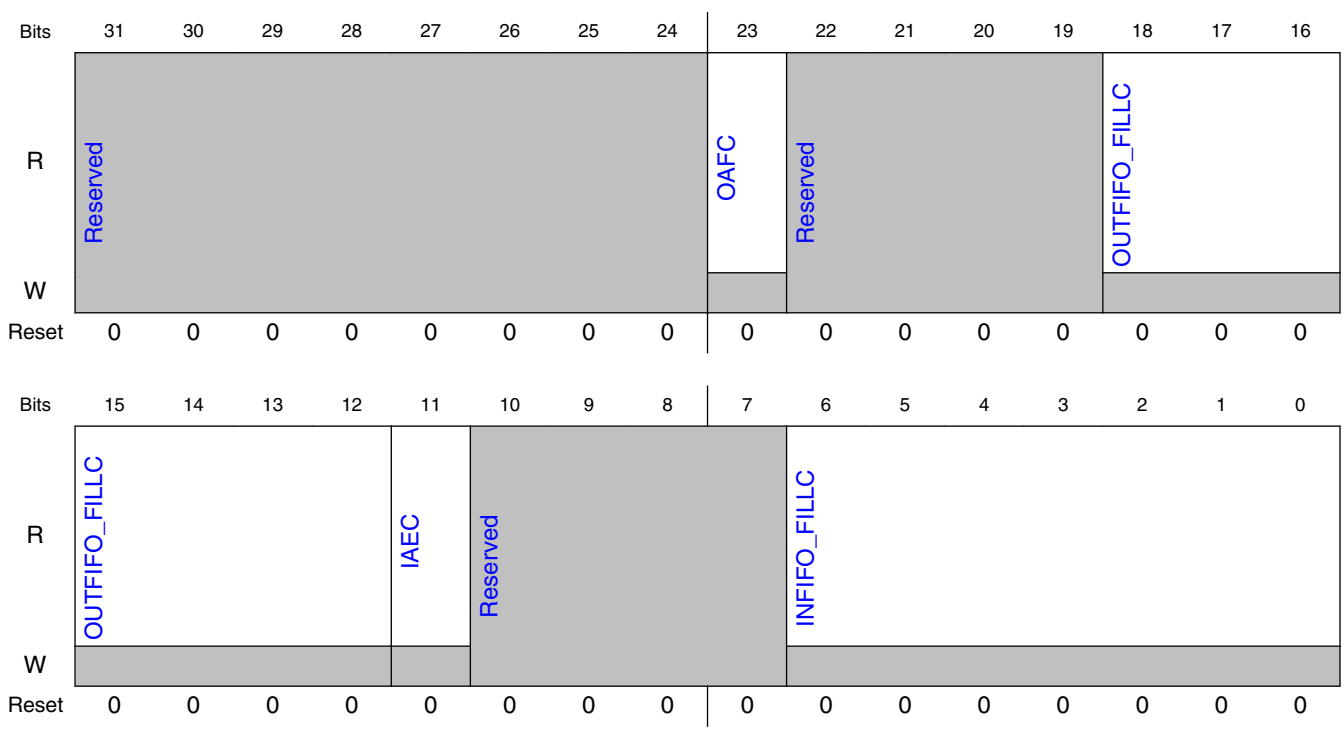
55.6.1.28 ASRC FIFO Status Register for Pair C (ASRFSTC)

The register (ASRFSTC) is used to show Pair C internal FIFO conditions.

55.6.1.28.1 Offset

Register	Offset
ASRFSTC	B4h

55.6.1.28.2 Diagram



55.6.1.28.3 Fields

Field	Description
31-24 —	- Reserved
23 OAFC	OAFC Output FIFO is near Full for Pair C This bit is to indicate whether the output FIFO of Pair C is near full. 0 - Output FIFO is not near full for Pair C 1 - Output FIFO is near full for Pair C
22-19 —	- Reserved. Should be written as zero for future compatibility.
18-12 OUTFIFO_FILLC	OUTFIFO_FILLC The fillings for Pair C's output FIFO per channel These bits stand for the fillings for Pair C's output FIFO per channel. Possible range is [0,64].
11 IAEC	IAEC Input FIFO is near Empty for Pair C This bit is to indicate whether the input FIFO of Pair C is near empty. 0 - Input FIFO is not near empty for Pair C 1 - Input FIFO is near empty for Pair C
10-7 —	- Reserved. Should be written as zero for future compatibility.
6-0 INFIFO_FILLC	INFIFO_FILLC The fillings for Pair C's input FIFO per channel These bits stand for the fillings for Pair C's input FIFO per channel. Possible range is [0,64].

55.6.1.29 ASRC Misc Control Register 1 for Pair X (ASRMCR1A - ASRMCR1C)

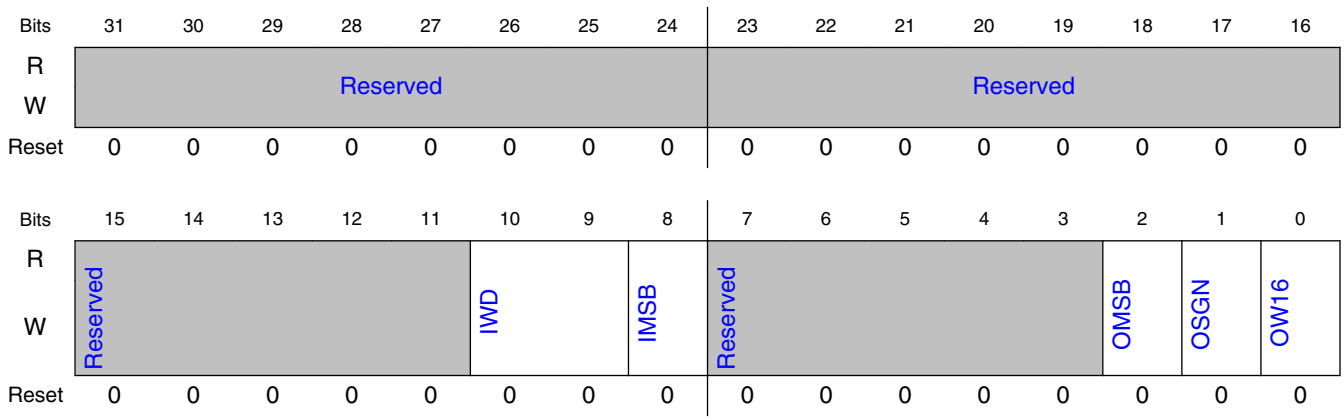
The register (ASRMCR1x) is used to control Pair *x* internal logic (for data alignment etc.).

The bit assignment for all the input data formats is the same as that supported by the SAI.

55.6.1.29.1 Offset

Register	Offset
ASRMCR1A	C0h
ASRMCR1B	C4h
ASRMCR1C	C8h

55.6.1.29.2 Diagram



55.6.1.29.3 Fields

Field	Description
31-24 —	- Reserved
23-11 —	- Reserved. Should be written as zero for future compatibility.
10-9 IWD	IWD Data Width of the input FIFO These three bits determines the bitwidth for the audio data into ASRC 00 - 24-bit audio data. 01 - 16-bit audio data. 10 - 8-bit audio data. 11 - Reserved.
8 IMSB	IMSB Data Alignment of the input FIFO This bit determines the data alignment of the input FIFO. 0 - LSB aligned.

Table continues on the next page...

Field	Description
	1 - MSB aligned.
7-3 —	- Reserved. Should be written as zero for future compatibility.
2 OMSB	OMSB Data Alignment of the output FIFO This bit determines the data alignment of the output FIFO. 0 - LSB aligned. 1 - MSB aligned.
1 OSGN	OSGN Sign Extension Option of the output FIFO This bit determines the sign extension option of the output FIFO. 0 - No sign extension. 1 - Sign extension.
0 OW16	OW16 Bit Width Option of the output FIFO This bit determines the bit width option of the output FIFO. 0 - 24-bit output data. 1 - 16-bit output data

Chapter 56
PDM Microphone Interface (PDM)

56.1 Chip-specific PDM Microphone Interface (PDM) information

Table 56-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

56.2 Overview

The Pulse Density Modulated Microphone Interface (PDM) is a popular way to deliver audio from microphones to the processor in several applications, such as mobile telephones. However, current digital-audio systems use multibit audio signal (also known as multibit PCM) to represent the signal. This block implements the required digital interface to provide a 24-bits audio signal from a PDM microphone bitstream in a configurable output sampling rate.

The implementation of this digital interface is based on the application of digital signal processing techniques in hardware. The PDM Microphone Interface architecture was designed to save gate count and minimize power consumption. It implements a series of filters to transform a PDM bitstream to a 24-bit PCM signal in the audio band.

The whole module is implemented to work in a multichannel mode. All channels have the same configuration but each input channel could be turned on/off independently.

56.2.1 Block Diagram

Figure 56-1 shows the block diagram for the PDM Microphone Interface module. It consists of the following:

- an input interface for each pair of PDM microphones.
- a decimation filter by channel.
- a FIFO by channel.
- a time generation unit.
- shared interfaces to DMA, interrupts and SoC.
- A Hardware Voice Activity Detector (HWVAD).

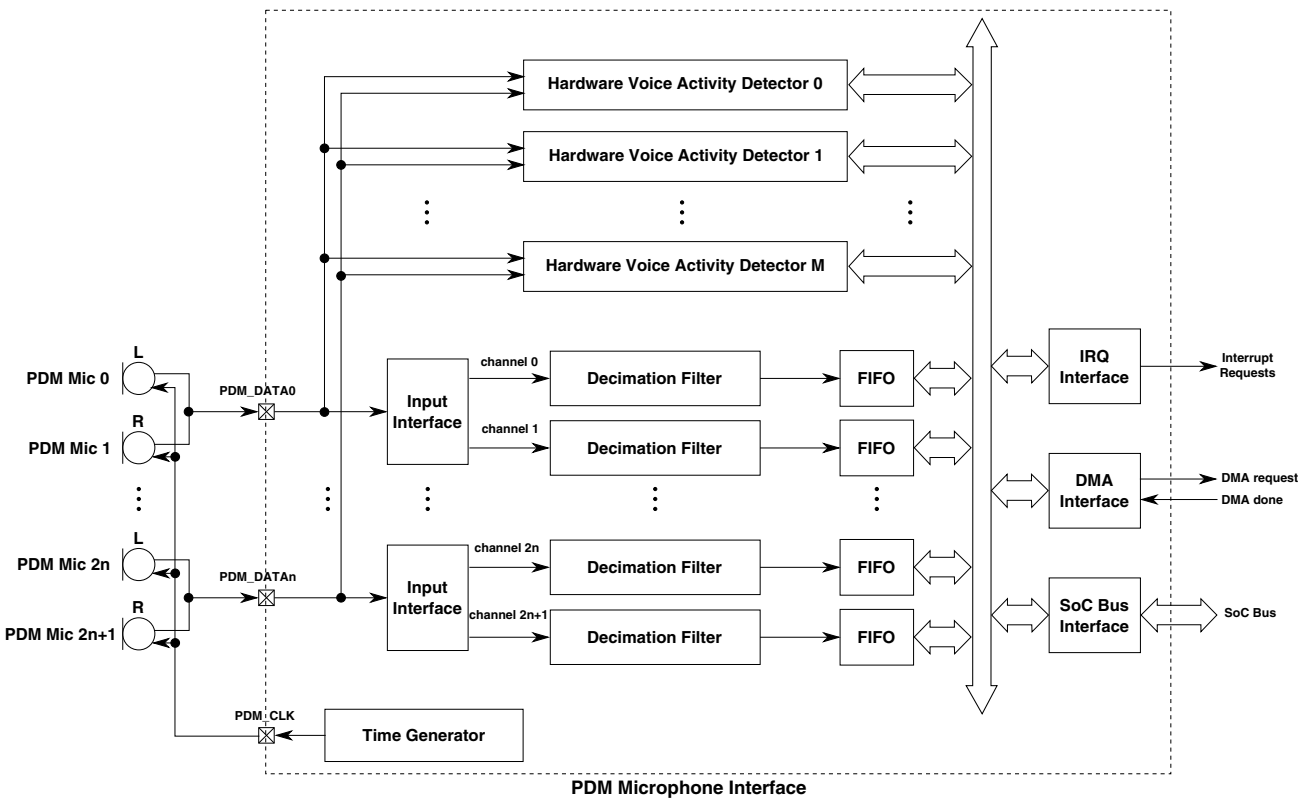


Figure 56-1. PDM Microphone Interface Block Diagram

The Time Generator unit delivers the PDM_CLK to all microphones. All microphones must operate at the same clock frequency. Each input interface receives a time multiplexed PDM bitstream from two PDM microphones and it separates the two audio information in two channels: left (0) and right (1). Each channel is processed separately.

The Decimation Filter is composed internally by a CIC filter, a DC remover and Half Band filters. The filtering results are stored in individual FIFOs; a FIFO by channel. The FIFOs have overflow and underflow detectors to deliver an error interrupt request.

An interrupt request or DMA request could be delivered by the PDM. Then the filters results stored in FIFOs could be accessed by the SoC or DMA respectively via the internal bus interface .

The HWVAD takes data from the input of a selected PDM microphone to detect if there is any voice activity. When a voice activity is detected, an interrupt could be delivered to the system.

56.2.2 Features

The PDM Microphone Interface (PDM) includes the following features:

- Decimation filters:
  - Fixed filtering characteristics for audio application.
  - 24-bit signed filter output.
  - Internal clock divider for a programmable PDM clock generation.
  - Full or partial set of channels operation with individual enable control.
  - Programmable decimation rate.
  - Programmable DC remover.
  - Range adjustement capability.
  - FIFOs with interrupt and DMA capability.
    - Each FIFO with 8 entries length.
- Hardware Voice Activity Detector (HWVAD).
  - Interrupt capability.
  - Zero-Crossing Detection (ZCD) option.

56.3 Functional Description

Figure 56-1 shows the PDM Microphone Interface (PDM) block diagram. Basically this module is composed by a Decimation Filter and a FIFO by channel, an Input Interface for each pair of microphones, a shared Time Generator unit, a shared bus interface, a shared DMA interface and a shared interrupt interface.

The Decimation Filter implements a low-pass filter in the audio band (20Hz-22.5kHz @48kHz output sampling rate by default) with a configurable decimation rate. It is implemented using a series of CIC, Half Band and DC remover filters.

The Time Generator unit generates the PDM_CLK to the microphones. This clock is the same for all the PDM microphones and it is active for all the microphones, i.e. there is not possiblity to turn off the PDM_CLK for one microphone only.

The output of each Decimation Filter is stored into a FIFO buffer. Each FIFO is mapped in the DATACHn registers. It is possible to generate either an interrupt or a DMA request when, in each FIFO of all enabled channels, the number of data stored surpasses a configured watermark.

Also, independently on decimation filters, there is a Hardware Voice Activity Detector(HWVAD) which implements voice-detection algorithms to generate wake-up interrupts.

56.3.1 Modes of operation

This section describes the PDM operation modes which depends on the SoC power mode , the SoC Doze mode , the SoC Debug mode and MDIS, DBGE, DBG and DOZEN bitfields as summarized in Table 56-2.

Table 56-2. Operation Mode Selection

<table>
<tr><th>Mode</th><th>MDIS</th><th>SoC Doze & DOZEN= 1</th><th>SoC Power Mode</th><th>DOZEN</th><th>SoC Debug | DBG</th><th>DBG E</th><th>CPU clock</th><th>PDM internal clock</th><th>Decimation Filter state</th><th>HWVAD[n] state</th></tr>
<tr><td>Normal</td><td rowspan="8">0</td><td rowspan="4">0</td><td rowspan="3">RUN</td><td rowspan="3">Any</td><td>0</td><td>Any</td><td rowspan="3">Enabled</td><td rowspan="4">Enabled</td><td>Run (Depends on PDMIEN and CHENn)</td><td>Run (Depends on PDMIEN and VADEN)</td></tr>
<tr><td>Debug</td><td rowspan="2">1</td><td>1</td><td>Stop</td><td>Stop</td></tr>
<tr><td>Low Power</td><td>0</td><td rowspan="3">Any</td><td rowspan="3">Any</td><td rowspan="3">Gated (Externally)</td><td>Run (Depends on PDMIEN and CHENn)</td><td rowspan="2">Run (Depends on PDMIEN and VADEN)[1]</td></tr>
<tr><td rowspan="2">SLEEP</td><td rowspan="2">0</td><td>DEEPSLEEP</td><td rowspan="3">Stop</td></tr>
<tr><td>Disable/Low Leakage</td><td>SLEEP/ DEEPSLEEP</td><td>1</td><td rowspan="2">Gated</td><td rowspan="2">Stop</td></tr>
<tr><td rowspan="2">1</td><td>1</td><td rowspan="2">Any</td><td rowspan="2">Any</td><td>Enabled</td></tr>
<tr><td>Any</td><td>Gated (Internally)</td></tr>
</table>

1. HWVAD[n] will run in DEEPSLEEP only if it is placed on an always-on power domain in the SoC.

56.3.1.1 Normal mode

This is the default PDM operational mode. In this mode the channels can be enabled to perform filtering operation.

56.3.1.2 Low Power mode

In Low Power mode the PDM is able to filter the PDM microphone data and perform voice detection such as in Normal mode and can generate a wake up event through asynchronous DMA/IRQ requests depending on CHENn, VADEN and PDMIEN values.

The Low Power mode is entered when the SoC enters to SLEEP or DEEPSLEEP modes and DOZEN =0. In this mode the CPU clock is gated externally and PDM could be kept running.

In case a SoC SLEEP mode is requested, the decimation filter is kept running such as Normal mode and the SoC SLEEP mode is acknowledged immediately.

In case a SoC DEEPSLEEP mode is requested it is waited until the last input samples are processed to acknowledge the SoC mode and stop the decimation filter.

During this mode, VAD will continue running if it was previously enabled (VADEN=1) during Normal mode.

The DMA and IRQ requests can be generated asynchronously to wake up the CPU. .

NOTE

The asynchronous DMA or asynchronous interrupt requests are generated only in this mode.

56.3.1.3 Debug mode

This mode is used to check the status of the IP when requested. In this mode the PDM can be stopped after the remaining data processing has finished depending on the DBGE bit value. This mode can be actived by a SoC debug request or asserting the DBG bitfield.

In DBGE=0 case, if a debug request is received or DBG is asserted while the module is running, it will be stopped after finish to process the remaining data. Otherwise, if DBGE=1, the module will be kept running.

The PDM Microphone Interface cannot enter Debug mode once in Disable/Low Leakage or Low Power mode.

The Debug mode request affects all channels in the PDM. Access to output buffers remain operational, as well as their related flags.

NOTE

Once a debug request is issued, it can not be aborted.

56.3.1.4 Disable/Low Leakage mode

In this mode, PDM is stopped after the remaining data processing has finished. PDM is in Disable/Low Leakage (DLL) mode when:

- The SoC is in SLEEP and DEEPSLEEP modes and DOZEN=1.
- The SoC is in Doze and DOZEN bit is asserted.
- MDIS is asserted.

When Disable/Low Leakage (DLL) mode is requested while the module is running, it will be stopped after finish to process the remaining data.

Writes only to CTRL_1 and CTRL_2 registers are allowed with the exception of writes to the DBG and SRES bits, which are ignored, but read is still available for these bits.

56.3.2 Output Rate

The output rate is the sampling rate at the decimation filter output. It depends on the sampling rate of the microphone input and the configured quality mode (refer to Quality Modes) . Common audio sampling rates are 48kHz and 16kHz. On this document the output rate is referred sometimes as FS.

56.3.3 Quality Modes

Independently on the power mode selected, if the decimation filter is running, the PDM can operate in 6 different quality modes, those can be selected by the QSEL bitfield. The decimation and interpolation rates in internal blocks are shown in Table 56-3 where OSR depends on the CICOSR as shown in Equation 1 on page 3418.

$$OSR = 16\text{-}CICOSR$$

Equation 1. Oversampling Rate (OSR)

The quality selected can be configured dinamically when the Decimation Filter is running, the output rate is mantained in every quality selected.

The passband of the Decimation Filter depends on the quality mode, refer to Overall Passband for more information.

In order to mantain the same output rate in the Decimation Filter when the quality is changed, the PDM_CLK rate changes depending on the quality selected. For more details about the PDM_CLK, refer to Time Generator.

The Sampler and the Half Band filters are whether enabled or not depending on the quality selected, achieving a low power consumption when they are disabled. The CIC filter decimation rate is also whether doubled or not depending on the quality selected. For more information, refer to Decimation Filter.

NOTE

In order to get a better audio performance, the RANGEADJn bitfield need to be adjusted depending on the quality mode selected as pointed out on its description. For more information, refer to Overall Filter Gain.

Table 56-3. Quality Modes

Quality Mode	QSEL bitfield	Sampler Interpolation	CIC Filter Decimation	1st Half Band Filter Decimation	2nd Half Band Filter Decimation	PDM_CLK Rate	Passband
High Quality	001	-	:(2OSR)	:2	:2	Output Rate x 8 x OSR	To ~0.5 x Output Rate
Medium Quality	000	-	:OSR	:2	:2	Output Rate x 4 x OSR	To ~0.5 x Output Rate
Low Quality	111	x2	:OSR	:2	:2	Output Rate x 2 x OSR	To ~0.5 x Output Rate
Very Low Quality 0	110	-	:(2OSR)	:2	-	Output Rate x 4 x OSR	To ~0.25 x Output Rate
Very Low Quality 1	101	-	:OSR	:2	-	Output Rate x 2 x OSR	To ~0.25 x Output Rate
Very Low Quality 2	100	x2	:OSR	:2	-	Output Rate x OSR	To ~0.25 x Output Rate

56.3.4 Time Generator

The Time Generator is the responsible to generate the PDM_CLK to the microphones.

NOTE

The Time Generator is affected by soft-reset. The PDM_CLK starts with a rising edge after a soft reset is issued.

56.3.4.1 Clock divider

The sampling rate of the microphone input depends on the PDM_CLK rate. The PDM_CLK rate can be trimmed using an internal clock divider, configuring the quality mode (refer to Table 56-3) or trimming the PDM's root clock frequency (PDM_CLK_ROOT rate).

The internal clock divider frequency is defined by the CLKDIV bitfield value which must be configured following Equation 2 on page 3420.

$$CLKDIV = \frac{\text{PDM_CLK_ROOT rate}}{8(16\text{-CICOSR})(\text{Output Rate})}$$

Equation 2. CLKDIV bitfield value

Once calculated the CLKDIV value, the generated PDM_CLK rate can be estimated as shown in Equation 3 on page 3420, where the K factor value depends on the quality mode as shown in Table 56-4 and floor(x) is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x.

$$\text{PDM_CLK rate} = \frac{\text{PDM_CLK_ROOT rate}}{2 \cdot \text{floor}(K \cdot CLKDIV)}$$

Equation 3. PDM_CLK rate

Table 56-4. K factor value

Quality mode	K factor
High Quality	1/2
Medium Quality, Very Low Quality 0	1
Low Quality, Very Low Quality 1	2
Very Low Quality 2	4

56.3.4.1.1 Minimum Required CLKDIV

NOTE

This minimum CLKDIV condition is required only if any decimation filter channel is enabled and the clock divider is enabled, otherwise this condition should be ignored.

To work properly the CLKDIV bitfield value must have a minimum value according to the quality mode configured (see Quality Modes), the number of enabled channels (EC), Equation 1 on page 3418 and K factor (see Input/Output Timing Requirements). When this condition is not met the LOWFREQF bit is asserted and an error interrupt can be issued (in case ERREN=1).

The equations to estimate the minimum required CLKDIV bitfield value depending on the configured quality mode are shown in Equation 4 on page 3421 (for medium, high and low quality modes) and Equation 5 on page 3421 (for very low quality modes), on these equations floor(x) is the function that takes as input a real number x and gives as output the greatest integer less than or equal to x.

$$\text{floor}(\text{K x CLKDIV}) >= \text{K x } \frac{10+93EC}{8 \text{ x OSR}}$$

Equation 4. Minimum required CLKDIV value in Medium, High and Low Quality modes

$$\text{floor}(\text{K x CLKDIV}) >= \text{K x } \frac{10+43EC}{8 \text{ x OSR}}$$

Equation 5. Minimum required CLKDIV value in Very Low Quality modes

56.3.4.2 PDM_CLK_ROOT rate and CLKDIV calculation examples

Example 1:

If it is required an output rate=16kHz and enable 4 channels (EC=4). A valid configuration is following:

1. Configure High Quality mode and CIC decimation by 32 (CICOSR =0, then OSR=16), .
2. Check the PDM_CLK rate for High Quality mode in Table 56-3. PDM_CLK rate = 16kHz x 8 x OSR = 16kHz x 8 x 16 = 2.048MHz.
3. Check the K factor value for High Quality mode in Table 56-4. K factor is 1/2.
4. Calculate minimum CLKDIV value (see Equation 4 on page 3421): floor(K x CLKDIV) >= K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) >= 2.984375. Finally, CLKDIV >= 5.96875.
5. Calculate the minimum PDM_CLK_ROOT rate:
   - From Equation 2 on page 3420 : PDM_CLK_ROOT rate = 8 x (16-CICOSR) x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV>=4, then PDM_CLK_ROOT rate >= 8 x (16-CICOSR) x 5.96875 x 16kHz.
   - Finally: PDM_CLK_ROOT rate >= 12224kHz
   - Select the PDM_CLK_ROOT rate in that range.
6. Check if selected PDM_CLK_ROOT rate and CLKDIV value satisfies the Table 56-5.
7. Finally, check again if the selected PDM_CLK_ROOT rate satisfies Equation 2 on page 3420. If not, adjust CLKDIV or change PDM_CLK_ROOT rate accordingly.

Example 2:

If it is required an output rate=48kHz and enable 4 channels (EC=4). A valid configuration is following:

1. Configure Medium Quality mode and CIC decimation by 16 (CICOSR =0, then OSR=16), .
2. Check the PDM_CLK rate for Medium Quality mode in Table 56-3. PDM_CLK rate = 48kHz x 4 x OSR = 48kHz x 4 x 16 = 3.072MHz.
3. Check the K factor value for Medium Quality mode in Table 56-4. K factor is 1.
4. Calculate minimum CLKDIV value (see Equation 4 on page 3421): floor(K x CLKDIV) >= K x (10 + 93EC)/(8 x OSR) then floor(K x CLKDIV) >= 2.984375. Finally, CLKDIV >= 2.984375.
5. Calculate the minimum PDM_CLK_ROOT rate:
   - From Equation 2 on page 3420 : PDM_CLK_ROOT rate = 8 x (16-CICOSR) x CLKDIV x (Output rate)
   - From step 4 result: CLKDIV>=2.984375, then PDM_CLK_ROOT rate >= 8 x (16-CICOSR) x 2.984375 x 48kHz.
   - Finally: PDM_CLK_ROOT rate >= 18336kHz
   - Select the PDM_CLK_ROOT rate in that range.
6. Check if selected PDM_CLK_ROOT rate and CLKDIV value satisfies the Table 56-5.
7. Finally, check again if the selected PDM_CLK_ROOT rate satisfies Equation 2 on page 3420. If not, adjust CLKDIV or change PDM_CLK_ROOT rate accordingly.

56.3.4.3 Input/Output Timing Requirements

The PDM microphones must meet the setup and hold timing requirements shown in Table 56-5 and Figure 56-2. The "k" factor value in Table 56-5 depends on the selected quality mode as shown in Table 56-4.

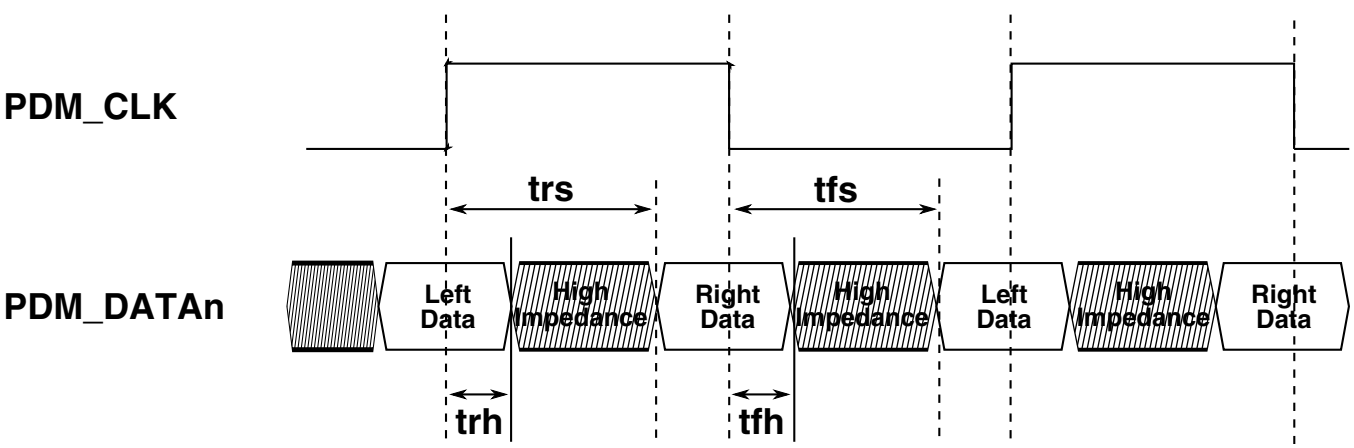


Figure 56-2. Input/Output TIming Requirements

Table 56-5. Timing parameters

Parameter	Value
trs, tfs	$<= \frac{\text{floor}(K \times CLKDIV) - 1}{\text{PDM_CLK_ROOT rate}}$ [1]
trh. tfh	>=0

1. Depending on K value, the user must make sure floor(K x CLKDIV) > 1 to avoid timing problems.

56.3.5 Decimation Filter

Figure 56-3 shows the block diagram of the Decimation Filter. This block is composed by a CIC filter, a DC remover, two Half Band filters for each channel.

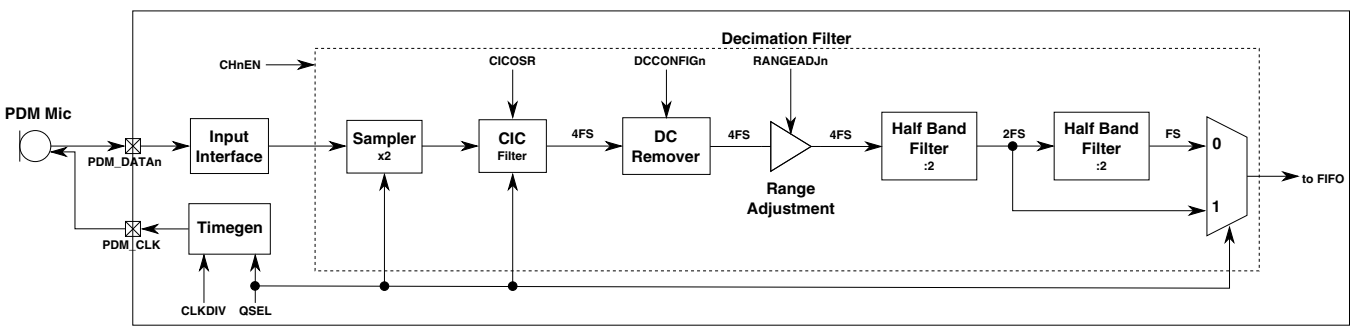


Figure 56-3. Decimation Filter Block Diagram

The Sampler block is activated depending on the quality mode selected (QSEL bitfield). This one samples twice the PDM_DATAn incoming from the Input Interface.

The CIC filter (Cascaded Integrator Comb filter) is a low pass filter commonly used to filter the sigma-delta modulator output signal.

The DC remover is a high pass filter used to remove the DC component of the processed signal with configurable cut-off frequency.

Each Half Band filter implements a low pass digital filter with decimation by 2. They are used to compensate the high CIC drop in passband and helps the FIR filter block to flat the overall passband response.

The Decimation Filter output depends on the quality mode selected by the QSEL bitfield, so it can come from any of the Half Band filters output. In case, the last Half Band filter stage is bypassed, power consumption should be reduced.

The range of each Decimation Filter can be adjusted by a Range Adjustment block so that, in case a adjustment overflows or underflows is detected. an error interrupt is raised.

56.3.5.1 Filter Performance

56.3.5.1.1 Overall Filter Gain

The overall filter gain depends on the quality mode, the Equation 8 on page 3425 and the RANGEADJn bitfield as shown in Equation 6 on page 3424.

$$OverallFilterGain\ (dB) = \begin{cases} 140 \cdot \log_{10}(32 - 2CICOSR) + 6.02RANGEADJn - 222.74 & ;if\ QSEL \in [HQ, VLQ0] \\ 140 \cdot \log_{10}(16 - CICOSR) + 6.02RANGEADJn - 222.74 & ;others \end{cases}$$

Equation 6. Overall Filter Gain

Finally, the input and output dBFS levels are related as shown in Equation 7 on page 3424.

$$Output(dBFS) = Input(dBFS) + OverallFilterGain(dB)$$

Equation 7. Filter input and output relationship

56.3.5.1.2 Overall Passband

The Decimation Filter passband depends on the quality mode and the output rate (FS). For instance, the overall filter passband for different output data rates when the PDM is in High, Medium and Low Quality modes is shown in Table 56-6.

Table 56-6. Filter Passband in HQ, MQ and LQ modes (DCCONFIGn=00)

Output Data Rate	Overall Filter Passband
FS	**0.0004FS - 0.4688FS**
16kHz	6.67Hz - 7.50kHz
48kHz	20Hz - 22.5kHz

The overall filter passband for different output data rates when the PDM is in Very Low Quality modes is shown in Table 56-7.

Table 56-7. Filter Passband in VLQn modes (DCCONFIGn=00)

Output Data Rate	Overall Filter Passband
FS	**0.0002FS - 0.2344FS**
16kHz	3.33Hz - 3.75kHz
48kHz	10.00Hz - 11.25kHz
96kHz	20Hz - 22.5kHz

56.3.5.2 Input Interface

The timing diagram of the input interface signals is shown in Figure 56-4. The bitstream incoming from the microphone data input "n" (PDM_DATAn) in the first half (Right microphone) of the PDM_CLK is directed to the Channel "2n+1" and the data generated during the last half (Left microphone) is directed to the Channel "2n".

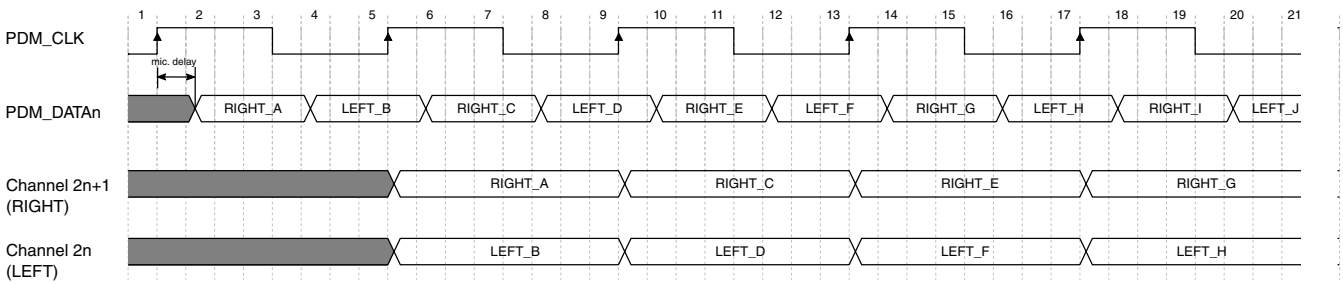


Figure 56-4. Input Interface signals

56.3.5.3 CIC Filter

The CIC filter is an optimized implementation of the low pass Sinc filter with multiple stages. It is widely used because it requires just adders and subtractors. The CIC filter is composed by sub-blocks called: integrator and comb. The CIC decimation rate is defined by the CICOSR bitfield and the selected quality mode as shown in Equation 8 on page 3425.

$$\text{CIC Decimation Rate} = \begin{cases} 2*(16-\text{CICOSR}) & ;\text{if HQ or VLQ0} \\ 16-\text{CICOSR} & ;\text{others} \end{cases}$$

Equation 8. CIC Decimation Rate

56.3.5.4 DC Remover

There are two DC removers in the filter chain, one in the input and another in the output. The former is integrated in the CIC filter, and the later is placed just before the FIFO. Both can be bypassed independently.

The block removes the DC level component of the signal using a high-pass filter with a very low cut-off frequency selected by the DCCONFIGn bitfield. This filter is bypassed when DCCONFIGn=11.

56.3.5.5 Half Band Filters

A half band filter divides the signal base-band into two equal subbands before to be decimated by 2. It also compensates the CIC filter drop in passband and helps to flatten the overall passband response. PDM has two half band filter stages, with the possibility of bypass the last one.

56.3.6 Hardware Voice Activity Detector

The Hardware Voice Activity Detector (HWVAD) is a block responsible for detect voice activity in a channel selected by the user as shown in the Figure 56-5 and Figure 56-6. It can be configured in Envelope-based or Energy-based mode and it should be configured as described in Initialization.

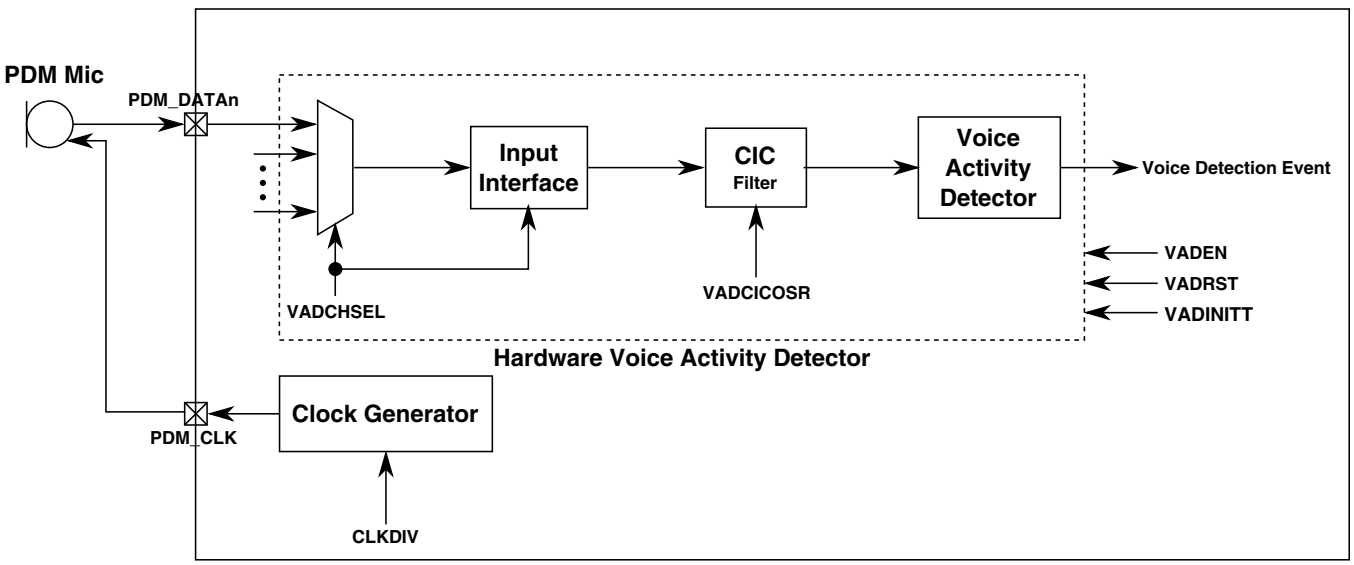


Figure 56-5. HWVAD Block Diagram

The HWVAD can divide the incoming PCM signal from its internal CIC in frames, the frame energy is used to estimate the background noise. Both frame energy and noise energy are examinated, a voice activity is detected when a specific delta between them is detected, then a interrupt will be issued. Optionally a Zero-Crossing Detection block (ZCD) could be enabled to avoid low energy voiced speech be missed, improving the voice detection performance.

The HWVAD detector can be enabled by the VADEN bit (VAD_CTRL_1 register) and could be reset by software through the VADRST bit (VAD_CTRL_1 register) . It is important to remark that the HWVAD detector could be enabled and/or reset only when the PDM isn't running i.e. when the BSY_FIL bit in STAT register is cleared.

To perform voice detection a Noise Filter and Signal Filter are used as shown Figure 56-6. In addition, a minimum block in the output of the Noise Filter and a maximum block in the ouput of the Signal Filter can be used. The result of both blocks can be scaled by gain blocks. Finally, the signal and noise estimated are compared, voice is detected when signal is greater than noise.

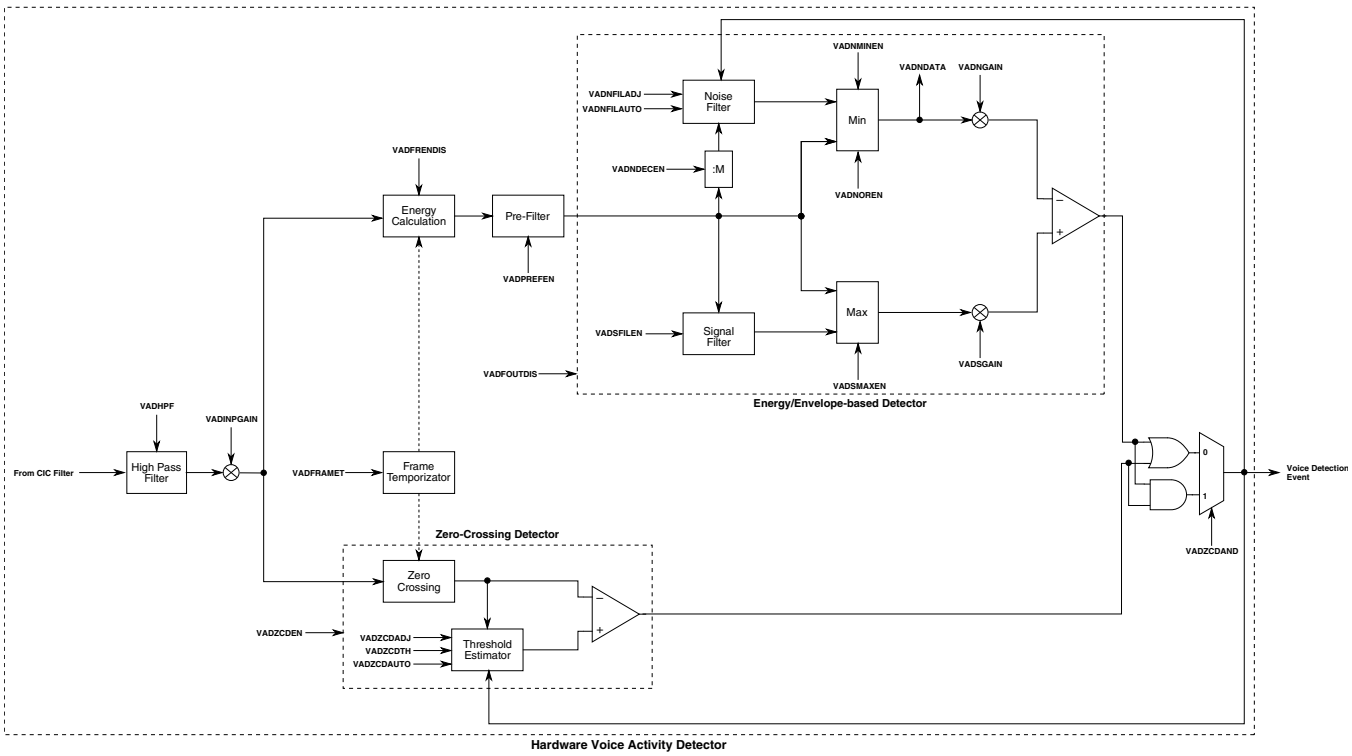


Figure 56-6. HWVAD Block Diagram (cont.)

56.3.6.1 High-Pass Filter

The purpose of the High-Pass Filter is to remove the low frequency components. The cut-off frequency is selected by the VADHPF bitfield. This filter is bypassed when VADHPF=00.

56.3.6.2 Input Setting

The HWVAD input has internally a user configurable High-Pass filter (see VADHPF bit description) and a user configurable gain (see VADINPGAIN). Both controls could be used to configure the module based on external noise conditions.

56.3.6.3 Frame Time Setting

In order to perform voice detection the signal frames incoming from the CIC filter can be divided in frames whose duration can be configured. The frame time will depend on the CIC output rate, the value of the VADFRAMET bitfield and a constant as is defined in Equation 9 on page 3428.

$$frame\ \ time = \frac{VADFRAMET \times 192}{CIC\ \ output\ \ rate}$$

Equation 9. Frame Time

For instance, in the typical case of the CIC filter output rate would be 192kHz, the frame time would be 10ms when VADFRAMET is 10.

56.3.6.4 Initialization Time

An initialization time can be configured by the VADINITT bitfield. During this initialization time no voice detection event is issued, all detectors are kept disabled.

$$initialization\ \ time = (\mathrm{VADINITT}+1) \times frame\ \ time$$

Equation 10. Initialization Time

56.3.6.5 Energy Calculation

In case VADFRENDIS = 0, the sum of the absolute values of all the samples of each frame are used to detect voice. Otherwise, if VADFRENDIS=1, the absolute value of the samples are used to detect voice i.e. the data incoming from CIC is not divided in frames.

56.3.6.6 Pre-Filter

The Pre-FIlter block is a low-pass filter which can be enabled when VADPREFEN=1 .

56.3.6.7 Energy/Envelope-based Detector

To perform voice detection a Noise Filter and Signal Filter are used as shown Figure 56-6. In addition, a minimum block in the output of the Noise Filter and a maximum block in the ouput of the Signal Filter can be used. The result of both blocks can be scaled by gain blocks. Finally, the signal and noise estimated are compared, voice is detected when signal is greater than noise.

56.3.6.7.1 Noise Filter

The Noise Filter is a low-pass filter . This filter could be adjusted by the VADNFILADJ bitfield . In case VADNFILAUTO=1 the noise filter is enabled only when voice is not detected, when voice is detected the filter output does not change. In case VADNFILAUTO=0 the filter is always enabled.

A decimation stage in the input of the Noise Filter can be enabled by the VADNDECEN bitfield.

56.3.6.7.2 Minimum Noise

In order to estimate the noise, a block which calculates the minimum value between the input and output of the noise filter is used. This block can be enabled by VADNMINEN, otherwise the output of the noise filter is passed directly.

The estimated noise of the output of this block is stored in the VADNDATA register for post-processing via software.

56.3.6.7.3 Signal Filter

The Signal Filter is a low-pass filter . This filter will be enabled when VADSFILEN=1, otherwise the filter will be bypassed.

56.3.6.7.4 Maximum Signal

In order to estimate the signal, a block which calculates the maximum value between the input and output of the signal filter can be used. This block can be enabled by VADSMAXEN.

56.3.6.7.5 Filter Result Gain Setting

There are two gain adjustment controls, one for the estimated noise (VADNGAIN) and another for signal estimated (VADSGAIN). Both gain adjustments outputs are compared, the comparison result determines whether or not there is voice in the channel.

56.3.6.8 Zero-Crossing Detector

Optionally, the HWVAD performance can be increased by the use of a Zero-Crossing Detection (ZCD) circuit. It is enabled by a VADZCDEN bit.

The threshold of the ZCD could be static or calculated dynamically depending on the VADZCDAUTO bit. When it is dynamically determined (VADZCDAUTO = 1) the threshold is calculated automatically taking the VADZCDTH bit-field as initial value, otherwise (VADZCDAUTO=0) the threshold is static in the VADZCDTH user configurable value.

The ZCD result can be AND'ed or OR'ed with the energy-based or envelope-based detection, it is selected by the VADZCDAND bit.The filter coefficient is equal to (VADZCDADJ+1)*2^(-4).

56.3.6.9 Interrupt

The HWVAD issues an interrupt when voice activity is detected and the VADIE bit in the VAD_CTRL_1 register is enabled. The voice activity event also will set the VADIF flag in the VAD_STAT register, this bit will be cleared when it is written 1.

56.3.6.10 Extended Operation

The VADIF bit could be used for a HWVAD extended operation i.e. using the VADNDATA bitfield value to determine a more accurate voice detection event by software. The procedure to follow will be:

1. An interrupt is issued by the HWVAD, the VADIF is set.
2. Leave the VADIF asserted while sofware is processing the VADNDATA bitfield content.
3. When software finishes to process the VADNDATA and detects voice activity:
   - If Envelope-based Detection is used, pulse VADST10 by more than 2-cycles and clear VADIF flag.
   - If Energy-based Detection is used, clear VADIF flag.

56.3.6.11 VAD Soft-reset

The VAD soft-reset is requested by writing 1 to the VADRST bit. It resets the following blocks and/or registers:

- VADIF, VADINSATF bitfields.
- All internal VAD buffers.

56.3.7 Bus Interface

The Bus Interface manages the Direct Memory Access (DMA) transactions, the interrupt requests and the user register interface with the SoC.

56.3.7.1 FIFO

The FIFO is a First-In, First-Out buffer that saves the overall filtering results. Each FIFO is 8 entries length. There is one FIFO for each channel. A push writes data into the FIFO, and a read pops data from FIFO. The push and pop operations from FIFO are in different clock domains, the push operation is made using the PDM clock while the pop operation uses the CPU clock . The FIFO has a configurable watermark FIFOWMK in the FIFO_CTRL register. When the amount of data in all the FIFOs of the enabled channels is greater than the watermark value (FIFO_CTRL[FIFOWMK]) a DMA/IRQ is requested depending on the DISEL field in the CTRL_1 register. After the filters are enabled by PDMIEN, the initial FIR results are discarded and not written into the FIFOs. About 68 initial FIR results are not stored on FIFOs (FIR_RDY=0). After the STAT[FIR_RDY] register is set the data generated by the filter is valid and starts to push the results in FIFO.

A new filtered result is discarded (not stored) when the FIFO is full and overflow flag FIFOOVFn rises. Similarly, when trying a pop and FIFO is empty no valid data is read and underflow flag rises. In both cases, when overflow or underflow occurs they are flagged in the FIFO_STAT register to the respective channel, and an error interrupt is generated if the ERREN in the CTRL_1 register is asserted.

When the FIFO is reset by asserting the SRES soft-reset bit, zero values fills all FIFO positions, as well as push and pop pointers are set to initial position.

56.3.7.2 DMA

If DISEL=01 the FIFO stored samples will be read through DMA transactions. Then if there is at least one enabled channel and all FIFOs of all enabled channels reach their watermark the output interface makes a request for DMA transaction. When the PDMIEN is set and the filters start to operate, there is a period to wait the filter results to stabilize and no filter results are stored on the FIFOs. Therefore the first DMA request takes longer to occur than the following ones.

The CHnF flags in the STAT register are cleared when a DMA transaction is finished.

When the PDM is in Low Power mode and the watermark level is surpassed an asynchronous DMA request is delivered to the system, then the system wakes up the CPU clock, returning to Normal mode, to read FIFOs data.

56.3.7.3 Interrupt Requests

If DISEL=10 then an interrupt request is used to indicate that filtering results are stored in the FIFOs to be read by CPU. Also in this case the data can be read in the DATACHn registers and the flag CHnF of the STAT register are asserted for the enabled channels when the CHn FIFO surpasses the watermark level.

When the module is in Low Power mode the CPU clock is disabled, but the PDM clock remains enabled. In this case, an asynchronous interrupt is delivered to the system, it wakes up the CPU clock, returning to Normal mode. The system could read the FIFOs data and after should clear the CHnF flags to get a new interrupt request.

An error interrupt request can be generated also when an exceptional condition happens, such as a overflow or underflow in any channel output, an overflow or underflow of any FIFO, or the PDM clock rate is not the minimum value required to the PDM working (see Minimum Required CLKDIV). This one is enabled by the ERREN bit and the type of error is signalized by its respective status flag: OUTOVFn and OUTUNFn for channel outputs errors, FIFOOVF and FIFOUND flags for FIFO errors, and LOWFREQF for PDM clock rate.

56.4 External Signals

Note

The PDM Microphone Interface module does not provide metastability protection or filtering for input data. The data must be generated using the provided PDM_CLK.

Table 56-8. PDM Microphone Interface External Signals

Signal	Description	Direction
PDM_DATAn	The data bitstreams received from *n*th PDM microphone pair throughout the input interface	I
PDM_CLK	The clock delivered to the PDM microphones from the clock generator	O

56.5 Initialization

Following are some simple initialization steps to be done before using the PDM block.

56.5.1 Initialization Procedure without HWVAD

The sequence of steps for the block initialization without enabling voice activity detector, i.e. using decimation filters only, is as follows:

1. Program the bit-fields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see PDM_CLK_ROOT rate and CLKDIV calculation examples.
2. Start the overall module asserting the PDMIEN bit.
3. The module is ready to receive data from the input ports of the enabled channels.

56.5.2 Initialization Procedure with HWVAD

In this section is described the initialization procedures to initialize PDM with the voice activity detector.

The voice activity detector can work in two modes: Energy-based mode or Envelope-based mode. Optionally, a Zero-Crossing Detector can be enabled to improve the voice detection.

The Energy-based method separates the audio signal in frames and estimates the energy of these frames to detect voice activity. In the other hand, the Envelope-based method calculates the audio signal envelope to detect voice.

Both methods have the same detection performance, but the Energy-based is recommended for low power applications where is expected that the chip is asleep for long periods. Envelope-based detection is recommended for applications where the device require to wake faster and it is not asleep for long periods.

The Zero-Crossing Detector is recommended to improve detection with fricatives phonemes (/e.g. /f/ or /s/) in noisy environments.

56.5.2.1 HWVAD in Energy-based mode

The sequence of steps to initialize the voice activity detector in Energy-based mode is as follows:

1. Program the control bitfields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see PDM_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
   a. CIC oversampling rate (VADCICOSR bitfield).
   b. Configure the voice source channel (VADCHSEL bitfield).
   c. Configure the internal initialization timer (VADINITT bitfield) if required. More details in Initialization Time.
   d. Assert VADEN bitfield.
3. Configure input, noise and signal gains according to the environmental noise conditions:
   a. Configure the input gain (VADINPGAIN bitfield).
   b. Configure the signal gain (VADSGAIN bitfield).
   c. Configure the noise gain (VADNGAIN bitfield).
4. Enable the high-pass filter if required (VADHPF bitfield).
5. Configure the VADNFILADJ bitfield according to environmental noise conditions (VADNFILADJ = 16 is recommended).
6. Put HWVAD in energy-based mode (default values):
   a. Keep the VADFRENDIS bitfield cleared.
   b. Keep the VADPREFEN bitfield cleared.
   c. Keep the VADSFILEN bitfield cleared.
   d. Keep the VADSMAXEN bitfield cleared.
   e. Keep the VADNFILAUTO bitfield asserted.
   f. Keep the VADNMINEN bitfield cleared.
   g. Keep the VADNDECEN bitfield cleared.
   h. Keep the VADNOREN bitfield cleared.
7. Initialize the Zero-Crossing Detector (Zero-Crossing Detector Initialization) if required.
8. Enable interrupts if required (VADIE and VADERIE bitfields).
9. Start the overall module asserting the PDMIEN bit.
10. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when voice is detected.

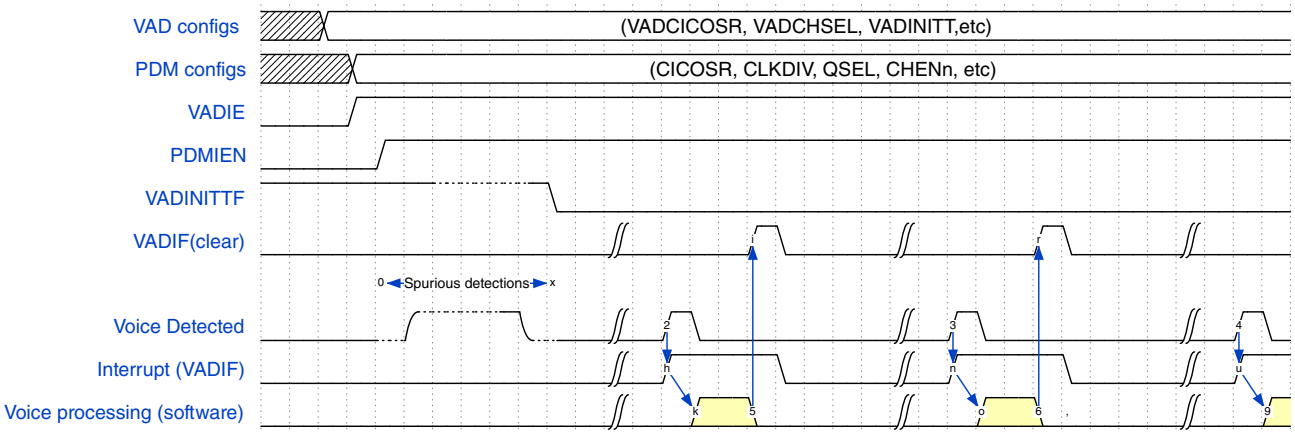


Figure 56-7. Energy-based initialization sequence

56.5.2.2 HWVAD in Envelope-based mode

The sequence of steps to initialize MICIL with the voice activity detector in Envelope-based mode is as follows:

1. Program the control bitfields as desired for your application (CLKDIV, CHnEN, FIFOWMK, DISEL, ERREN etc.). For more details see PDM_CLK_ROOT rate and CLKDIV calculation examples.
2. Configure general HWVAD parameters:
   a. CIC oversampling rate (VADCICOSR bitfield).
   b. Configure the voice source channel (VADCHSEL bitfield).
   c. Configure the internal initialization timer (VADINITT bitfield) if required. More details in Initialization Time.
   d. Assert VADEN bitfield.
3. Configure input, noise and signal gains according the environment noise conditions:
   a. Configure the input gain (VADINPGAIN bitfield).
   b. Configure the signal gain (VADSGAIN bitfield).
   c. Configure the noise gain (VADNGAIN bitfield).
4. Enable the high-pass filter if required (VADHPF bitfield).
5. Put HWVAD in envelope-based mode:
   a. Keep VADNFILADJ bitfield cleared.
   b. Assert the VADFRENDIS bitfield.
   c. Assert the VADPREFEN bitfield.
   d. Assert the VADSFILEN bitfield.
   e. Assert the VADSMAXEN bitfield.
   f. Clear the VADNFILAUTO bitfield.
   g. Assert the VADNMINEN bitfield.
   h. Assert the VADNDECEN bitfield.

i. Assert the VADNOREN bitfield.

6. Initialize the Zero-Crossing Detector (Zero-Crossing Detector Initialization) if required.
7. Start the overall module asserting the PDMIEN bit.
8. Wait for VADINITF to be cleared.
9. Write VADST10 bitfield three times.
10. Enable interrupts if required (VADIE and VADERIE bitfields).
11. The module is ready to receive data from the input ports of the enabled channels and to trigger interrupts when voice is detected.

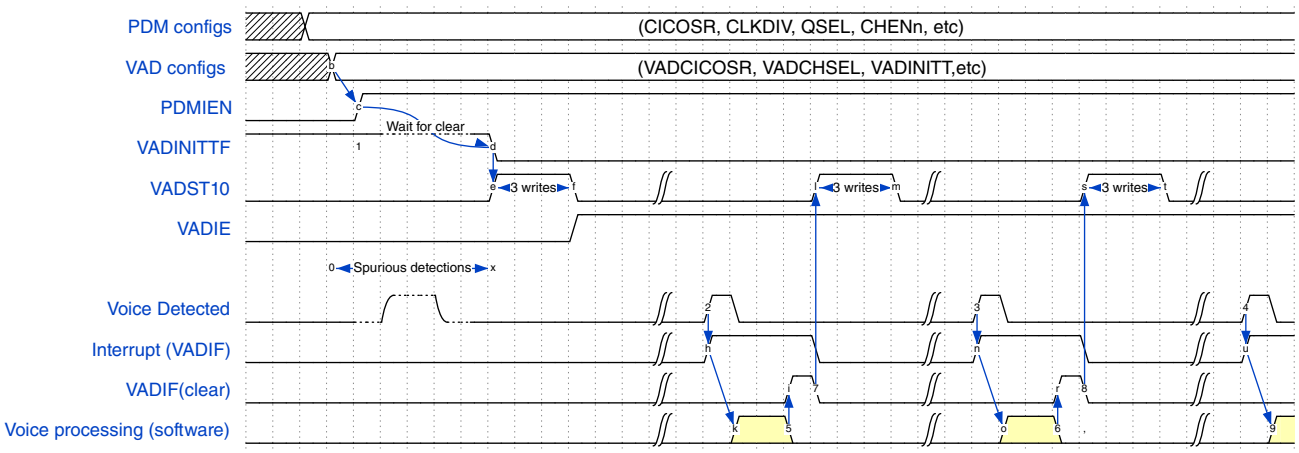


Figure 56-8. Envelope-based initialization sequence

56.5.2.3 Zero-Crossing Detector Initialization

The sequence of steps to initialize the Zero-Crossing Detector in Automatic mode is as follows:

1. Configure VADZCDADJ bitfield.
2. Configure an initial threshold in VADZCDTH according to the environmental noise conditions if required.
3. Configure VADZCDAND bitfield.
4. Keep VADZCDAUTO bitfield asserted.
5. Assert VADZCDEN.

The sequence of steps to initialize the Zero-Crossing Detector in Manual mode is as follows:

1. Configure a threshold in VADZCDTH according to the environmental noise conditions if required.
2. Configure VADZCDAND bitfield.
3. Clear VADZCDAUTO bitfield.

4. Assert VADZCDEN.

56.6 Application Information

56.6.1 Reconfiguration Procedure

The sequence of steps for the block reconfiguration is as follows:

1. Deassert PDMIEN bit.
2. Wait to BSY_FIL to be cleared.
3. Run a soft-reset cycle.
4. Run a initilialization procedure as described in Initialization Procedure without HWVAD or Initialization Procedure with HWVAD.

56.7 PDM register descriptions

This section provides the memory map and detailed description of all PDM registers. Any access to reserved areas can issue a slave bus error indication.

The registers allocated in this memory map are implemented for up to 4 pairs of microphones, that means 8 channels are available.

56.7.1 PDM memory map

PDM base address: 40C2_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PDM Control register 1 (CTRL_1)	32	RW	0000_0000
4	PDM Control register 2 (CTRL_2)	32	RW	0000_0000
8	PDM Status register (STAT)	32	W1C	0000_0000
10	PDM FIFO Control register (FIFO_CTRL)	32	RW	0000_0007
14	PDM FIFO Status register (FIFO_STAT)	32	W1C	0000_0000
24 - 40	PDM Output Result Register (DATACH0 - DATACH7)	32	RO	0000_0000
64	PDM DC Remover Control register (DC_CTRL)	32	RW	0000_0000
74	PDM Range Control register (RANGE_CTRL)	32	RW	0000_0000

Table continues on the next page...

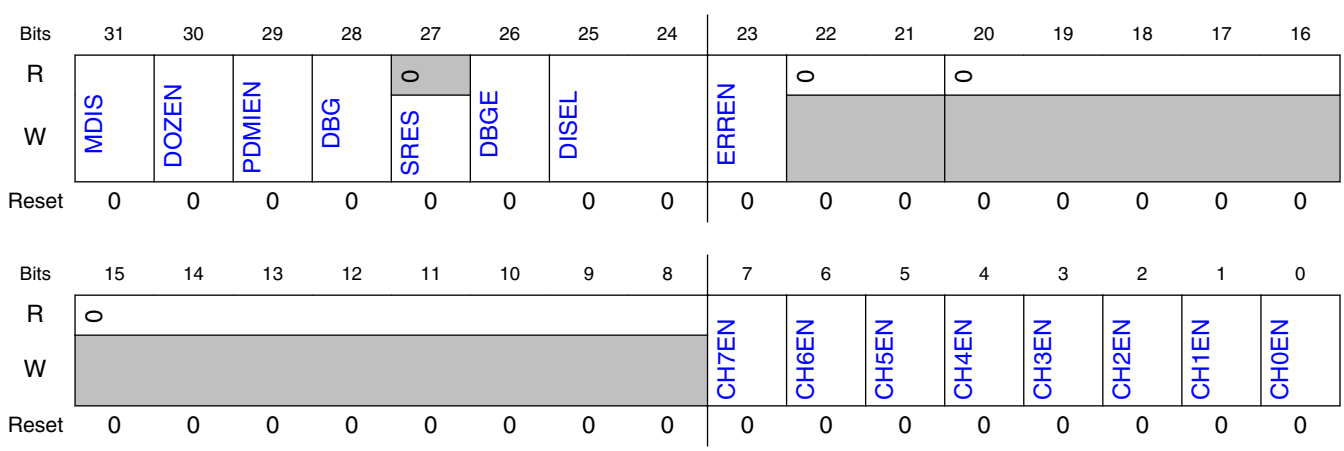
Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
7C	PDM Range Status register (RANGE_STAT)	32	W1C	0000_0000
90	Voice Activity Detector 0 Control register (VAD0_CTRL_1)	32	RW	0000_0000
94	Voice Activity Detector 0 Control register (VAD0_CTRL_2)	32	RW	000A_0000
98	Voice Activity Detector 0 Status register (VAD0_STAT)	32	W1C	8000_0000
9C	Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)	32	RW	0000_0000
A0	Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)	32	RW	8000_0000
A4	Voice Activity Detector 0 Noise Data (VAD0_NDATA)	32	RO	0000_0000
A8	Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)	32	RW	0000_0004

56.7.2 PDM Control register 1 (CTRL_1)

56.7.2.1 Offset

Register	Offset
CTRL_1	0h

56.7.2.2 Diagram



56.7.2.3 Fields

Field	Description
31 MDIS	Module Disable Puts the whole PDM in Disable/Low Leakage mode. The generated PDM_CLK is stopped and consequently no data is generated by the microphones. **NOTE:** When MDIS=1, writes to CTRL_1 and CTRL_2 only are allowed with the exception of writes to the DBG and SRES bits. A slave bus error indication will be issued when any register except CTRL_1 and CTRL_2 is written when MDIS=1. 0 - Normal Mode 1 - Disable/Low Leakage Mode
30 DOZEN	DOZE enable When SoC is in Doze mode and DOZEN is active the module enters to Disable/Low Leakage mode.
29 PDMIEN	PDM Enable Starts the module operation. Only the channels enabled by CHnEN or VAD enabled by VADEN will start to operate. When it is cleared, the whole module is stopped after to finish the pending filtering. **NOTE:** This field must be the latest to be enabled, for more information refer to Initialization . 0 - PDM stopped 1 - PDM operation started
28 DBG	Debug Mode Enables the Debug mode. The module can run on this mode depending on the DBGE bit value. **NOTE:** This bit is writable only when MDIS is cleared. 0 - Normal Mode 1 - Debug Mode
27 SRES	Software-reset bit Provides the CPU with the capability to initialize the PDM module through the slave-bus interface. This bit always reads as zero, and this bit is only effective when MDIS is cleared. This bit is self-cleared. **NOTE:** This bit must be asserted only when BSY_FIL=0 This bit can be used to clear all data in the filters. It resets the following blocks and/or registers: • All internal buffers in decimation filters and Voice Activity Detectors. • All FIFOs. • All CHnF flags and LOWFREQF flag on the STAT register, and the FIFO_STAT register. • Time Generator. 0 - No action 1 - Software reset
26 DBGE	Module Enable in Debug Enables operation in Debug mode. 0 - Disabled after completing the current frame 1 - Enabled
25-24 DISEL	DMA Interrupt Selection

Table continues on the next page...

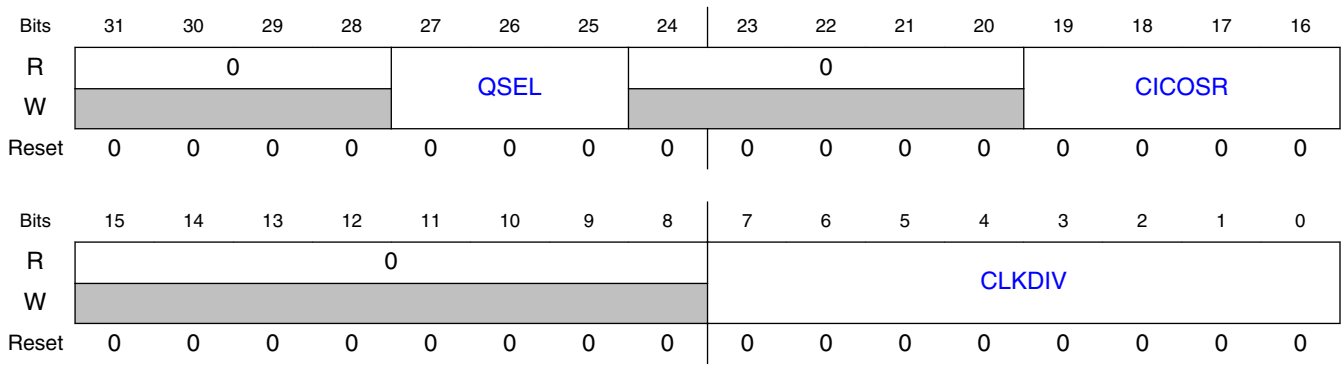
Field	Description
	Selects if the module performs DMA requests or filter interrupt requests when the number of FIFO elements surpasses the configured watermark. Note that requests are only generated from the channels whose CHnEN bits are asserted. **NOTE:** This bit mustn't change when BSY_FIL bit is asserted 00 - DMA and interrupt requests disabled 01 - DMA requests enabled 10 - Interrupt requests enabled 11 - Reserved
23 ERREN	Error Interruption Enable Enables error interrupts (exceptions). This error could be caused by: • an overflow or underflow in the FIFOs (FIFOOVF or FIFOUND bits), • an overflow or underflow in the decimations filters (the RANGEOVF or RANGEUNF bits), or • not enough PDM internal clock frequency (LOWFREQF bit). **NOTE:** This bit mustn't change when BSY_FIL bit is asserted 0 - Error Interrupts disabled 1 - Error Interrupts enabled
22-21 —	Reserved
20-16 —	Reserved
15-8 —	Reserved
7-0 CHnEN	Channel n Enable Enables the decimation filter channel n. Refer to Initialization for more information. **NOTE:** This bit mustn't change when BSY_FIL bit is asserted.

56.7.3 PDM Control register 2 (CTRL_2)

56.7.3.1 Offset

Register	Offset
CTRL_2	4h

56.7.3.2 Diagram



56.7.3.3 Fields

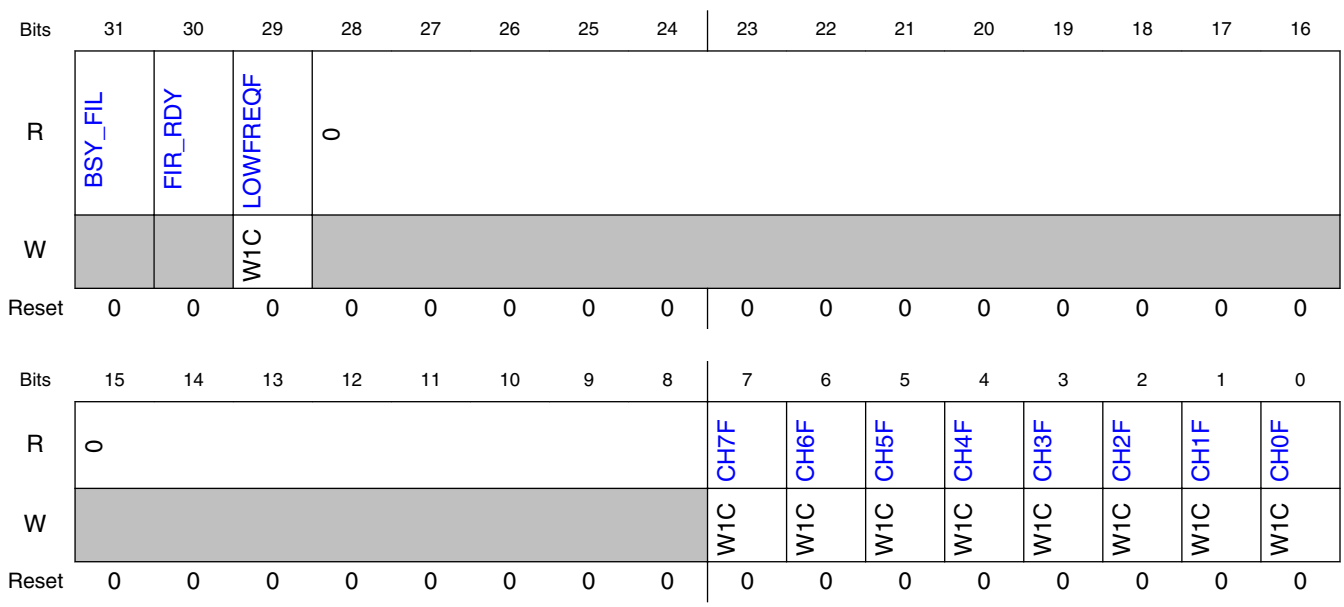
Field	Description
31-28 —	Reserved
27-25 QSEL	Quality Mode Defines the actual quality mode of the decimation filter. **NOTE:** This bitfield mustn't change when BSY_FIL bit is asserted. 000 - Medium quality mode 001 - High quality mode 100 - Very low quality 2 mode 101 - Very low quality 1 mode 110 - Very low quality 0 mode 111 - Low quality mode
24-20 —	Reserved
19-16 CICOSR	CIC Decimation Rate Helps to configure the Equation 8 on page 3425. **NOTE:** This bitfield mustn't change when BSY_FIL bit is asserted.
15-8 —	Reserved
7-0 CLKDIV	Clock Divider Configures the internal clock divider value. **NOTE:** This bitfield mustn't change when BSY_FIL bit is asserted. **NOTE:** CLKDIV=0 has the same effect than CLKDIV=1. CLKDIV=0 or 1 has the same effect than CLKDIV=2 during Medium Quality mode. CLKDIV=0 or 1 is not allowed during High Quality mode.

56.7.4 PDM Status register (STAT)

56.7.4.1 Offset

Register	Offset
STAT	8h

56.7.4.2 Diagram



56.7.4.3 Fields

Field	Description
31 BSY_FIL	Busy Flag Signalizes when at least a decimation filter channel or at least a HWVAD is running and the PDM_CLK is being generated. This bit could be used as a confirmation that all decimation filters are stopped in a transition to Debug (when DBGE=0) or Disable/Low Leakage mode. 0 - PDM is stopped

Table continues on the next page...

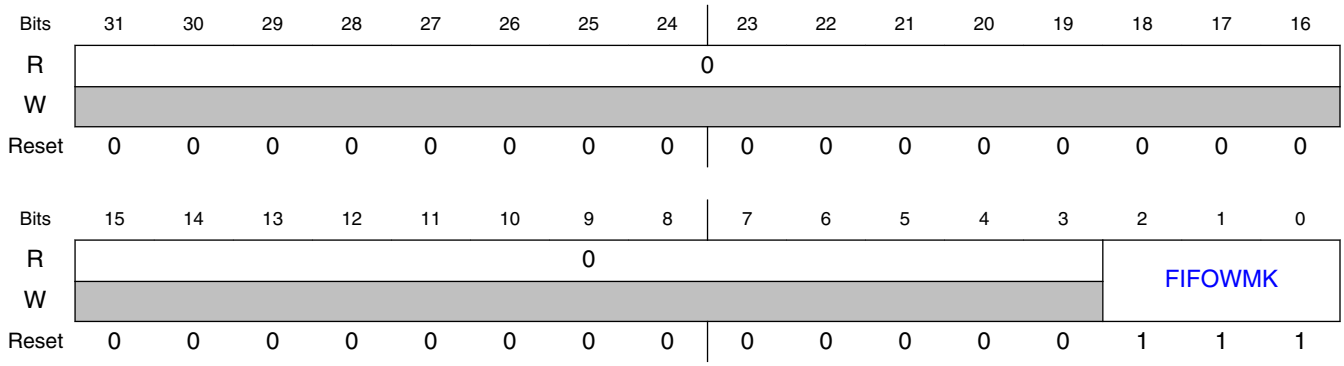
Field	Description
	1 - PDM is running
30 FIR_RDY	Filter Data Ready Indicates that all the decimation filter buffers have been updated, and all data provided from now on will be reliable. 0 - Filter data is not reliable 1 - Filter data is reliable
29 LOWFREQF	Low Frequency Flag Indicates that the CLKDIV bitfield value is not high enough to filter all enabled channels , and the half band filters didn't finish their operations before the next sample arrived. See Minimum Required CLKDIV . Write a logic 1 to this field to clear this flag. **NOTE:** The filtering results in this condition are unpredictable. 0 - CLKDIV value is OK 1 - CLKDIV value is too low
28-16 —	Reserved
15-8 —	Reserved
7-0 CHnF	Channel n Output Data Flag Indicates that the FIFO from this channel has surpassed its watermark value and the data is available at the DATACHn register. This flag contributes to generate an Interrupt or DMA request if enabled by DISEL in the CTRL_1. Write a logic 1 to this field to clear this flag. If DMA request is set (DISEL = 01) the CHnF flag will be set according to watermark level and will be cleared , in the other cases (IRQ included) the CHnF flag will be set according to the watermark level and will be cleared with the w1c functionality of the bit. **NOTE:** If DMA is enabled, this flag is self-cleared when the DMA transaction is finished . 0 - Channel's FIFO did not reach the number of elements configured in watermark bit-field 1 - Channel's FIFO reached the number of elements configured in watermark bit-field

56.7.5 PDM FIFO Control register (FIFO_CTRL)

56.7.5.1 Offset

Register	Offset
FIFO_CTRL	10h

56.7.5.2 Diagram



56.7.5.3 Fields

Field	Description
31-3 —	Reserved
2-0 FIFOWMK	FIFO Watermark Control Controls the watermark of the FIFO used to set CHnF. When the number of results in the FIFO is greater than FIFOWMK value then the CHnF flag is set. A DMA request or interrupt can be also generated according to DISEL bit-field. **NOTE:** This mustn't change when BSY_FIL is asserted. Before changing the watermark value service all filtered data and clear channel flags.

56.7.6 PDM FIFO Status register (FIFO_STAT)

56.7.6.1 Offset

Register	Offset
FIFO_STAT	14h

56.7.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOUND7	FIFOUND6	FIFOUND5	FIFOUND4	FIFOUND3	FIFOUND2	FIFOUND1	FIFOUND0	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.7.6.3 Fields

Field	Description
31-16 —	Reserved
15-8 FIFOUNDn	FIFO Underflow Exception flag for Channel n Indicates an exceptional underflow condition in the FIFO. If the FIFO is empty and a read operation is requested then this flag is set. Write a logic 1 to this field to clear this flag. 0 - No exception by FIFO Underflow 1 - Exception by FIFO underflow
7-0 FIFOOVFn	FIFO Overflow Exception flag for Channel n Indicates an exceptional overflow condition in the FIFO. If the FIFO is full and an additional result is received to be written then this flag is set. Write a logic 1 to this field to clear this flag. 0 - No exception by FIFO overflow 1 - Exception by FIFO overflow

56.7.7 PDM Output Result Register (DATACH0 - DATACH7)

56.7.7.1 Offset

For a = 0 to 7:

Register	Offset
DATACHa	24h + (a × 4h)

56.7.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.7.7.3 Fields

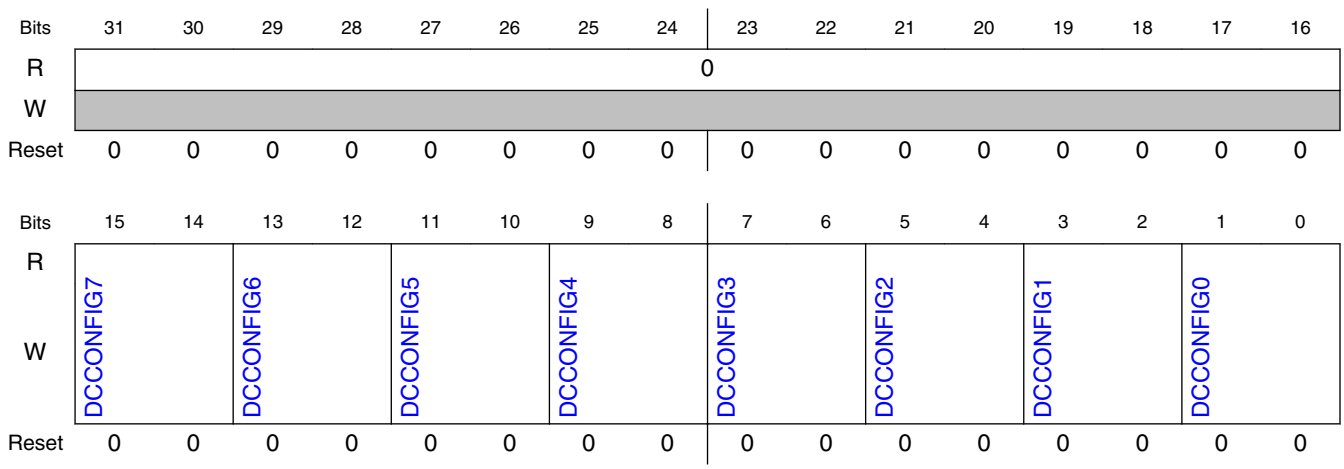
Field	Description
31-0 DATA	Channel n Data Reflects the word value stored at the top of the Channel's n FIFO. The samples are integer signed numbers in two's complement format. **NOTE:** Even FIFO data width is 32-bits, only the 24 more significative bits have information, the other bits are always zero.

56.7.8 PDM DC Remover Control register (DC_CTRL)

56.7.8.1 Offset

Register	Offset
DC_CTRL	64h

56.7.8.2 Diagram



56.7.8.3 Fields

Field	Description
31-16 —	Reserved
15-14: DCCONFIG7 13-12: DCCONFIG6 11-10: DCCONFIG5 9-8: DCCONFIG4 7-6: DCCONFIG3 5-4: DCCONFIG2 3-2: DCCONFIG1	Channel n DC Remover Configuration Defines the value of the cut-off frequency of the DC Remover. 00 - DC Remover cut-off at 21Hz 01 - DC Remover cut-off at 83Hz 10 - DC Remover cut-off at 152Hz 11 - DC Remover is bypassed

Field	Description
1-0: DCCONFIG0	

56.7.9 PDM Range Control register (RANGE_CTRL)

56.7.9.1 Offset

Register	Offset
RANGE_CTRL	74h

56.7.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RANGEADJ7				RANGEADJ6				RANGEADJ5				RANGEADJ4			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RANGEADJ3				RANGEADJ2				RANGEADJ1				RANGEADJ0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.7.9.3 Fields

Field	Description
31-28: RANGEADJ7 27-24: RANGEADJ6 23-20: RANGEADJ5 19-16: RANGEADJ4	Channel n Range Adjustment Adjusts the dynamic range of the CIC filter. It is a unsigned positive value. This bitfield must be set within the following ranges: If QSEL in [HQ,LQ]: $RANGEADJn <= 37 - \text{ceil}(7\log_2(32 - 2CICOSR))$ If QSEL in [MQ,VLQ1]: $RANGEADJn <= 37 - \text{ceil}(7\log_2(16 - CICOSR))$

Field	Description
15-12: RANGEADJ3 11-8: RANGEADJ2 7-4: RANGEADJ1 3-0: RANGEADJ0	If QSEL in [LQ,VLQ2]: $RANGEADJn <= 36 - \text{ceil}(7\log_2(16 - CICOSR))$ where OSR is the Equation 1 on page 3418 and the ceiling function, ceil(x), maps to the least integer greater than or equal to x.

56.7.10 PDM Range Status register (RANGE_STAT)

56.7.10.1 Offset

Register	Offset
RANGE_STAT	7Ch

56.7.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RANGEUNF7	RANGEUNF6	RANGEUNF5	RANGEUNF4	RANGEUNF3	RANGEUNF2	RANGEUNF1	RANGEUNF0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RANGEOVF7	RANGEOVF6	RANGEOVF5	RANGEOVF4	RANGEOVF3	RANGEOVF2	RANGEOVF1	RANGEOVF0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.7.10.3 Fields

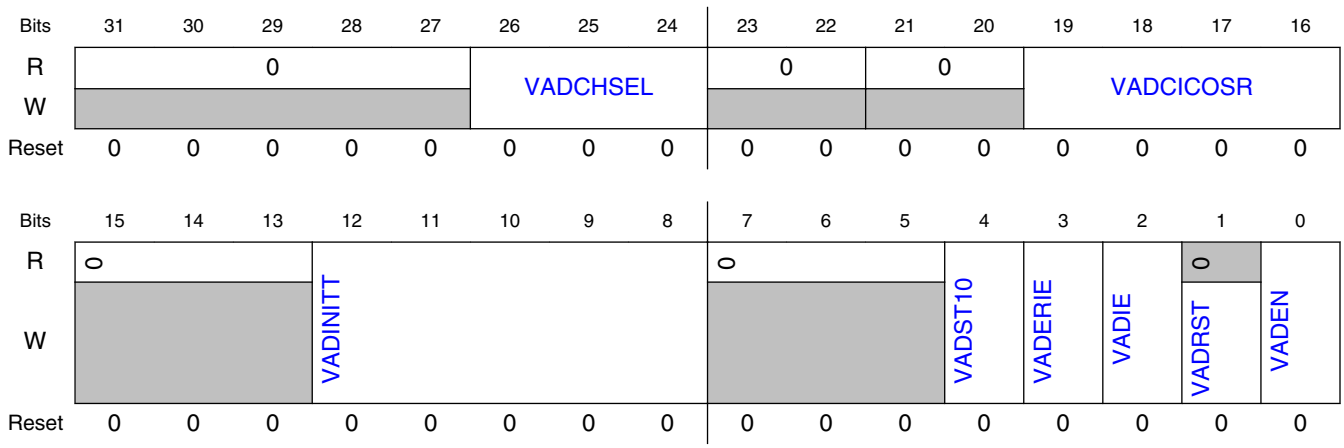
Field	Description
31-24 —	Reserved
23-16 RANGEUNFn	Channel n Range Underflow Error Flag Indicates an exceptional underflow condition in PDM range. **NOTE:** If this error flag is asserted, channel n data is not reliable. Require to adjust the RANGEADJn value until this flag is not asserted anymore. 0 - No exception by range underflow 1 - Exception by range underflow
15-8 —	Reserved
7-0 RANGEOVFn	Channel n Range Overflow Error Flag Indicates an exceptional overflow condition in PDM range. **NOTE:** If this error flag is asserted, channel n data is not reliable. Require to adjust the RANGEADJn value until this flag is not asserted anymore. 0 - No exception by range overflow 1 - Exception by range overflow

56.7.11 Voice Activity Detector 0 Control register (VAD0_CTRL_1)

56.7.11.1 Offset

Register	Offset
VAD0_CTRL_1	90h

56.7.11.2 Diagram



56.7.11.3 Fields

Field	Description
31-27 —	Reserved
26-24 VADCHSEL	Voice Activity Detector Channel Selector Selects the number of channel which the hardware voice activity detector will work. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
23-22 —	Reserved
21-20 —	Reserved
19-16 VADCICOSR	Voice Activity Detector CIC Oversampling Rate Defines the oversampling rate of the CIC filter. The oversampling rate is (16-CICOSR). **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
15-13 —	Reserved
12-8 VADINITT	Voice Activity Detector Initialization Time Selects the number of frames to be used to initialize the voice detection. During this period the output of the voice activity detector is forced to inactive. For more information, refer to Initialization Time . **NOTE:** The initialization time will be the time taken by [VADINITT+1] frames. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted.
7-5 —	Reserved
4	Voice Activity Detector Internal Filters Initialization

Table continues on the next page...

Field	Description
VADST10	Initializes the signal and noise filter with pre-filter value. **NOTE:** If the use of this bit is required, it should be stay pulsed by more than 2 cycles of system clock . **NOTE:** There is not voice activity detection when VADST10 is asserted. 0 - Normal operation. 1 - Filters are initialized.
3 VADERIE	Voice Activity Detector Error Interruption Enable Enables Error Interrupts in the PDM. This error could be caused by an overflow in the input of the HWVAD and it is signalized by the VADINSATF flag. 0 - HWVAD Error Interrupts disabled 1 - HWVAD Error Interrupts enabled
2 VADIE	Voice Activity Detector Interruption Enable Enables interrupts in the PDM when voice activity event has been detected by the Hardware Voice Activity Detector (HWVAD). See the VADIF for more information. 0 - HWVAD Interrupts disabled 1 - HWVAD Interrupts enabled
1 VADRST	Voice Activity Detector Reset Provides the CPU with the capability to initialize the VAD through the slave-bus interface. This bit always reads as zero. This bit is self-cleared. See VAD Soft-reset.
0 VADEN	Voice Activity Detector Enable Enables the use of the Hardware Voice Activity Detector. **NOTE:** This bitfield mustn't change when BSY_FIL is asserted. 0 - The HWVAD is disabled 1 - The HWVAD is enabled

56.7.12 Voice Activity Detector 0 Control register (VAD0_CTRL_2)

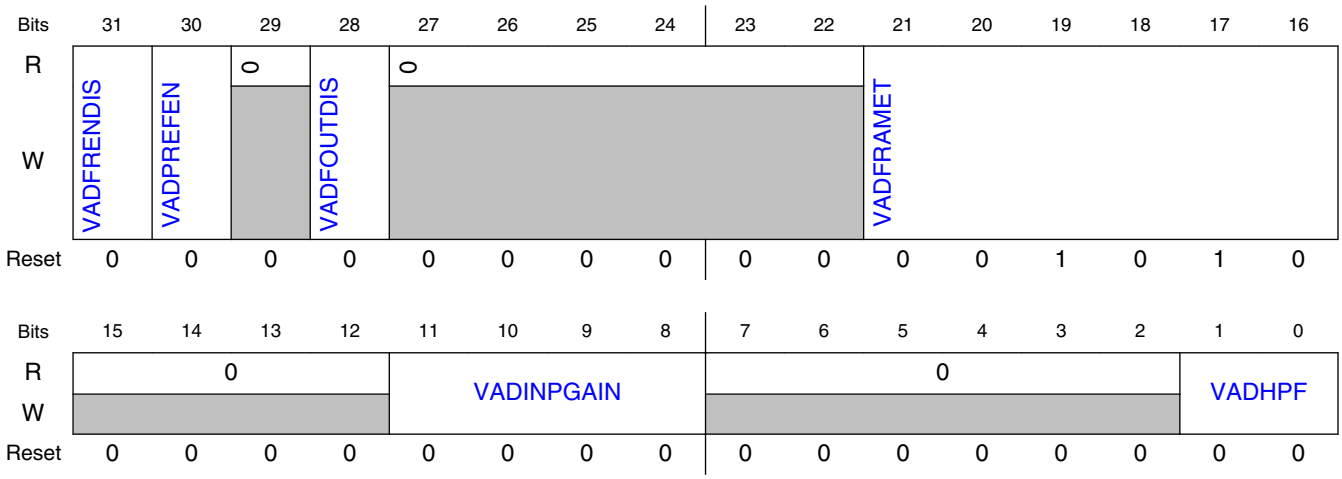
NOTE

This register mustn't change when BSY_FIL is asserted.

56.7.12.1 Offset

Register	Offset
VAD0_CTRL_2	94h

56.7.12.2 Diagram



56.7.12.3 Fields

Field	Description
31 VADFRENDIS	Voice Activity Detector Frame Energy Disable Disables the calculus of energy in a frame. For more information, refer to Noise Filter . 0 - Frame energy calculus enabled. 1 - Frame energy calculus disabled.
30 VADPREFEN	Voice Activity Detector Pre Filter Enable Enables a pre-filter in the input of Noise and Signal filters. For more information, refer to Pre-Filter . 0 - Pre-filter is bypassed. 1 - Pre-filter is enabled.
29 —	Reserved
28 VADFOUTDIS	Voice Activity Detector Force Output Disable Disables the output of the Energy/Envelope-based Detector. 0 - Output is enabled. 1 - Output is disabled.
27-22 —	Reserved
21-16 VADFRAMET	Voice Activity Detector Frame Time Selects the scale value to control the frame time duration as is described in Frame Time Setting . **NOTE:** VADFRAMET=0 is not allowed.
15-12	Reserved

Table continues on the next page...

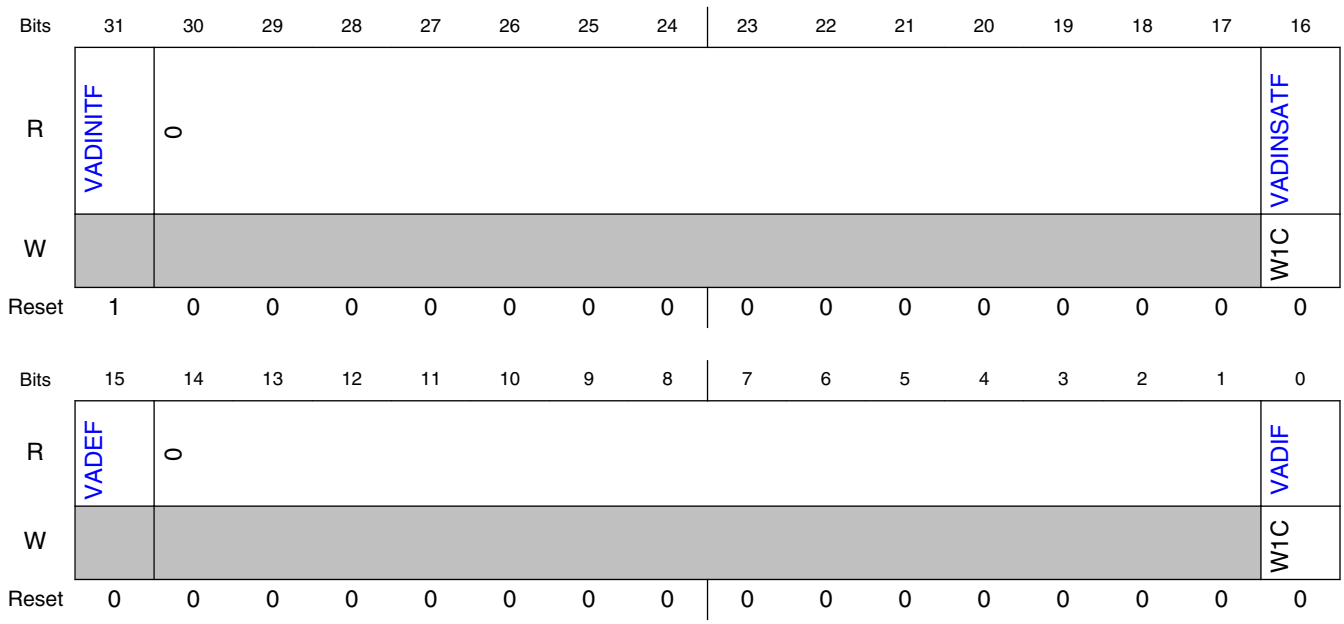
Field	Description
—	
11-8 VADINPGAIN	Voice Activity Detector Input Gain Configures the gain value in the HWVAD input as a positive or negative (two's complement) number of bits to shift.
7-2 —	Reserved
1-0 VADHPF	Voice Activity Detector High-Pass Filter Configures the internal high-pass filter. 00 - Filter bypassed. 01 - Cut-off frequency at 1750Hz. 10 - Cut-off frequency at 215Hz. 11 - Cut-off frequency at 102Hz.

56.7.13 Voice Activity Detector 0 Status register (VAD0_STAT)

56.7.13.1 Offset

Register	Offset
VAD0_STAT	98h

56.7.13.2 Diagram



56.7.13.3 Fields

Field	Description
31 VADINITF	Voice Activity Detector Initialization Flag Signalizes when the HWVAD is being initialized according to the VADINITT bitfield value. 0 - HWVAD is not being initialized. 1 - HWVAD is being initialized.
30-17 —	Reserved
16 VADINSATF	Voice Activity Detector Input Saturation Flag Indicates an exceptional saturation condition by an overflow or underflow in the HWVAD input. 0 - No exception 1 - Exception
15 VADEF	Voice Activity Detector Event Flag Indicates when voice activity event has been detected. 0 - Voice activity not detected 1 - Voice activity detected
14-1 —	Reserved
0	Voice Activity Detector Interrupt Flag

Field	Description
VADIF	Indicates when voice activity has been detected by the HWVAD. Write a logic 1 to this field to clear this flag. 0 - Voice activity not detected 1 - Voice activity detected

56.7.14 Voice Activity Detector 0 Signal Configuration (VAD0_SCONFIG)

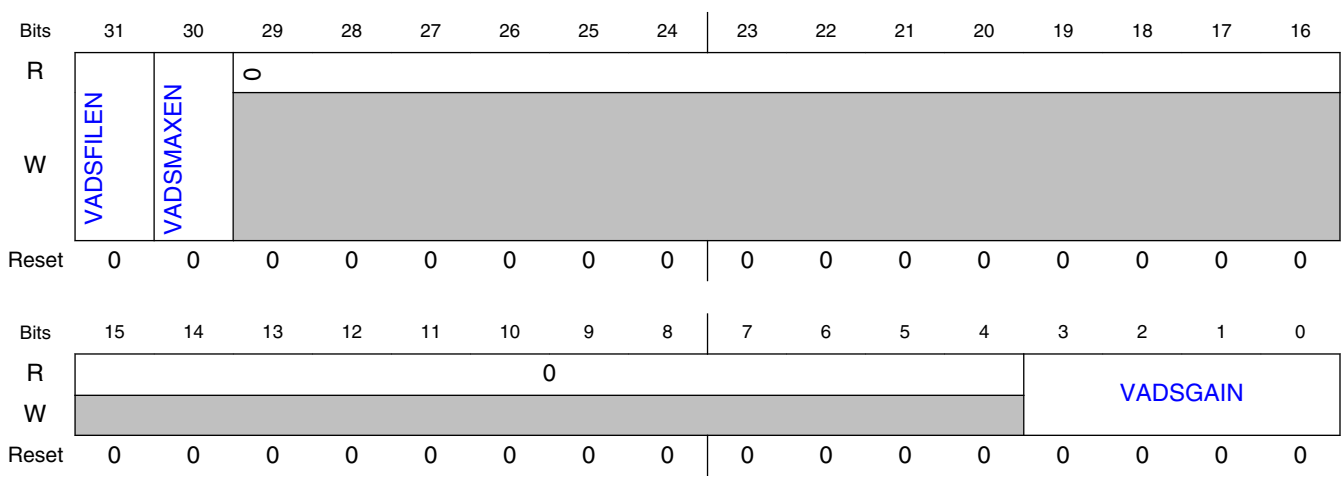
NOTE

These bit-fields must be configured when the PDM is not running.

56.7.14.1 Offset

Register	Offset
VAD0_SCONFIG	9Ch

56.7.14.2 Diagram



56.7.14.3 Fields

Field	Description
31 VADSFILEN	Voice Activity Detector Signal Filter Enable Enables the signal filter. For more information, refer to Signal Filter . 0 - Signal filter is disabled. 1 - Signal filter is enabled.
30 VADSMAXEN	Voice Activity Detector Signal Maximum Enable Selects whether or not a block to calculate the maximum value between the input and output of signal filter is used as signal estimation. For more information, refer to Signal Filter . 0 - Maximum block is bypassed. 1 - Maximum block is enabled.
29-4 —	Reserved
3-0 VADSGAIN	Voice Activity Detector Signal Gain Configures the gain value for the signal energy or envelope estimated. It is a unsigned value which is multiplied with the estimated signal data. **NOTE:** VADSGAIN=0 gain corresponds to 1.

56.7.15 Voice Activity Detector 0 Noise Configuration (VAD0_NCONFIG)

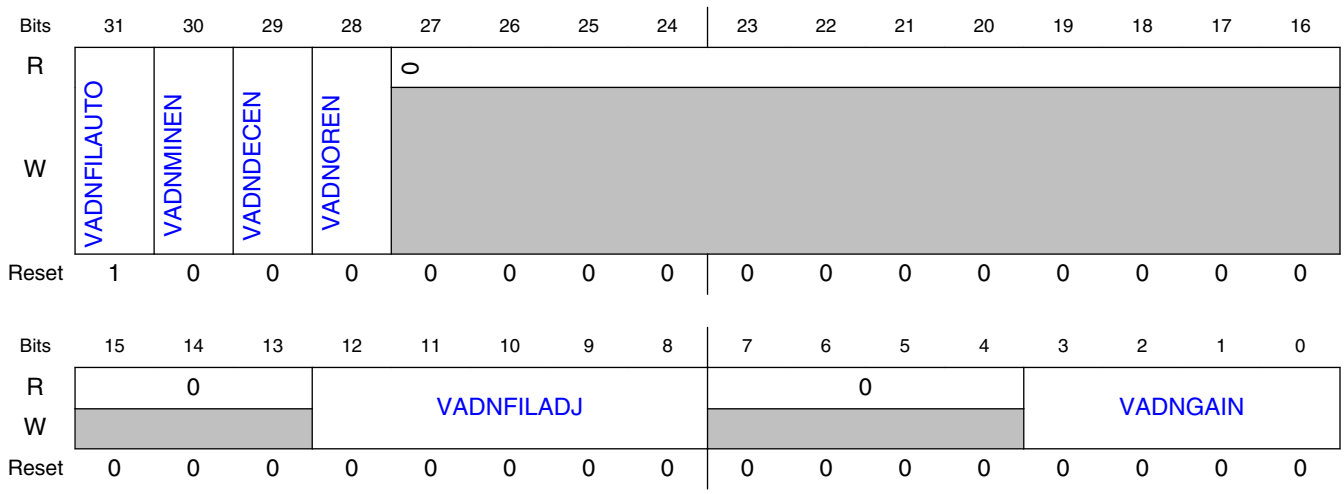
NOTE

These bit-fields must be configured when the PDM is not running.

56.7.15.1 Offset

Register	Offset
VAD0_NCONFIG	A0h

56.7.15.2 Diagram



56.7.15.3 Fields

Field	Description
31 VADNFILAUTO	Voice Activity Detector Noise Filter Auto Selects whether or not the noise filter is activated automatically based in voice activity information. For more information, refer to Noise Filter . 0 - Noise filter is always enabled. 1 - Noise filter is enabled/disabled based on voice activity information.
30 VADNMINEN	Voice Activity Detector Noise Minimum Enable Selects whether or not a block to calculate the minimum value between the input and output of noise filter is used as noise estimation. For more information, refer to Noise Filter . 0 - Minimum block is bypassed. 1 - Minimum block is enabled.
29 VADNDECEN	Voice Activity Detector Noise Decimation Enable Selects whether or not the input of the noise filter is decimated. For more information, refer to Noise Filter . 0 - Noise input is not decimated. 1 - Noise input is decimated.
28 VADNOREN	Voice Activity Detector Noise OR Enable Enables a OR logic in the output of minimum noise estimator block. For more information, refer to Filter Result Gain Setting . 0 - Noise input is not decimated. 1 - Noise input is decimated.
27-13 —	Reserved

Table continues on the next page...

Field	Description
12-8 VADNFILADJ	Voice Activity Detector Noise Filter Adjustment Selects the adjustment value of the noise filter. For more information, refer to Noise Filter .
7-4 —	Reserved
3-0 VADNGAIN	Voice Activity Detector Noise Gain Configures the gain value for the noise energy or envelope estimated. It is an unsigned value which is multiplied with the estimated noise data. **NOTE:** VADNGAIN=0 gain corresponds to 1.

56.7.16 Voice Activity Detector 0 Noise Data (VAD0_NDATA)

56.7.16.1 Offset

Register	Offset
VAD0_NDATA	A4h

56.7.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VADNDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.7.16.3 Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Field	Description
—	
15-0 VADNDATA	Voice Activity Detector Noise Data Stores the noise energy or noise envelope calculated by the HWVAD. It can be used by software for a further voice activity detection. **NOTE:** This field value should be ignored when VADFOUTDIS=1 or VADST10=1.

56.7.17 Voice Activity Detector 0 Zero-Crossing Detector (VAD0_ZCD)

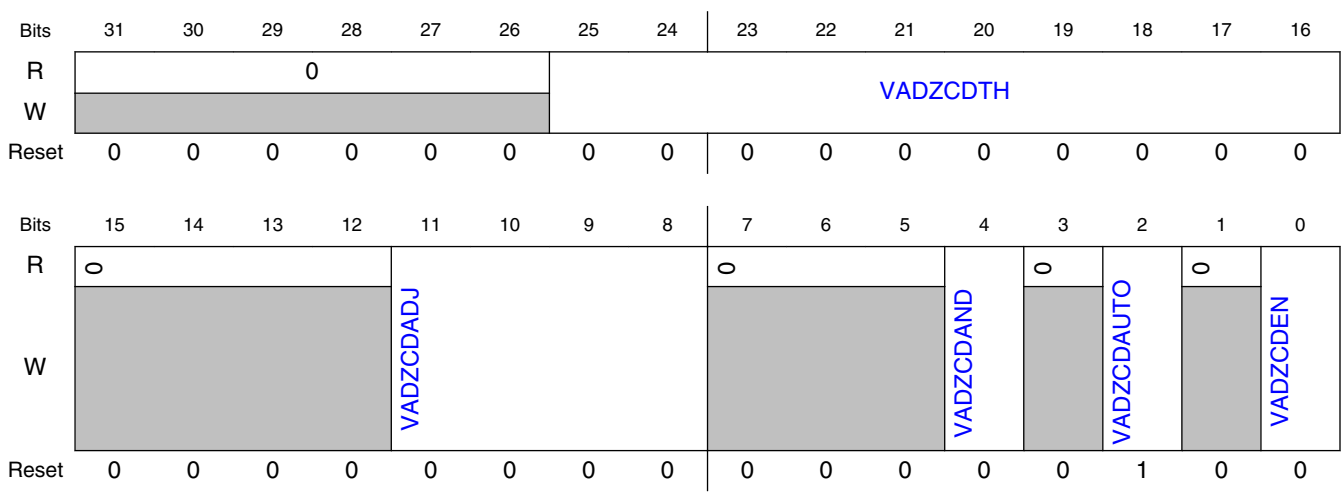
NOTE

These bit-fields must be configured when the PDM is not running.

56.7.17.1 Offset

Register	Offset
VAD0_ZCD	A8h

56.7.17.2 Diagram



56.7.17.3 Fields

Field	Description
31-26 —	Reserved
25-16 VADZCDTH	Zero-Crossing Detector Threshold Configures the initial value taken to estimate the ZCD threshold when it is estimated automatically (VASDZCDAUTO = 1), otherwise (VADZCDAUTO = 0) this bit-field is the static value of the ZCD threshold.
15-12 —	Reserved
11-8 VADZCDADJ	Zero-Crossing Detector Adjustment Selects the adjustment value of the threshold estimator. For more information, refer to Zero-Crossing Detector .
7-5 —	Reserved
4 VADZCDAND	Zero-Crossing Detector AND Behavior Selects whether the Zero-Crossing detector results will be AND'ed or OR'ed with the energy-based detection to form the voice activity detection event. 0 - The ZCD result is OR'ed with the energy-based detection. 1 - The ZCD result is AND'ed with the energy-based detection.
3 —	Reserved
2 VADZCDAUTO	Zero-Crossing Detector Automatic Threshold Enables the automatic estimation of the ZCD threshold. 0 - The ZCD threshold is not estimated automatically 1 - The ZCD threshold is estimated automatically
1 —	Reserved
0 VADZCDEN	Zero-Crossing Detector Enable Enables the Zero-Crossing Detector (ZCD) in the Hardware Voice Activity Detector (HWVAD). 0 - The ZCD is disabled 1 - The ZCD is enabled

Chapter 57
Medium Quality Sound (MQS)

57.1 Chip-specific MQS information

Table 57-1. Reference links to related information

Topic	Related module or subsystem	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Audio Subsystem	Audio Subsystem	Audio Subsystem Overview
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

57.2 Overview

Medium quality sound (MQS) is used to generate medium quality audio via a standard GPIO in the pinmux, allowing the user to connect stereo speakers or headphones to a power amplifier without an additional DAC chip.

57.2.1 Block Diagram

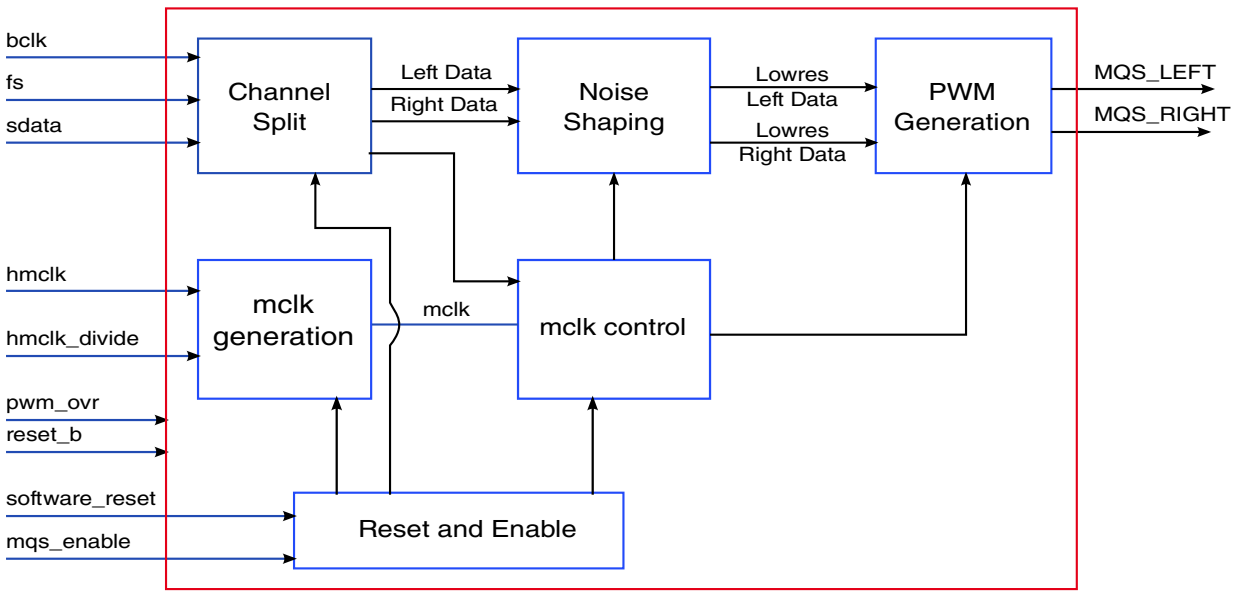


Figure 57-1. Block Diagram

57.2.2 Features

MQS has the following sub-modules:

- Channel Split: Splits the I2S signals into separate left channel and right channel audio data.
- Noise Shaping: Uses the sigma-delta algorithm to generate low-resolution, very high sampling audio, while the audio sampling rate is increased.
- PWM Generation: Generates the bit stream to the GPIO, which is then used to drive the amplifier and then to drive the external speakers or headphones.
- Mclk Generation: Used to generate the master clock (mclk). The frequency of mclk is determined by the final bit duration of PWM generation module.
- Mclk Control:Used as a metronome to co-ordinate the different functional blocks working synchronously.
- Reset and Enable: Used to generate the reset and enable logic to different clock domains.

57.3 Functional Description

MQS accepts the following inputs:

- 2-channel, LSB-valid 16-bit, MSB shift-out first serial data (sdata)

- Frame sync asserting with the first bit of the frame (fs)
- Bit clock used to shift data out on the positive clock edge (bclk)

The signals from the SAI3 are in left_justified format. MQS provides the SNR target as no more than 20dB for the signals below 10 kHz. The signals above 10 kHz will have worse THD+N values.

MQS provides only simple audio reproduction. No internal pop, click or distortion artifact reduction methods are provided.

57.4 External Signals

The following table describes the external signals of MQS:

Table 57-2. MQS External Signals

Signal	Description	Direction
MQS_LEFT	Left signal output	O
MQS_RIGHT	Right signal output	O

57.4.1 Interface Signals

MQS module has the following interface signals.

Signal Name	In/Out	BitWidth	Description	Comments
reset_b	In	1	asynchronous reset	
software_reset	In	1	Software reset	From IOMUXC_GPR3
mqs_enable	In	1	module enable	From IOMUXC_GPR3
pwm_ovr	In	1	PWM oversampling ratio 1=64, 0=32	From IOMUXC_GPR3
hmclk	In	1	Maximum bit clock, used to generate the mclk, divider ratio is controlled by mqs_hmclk_divide	Max 66.5MHzTypical 24.576MHz
hmclk_divide	In	8	Divider ration control for mclk from hmclk	From IOMUXC_GPR3
bclk	In	1	bit clock from I2S signal	
fs	In	1	frame sync clock from I2S signal	
sdata	In	1	serial audio data from I2S signal	

57.5 Application Information

MQS has no internal programmable registers. But it does have some programmability from IOMUXC_GPR3.

Register Bitfields	Description
IOMUXC_GPR3[MQS_OVERSAMPLE]	Used to control the PWM oversampling rate compared with mclk. 1—64, 0—32.
IOMUXC_GPR3[MQS_EN]	MQS enable. 1—Enable MQS, 0—Disable MQS
IOMUXC_GPR3[MQS_SW_RST]	MQS software reset. 1—Enable software reset for MQS 0—Exit software reset for MQS
IOMUXC_GPR3[MQS_CLK_DIV]]	Divider ration control for mclk from hmclk. 0—mclk frequency = hmclk frequency; 1—mclk frequency = ½*hmclk frequency; 2—mclk frequency = 1/3*hmclk frequency; …; n—mclk frequency = 1/(n+1)*hmclk frequency

57.5.1 Usage Model

The user needs to program the SAI3 to output 2-channel LSB-16 bit active left-justified signal, and then program the related IOMUXC_GPR3 bits.

Due to the different devices connected to MQS, and different high frequency behaviors of the connected analog circuits, the user needs choose the appropriate MQS_CLK_DIV and MQS_OVERSAMPLE values for the best audible effects.

Chapter 58
Synchronous Audio Interface (SAI)

58.1 Chip-specific SAI information

Table 58-1. Reference links to related information

Topic	Related module or subsystem	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Audio Subsystem	Audio Subsystem	Audio Subsystem Overview
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

Number of TX and RX data lines for SAI1 and SAI2/SAI3 are 4, and 1 respectively.

NOTE

In this device, it does NOT support synchronous mode between different SAI peripherals.

MCLK sources are determined outside of the SAI block, as defined in the below table. For additional IOMUXC GPR information, please see the IOMUX chapter. For more details about the features and clocking of the different SAI instances, see theAudio Subsystem Overview chapter.

Table 58-2. SAI MCLK Information

SAI Instance	MCLK	MCLK sources
SAI1	MCLK1/MCLK2	From IOMUX GPR0: • sai1_clk_root • sai2_clk_root • sai3_clk_root • sai1_ipg_clk_sai_mclk • sai2_ipg_clk_sai_mclk • sai3_ipg_clk_sai_mclk
	MCLK3	From IOMUX GPR0: • spdif_clk_root • spdif_tx_clk2 • spdif_srclk • spdif_outclock
SAI2	MCLK1	sai2_clk_root
	MCLK2	SAI2_MCLK (pin)
	MCLK3	From IOMUX GPR1: • spdif_clk_root • spdif_tx_clk2 • spdif_srclk • spdif_outclock
SAI3	MCLK1	sai2_clk_root
	MCLK2	SAI3_MCLK (pin)
	MCLK3	From IOMUX GPR2: • spdif_clk_root • spdif_tx_clk2 • spdif_srclk • spdif_outclock
SAI4	MCLK	SAI4_MCLK (pin)

58.2 Overview

The Synchronous Audio Interface (SAI) provides an interface that supports full-duplex serial interfaces with frame synchronization such as I^2S, AC97, TDM, and codec/DSP interfaces.

58.2.1 Block diagram

The following block diagram also shows the module clocks.

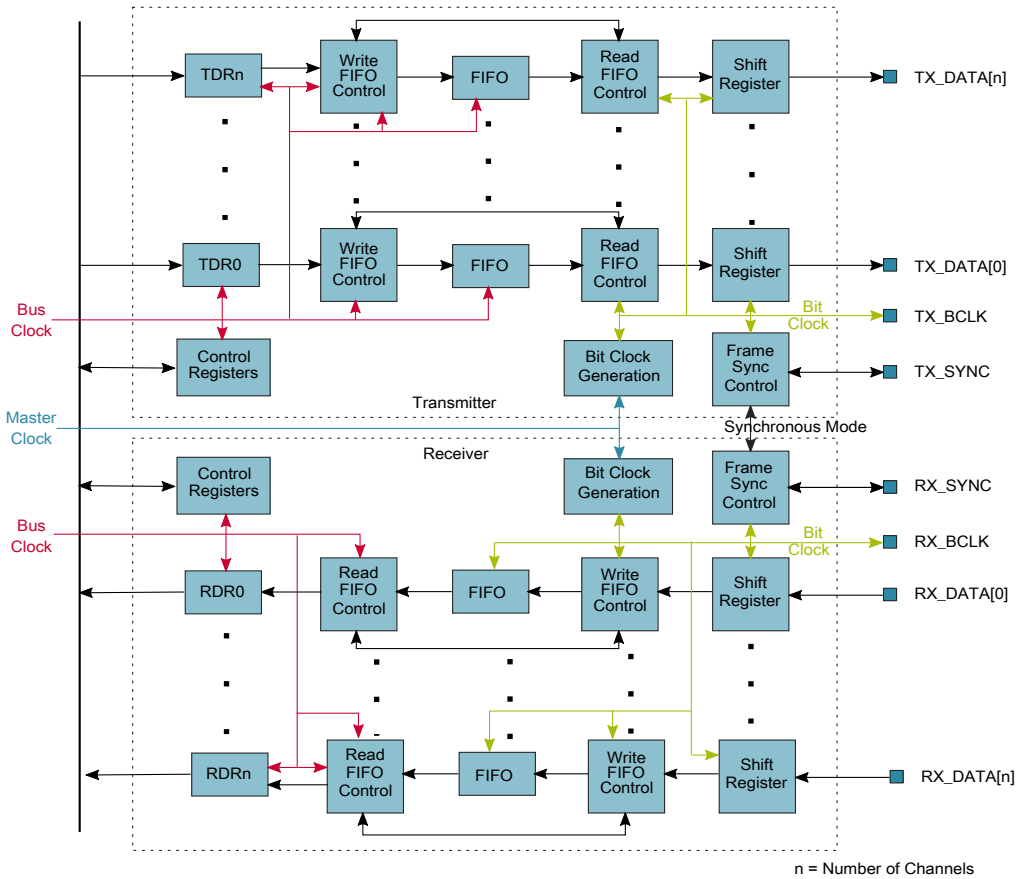


Figure 58-1. SAI block diagram

58.2.2 Features

Some of the features are not supported across all SAI instances; see the chip-specific SAI information in the first section of this chapter.

- Transmitter with independent bit clock and frame sync supporting 4 data lines
- Receiver with independent bit clock and frame sync supporting 4 data lines
- Each data line can support a maximum frame size of 32 words
- 8- to 32-bit word size
- Word size configured separately for the first word and remaining words in a frame
- Asynchronous 32 x 32-bit FIFO for each transmit and receive data line supporting:
  - Graceful restart after FIFO error
  - Automatic restart after FIFO error without software intervention
  - 8-bit and 16-bit data packing into each 32-bit FIFO word
  - Multiple data line FIFOs combining into single data line FIFO

58.3 Functional description

This section provides a complete functional description of the block.

58.3.1 Modes of operation

Module power modes include Run mode, Stop modes, and Debug mode.

58.3.1.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

58.3.1.2 Stop modes

In Stop mode, the SAI transmitter and/or receiver can continue operating provided the appropriate Stop Enable bit = 1 (TCSR[STOPE] and/or RCSR[STOPE], respectively), and provided the transmitter and/or receiver is/are using an externally generated bit clock or an Audio Master Clock that remains operating in Stop mode. The SAI transmitter and/or receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

In Stop mode, if the Transmitter Stop Enable (TCSR[STOPE]) bit is clear, the transmitter is disabled after completing the current transmit frame, and, if the Receiver Stop Enable (RCSR[STOPE]) bit is clear, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented (not acknowledged) while waiting for the transmitter and receiver to be disabled at the end of the current frame.

58.3.1.3 Debug mode

In Debug mode, the SAI transmitter and/or receiver continues operating if the Debug Enable bit is set. When TCSR[DBGE] or RCSR[DBGE] bit is clear and Debug mode is entered, the SAI is disabled after completing the current transmit or receive frame. Debug mode does not affect the transmitter and receiver bit clocks.

58.3.2 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other.

58.3.2.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If both the transmitter and receiver use the transmitter bit clock and frame sync:

- Configure the transmitter for asynchronous operation and the receiver for synchronous operation.
- Enable the receiver in synchronous mode only after both the transmitter and receiver are enabled.
- Enable the transmitter last and disable the tranmitter first.

If both the transmitter and receiver use the receiver bit clock and frame sync:

- Configure the receiver for asynchronous operation and the transmitter for synchronous operation.
- Enable the transmitter in synchronous mode only after both the receiver and transmitter are enabled.
- Enable the receiver last and disable the receiver first.

When operating in synchronous mode, the transmitter and receiver share only the bit clock, frame sync, and transmitter/receiver enable. Otherwise, the transmitter and receiver operate independently, although the configuration registers must be configured consistently across both the transmitter and receiver.

58.3.3 Frame sync configuration

When enabled, the SAI continuously transmits and/or receives frames of data. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. Within each frame, any given word can be masked causing the receiver to ignore that word and the transmitter to tri-state for the duration of that word.

The frame sync signal indicates the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Generate externally or internally
- Require active high or active low
- Assert with the first bit in frame or assert one bit early
- Assert for a duration of between 1 bit clock and the first word length
- Set frame length from 1 to 32 words per frame
- Support word length of 8 to 32 bits per word
    - First word length and remaining word lengths can be configured separately
- Transmit/Receive words MSB first or LSB first

These configuration options cannot be changed after enabling the SAI transmitter or receiver.

58.3.4 Data FIFO

Each transmit and receive channel includes a 32 x 32-bit size FIFO. To access the FIFO data use the SAI Transmit/Receive Data Registers.

58.3.4.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 58-2 for LSB First configurations and Figure 58-3 for MSB First configurations.

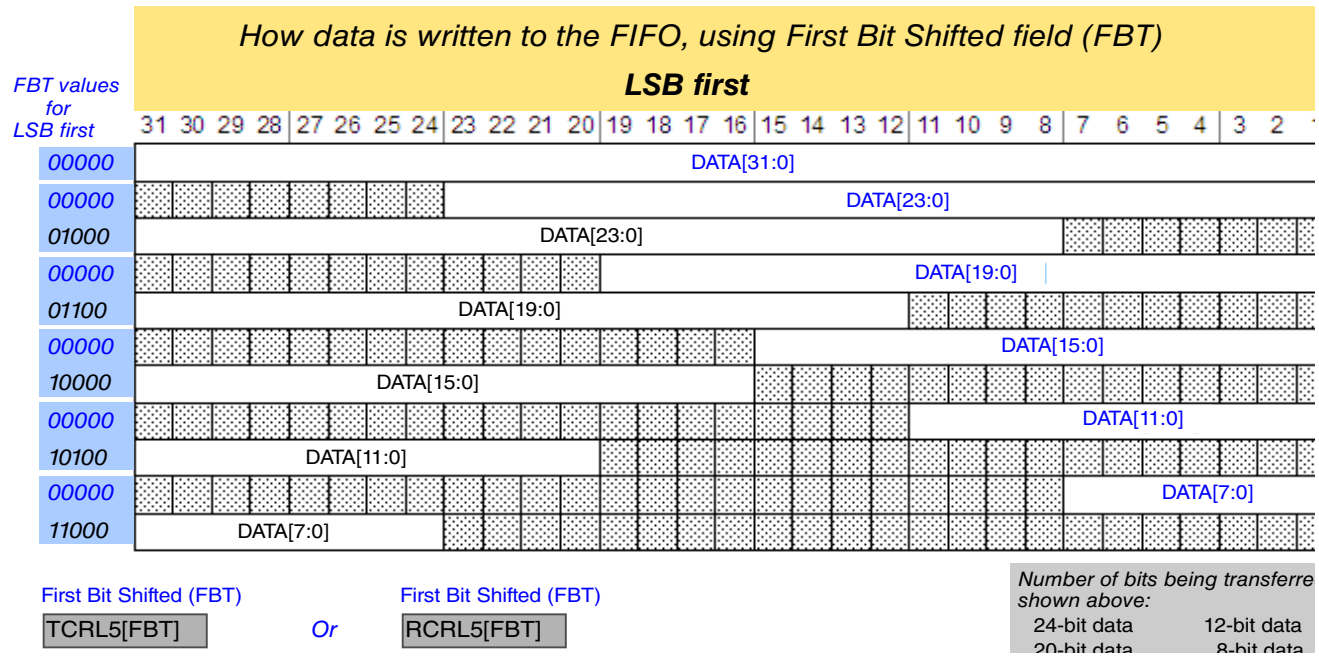


Figure 58-2. SAI first bit shifted, LSB first

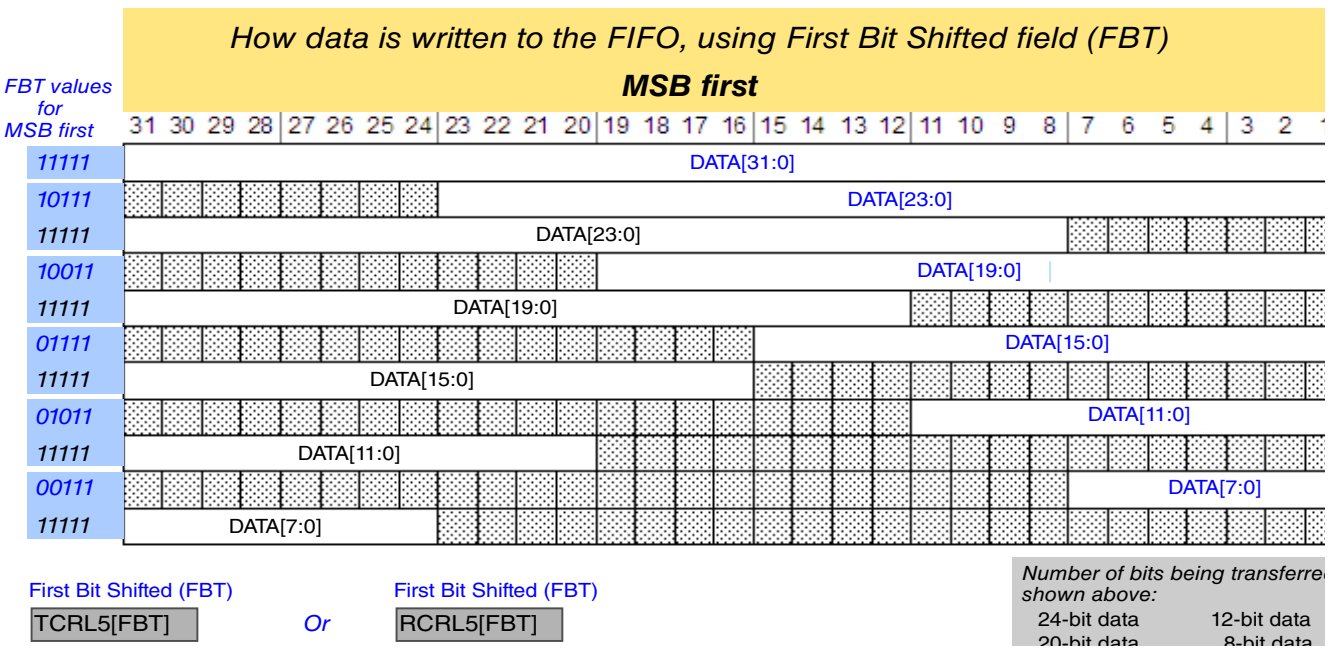


Figure 58-3. SAI first bit shifted, MSB first

58.3.4.2 FIFO pointers

When writing to a Transmit Data Register (TDR*n*), the Write FIFO Pointer (WFP) of the corresponding Transmit FIFO Register (TFR*n*) increments after each valid write. The SAI supports 8-bit, 16-bit, and 32-bit writes to the Transmit Data Register and the FIFO

pointer increments after each individual write. Note that 8-bit writes should only be used when transmitting up to 8-bit data; 16-bit writes should only be used when transmitting up to 16-bit data.

- If the Transmit FIFO is full, writes to a Transmit Data Register are ignored.
- If the Transmit FIFO is empty, to avoid a FIFO underrun, write to the Transmit Data Register at least 3 bit clocks before the start of the next unmasked word. Before enabling the transmitter, initialize the Transmit FIFO with data (because after enabling the transmitter, the transmitter starts a new frame, and if no data is in the FIFO, then the transmitter immediately give an error).

When reading a Receive Data Register (RDR*n*), the Read FIFO Pointer (RFP) of the corresponding Receive FIFO Register (RFR*n*) increments after each valid read. The SAI supports 8-bit, 16-bit and 32-bit reads from the RDR and the FIFO pointer will increment after each individual read. Note that 8-bit reads should only be used when receiving up to 8-bit data; 16-bit reads should only be used when receiving up to 16-bit data.

- If the Receive FIFO is empty, reads from a Receive Data Register are ignored.
- If the Receive FIFO is full, to avoid a FIFO overrun, read the Receive Data Register at least 3 bit clocks before the end of an unmasked word.

58.3.4.3 FIFO packing

FIFO packing supports storing multiple 8-bit or 16-bit data words in one 32-bit FIFO word for the transmitter and/or receiver. While this can be emulated by adjusting the number of bits per word and number of words per frame (for example, one 32-bit word per frame versus two 16-bit words per frame), FIFO packing does not require even multiples of words per frame and fully supports word masking. When FIFO packing is enabled, the FIFO pointers only increment when the full 32-bit FIFO word has been written (transmit) or read (receive) by software, supporting scenarios where different words within each frame are loaded/stored in different areas of memory.

Using 16-bit FIFO packing for transmit, the transmit shift register loads at the start of each frame and after every second unmasked transmit word. The first word transmitted is taken from 16-bit word at byte offset 0x0 (the first bit is selected by TCR5[FBT] and must be configured within this 16-bit word) and the second word transmitted is taken from the 16-bit word at byte offset 0x2 (the first bit is selected by TCR5[FBT][3:0]). The transmitter transmits logic zero until the start of the next word once the 16-bit word has been transmitted.

Using 16-bit FIFO packing for receive, the receive shift register is stored after every second unmasked received word, and at the end of each frame if there is an odd number of unmasked received words in each frame. The first word received is stored in the 16-bit word at byte offset 0x0 (rhe first bit is selected by RCR5[FBT] and must be configured

within this 16-bit word) and the second word received is stored in the 16-bit word at byte offset 0x2 (the first bit is selected by RCR5[FBT][3:0]). The receiver ignores received data until the start of the next word once the 16-bit word has been received.

The 8-bit FIFO packing is similar to 16-bit packing except four words are loaded or stored into each 32-bit FIFO word. The first word is loaded/stored in byte offset 0x0, second word in byte offset 0x1, third word in byte offset 0x2 and fourth word in byte offset 0x3. The TCR5[FBT] and/or RCR5[FBT] must be configured within byte offset 0x0.

58.3.4.4 FIFO Combine

FIFO Combining mode allows the separate FIFOs for multiple data channels to be used as a single FIFO for either software accesses or a single data channel or both. Note that the enabled data channels must be contiguous and data channel 0 must be enabled when using FIFO Combine mode.

Combining FIFOs for software access (writing transmit FIFO registers, reading receive FIFO registers) allows a DMA controller or software to read or write multiple FIFOs without incrementing the address that is accessed. Once enabled, the first software access to a FIFO register accesses the first enabled channel FIFO, while the second access to a FIFO register accesses the second enabled channel FIFO. This continues until software accesses the last enabled channel FIFO and the pointer resets back to the first enabled channel FIFO. To reset the pointer manually, software can reset the FIFOs or disable FIFO combining on software accesses.

Combining FIFOs for transmit data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with identical data output on each enabled data channel. The transmit shift registers for all enabled data channels load at the start of each frame and every N unmasked words (where N is the number of enabled data channels). The first word transmitted loads from the first enabled channel FIFO, while the second word transmitted loads from the second enabled channel FIFO, and so on until the end of the frame. Because the first word in each frame always loads from the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Combining FIFOs for receive data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with received data from channel 0 stored into each enabled data channel. The receive shift register for all enabled data channels are stored after every N unmasked words (where N is the number of enabled data channels). The first word received stores to the first enabled channel FIFO, while the second word received stores to the second enabled channel FIFO, and so on until the end of the frame. Because the

first word in each frame always stores to the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Note that combining FIFOs for data channels loads or stores each channel FIFO at the same time. This means that FIFO error conditions are only checked every N words (where N is the number of enabled data channels) and that the FIFO warning and request flags assert if any of the enabled data channel meets the warning flag or request flag conditions.

58.3.5 Word mask register

The SAI transmitter and receiver each contain a word mask register (TMR and RMR) that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

58.3.6 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags. Asynchronous versions of the transmitter and receiver interrupts generate to wake up the CPU from Stop mode. Asynchronous interrupts are only generated when the system clock is gated provided the corresponding BCLK continues.

58.3.6.1 FIFO request flag

The FIFO request flag sets based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit FIFO request flag sets when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and clears when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive FIFO request flag sets when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and clears when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO request flag can generate an interrupt or a DMA request.

58.3.6.2 FIFO warning flag

The FIFO warning flag sets based on the number of entries in the FIFO.

The transmit warning flag sets when the number of entries in any of the enabled transmit FIFOs is empty and clears when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag sets when the number of entries in any of the enabled receive FIFOs is full and clears when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

58.3.6.3 FIFO error flag

The Transmit FIFO Error Flag (TCSR[FEF] = 1]) sets when any of the enabled transmit FIFOs underrun. After the error flag sets, all enabled transmit channels transmit zero data before TCSR[FEF] clears.

When FIFO Continue on Error sets (TCR4[FCONT] = 1), the FIFO continues transmitting data following an underrun without software intervention. To ensure that data transmits in the correct order, the transmitter continues from the same word number in the frame that caused the FIFO to underrun, but only after new data writes to the transmit FIFO. Software should still clear the TCSR[FEF] flag, but without reinitializing the transmit FIFOs.

The Receive FIFO Error Flag sets (RCSR[FEF] = 1) when any of the enabled receive FIFOs overflow. After the flag is set, all enabled receive channels discard received data until RCSR[FEF] clears and the next next receive frame starts. Empty all enabled receive FIFOs before RCSR[FEF] clears.

When Receive FIFO Continue on Error sets (RCR4[FCONT] = 1) the FIFO continues receiving data following an overflow without software intervention. To ensure that data is received in the correct order, the receiver continues from the same word number in the

frame that caused the FIFO to overflow, but only after data has been read from the receive FIFO. Software should still clear the RCSR[FEF] flag, but without emptying the receive FIFOs.

The FIFO Error Flag can generate only an interrupt.

58.3.6.4 Sync error flag

The Sync Error Flag, TCSR[SEF] or RCSR[SEF], sets when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag sets. When the Sync Error Flag sets, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The Sync Error Flag can only generate an interrupt.

58.3.6.5 Word start flag

The Word Start Flag sets (TCSR[WSF] = 1) at the start of the second bit clock for the selected word, as configured by the Word Flag Configuration field (TCR3[WDFL]).

The Word Start Flag can generate an interrupt only.

58.3.7 SAI clocking

The SAI clocks include:

- Audio master clock
- Bit clock
- Bus clock

58.3.7.1 Audio master clock

The audio master clock generates the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated.

58.3.7.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

- If both transmitter and receiver are configured for asynchronous operation, then the transmitter and receiver will each use *their own* bit clock and frame sync.
- If the *transmitter* is configured for asynchronous mode and the receiver is configured for synchronous mode, then both transmitter and receiver will use the *transmitter* bit clock and frame sync.
- If the *receiver* is configured for asynchronous mode and the transmitter is configured for synchronous mode, then both transmitter and receiver will use the *receiver* bit clock and frame sync.

Note that the software configures synchronous or asynchronous mode, and that choice selects the bit clock/frame sync used.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

If the SAI transmitter or receiver uses an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously.

58.3.7.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

NOTE

Although there is no minimum bus clock frequency specified, the bus clock frequency must be fast enough (relative to the bit clock frequency) to ensure that the FIFOs can be serviced,

without generating either a transmitter FIFO underrun or receiver FIFO overflow condition.

58.3.8 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

58.3.8.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. This software reset does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. This software reset does not reset the configuration registers. The software reset remains asserted until cleared by software.

58.3.8.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This FIFO reset empties the FIFO contents and is used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI transmitter can also reset the FIFO of individual data channels by setting the appropriate TCR3[CFR] bit. This should only be done when the corresponding TCR3[TCE] bit is clear.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver can also reset the FIFO of individual data channels by setting the appropriate RCR3[CFR] bit. This should only be done when the corresponding RCR3[RCE] bit is clear.

58.4 External signals

Name	Function	I/O
TX_BCLK	Transmit Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
TX_SYNC	Transmit Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
TX_DATA[3:0]	Transmit Data. The transmit data is generated synchronously by the bit clock and is tristated whenever not transmitting a word.	O
RX_BCLK	Receive Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
RX_SYNC	Receive Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
RX_DATA[3:0]	Receive Data. The receive data is sampled synchronously by the bit clock.	I

58.5 Memory map and register definition

A read or write access to an address from offset 0x100 and above will result in a bus error.

58.5.1 SAI register descriptions

58.5.1.1 SAI memory map

SAI1 base address: 4040_4000h

SAI2 base address: 4040_8000h

SAI3 base address: 4040_C000h

SAI4 base address: 40C4_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	RO	0300_0000
4	Parameter (PARAM)	32	RO	Table 58-2
8	Transmit Control (TCSR)	32	RW	0000_0000
C	Transmit Configuration 1 (TCR1)	32	RW	0000_0000
10	Transmit Configuration 2 (TCR2)	32	RW	0000_0000
14	Transmit Configuration 3 (TCR3)	32	RW	0000_0000
18	Transmit Configuration 4 (TCR4)	32	RW	0000_0000
1C	Transmit Configuration 5 (TCR5)	32	RW	0000_0000
20 - 2C	Transmit Data (TDR0 - TDR3)	32	WORZ	See description
40 - 4C	Transmit FIFO (TFR0 - TFR3)	32	RO	See description
60	Transmit Mask (TMR)	32	RW	0000_0000
88	Receive Control (RCSR)	32	RW	0000_0000
8C	Receive Configuration 1 (RCR1)	32	RW	0000_0000
90	Receive Configuration 2 (RCR2)	32	RW	0000_0000
94	Receive Configuration 3 (RCR3)	32	RW	0000_0000
98	Receive Configuration 4 (RCR4)	32	RW	0000_0000
9C	Receive Configuration 5 (RCR5)	32	RW	0000_0000
A0 - AC	Receive Data (RDR0 - RDR3)	32	RO	See description
C0 - CC	Receive FIFO (RFR0 - RFR3)	32	RO	See description
E0	Receive Mask (RMR)	32	RW	0000_0000

58.5.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

58.5.1.2.1 Offset

Register	Offset
VERID	0h

58.5.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

58.5.1.2.3 Fields

Field	Description
31-24 MAJOR	Major Version Number This read only field returns the major version number for the specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000000 - Standard feature set.

58.5.1.3 Parameter (PARAM)

Contains parameter values that were implemented in the module.

58.5.1.3.1 Offset

Register	Offset
PARAM	4h

58.5.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												FRAME			
W																
Reset	See #d31617e821a1310.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO				0				DATALINE			
W																
Reset	See #d31617e821a1310.															

58.5.1.3.3 Register reset values

Register	Reset value
PARAM	SAI1: 0005_0504h SAI2–SAI4: 0005_0501h

58.5.1.3.4 Fields

Field	Description
31-20 —	Reserved
19-16 FRAME	Frame Size The maximum number of slots per frame is 2^FRAME.
15-12 —	Reserved
11-8 FIFO	FIFO Size The number of words in each FIFO is 2^FIFO.
7-4 —	Reserved
3-0 DATALINE	Number of Datalines The number of datalines implemented.

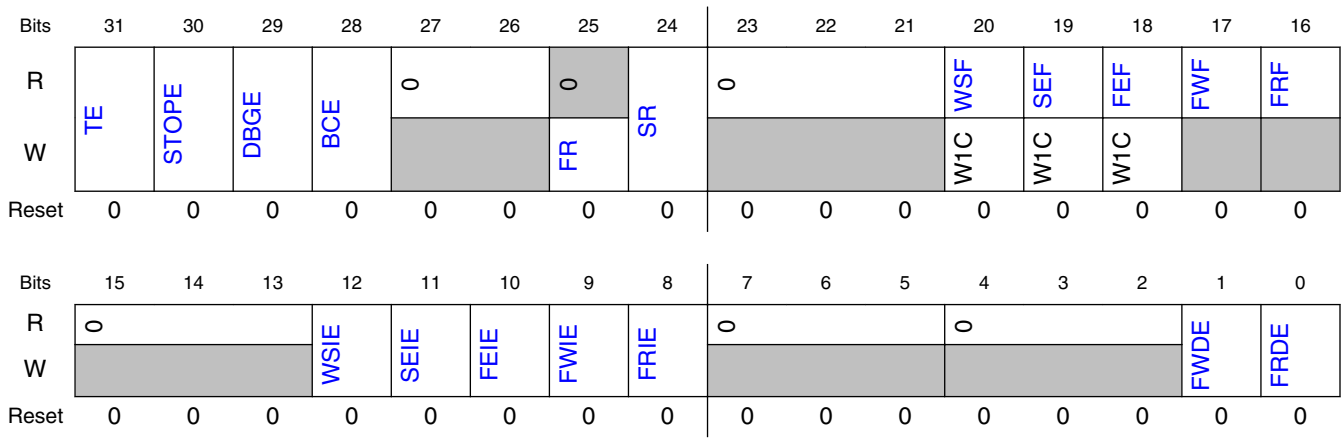
58.5.1.4 Transmit Control (TCSR)

Contains transmitter enable fields including resets, error and interrupt enable fields, and error flag fields.

58.5.1.4.1 Offset

Register	Offset
TCSR	8h

58.5.1.4.2 Diagram



58.5.1.4.3 Fields

Field	Description
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0 - Transmitter is disabled. 1 - Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures transmitter operation in Stop mode. 0 - Transmitter disabled in Stop mode. 1 - Transmitter enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables transmitter operation in Debug mode. The transmit bit clock is not affected by debug mode. 0 - Transmitter is disabled in Debug mode, after completing the current frame. 1 - Transmitter is enabled in Debug mode.

Table continues on the next page...

Field	Description
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0 - Transmit bit clock is disabled. 1 - Transmit bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the transmitter is disabled or the FIFO error flag is set. 0 - No effect. 1 - FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO read and write pointers. Software-visible registers are not affected, except for the status registers. 0 - No effect. 1 - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 - Start of word not detected. 1 - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 - Sync error not detected. 1 - Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0 - Transmit underrun not detected. 1 - Transmit underrun detected.
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty. 0 - No enabled transmit FIFO is empty. 1 - Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark.

Table continues on the next page...

Field	Description
	0 - Transmit FIFO watermark has not been reached. 1 - Transmit FIFO watermark has been reached.
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.

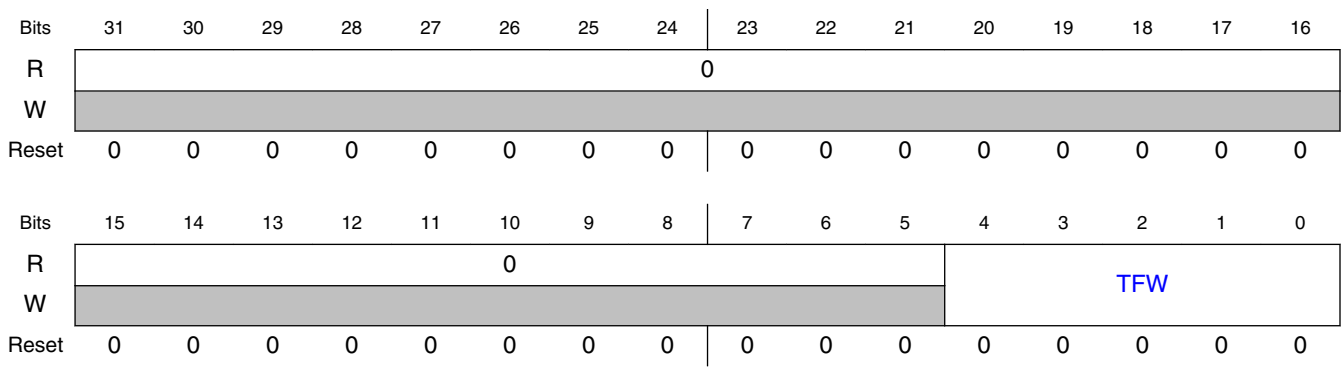
58.5.1.5 Transmit Configuration 1 (TCR1)

Configures the watermark level for all enabled transmit channels.

58.5.1.5.1 Offset

Register	Offset
TCR1	Ch

58.5.1.5.2 Diagram



58.5.1.5.3 Fields

Field	Description
31-5 —	Reserved
4-0 TFW	Transmit FIFO Watermark

58.5.1.6 Transmit Configuration 2 (TCR2)

Contains the SYNC mode and clock setting fields.

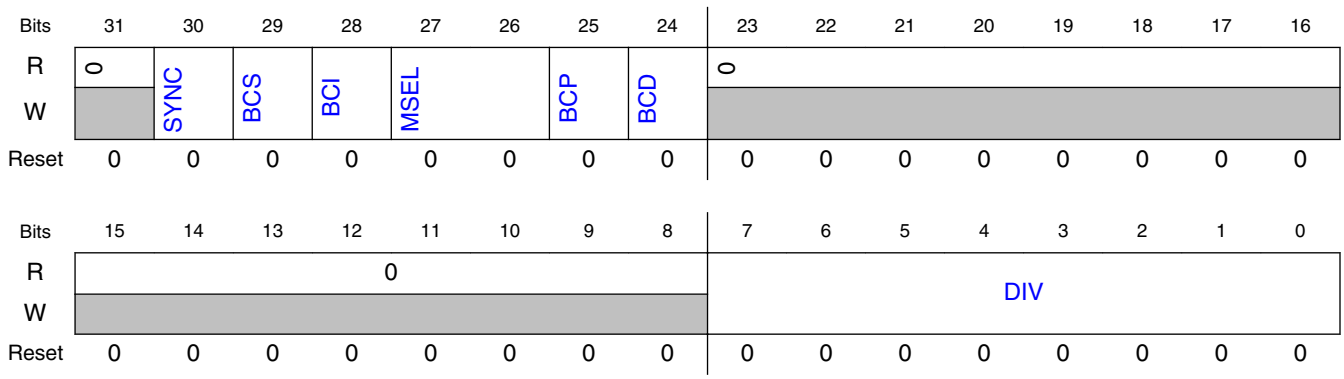
NOTE

This register must not be altered when TCSR[TE] is set.

58.5.1.6.1 Offset

Register	Offset
TCR2	10h

58.5.1.6.2 Diagram



58.5.1.6.3 Fields

Field	Description
31 —	Reserved
30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver must be configured for asynchronous operation. 0 - Asynchronous mode. 1 - Synchronous with receiver.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (RX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (TX_SYNC). When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (TX_BCLK) but use the receiver frame sync (RX_SYNC). 0 - Use the normal bit clock source. 1 - Swap the bit clock source.
28 BCI	Bit Clock Input

Table continues on the next page...

Field	Description
	When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the transmitter is delayed by the pad output delay (the transmitter is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the transmitter when this bit is set. In synchronous mode, this bit allows the transmitter to use the slave mode timing from the datasheet, while the receiver uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 - No effect. 1 - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the meaning of each option. 00 - Bus Clock selected. 01 - Master Clock (MCLK) 1 option selected. 10 - Master Clock (MCLK) 2 option selected. 11 - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 - Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 - Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Bit clock is generated externally in Slave mode. 1 - Bit clock is generated internally in Master mode.
23-8 —	Reserved
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

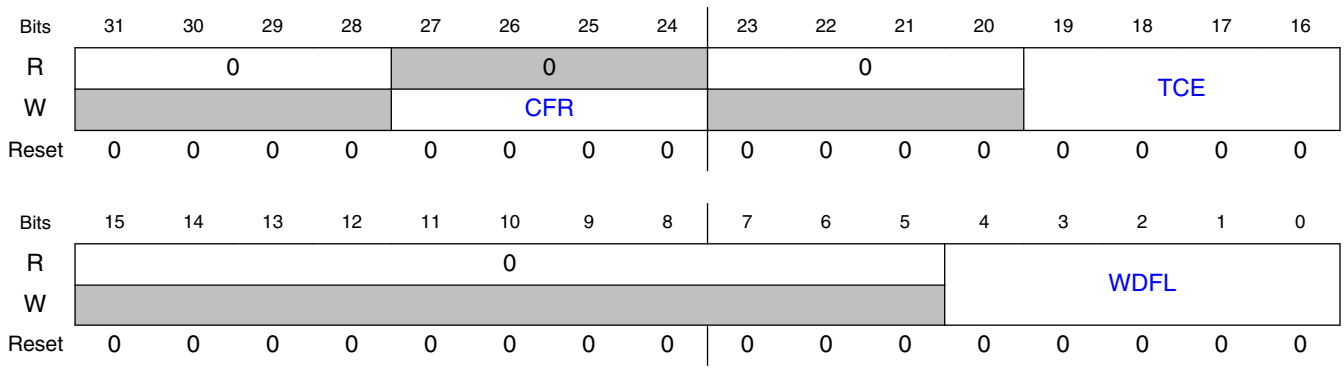
58.5.1.7 Transmit Configuration 3 (TCR3)

Contains the transmit channel settings.

58.5.1.7.1 Offset

Register	Offset
TCR3	14h

58.5.1.7.2 Diagram



58.5.1.7.3 Fields

Field	Description
31-28 —	Reserved
27-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field always returns zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set. The width of CFR field = the number of transmit channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to transmit channel 1 FIFO pointer and bit position 25 refers to transmit channel 2 FIFO pointer. Setting bit 24 resets transmit channel 1 FIFO pointer, and setting bit 25 enables transmit channel 2 FIFO pointer. Setting bit N will reset transmit channel N FIFO pointer. **NOTE:** When there is only a single channel, there is no need for individual channel FIFO reset (TCR3[CFR]). In the case of a single channel, use the global FIFO reset (TCSR[FR]). 0b - No effect. 1b - Transmit data channel N FIFO is reset. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Field supported in	Field not supported in
SAI1_TCR3	—
—	SAI2_TCR3
—	SAI3_TCR3
—	SAI4_TCR3

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>23-20
—</td><td>Reserved</td></tr>
<tr><td>19-16
TCE</td><td>Transmit Channel Enable
Enables the corresponding data channel for transmit operation. Changing TCE field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for transmit operation.
The width of TCE field = the number of transmit channels (call it N). For example, if TCE field is 2 bits wide, then bit position 16 refers to transmit channel 1 and bit position 17 refers to transmit channel 2. Setting bit 16 enables transmit channel 1, and setting bit 17 enables transmit channel 2. Setting bit N will enable transmit channel N.
0b - Transmit data channel N is disabled.
1b - Transmit data channel N is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1_TCR3</td><td>—</td></tr>
<tr><td>SAI2_TCR3[16]</td><td>SAI2_TCR3[19–17]</td></tr>
<tr><td>SAI3_TCR3[16]</td><td>SAI3_TCR3[19–17]</td></tr>
<tr><td>SAI4_TCR3[16]</td><td>SAI4_TCR3[19–17]</td></tr>
</table>
</td></tr>
<tr><td>15-5
—</td><td>Reserved</td></tr>
<tr><td>4-0
WDFL</td><td>Word Flag Configuration
Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.</td></tr>
</table>

58.5.1.8 Transmit Configuration 4 (TCR4)

Contains the transmit fields for FIFO Combine Mode, FIFO Packing Mode, and frame sync settings.

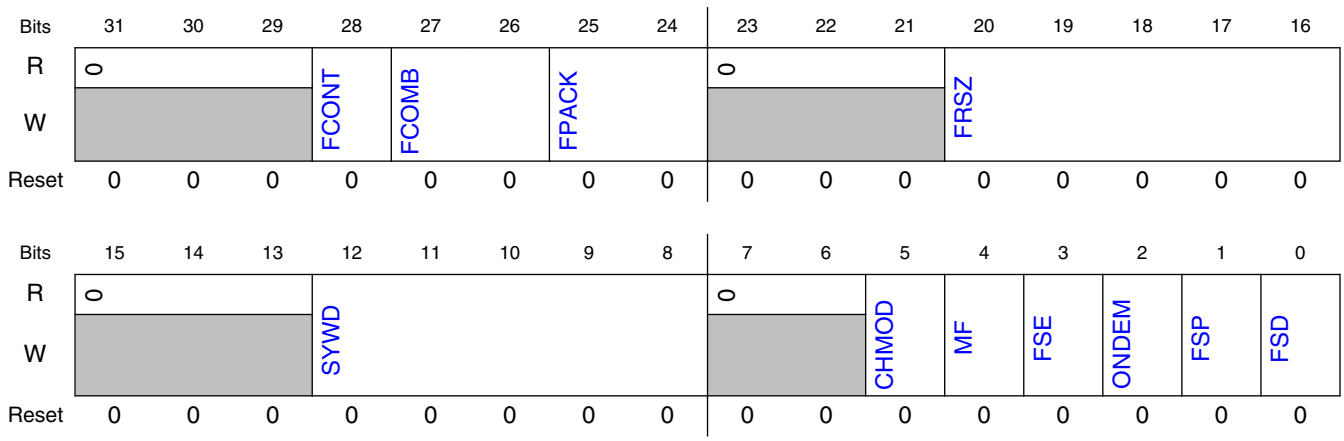
NOTE

This register must not be altered when TCSR[TE] is set.

58.5.1.8.1 Offset

Register	Offset
TCR4	18h

58.5.1.8.2 Diagram



58.5.1.8.3 Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue transmitting after a FIFO error has been detected. 0 - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared. 1 - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.
27-26 FCOMB	FIFO Combine Mode When FIFO combine mode is enabled for FIFO writes, software writing to any FIFO data register will alternate the write among the enabled data channel FIFOs. For example, if two data channels are enabled then the first write will be performed to the first enabled data channel FIFO and the second write will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO writes will reset the pointer back to the first enabled data channel. When FIFO combine mode is enabled for FIFO reads from the transmit shift registers, the transmit data channel output will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked word will be transmitted from the first enabled data channel FIFO and the second unmasked word will be transmitted from the second enabled data channel FIFO. Since the first word of the frame is always transmitted from the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels.

Table continues on the next page...

Field	Description
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Field supported in** / **Field not supported in** SAI1_TCR4 / — — / SAI2_TCR4 — / SAI3_TCR4 — / SAI4_TCR4 00 - FIFO combine mode disabled. 01 - FIFO combine mode enabled on FIFO reads (from transmit shift registers). 10 - FIFO combine mode enabled on FIFO writes (by software). 11 - FIFO combine mode enabled on FIFO reads (from transmit shift registers) and writes (by software).
25-24 FPACK	FIFO Packing Mode Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are loaded from the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO write pointer will only increment when the full 32-bit FIFO word has been written by software. 00 - FIFO packing is disabled. 01 - Reserved 10 - 8-bit FIFO packing is enabled. 11 - 16-bit FIFO packing is enabled.
23-21 —	Reserved
20-16 FRSZ	Frame size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 CHMOD	Channel Mode Configures if transmit data pins are configured for TDM mode or Output mode. 0 - TDM mode, transmit data pins are tri-stated when slots are masked or channels are disabled. 1 - Output mode, transmit data pins are never tri-stated and will output zero when slots are masked or channels are disabled.

Table continues on the next page...

Field	Description
4 MF	MSB First Configures whether the LSB or the MSB is transmitted first. 0 - LSB is transmitted first. 1 - MSB is transmitted first.
3 FSE	Frame Sync Early 0 - Frame sync asserts with the first bit of the frame. 1 - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear. 0 - Internal frame sync is generated continuously. 1 - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 - Frame sync is active high. 1 - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Frame sync is generated externally in Slave mode. 1 - Frame sync is generated internally in Master mode.

58.5.1.9 Transmit Configuration 5 (TCR5)

Contains transmit word width and bit index settings.

NOTE

This register must not be altered when TCSR[TE] is set.

58.5.1.9.1 Offset

Register	Offset
TCR5	1Ch

58.5.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

58.5.1.9.3 Fields

Field	Description
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

58.5.1.10 Transmit Data (TDR0 - TDR3)

Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

58.5.1.10.1 Offset

Register	Offset
TDR0	20h
TDR1	24h
TDR2	28h
TDR3	2Ch

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_TDR0–TDR3	—
SAI2_TDR0	SAI2_TDR1–TDR3
SAI3_TDR0	SAI3_TDR1–TDR3
SAI4_TDR0	SAI4_TDR1–TDR3

58.5.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	TDR															
Reset	See #d31617e3739a1310.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	TDR															
Reset	See #d31617e3739a1310.															

58.5.1.10.3 Register reset values

Register	Reset value
TDR0	SAI1–SAI4: 0000_0000h

Table continues on the next page...

Register	Reset value
TDR1–TDR3	0000_0000h

58.5.1.10.4 Fields

Field	Description
31-0 TDR	Transmit Data Register

58.5.1.11 Transmit FIFO (TFR0 - TFR3)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

58.5.1.11.1 Offset

Register	Offset
TFR0	40h
TFR1	44h
TFR2	48h
TFR3	4Ch

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_TFR0–TFR3	—
SAI2_TFR0	SAI2_TFR1–TFR3
SAI3_TFR0	SAI3_TFR1–TFR3
SAI4_TFR0	SAI4_TFR1–TFR3

58.5.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WCP	0									WFP					
W																
Reset	See #d31617e4032a1310.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										RFP					
W																
Reset	See #d31617e4032a1310.															

58.5.1.11.3 Register reset values

Register	Reset value
TFR0	SAI1–SAI4: 0000_0000h
TFR1–TFR3	0000_0000h

58.5.1.11.4 Fields

Field	Description
31 WCP	Write Channel Pointer When FIFO Combine mode is enabled for writes, indicates that this data channel is the next FIFO to be written. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Field supported in** / **Field not supported in** SAI1_TFR0–TFR3 / — — / SAI2_TFR0 — / SAI3_TFR0 — / SAI4_TFR0 0 - No effect. 1 - FIFO combine is enabled for FIFO writes and this FIFO will be written on the next FIFO write.
30-22 —	Reserved
21-16	Write FIFO Pointer

Table continues on the next page...

Field	Description
WFP	FIFO write pointer for transmit data channel.
15-6 —	Reserved
5-0 RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

58.5.1.12 Transmit Mask (TMR)

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

58.5.1.12.1 Offset

Register	Offset
TMR	60h

58.5.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

58.5.1.12.3 Fields

Field	Description
31-0	Transmit Word Mask

Field	Description
TWM	Configures whether the transmit word is masked (transmit data pins are tri-stated or drive zero and transmit data not read from FIFO) for the corresponding word in the frame. 00000000000000000000000000000000 - Word N is enabled. 00000000000000000000000000000001 - Word N is masked. The transmit data pins are tri-stated or drive zero when masked.

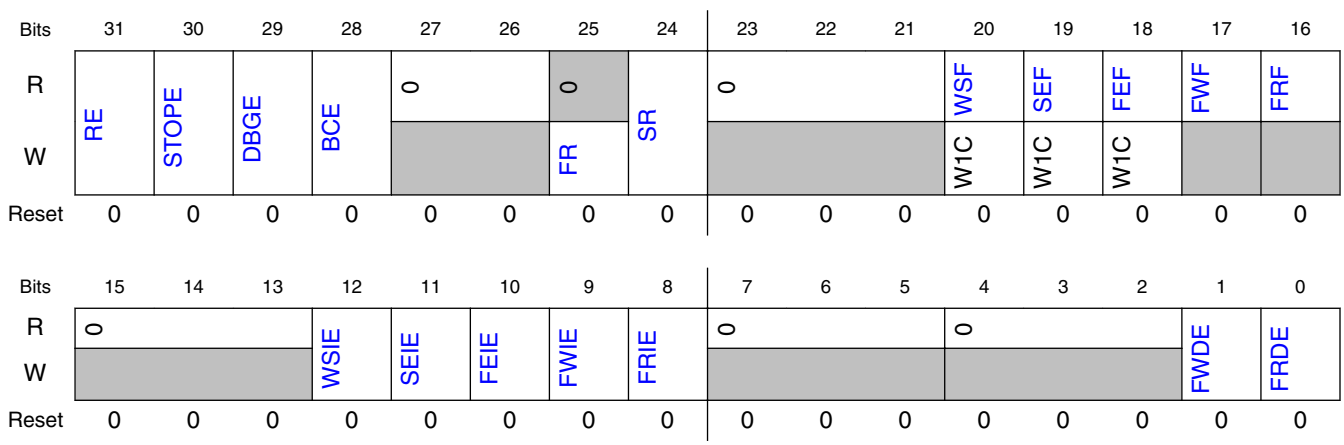
58.5.1.13 Receive Control (RCSR)

Contains receiver enable fields including resets, error and interrupt enable fields, and error flag fields.

58.5.1.13.1 Offset

Register	Offset
RCSR	88h

58.5.1.13.2 Diagram



58.5.1.13.3 Fields

Field	Description
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame.

Table continues on the next page...

Field	Description
	0 - Receiver is disabled. 1 - Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures receiver operation in Stop mode. 0 - Receiver disabled in Stop mode. 1 - Receiver enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables receiver operation in Debug mode. The receive bit clock is not affected by Debug mode. 0 - Receiver is disabled in Debug mode, after completing the current frame. 1 - Receiver is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0 - Receive bit clock is disabled. 1 - Receive bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the receiver is disabled or the FIFO error flag is set. 0 - No effect. 1 - FIFO reset.
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 - No effect. 1 - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 - Start of word not detected. 1 - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 - Sync error not detected. 1 - Frame sync error detected.

Table continues on the next page...

Field	Description
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0 - Receive overflow not detected. 1 - Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0 - No enabled receive FIFO is full. 1 - Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 - Receive FIFO watermark not reached. 1 - Receive FIFO watermark has been reached.
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 - Disables interrupt. 1 - Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 - Disables the interrupt. 1 - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved

Table continues on the next page...

Field	Description
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 - Disables the DMA request. 1 - Enables the DMA request.

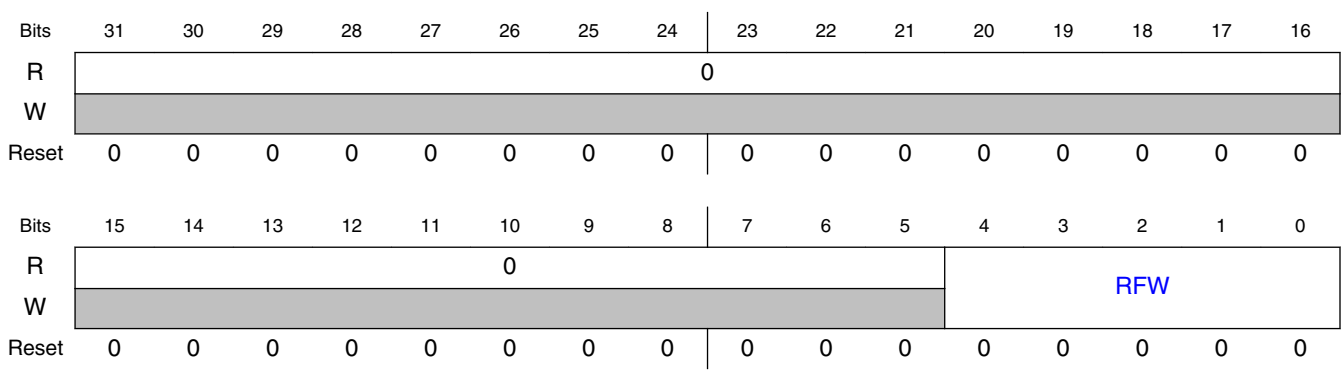
58.5.1.14 Receive Configuration 1 (RCR1)

Configures the watermark level for all enabled receiver channels.

58.5.1.14.1 Offset

Register	Offset
RCR1	8Ch

58.5.1.14.2 Diagram



58.5.1.14.3 Fields

Field	Description
31-5	Reserved

Table continues on the next page...

Field	Description
—	
4-0 RFW	Receive FIFO Watermark

58.5.1.15 Receive Configuration 2 (RCR2)

Contains the SYNC mode and clock setting fields.

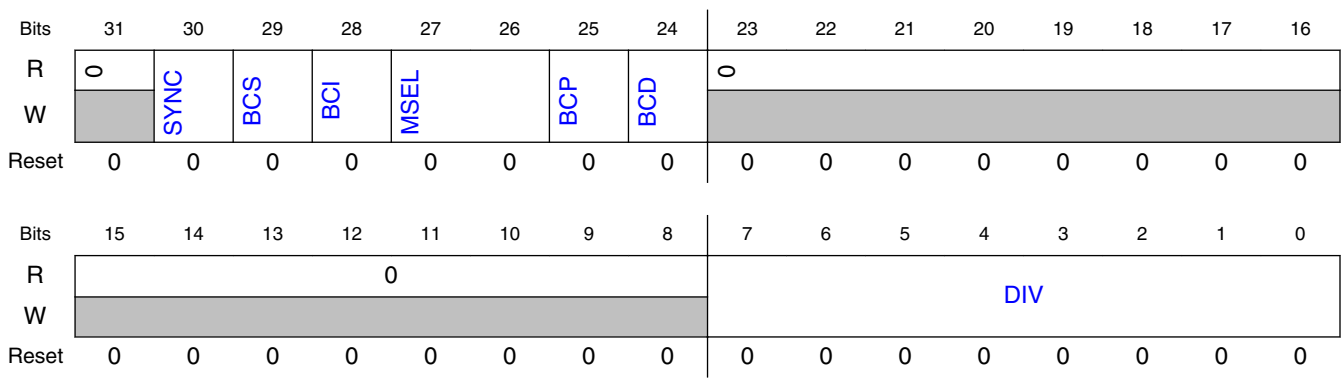
NOTE

This register must not be altered when RCSR[RE] is set.

58.5.1.15.1 Offset

Register	Offset
RCR2	90h

58.5.1.15.2 Diagram



58.5.1.15.3 Fields

Field	Description
31 —	Reserved
30	Synchronous Mode

Table continues on the next page...

Field	Description
SYNC	Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter must be configured for asynchronous operation. 0 - Asynchronous mode. 1 - Synchronous with transmitter.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the receiver. When the receiver is configured in asynchronous mode and this bit is set, the receiver is clocked by the transmitter bit clock (TX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receiver frame sync (RX_SYNC). When the receiver is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the receiver bit clock (RX_BCLK) but use the transmitter frame sync (TX_SYNC). 0 - Use the normal bit clock source. 1 - Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the receiver is delayed by the pad output delay (the receiver is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the receiver when this bit is set. In synchronous mode, this bit allows the receiver to use the slave mode timing from the datasheet, while the transmitter uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 - No effect. 1 - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the availability and chip-specific meaning of each option. 00 - Bus Clock selected. 01 - Master Clock (MCLK) 1 option selected. 10 - Master Clock (MCLK) 2 option selected. 11 - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 - Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 - Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 - Bit clock is generated externally in Slave mode. 1 - Bit clock is generated internally in Master mode.
23-8	Reserved

Table continues on the next page...

Field	Description
—	
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

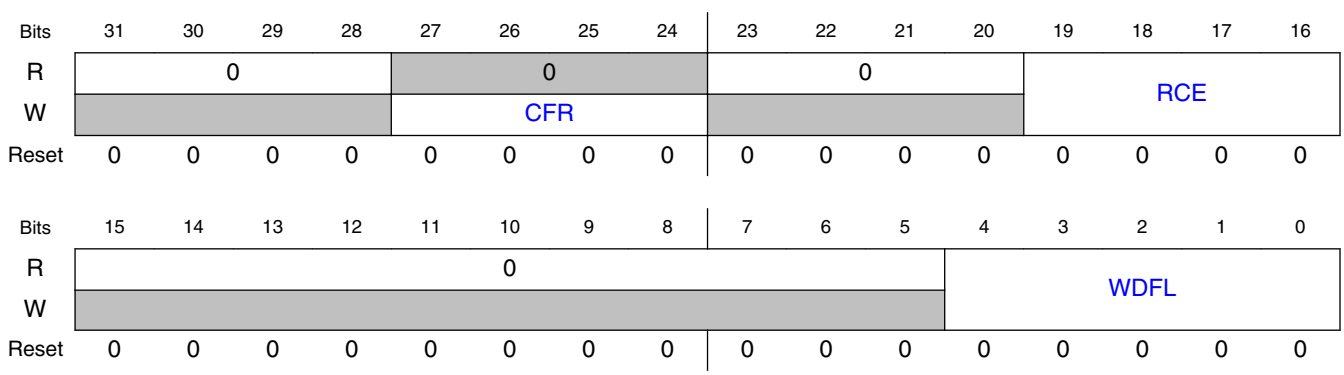
58.5.1.16 Receive Configuration 3 (RCR3)

Contains the receive channel settings.

58.5.1.16.1 Offset

Register	Offset
RCR3	94h

58.5.1.16.2 Diagram



58.5.1.16.3 Fields

Field	Description
31-28 —	Reserved
27-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field will always return zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>The width of CFR field = the number of receive channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to receive channel 1 FIFO pointer and bit position 25 refers to receive channel 2 FIFO pointer. Setting bit 24 resets receive channel 1 FIFO pointer, and setting bit 25 enables receive channel 2 FIFO pointer. Setting bit N will reset receive channel N FIFO pointer.
0b - No effect.
1b - Receive data channel N FIFO is reset.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1_RCR3</td><td>—</td></tr><tr><td>—</td><td>SAI2_RCR3</td></tr><tr><td>—</td><td>SAI3_RCR3</td></tr><tr><td>—</td><td>SAI4_RCR3</td></tr></table></td></tr>
<tr><td>23-20
—</td><td>Reserved</td></tr>
<tr><td>19-16
RCE</td><td>Receive Channel Enable
Enables the corresponding data channel for receive operation. Changing this field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for receive operation.
The width of RCE field = the number of receive channels (call it N). For example, if RCE field is 2 bits wide, then bit position 16 refers to receive channel 1 and bit position 17 refers to receive channel 2. Setting bit 16 enables receive channel 1, and setting bit 17 enables receive channel 2. Setting bit N will enable receive channel N.
NOTE: When there is only a single channel, there is no need for individual channel FIFO reset (RCR3[CFR]). In the case of a single channel, use the global FIFO reset (RCSR[FR]).
0b - Receive data channel N is disabled.
1b - Receive data channel N is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1_RCR3</td><td>—</td></tr><tr><td>SAI2_RCR3[16]</td><td>SAI2_RCR3[19–17]</td></tr><tr><td>SAI3_RCR3[16]</td><td>SAI3_RCR3[19–17]</td></tr><tr><td>SAI4_RCR3[16]</td><td>SAI4_RCR3[19–17]</td></tr></table></td></tr>
<tr><td>15-5
—</td><td>Reserved</td></tr>
<tr><td>4-0
WDFL</td><td>Word Flag Configuration
Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.</td></tr>
</table>

58.5.1.17 Receive Configuration 4 (RCR4)

Contains the receive fields for FIFO Combine Mode, FIFO Packing Mode, and frame sync settings.

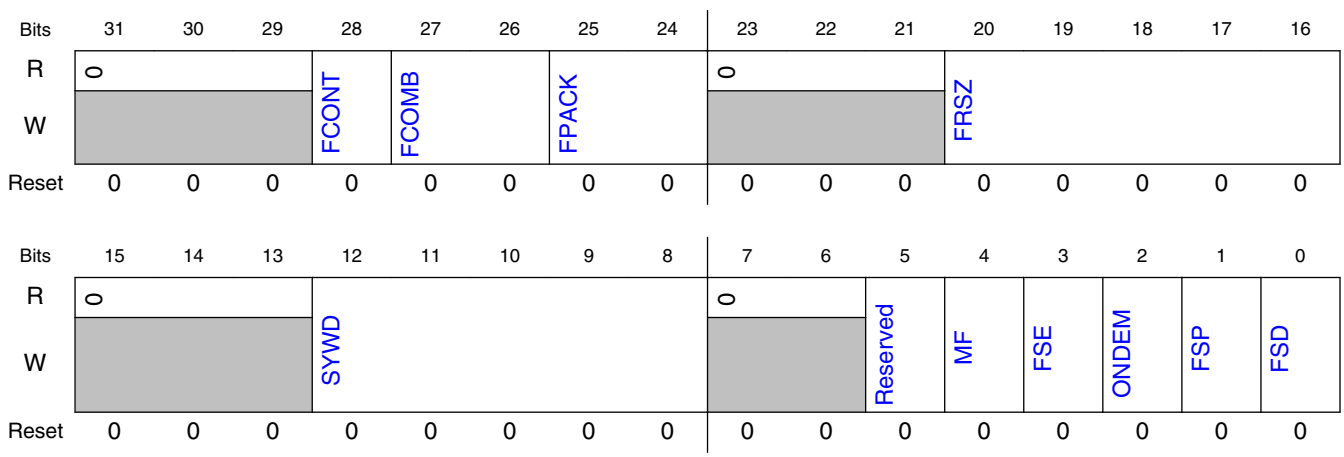
NOTE

This register must not be altered when RCSR[RE] is set.

58.5.1.17.1 Offset

Register	Offset
RCR4	98h

58.5.1.17.2 Diagram



58.5.1.17.3 Fields

Field	Description
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue receiving after a FIFO error has been detected. 0 - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>1 - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.</td></tr>
<tr><td>27-26
FCOMB</td><td>FIFO Combine Mode

When FIFO combine mode is enabled for FIFO reads, software reading any FIFO data register will alternate the read among the enabled data channel FIFOs. For example, if two data channels are enabled then the first read will be performed to the first enabled data channel FIFO and the second read will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO reads will reset the pointer back to the first enabled data channel.

When FIFO combine mode is enabled for FIFO writes from the receive shift registers, the first enabled data channel input will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked received word will be stored in the first enabled data channel FIFO and the second unmasked received word will be stored in the second enabled data channel FIFO. Since the first word of the frame is always stored in the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1_RCR4</td><td>—</td></tr>
<tr><td>—</td><td>SAI2_RCR4</td></tr>
<tr><td>—</td><td>SAI3_RCR4</td></tr>
<tr><td>—</td><td>SAI4_RCR4</td></tr>
</table>

00 - FIFO combine mode disabled.

01 - FIFO combine mode enabled on FIFO writes (from receive shift registers).

10 - FIFO combine mode enabled on FIFO reads (by software).

11 - FIFO combine mode enabled on FIFO writes (from receive shift registers) and reads (by software).</td></tr>
<tr><td>25-24
FPACK</td><td>FIFO Packing Mode

Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are stored to the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO read pointer will only increment when the full 32-bit FIFO word has been read by software.

00 - FIFO packing is disabled

01 - Reserved.

10 - 8-bit FIFO packing is enabled

11 - 16-bit FIFO packing is enabled</td></tr>
<tr><td>23-21
—</td><td>Reserved</td></tr>
<tr><td>20-16
FRSZ</td><td>Frame Size

Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.</td></tr>
<tr><td>15-13</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Field	Description
—	
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 —	Reserved Software should only write zero to this bit.
4 MF	MSB First Configures whether the LSB or the MSB is received first. 0 - LSB is received first. 1 - MSB is received first.
3 FSE	Frame Sync Early 0 - Frame sync asserts with the first bit of the frame. 1 - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear. 0 - Internal frame sync is generated continuously. 1 - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 - Frame sync is active high. 1 - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 - Frame Sync is generated externally in Slave mode. 1 - Frame Sync is generated internally in Master mode.

58.5.1.18 Receive Configuration 5 (RCR5)

This register must not be altered when RCSR[RE] is set.

58.5.1.18.1 Offset

Register	Offset
RCR5	9Ch

58.5.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

58.5.1.18.3 Fields

Field	Description
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

58.5.1.19 Receive Data (RDR0 - RDR3)

Reads from this register when the receive FIFO is not empty return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

58.5.1.19.1 Offset

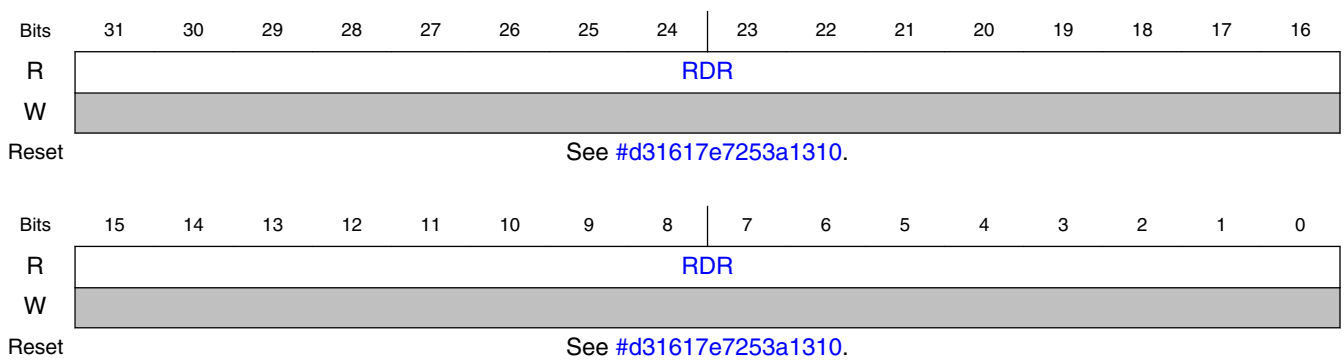
Register	Offset
RDR0	A0h
RDR1	A4h
RDR2	A8h
RDR3	ACh

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_RDR0–RDR3	—
SAI2_RDR0	SAI2_RDR1–RDR3
SAI3_RDR0	SAI3_RDR1–RDR3
SAI4_RDR0	SAI4_RDR1–RDR3

58.5.1.19.2 Diagram



58.5.1.19.3 Register reset values

Register	Reset value
RDR0	SAI1–SAI4: 0000_0000h
RDR1–RDR3	0000_0000h

58.5.1.19.4 Fields

Field	Description
31-0 RDR	Receive Data Register

58.5.1.20 Receive FIFO (RFR0 - RFR3)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

58.5.1.20.1 Offset

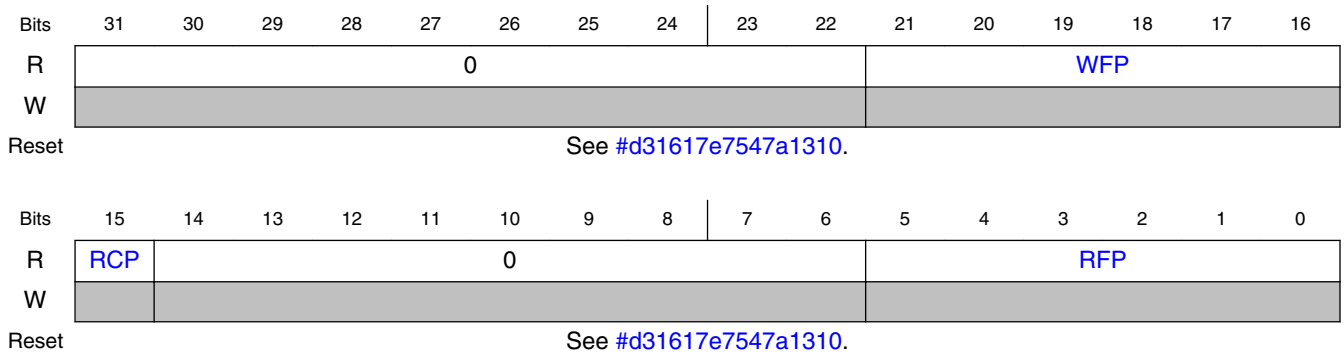
Register	Offset
RFR0	C0h
RFR1	C4h
RFR2	C8h
RFR3	CCh

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_RFR0–RFR3	—
SAI2_RFR0	SAI2_RFR1–RFR3
SAI3_RFR0	SAI3_RFR1–RFR3
SAI4_RFR0	SAI4_RFR1–RFR3

58.5.1.20.2 Diagram



58.5.1.20.3 Register reset values

Register	Reset value
RFR0	SAI1–SAI4: 0000_0000h
RFR1–RFR3	0000_0000h

58.5.1.20.4 Fields

Field	Description
31-22 —	Reserved
21-16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15 RCP	Receive Channel Pointer When FIFO Combine mode is enabled for reads, indicates that this data channel is the next FIFO to be read. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Field supported in** / **Field not supported in** SAI1_RFR0–RFR3 / — — / SAI2_RFR0 — / SAI3_RFR0 — / SAI4_RFR0 0 - No effect.

Table continues on the next page...

Field	Description
	1 - FIFO combine is enabled for FIFO reads and this FIFO will be read on the next FIFO read.
14-6 —	Reserved
5-0 RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

58.5.1.21 Receive Mask (RMR)

This register is double-buffered and updates:

1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

58.5.1.21.1 Offset

Register	Offset
RMR	E0h

58.5.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

58.5.1.21.3 Fields

Field	Description
31-0	Receive Word Mask

Field	Description
RWM	Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 00000000000000000000000000000000 - Word N is enabled. 00000000000000000000000000000001 - Word N is masked.

Chapter 59
Sony/Philips Digital Interface (SPDIF)

59.1 Chip-specific SPDIF information

Table 59-1. Reference links to related information

Topic	Related module or subsystem	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Audio Subsystem	Audio Subsystem	Audio Subsystem Overview
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

59.2 Overview

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio.

59.2.1 Block Diagram

The figure below shows a block diagram of the SPDIF transceiver data paths (receiver and transmitter) and its interface.

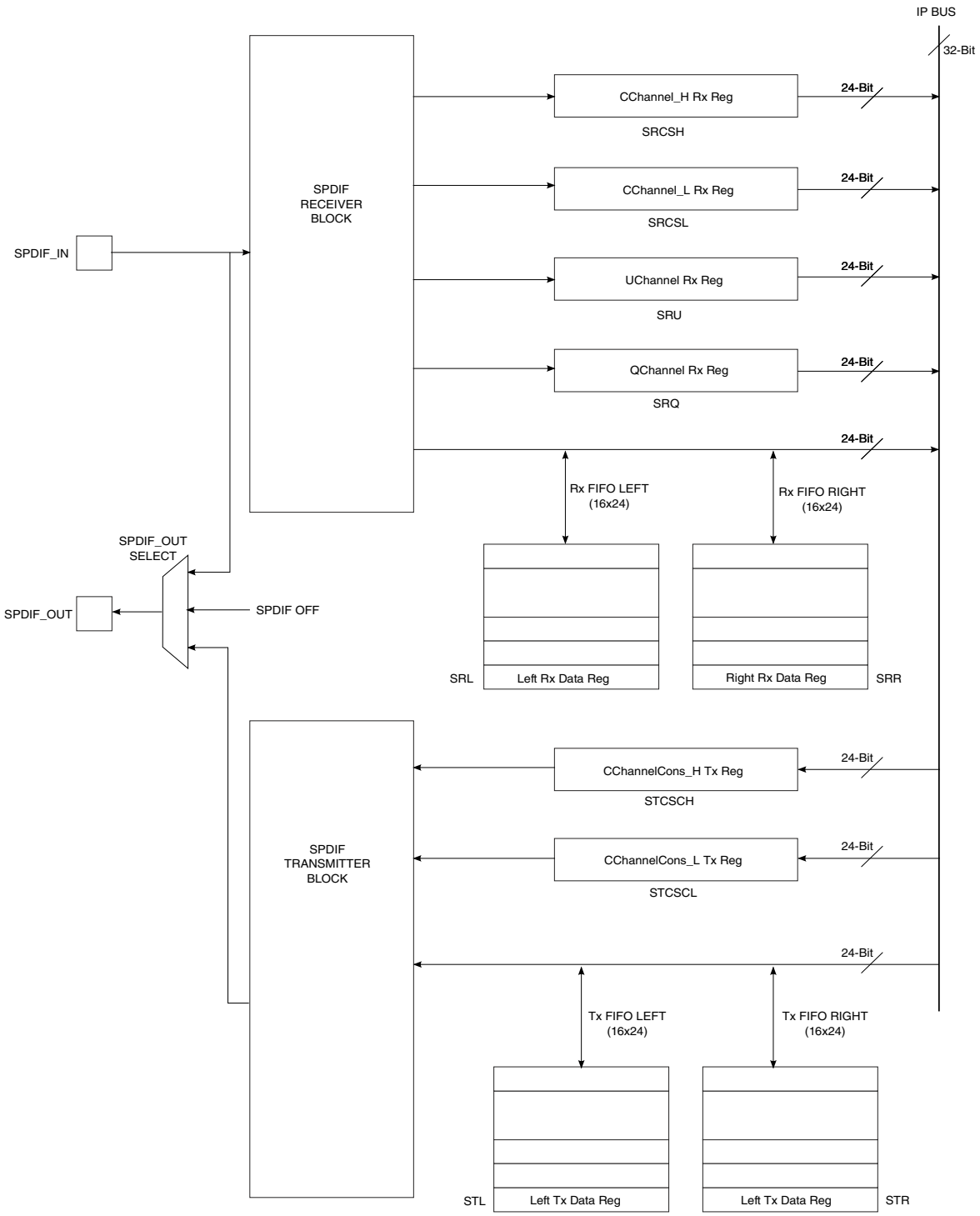


Figure 59-1. SPDIF Transceiver Data Interface Block Diagram

59.2.2 Features

- Allows the handling of both SPDIF channel status (CS) and User (U) data
- Includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency
- Provides a recovered clock to drive both internal components in the system, such as SAI ports, and external components, such as A/Ds or D/As, with clocking control provided via related registers

59.3 Functional Description

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter.

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates a SPDIF output bitstream in the biphase mark format (IEC60958), which consists of audio data, channel status and user bits.

In the SPDIF transmitter, the IEC60958 biphase bit stream is generated on both edges of the SPDIF Transmit clock. The SPDIF Transmit clock is generated by the SPDIF internal clock generate block and the sources are from outside of the SPDIF block. For the SPDIF receiver, it can recover the SPDIF Rx clock.

Figure 59-2 shows the clock structure of the SPDIF transceiver.

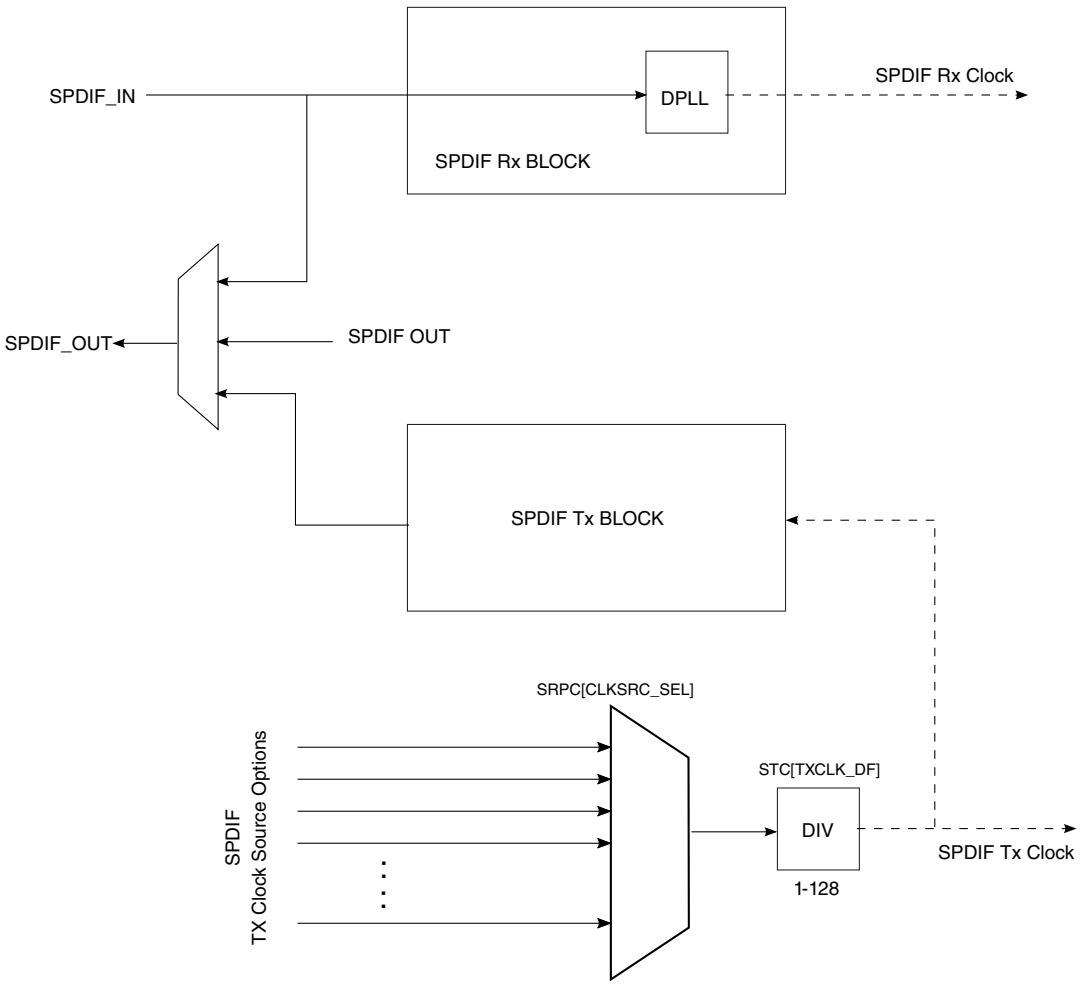


Figure 59-2. SPDIF Transceiver Clock Diagram

59.3.1 SPDIF Receiver

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in Rx left and right FIFOs.

The Tx left and right FIFOs are 16-deep and 24-bit-wide (equal to the audio data width). The Channel Status and User Bits are also extracted from each frame and placed in corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

The SPDIF receiver handles the main data audio stream and recovers the bit clock from the SPDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the SPDIF C and U channels. The SPDIF C and U channel data is interfaced directly to memory-mapped registers.

All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

59.3.1.1 Audio Data Reception

The SPDIF Receiver block extracts the audio data from the IEC60958 stream, and outputs this via Rx left and right FIFOs to the memory-mapped registers SPDIFRxLeft and SPDIFRxRight.

Data from the SPDIF receiver is buffered in receive FIFO, and can be read by the processor from the memory-mapped registers.

59.3.1.1.1 SPDIF receiver data registers - Behavior on overrun, underrun

The SPDIF Data Receive registers (SPDIFRxLeft and SPDIFRxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFOs. To prevent this from happening, hardware has been added to the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If a SPDIF Data Rx FIFO overrun occurs on e.g. the right half of the FIFO, the sample that caused the overrun is not written to the right half (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO. If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

59.3.1.1.2 SPDIF receiver data registers - Automatic resynchronization of FIFOs

An automatic FIFO resynchronization feature is available. It can be enabled and disabled separately for every FIFO. If it is enabled, the hardware will check to see if the left and right FIFOs are in sync. If that is not the case, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.

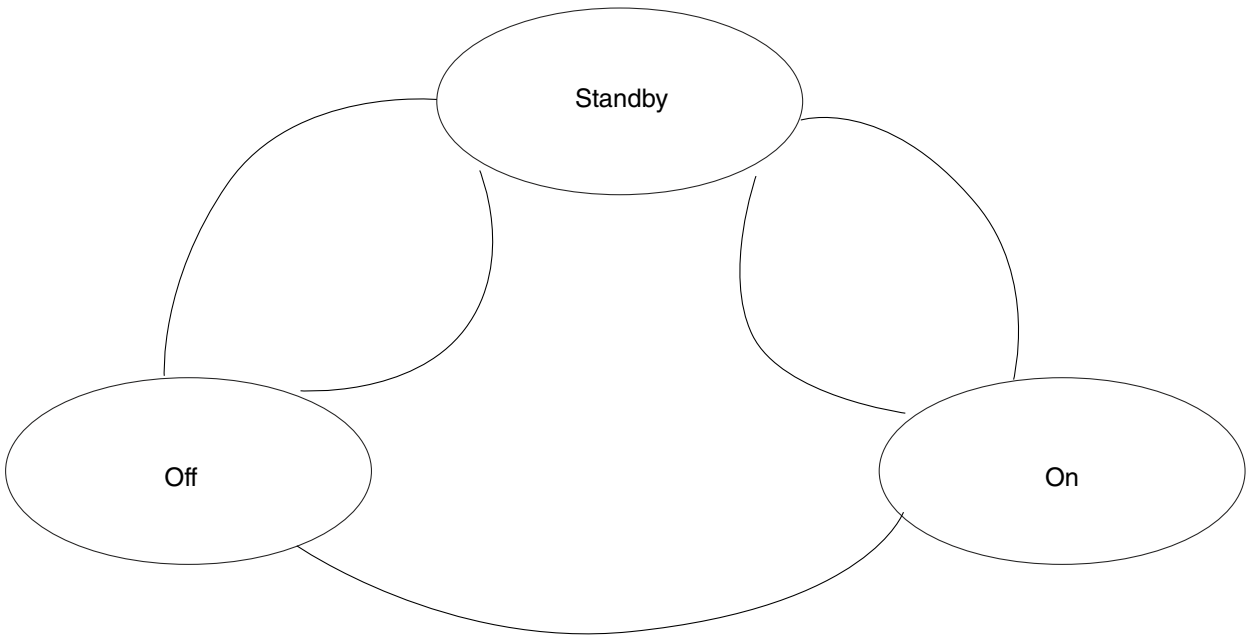


Figure 59-3. FIFO Auto-resync Controller State Machine

The operation is explained from the state diagram shown above. Every FIFO auto-resync controller has a state machine with 3 states: Off, StandBy and On. In the On state, the filling of the left FIFO is compared with the filling of right, and if they are not equal, right is made equal to left, and an interrupt is generated.

The controller will stay in Off state when the feature is disabled. When not disabled, the state machine will go to Off state on any processor read or write to the FIFO. It will go from On or Off to Standby on any left sample read from SPDIF Tx FIFOs, or on any left sample write to SPDIF Rx FIFOs. The controller will go from Standby to On on any right sample read from SPDIF Tx FIFO, or on any right sample write to SPDIF Rx FIFO. There is a control bit in the SPDIF Configuration Register (SCR) to enable/disable the feature for the SPDIF Rx FIFO and SPDIF Tx FIFO.

See for additional details.

59.3.1.2 Channel Status Reception

59.3.1.2.1 48bits Mode

A total of 48 channel status bits are received in two registers. No interpretation is performed by the SPDIF receiver block.

Channel Status Bits are ordered first bit left. CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFRxCChannel_h. CS-channel bit "23" is considered the LSB bit 0 in the register. C-channel bit 24 to 47 is seen as [23:0] bits of register SPDIFRxCChannel_l.

59.3.1.2.2 Channel Status Interrupt

When the value of a new SPDIF "CS" channel status frame is loaded in the register, an interrupt is generated. The interrupt is cleared when the processor writes the corresponding bit in the InterruptStat register.

59.3.1.3 User Bit Reception

There are two modes for U Channel reception, CD and non-CD. As is decided by USyncMode (bit 1 of CDText_Control register).

59.3.1.3.1 Behavior of U Channel receive interface on incoming CD U Channel Sub-code in SPDIF receiver

This mode is selected if UsyncMode, bit 1 in register CD Text control is set "1".

The CD sub-code stream embedded into the SPDIF U channel consists of a sequence of packets. Every packet is made up 98 "symbols". The first two symbols of every packet are "sync symbols", the other 96 symbols are "data symbols".

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

Data symbols are coming in MSB first. The MSB is the leading one.

When a "long pause" is seen between 2 subsequent "data symbols", the SPDIF receiver will assume the reception of one or more "sync symbols". Table below gives details.

Table 59-2. Sync Control Bits

Number of U Channel zero bits	Corresponding number of sync symbols
0-1	Unpredictable, not allowed
2-10	0
11-22	1
23-34	2
35-46	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols derives from the fact that the U channel transmitter in the CD channel decoder will transmit one symbol on average every 12 SPDIF channel bits. On this average rate, there is a maximum tolerance of 5%.

The SPDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be in error. The error may cause a change in data value, which is not detected by this interface, or it may cause a data symbol to be seen as a sync symbol, or a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The SPDIF U channel circuitry recognizes the 98-symbol packet structure, and sends the 96 symbol payload to the processor application. The 96 symbol payload is transmitted to the processor via 2 registers:

- The SPDIFRxUChannel register. In this register, data is presented 3 symbols at the time to the processor. Every time 3 new valid symbols, received on the SPDIF U Channel are present, the UChannelRxFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across SPDIFRxUChannel. To transfer all this data, 32 UChannelRxFull interrupts are generated.
- The QChannelReceive register. In this register, only the Q bit of the packet is accumulated.Operation is similar to UChannelReceive. Because only Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRxFull interrupts are generated. When QChannelRxFull occurs, it is coincident with UChannelRxFull. There is only one QChannelRxFull for every 8 UChannelRxFull. The convention is that most significant data is transmitted first, and is left-aligned in the registers.
- Timing regarding packet boundary is extracted by hardware. The last UChannelRxFull corresponding to a given packet should be coincident with the last QChannelRxFull. In this last U, Q channel interrupt, symbols 95-98 are received, Q channel bits 67-98. The interrupts are coincident with UQSyncFound, flagging last symbols of the current frame.
- When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read out UChannelReceive and QchannelReceive registers, discard the value, and assume the start of a new packet.
- As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence data-sync-sync-data in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous

occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.
- Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

59.3.1.3.2 Behavior of U Channel receive interface on incoming non-CD data

This mode is selected if UsyncMode, bit 1 in register CD Text control is set'0'.

In non-CD mode, the SPDIF U channel stream is recognized as a sequence of "data symbols". No packet recognition is done.

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

3 consecutive data symbols seen in the SPDIF U Channel stream are grouped together into the SPDIFRxUChannel register. First symbol is left, last symbol is right aligned. When SPDIFRxUChannel contains 3 new data symbols, UChannelRxFull is asserted.

In this mode, the operation of QchannelRx and associated interrupt QchannelRxFull is reserved, undefined. And the operation of UQFrameError and UQSyncFound is also reserved, undefined.

The U channel is extracted, and output by the SPDIF Rx on SPDIFRxUChannel-Stream.

When incoming SPDIF data parity error or bit error is detected, and if the next SPDIF word for that channel is error-free, the SPDIF word in error is replaced with the average of the previous word and next word. When incoming SPDIF data parity error or bit error is detected, and the next SPDIF word is in error, the previous SPDIF word is repeated.

59.3.1.4 Validity Flag Reception

An interrupt is associated with the Validity flag. (interrupt 16 - SPDIFValNoGood). This interrupt is set every time a frame is seen on the SPDIF interface with the validity bit set to "invalid".

59.3.1.5 SPDIF Receiver Interrupt Exception Definition

Several SPDIF exceptions can trigger an interrupt. They are:

- Control Status channel change. Set when SPDIFRxCChannel_1 register is updated. The register is updated for every new C-Channel received. The exception is reset on write to InterruptClear register.
- SPDIF Illegal Symbol. Set on reception of illegal symbol during SPDIF receive. Reset by writing register InterruptClear.[1]
- SPDIF bit error. Set on reception of bit error. (Parity bit does not match). Reset on write to InterruptClear register.
- Receive data FIFO full. Set when SPDIF receive data FIFO is full.
- Receive data FIFO underrun/overrun. Set when there is a underrun/overrun on the SPDIF receive data FIFO.
- Receive data FIFO resynchronization. Set when a resynchronization event occurs on the SPDIF receive data FIFO.
- Receive U Channel buffer full. Set when next 24 bits of U channel code are available.
- Receive Q Channel buffer overrun. Set when Q channel buffer overrun.
- Receive U Channel buffer overrun. Set on U channel buffer overrun.
- Receive Q Channel buffer full. Set when next 24 bits of Q channel code are available.
- Receive UQ sync found. Set when UQ channel sync found.
- Receive UQ frame error. Set when UQ frame error found.

59.3.1.6 Standards Compliance

The SPDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except clause 6.3.3 and the IEC60958-3 Ed2 for relevant topics.

Supported input frequency range is 12 Khz up to 96 Khz. (fully compliant) and 96 Khz up to 176 Khz (Can interface with compliant SPDIF transmitter within same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on SPDIF input signals are 0.25 bit peak-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode is capable of coping with single-symbol errors, and still retrieve U channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC60958 annex A.3 standard, and for this reason user channel

1. The SPDIF input is a biphase/mark modulated signal. The time between any two successive transitions of the SPDIF signal is always 1, 2 or 3 SPDIF symbol periods long. The SPDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. To give an example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error SPDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that(a) The SPDIF signal is destroyed by noise (b) The SPDIF frequency changed.

reception is not compliant with IEC60958 annex A.3. However, the interface is capable to receive U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel error than what is required by IEC60958 annex A.3.

59.3.1.7 SPDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the SPDIF Lock output pin SPDIF_LOCK will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is 128 x the sampling frequency.(For a 44.1 Khz input sampling frequency, the average pulse rate = 128 x 44.1 Khz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

59.3.1.8 Measuring Frequency of SPDIF_RxClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS_CLK.

FreqMeas[23:0] = FreqMeas_CLK / BUS_CLK * 2^{10} * GAIN.

For example, if the GAIN is selected as 8*(2**10) (PhaseConfig[5:3] = 3'b011), the actual result

FreqMeas_CLK / BUS_CLK is equal to FreqMeas[23:0] / 2^{23}.

59.3.2 SPDIF Transmitter

Audio data for the SPDIF transmitter is provided by processor via the SPDIFTxLeft and SPDIFTxRight registers.

Clocking for SPDIF transmitter is selected through a multiplexer from several clock sources (see register bitfield description STC[TxClk_Source] for transmit clock source inputs). The SPDIF transmitter clock source can be divided down as needed using Txclk_DF. The SPDIF transmitter output can be chosen from either the SPDIF transmitter block, directly from the SPDIF receiver (via the output multiplexer), or disabled.

The SPDIF transmitter generates a SPDIF output bitstream in IEC60958 biphase mark format, consisting of audio data, channel status.

59.3.2.1 Audio Data Transmission

Audio data for the SPDIF transmitter is provided by the processor via SPDIFTxLeft and SPDIFTxRight registers. They send audio data to Tx left and right FIFOs. The Tx left and right FIFOs are also 16-deep and 24-width (equal to the audio data width).

59.3.2.1.1 SPDIF transmitter data registers - Behavior on overrun, underrun

The SPDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, hardware has been added on the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If SPDIF Tx FIFO underruns on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe is generated). If the underrun occurs on the left half of the FIFO, next read strobe to the right FIFO is blocked.

59.3.2.1.2 SPDIF transmitter data registers - Automatic resynchronization of FIFOs

See Audio Data Reception and .

59.3.2.2 Channel Status Transmission

59.3.2.2.1 48bits Mode

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left.

CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFTxCChannelCons_h. CS-channel bit "23" is considered bit 0 in the register. C-channel bits 24-47 are seen as MSB-LSB bits of register SPDIFTxCChannelCons_l.

59.3.2.3 Validity Flag Transmission

The validity bit setting is performed via the ValCtrl bit of the SPDIF_SCR register.

59.3.3 Clocks

The table found here describes the clock sources for SPDIF.

Please see clock control block for clock setting, configuration and gating information.

Table 59-3. SPDIF Clocks

Clock name	Description
gclkw_t0	Global clock
ipg_clk_s	Peripheral access clock
tx_clk	Module Tx clock

59.4 External Signals

The following table describes the external signals of SPDIF:

Table 59-4. SPDIF External Signals

Signal	Description	Direction
SPDIF_EXT_CLK	External clock signal	I
SPDIF_IN	Input line	I
SPDIF_LOCK	Lock signal	O
SPDIF_OUT	Output line signal	O
SPDIF_SR_CLK	RX Clock	O
SPDIF_OUT_CLK	TX Clock	O

59.5 Application Information

The automatic FIFO resynchronization can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules:

1. When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right is 1/2 sample clock. (E.g. if sample frequency is 44 Khz, approximately 10 micro-seconds. For 88 Khz, approximately 5 micro-seconds.)
2. Write/read data to FIFOs at least 2 samples at the time. If there is a mismatch Left-Right, the resync logic may go on only 1 sample clock after last data is read/written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.
3. The first 192 frames (after DPLL lock) must be discarded.

59.5.1 SPDIF receiver details

There are three exceptions associated with the SPDIF Receiver FIFOs

- full
- under/overrun
- resync

When the "full" condition is set for processor data input registers, the processor should read data from the FIFO, before overrun occurs. When "full" is set, and the FIFO contains e.g. 6 samples, it is acceptable for the software to read first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 6 samples from the RIGHT address, followed by 6 samples from the LEFT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. There is no order specified.

The implementation for SPDIF Rx is a double FIFO, one for left and one for right. "full" is set when both FIFOs are full. "underrun, overrun" are set when one of the FIFOs do underrun or do overrun. The resync interrupt means hardware took special action to resynchronize left and right FIFOs.

The FIFO level at which the "full" interrupt is generated, is programmable via the Full Select field in the SPDIFConfigReg register.

59.5.2 Rx FIFO on and Rx FIFO reset

Two additional control fields of the SPDIF Rx FIFO are the on/off select and FIFO reset fields.

If on/off select is set to off, all-zero will be read from the FIFO, irrespective of the data received over the SPDIF interface.

If FIFO reset is set, the FIFO is blocked at "1 sample in FIFO". In this, the full interrupt will be on if FullSelect is set to "00". If FullSelect is set to any other value, interrupt will be off. The other interrupts are always off.

59.5.3 SPDIFTxLeft, SPDIFTxRight details

With SPDIF Tx FIFOs three exceptions are associated:

- empty
- under/overrun
- resync

When the empty condition is set for processor data output registers, the processor should write data to the FIFO, before underrun occurs. When empty is set and, for instance, 6 samples need to be written, it is acceptable for the software to write first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. Left should be written before right. The implementation of all data out FIFOs is a double FIFO, one for left and one for right. Empty is set when both FIFOs are empty. Underrun, overrun are set when one of the FIFOs do underrun or do overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving underrun, overrun interrupt, synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the underrun or overrun comes from the IEC60958 side of the FIFO. If the processor reads or writes more data from, for example, left than from right, synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

59.6 SPDIF Memory Map/Register Definition

59.6.1 SPDIF register descriptions

59.6.1.1 SPDIF memory map

ipv_spdif base address: 4040_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	SPDIF Configuration Register (SCR)	32	RW	0000_0400
4	CDText Control Register (SRCD)	32	RW	0000_0000
8	PhaseConfig Register (SRPC)	32	RW	0000_0000
C	InterruptEn Register (SIE)	32	RW	0000_0000
10	InterruptClear Register (SIC)	32	WO	0000_0000
10	InterruptStat Register (SIS)	32	RO	0000_0002
14	SPDIFRxLeft Register (SRL)	32	RO	0000_0000
18	SPDIFRxRight Register (SRR)	32	RO	0000_0000
1C	SPDIFRxCChannel_h Register (SRCSH)	32	RO	0000_0000
20	SPDIFRxCChannel_l Register (SRCSL)	32	RO	0000_0000
24	UchannelRx Register (SRU)	32	RO	0000_0000
28	QchannelRx Register (SRQ)	32	RO	0000_0000
2C	SPDIFTxLeft Register (STL)	32	WO	0000_0000
30	SPDIFTxRight Register (STR)	32	WO	0000_0000
34	SPDIFTxCChannelCons_h Register (STCSCH)	32	RW	0000_0000
38	SPDIFTxCChannelCons_l Register (STCSCL)	32	RW	0000_0000
44	FreqMeas Register (SRFM)	32	RO	0000_0000
50	SPDIFTxClk Register (STC)	32	RW	0002_0F00

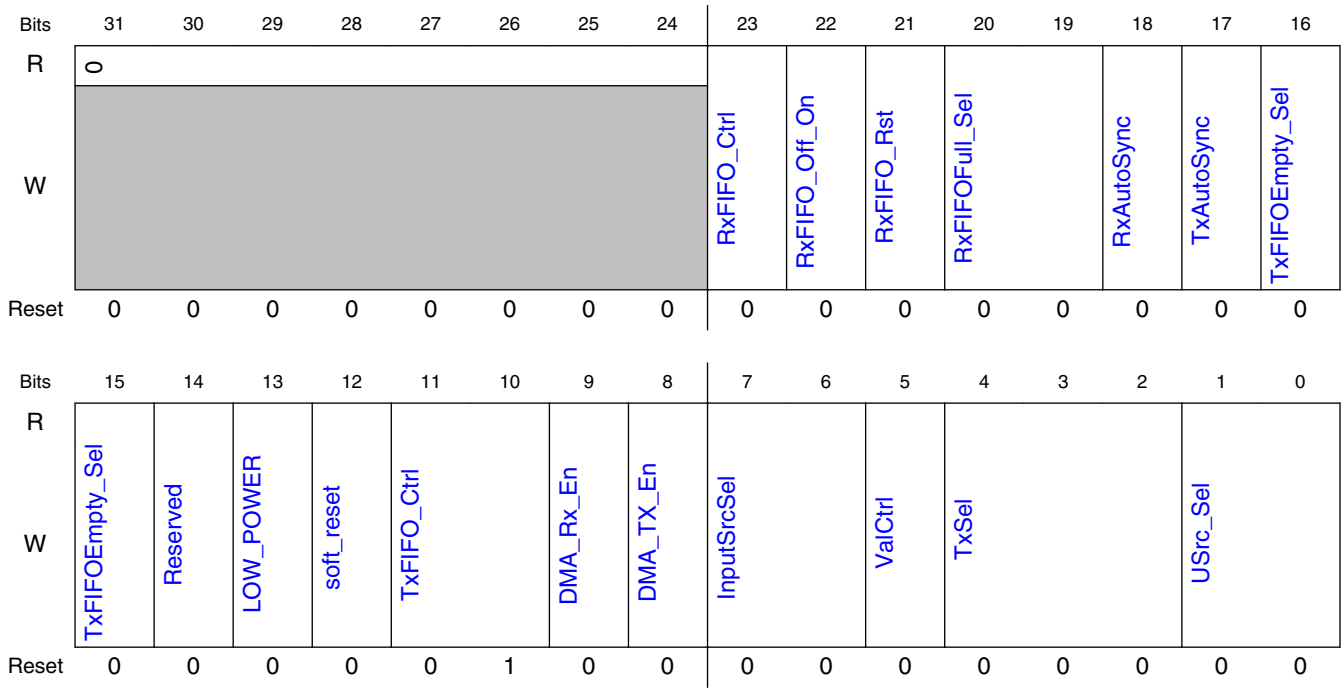
59.6.1.2 SPDIF Configuration Register (SCR)

Contains SPDIF configuration selections.

59.6.1.2.1 Offset

Register	Offset
SCR	0h

59.6.1.2.2 Diagram



59.6.1.2.3 Fields

Field	Description
31-24 —	Reserved
23 RxFIFO_Ctrl	RxFIFO_Ctrl 0 - Normal operation 1 - Always read zero from Rx data register
22 RxFIFO_Off_On	RxFIFO_Off_On 0 - SPDIF Rx FIFO is on 1 - SPDIF Rx FIFO is off. Does not accept data from interface
21 RxFIFO_Rst	RxFIFO_Rst 0 - Normal operation 1 - Reset register to 1 sample remaining
20-19 RxFIFOFull_Sel	RxFIFOFull_Sel 00 - Full interrupt if at least 1 sample in Rx left and right FIFOs 01 - Full interrupt if at least 4 sample in Rx left and right FIFOs 10 - Full interrupt if at least 8 sample in Rx left and right FIFOs 11 - Full interrupt if at least 16 sample in Rx left and right FIFO
18	RxAutoSync

Table continues on the next page...

Field	Description
RxAutoSync	0 - Rx FIFO auto sync off 1 - RxFIFO auto sync on
17 TxAutoSync	TxAutoSync 0 - Tx FIFO auto sync off 1 - Tx FIFO auto sync on
16-15 TxFIFOEmpty_Sel	TxFIFOEmpty_Sel 00 - Empty interrupt if 0 sample in Tx left and right FIFOs 01 - Empty interrupt if at most 4 sample in Tx left and right FIFOs 10 - Empty interrupt if at most 8 sample in Tx left and right FIFOs 11 - Empty interrupt if at most 12 sample in Tx left and right FIFOs
14 —	- Reserved
13 LOW_POWER	LOW_POWER When write 1 to this bit, it will cause SPDIF enter low-power mode. return 1 when SPDIF in Low-Power mode.
12 soft_reset	soft_reset When write 1 to this bit, it will cause SPDIF software reset. The software reset will last 8 cycles. When in the reset process, return 1 when read. else return 0 when read.
11-10 TxFIFO_Ctrl	TxFIFO_Ctrl 00 - Send out digital zero on SPDIF Tx 01 - Tx Normal operation 10 - Reset to 1 sample remaining 11 - Reserved
9 DMA_Rx_En	DMA_Rx_En DMA Receive Request Enable (RX FIFO full)
8 DMA_TX_En	DMA_TX_En DMA Transmit Request Enable (Tx FIFO empty)
7-6 InputSrcSel	InputSrcSel Input Source Selector 00 - SPDIF_IN 01-11 - None
5 ValCtrl	ValCtrl 0 - Outgoing Validity always set 1 - Outgoing Validity always clear
4-2	TxSel

Table continues on the next page...

Field	Description
TxSel	000 - Off and output 0 001 - Feed-through SPDIFIN 101 - Tx Normal operation
1-0 USrc_Sel	USrc_Sel U channel source select 00 - No embedded U channel 01 - U channel from SPDIF receive block (CD mode) 10 - Reserved 11 - U channel from on chip transmitter

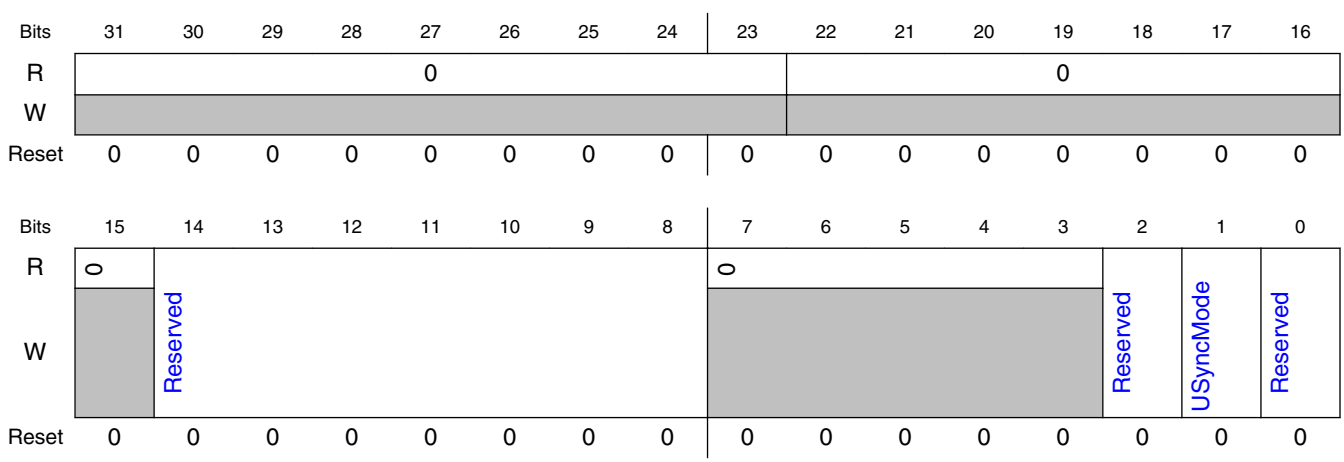
59.6.1.3 CDText Control Register (SRCD)

Contains CD text controls

59.6.1.3.1 Offset

Register	Offset
SRCD	4h

59.6.1.3.2 Diagram



59.6.1.3.3 Fields

Field	Description
31-23 —	This is a 24-bit register the upper byte is unimplemented.
22-15 —	- Return zeros when read
14-8 —	- Reserved. Set to zero.
7-3 —	- Return zeros when read
2 —	- Reserved
1 USyncMode	USyncMode 0 - Non-CD data 1 - CD user channel subcode
0 —	- Reserved

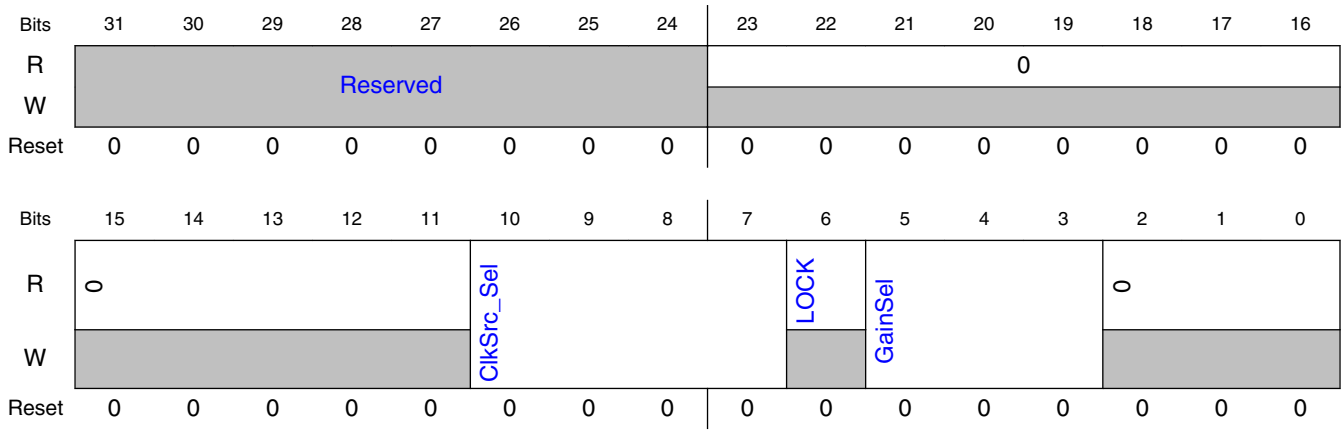
59.6.1.4 PhaseConfig Register (SRPC)

This register contains clocking and DPLL configuration and status details.

59.6.1.4.1 Offset

Register	Offset
SRPC	8h

59.6.1.4.2 Diagram



59.6.1.4.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-11 —	- Reserved, return zeros when read
10-7 ClkSrc_Sel	ClkSrc_Sel Clock source selection, all other settings not shown are reserved: 0000 - if (DPLL Locked) SPDIF_RxClk else REF_CLK_32K (XTALOSC) 0001 - if (DPLL Locked) SPDIF_RxClk else tx_clk (SPDIF0_CLK_ROOT) 0011 - if (DPLL Locked) SPDIF_RxClk else SPDIF_EXT_CLK 0101 - REF_CLK_32K (XTALOSC) 0110 - tx_clk (SPDIF0_CLK_ROOT) 1000 - SPDIF_EXT_CLK
6 LOCK	LOCK LOCK bit to show that the internal DPLL is locked, read only
5-3 GainSel	GainSel Gain selection: 000 - 24*(2**10) 001 - 16*(2**10) 010 - 12*(2**10) 011 - 8*(2**10) 100 - 6*(2**10) 101 - 4*(2**10) 110 - 3*(2**10)

Table continues on the next page...

Field	Description
2-0 —	- Reserved, return zeros when read

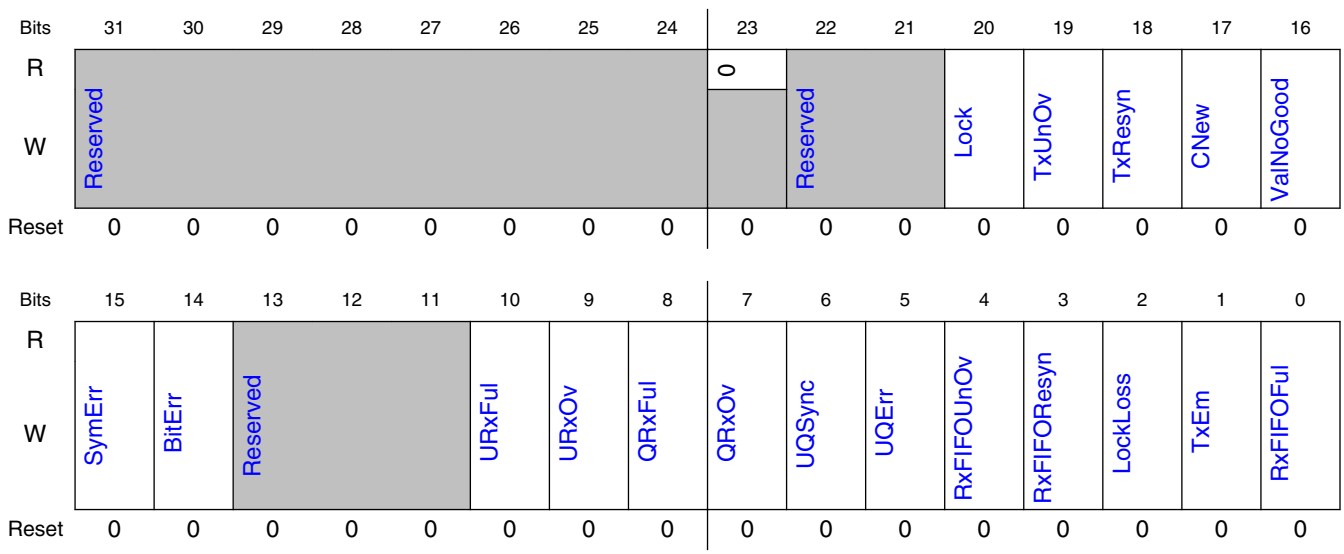
59.6.1.5 InterruptEn Register (SIE)

The InterruptEn register (SPDIF_SIE) provides control over the enabling of interrupts.

59.6.1.5.1 Offset

Register	Offset
SIE	Ch

59.6.1.5.2 Diagram



59.6.1.5.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23	-

Table continues on the next page...

Field	Description
—	Reserved
22-21	-
—	Reserved. Set to zero.
20	Lock
Lock	SPDIF receiver's DPLL is locked
19	TxUnOv
TxUnOv	SPDIF Tx FIFO under/overrun
18	TxResyn
TxResyn	SPDIF Tx FIFO resync
17	CNew
CNew	SPDIF receive change in value of control channel
16	ValNoGood
ValNoGood	SPDIF validity flag no good
15	SymErr
SymErr	SPDIF receiver found illegal symbol
14	BitErr
BitErr	SPDIF receiver found parity bit error
13-11	-
—	Reserved. Set to zero.
10	URxFul
URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9	URxOv
URxOv	U Channel receive register overrun
8	QRxFul
QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7	QRxOv
QRxOv	Q Channel receive register overrun
6	UQSync
UQSync	U/Q Channel sync found
5	UQErr
UQErr	U/Q Channel framing error
4	RxFIFOUnOv
RxFIFOUnOv	Rx FIFO underrun/overrun
3	RxFIFOResyn
RxFIFOResyn	Rx FIFO resync
2	LockLoss
LockLoss	SPDIF receiver loss of lock
1	TxEm
TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0	RxFIFOFul

Field	Description
RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

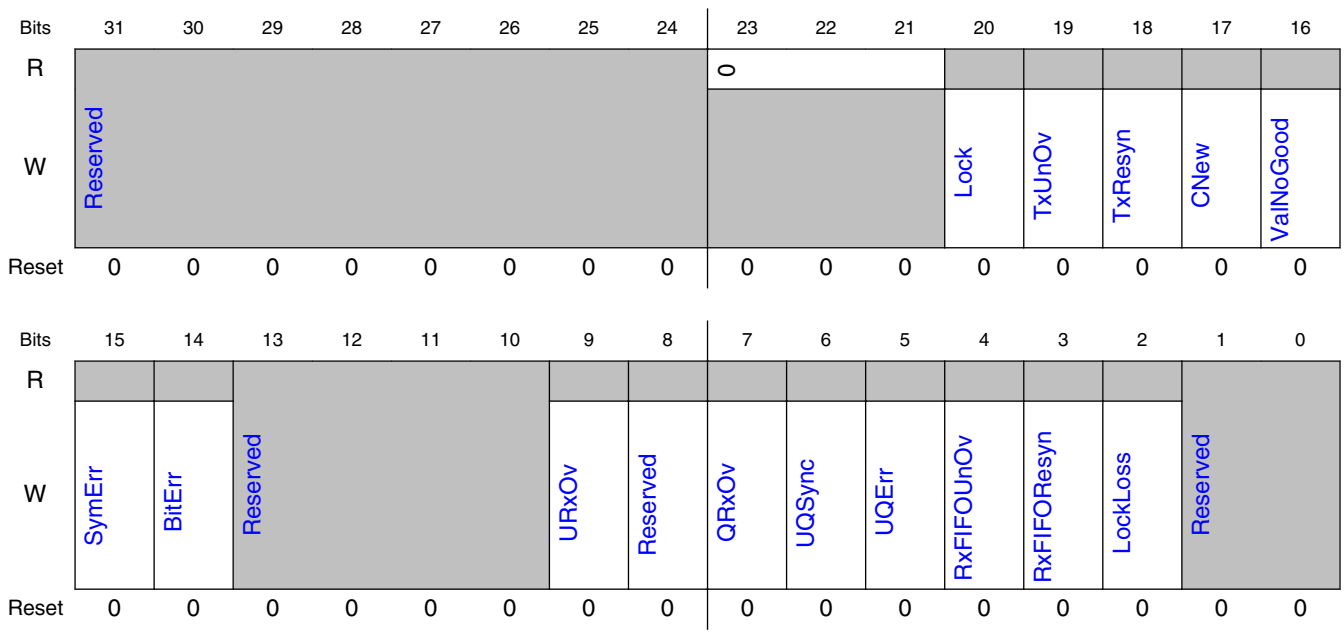
59.6.1.6 InterruptClear Register (SIC)

The InterruptClear (SPDIF_SIC) register is a write only register and is used to clear interrupts.

59.6.1.6.1 Offset

Register	Offset
SIC	10h

59.6.1.6.2 Diagram



59.6.1.6.3 Fields

Field	Description
31-24	This is a 24-bit register the upper byte is unimplemented.

Table continues on the next page...

Field	Description
—	
23-21	-
—	Reserved, for InterruptStat/Clear return zeros when read.
20	Lock
Lock	SPDIF receiver's DPLL is locked
19	TxUnOv
TxUnOv	SPDIF Tx FIFO under/overrun
18	TxResyn
TxResyn	SPDIF Tx FIFO resync
17	CNew
CNew	SPDIF receive change in value of control channel
16	ValNoGood
ValNoGood	SPDIF validity flag no good
15	SymErr
SymErr	SPDIF receiver found illegal symbol
14	BitErr
BitErr	SPDIF receiver found parity bit error
13-10	-
—	Reserved
9	URxOv
URxOv	U Channel receive register overrun
8	-
—	Reserved
7	QRxOv
QRxOv	Q Channel receive register overrun
6	UQSync
UQSync	U/Q Channel sync found
5	UQErr
UQErr	U/Q Channel framing error
4	RxFIFOUnOv
RxFIFOUnOv	Rx FIFO underrun/overrun
3	RxFIFOResyn
RxFIFOResyn	Rx FIFO resync
2	LockLoss
LockLoss	SPDIF receiver loss of lock
1-0	-
—	Reserved

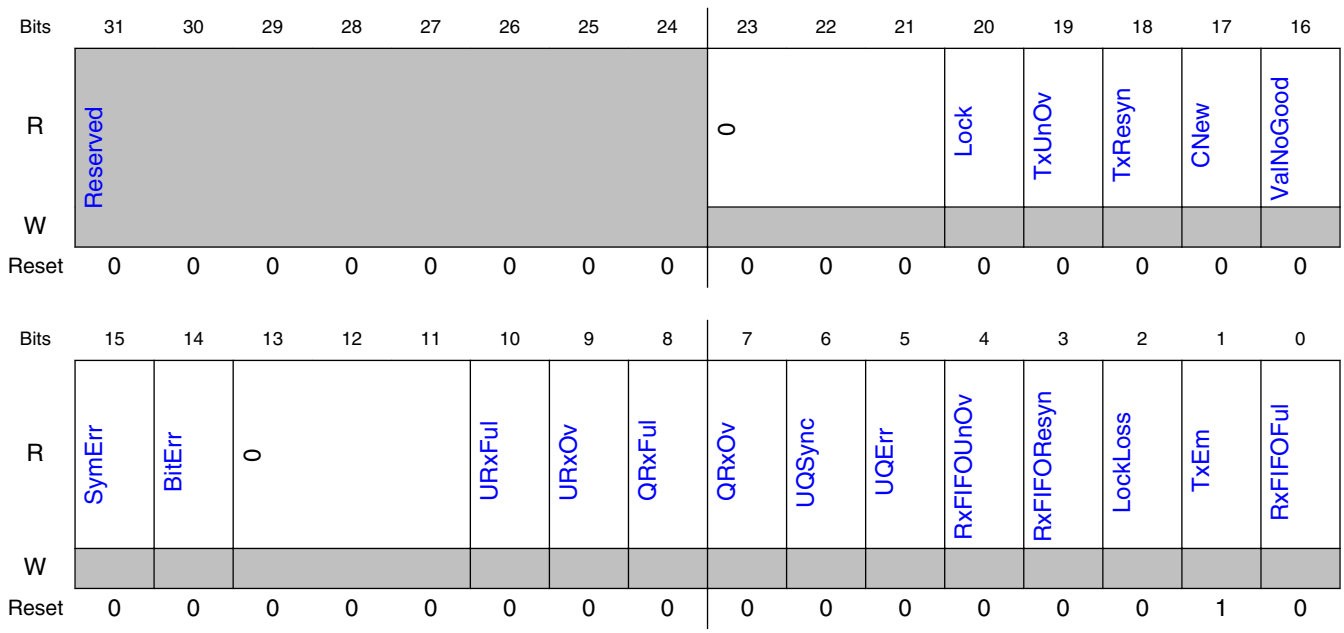
59.6.1.7 InterruptStat Register (SIS)

The InterruptStat (SPDIF_SIS) register is a read only register that provides the status on interrupt operations.

59.6.1.7.1 Offset

Register	Offset
SIS	10h

59.6.1.7.2 Diagram



59.6.1.7.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-21 —	- Reserved, for InterruptStat/Clear return zeros when read.
20 Lock	Lock SPDIF receiver's DPLL is locked

Table continues on the next page...

Field	Description
19 TxUnOv	TxUnOv SPDIF Tx FIFO under/overrun
18 TxResyn	TxResyn SPDIF Tx FIFO resync
17 CNew	CNew SPDIF receive change in value of control channel
16 ValNoGood	ValNoGood SPDIF validity flag no good
15 SymErr	SymErr SPDIF receiver found illegal symbol
14 BitErr	BitErr SPDIF receiver found parity bit error
13-11 —	- Reserved. Return zeros when read
10 URxFul	URxFul U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	URxOv U Channel receive register overrun
8 QRxFul	QRxFul Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	QRxOv Q Channel receive register overrun
6 UQSync	UQSync U/Q Channel sync found
5 UQErr	UQErr U/Q Channel framing error
4 RxFIFOUnOv	RxFIFOUnOv Rx FIFO underrun/overrun
3 RxFIFOResyn	RxFIFOResyn Rx FIFO resync
2 LockLoss	LockLoss SPDIF receiver loss of lock
1 TxEm	TxEm SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	RxFIFOFul SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

59.6.1.8 SPDIFRxLeft Register (SRL)

SPDIFRxLeft register is an audio data reception register.

59.6.1.8.1 Offset

Register	Offset
SRL	14h

59.6.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								RxDataLeft							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RxDataLeft															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

59.6.1.8.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 RxDataLeft	RxDataLeft Processor receive SPDIF data left

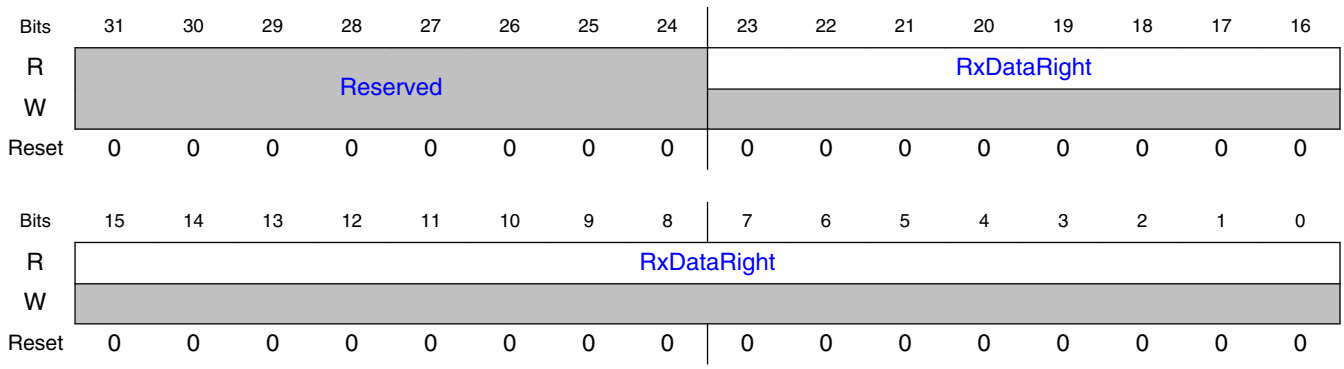
59.6.1.9 SPDIFRxRight Register (SRR)

SPDIFRxRight register is an audio data reception register.

59.6.1.9.1 Offset

Register	Offset
SRR	18h

59.6.1.9.2 Diagram



59.6.1.9.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 RxDataRight	RxDataRight Processor receive SPDIF data right

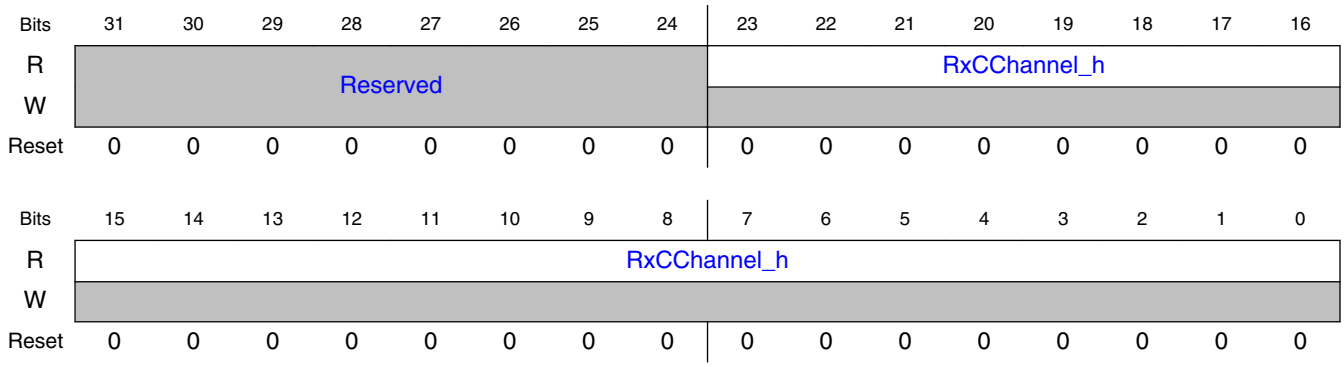
59.6.1.10 SPDIFRxCChannel_h Register (SRCSH)

SPDIFRxCChannel_h register is a channel status reception register.

59.6.1.10.1 Offset

Register	Offset
SRCSH	1Ch

59.6.1.10.2 Diagram



59.6.1.10.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 RxCChannel_h	RxCChannel_h SPDIF receive C channel register, contains first 24 bits of C channel without interpretation

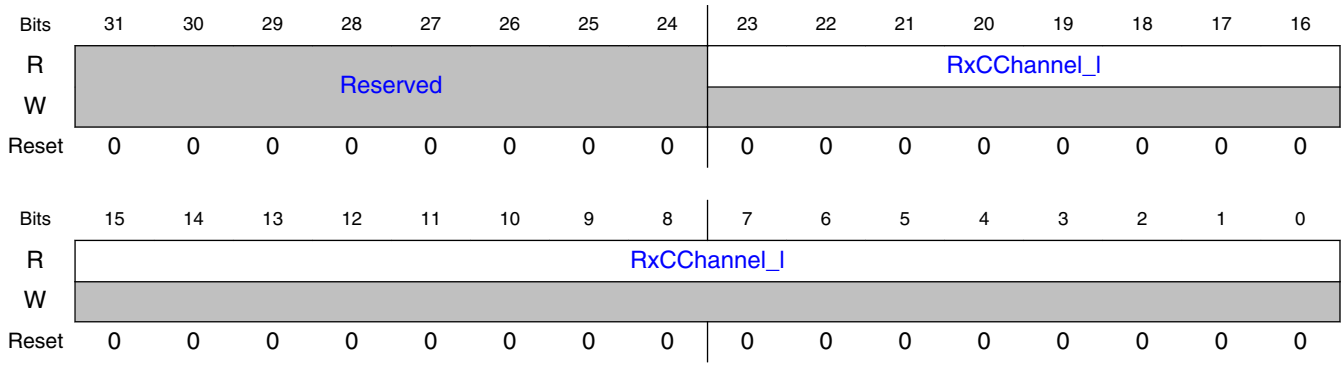
59.6.1.11 SPDIFRxCChannel_l Register (SRCSL)

SPDIFRxCChannel_l register is a channel status reception register.

59.6.1.11.1 Offset

Register	Offset
SRCSL	20h

59.6.1.11.2 Diagram



59.6.1.11.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 RxCChannel_l	RxCChannel_l SPDIF receive C channel register, contains next 24 bits of C channel without interpretation

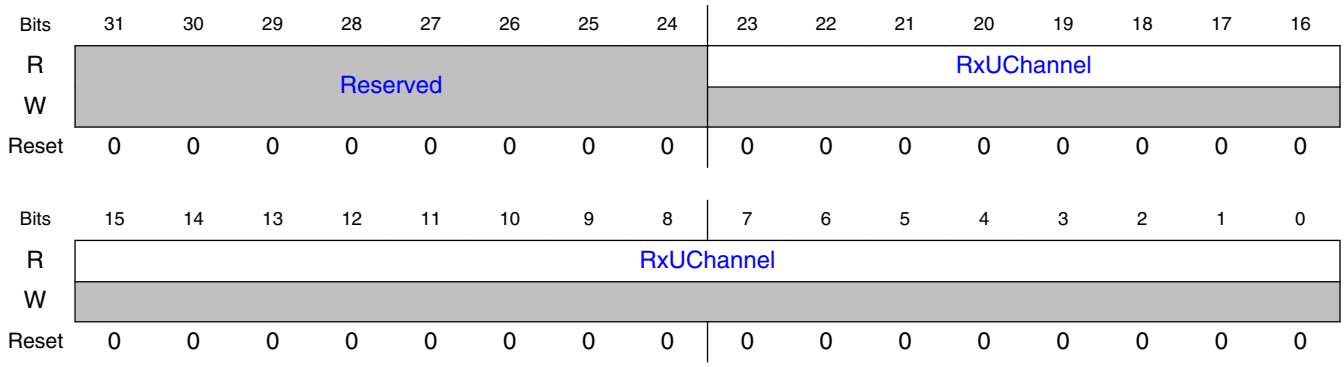
59.6.1.12 UchannelRx Register (SRU)

UChannelRx register is a user bits reception register.

59.6.1.12.1 Offset

Register	Offset
SRU	24h

59.6.1.12.2 Diagram



59.6.1.12.3 Fields

Field	Description
31-24 —	[unimplemented] This is a 24-bit register the upper byte is unimplemented.
23-0 RxUChannel	RxUChannel SPDIF receive U channel register, contains next 3 U channel bytes

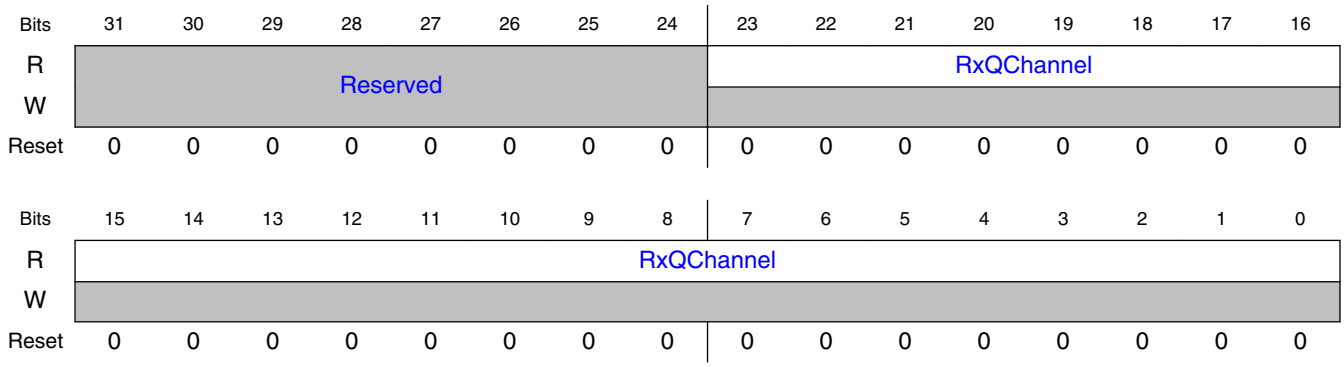
59.6.1.13 QchannelRx Register (SRQ)

QChannelRx register is a user bits reception register.

59.6.1.13.1 Offset

Register	Offset
SRQ	28h

59.6.1.13.2 Diagram



59.6.1.13.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 RxQChannel	RxQChannel SPDIF receive Q channel register, contains next 3 Q channel bytes

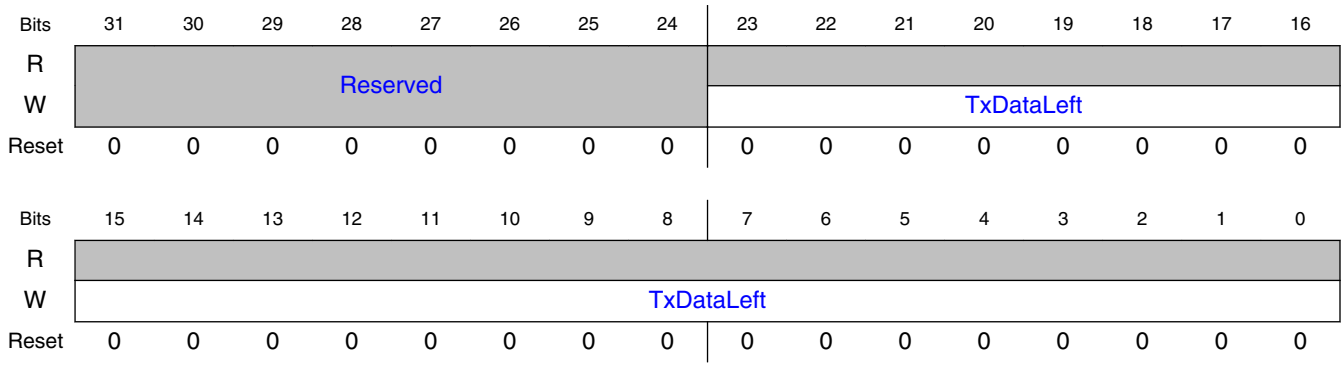
59.6.1.14 SPDIFTxLeft Register (STL)

SPDIFTxLeft register is an audio data transmission register.

59.6.1.14.1 Offset

Register	Offset
STL	2Ch

59.6.1.14.2 Diagram



59.6.1.14.3 Fields

Field	Description
31-24 —	[unimplemented] This is a 24-bit register the upper byte is unimplemented.
23-0 TxDataLeft	TxDataLeft SPDIF transmit left channel data. It is write-only, and always returns zeros when read

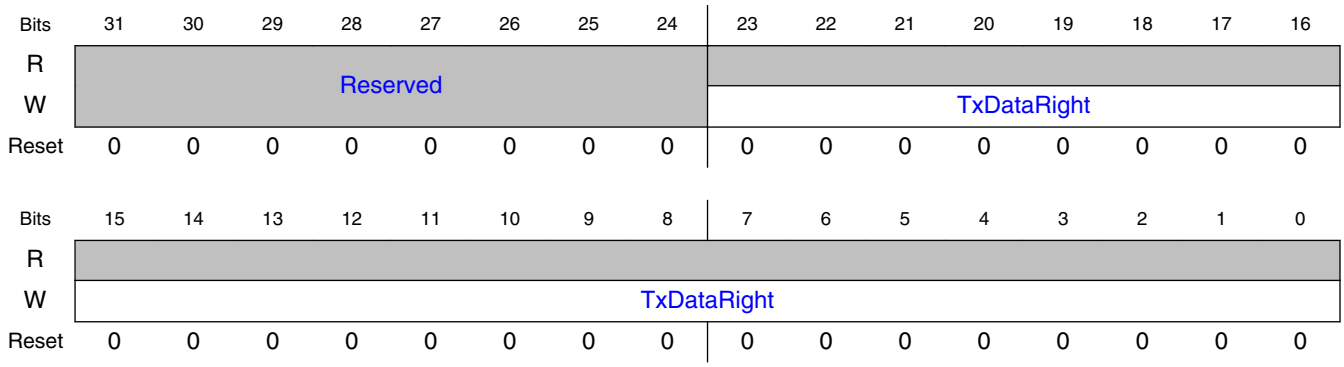
59.6.1.15 SPDIFTxRight Register (STR)

SPDIFTxRight register is an audio data transmission register.

59.6.1.15.1 Offset

Register	Offset
STR	30h

59.6.1.15.2 Diagram



59.6.1.15.3 Fields

Field	Description
31-24 —	[unimplemented] This is a 24-bit register the upper byte is unimplemented.
23-0 TxDataRight	TxDataRight SPDIF transmit right channel data. It is write-only, and always returns zeros when read

59.6.1.16 SPDIFTxCChannelCons_h Register (STCSCH)

SPDIFTxCChannelCons_h register is a channel status transmission register.

59.6.1.16.1 Offset

Register	Offset
STCSCH	34h

59.6.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TxCChannelCons_h							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TxCChannelCons_h															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

59.6.1.16.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 TxCChannelCons_h	TxCChannelCons_h SPDIF transmit Cons. C channel data, contains first 24 bits without interpretation. When read, it returns the latest data written by the processor

59.6.1.17 SPDIFTxCChannelCons_l Register (STCSCL)

SPDIFTxCChannelCons_l register is a channel status transmission register.

59.6.1.17.1 Offset

Register	Offset
STCSCL	38h

59.6.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TxCChannelCons_l							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TxCChannelCons_l															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

59.6.1.17.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 TxCChannelCons_l	TxCChannelCons_l SPDIF transmit Cons. C channel data, contains next 24 bits without interpretation. When read, it returns the latest data written by the processor

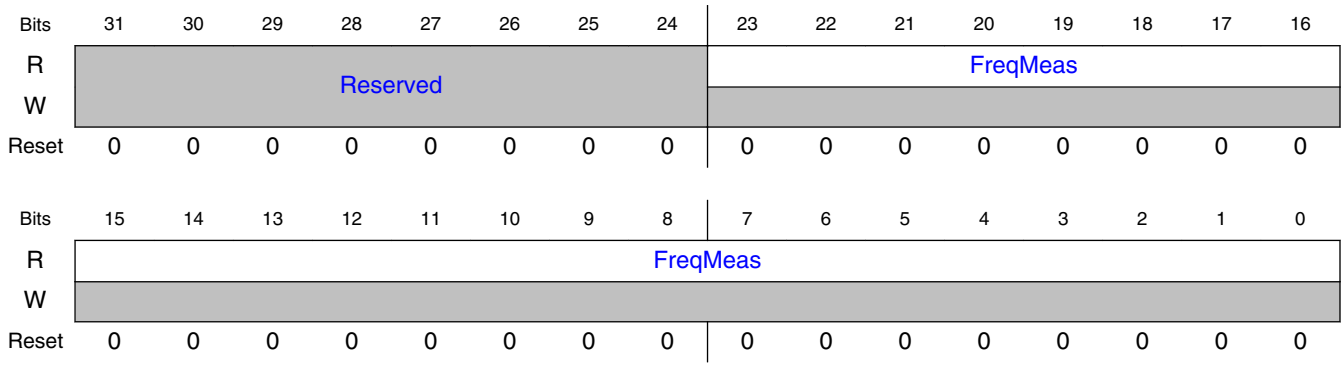
59.6.1.18 FreqMeas Register (SRFM)

Contains Frequency measurement data.

59.6.1.18.1 Offset

Register	Offset
SRFM	44h

59.6.1.18.2 Diagram



59.6.1.18.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-0 FreqMeas	FreqMeas Frequency measurement data

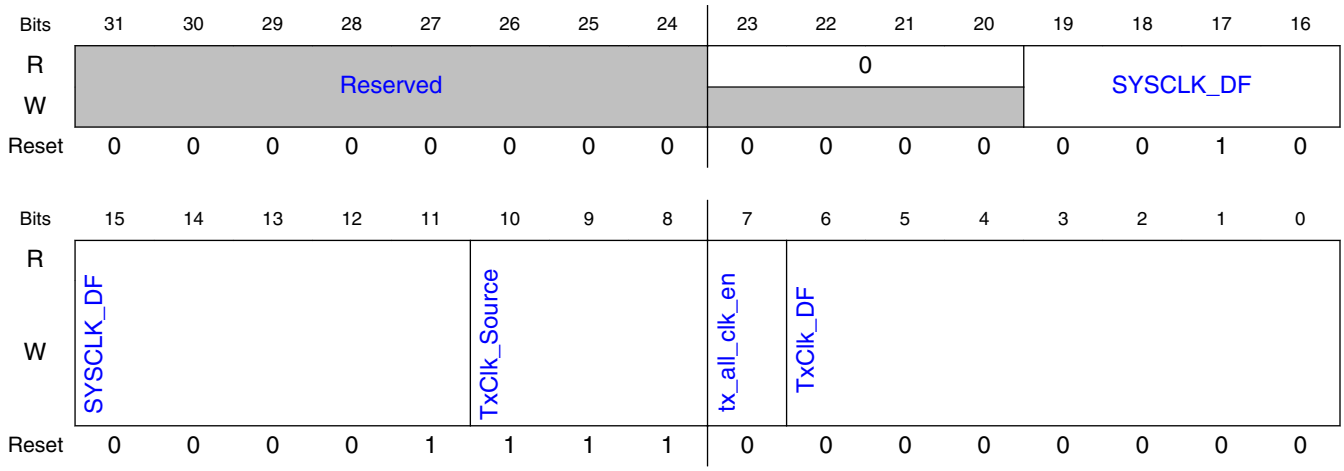
59.6.1.19 SPDIFTxClk Register (STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division.

59.6.1.19.1 Offset

Register	Offset
STC	50h

59.6.1.19.2 Diagram



59.6.1.19.3 Fields

Field	Description
31-24 —	This is a 24-bit register the upper byte is unimplemented.
23-20 —	- Reserved, return zeros when read
19-11 SYSCLK_DF	SYSCLK_DF system clock divider factor, 2~512. 000000000 - no clock signal 000000001 - divider factor is 2 111111111 - divider factor is 512
10-8 TxClk_Source	TxClk_Source 000 - REF_CLK_32K input (XTALOSC 32 kHz clock) 001 - tx_clk input (from SPDIF0_CLK_ROOT. See clock control block for more information.) 011 - SPDIF_EXT_CLK, from pads 101 - ipg_clk input (frequency divided)
7 tx_all_clk_en	tx_all_clk_en Spdif transfer clock enable.When data is going to be transfered, this bit should be set to1. 0 - disable transfer clock. 1 - enable transfer clock.
6-0 TxClk_DF	TxClk_DF Divider factor (1-128) 0000000 - divider factor is 1 0000001 - divider factor is 2 1111111 - divider factor is 128

Chapter 60
10/100 /1000 -Mbps Ethernet MAC (ENET/ENET1G)

60.1 Chip-specific ENET information

Table 60-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

This chip supports two instances of ENET IP:

ENET supports 10M/100M Ethernet, and has the following key features:

- 3.3V RMII and MII operation
- IEEE1588

ENET_1G supports 1G Ethernet, and has the following features:

- RGMII/RMII/MII operation
- IEEE1588
- AVB

60.2 Overview

The core implements a triple-speed 10/100/1000-Mbit/s ENET compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s and full-duplex gigabit Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable ENET with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

60.2.1 Block diagram

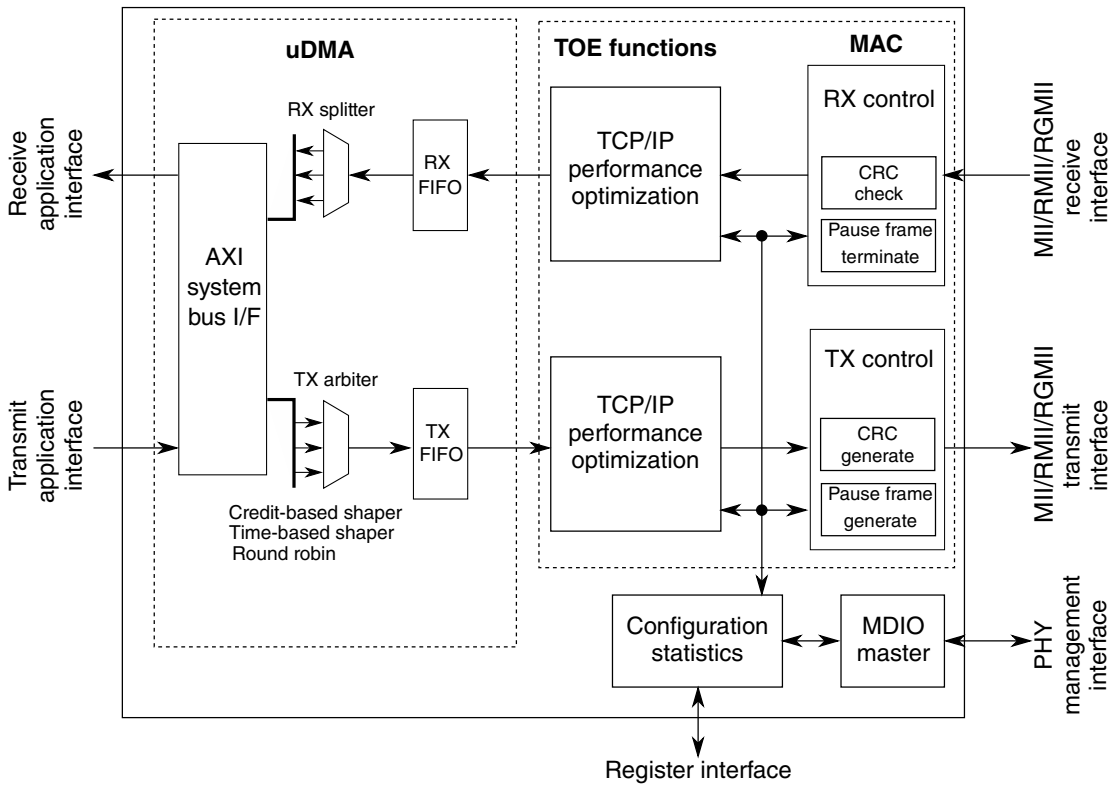


Figure 60-1. Ethernet MAC-NET core block diagram

60.2.2 Features

The MAC-NET core includes the following features.

60.2.2.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s and gigabit operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Supports gigabit full-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial Ethernet PHY devices via one of the following:
  - a 4-bit Media Independent Interface (MII) operating at 2.5/25 MHz.

    - a 4-bit non-standard MII-Lite (MII without the CRS and COL signals) operating at 2.5/25 MHz.
    - a 2-bit Reduced MII (RMII) operating at 50 MHz.
    - a (double data rate) 4-bit Reduced GMII (RGMII) operating at 125 MHz.
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
    - Supports Ethernet pause frame (802.3 Annex 31A) providing fully automated flow control without any user application overhead
    - Pause quanta used to form pause frames, dynamically programmable
    - Pause frame generation additionally controllable by user application offering flexible traffic flow control
    - Optional forwarding of received pause frames to the user application
    - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- Supports internal loopback option
- Programmable Clause 22 and Clause 45 MDIO Master interface for PHY device configuration and management
- Supports legacy FEC buffer descriptors
- Supports interrupt coalescing

- Supports credit-based shaping and round robin QoS scheme for traffic shaping bandwidth distribution. Either policy can be combined with time-based shaping.
- AVB (Audio Video Bridging, IEEE 802.1Qav) features:
  - Credit-based bandwidth distribution policy can be combined with time-based shaping
  - AVB endpoint talker and listener support
  - Support for arbitration between different priority traffic (for example, AVB class A, AVB class B, and non-AVB)

60.2.2.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and insertion on transmit. Configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked
- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive

- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

60.2.2.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

60.3 Functional description

This section provides a complete functional description of the MAC-NET core.

60.3.1 Ethernet MAC frame formats

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields

- Length or type field
- Data field
- Frame check sequence (CRC value)

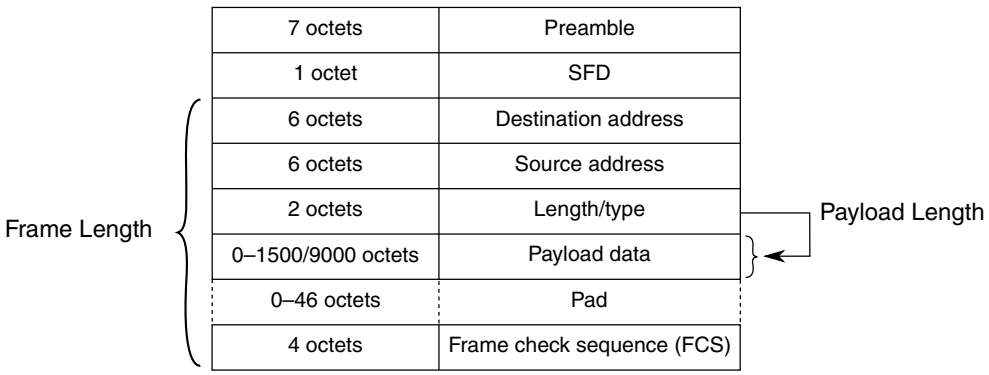


Figure 60-2. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

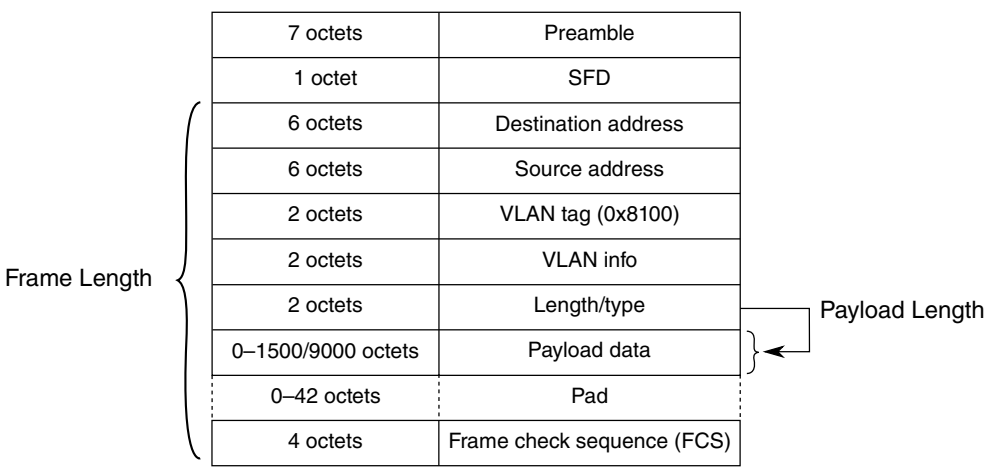


Figure 60-3. VLAN-tagged MAC frame format overview

Table 60-2. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first.

Table continues on the next page...

Table 60-2. MAC frame definition (continued)

Term	Description
	• If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

60.3.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 60-3. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

60.3.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

60.3.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

60.3.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 60-4. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.

Table continues on the next page...

Table 60-4. Ethernet type value examples (continued)

Type	Description
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

60.3.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 60-5. IPv4 header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 60-6. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.

Table continues on the next page...

Table 60-6. IPv4 header fields (continued)

Field name	Description
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

60.3.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly. The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

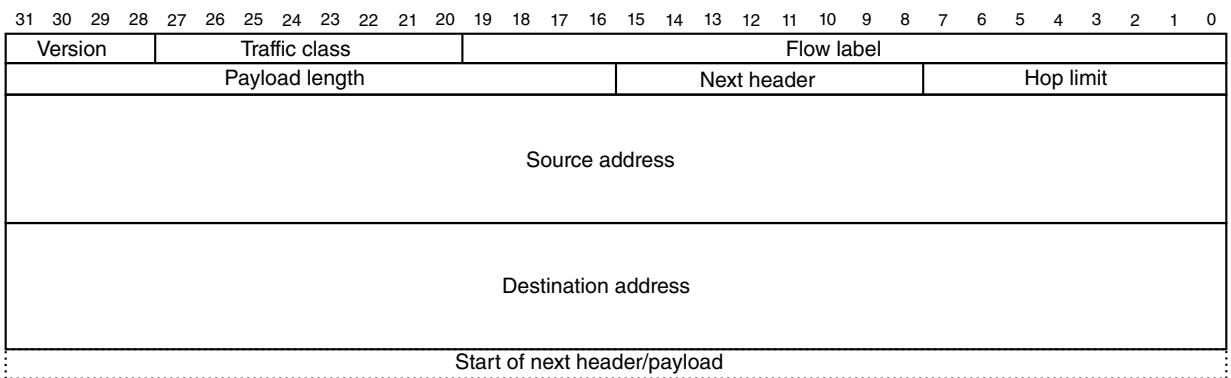


Figure 60-4. IPv6 header format

Table 60-7. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.

Table continues on the next page...

Table 60-7. IPv6 header fields (continued)

Field name	Description
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

60.3.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 60-8. ICMP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Type		Code		Checksum			
ICMP message data							

Table 60-9. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

60.3.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 60-10. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 60-11. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

60.3.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 60-12. TCP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Sequence number							
Acknowledgement number							
Offset	Reserved		Flags	Window			
Checksum				Urgent pointer			
Options							

Table 60-13. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

60.3.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

60.3.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

60.3.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

The table below shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 60-14. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 60-15. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

60.3.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 60-16. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

60.3.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

60.3.3.2.1 PTPv1 header

Table 60-17. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 60-18 :

Table 60-18. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

60.3.3.2.2 PTPv2 header

Table 60-19. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 60-20. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message

Table continues on the next page...

Table 60-20. PTPv2 message type identification (continued)

messageId	Message name	Message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 60-21. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

60.3.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists

- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

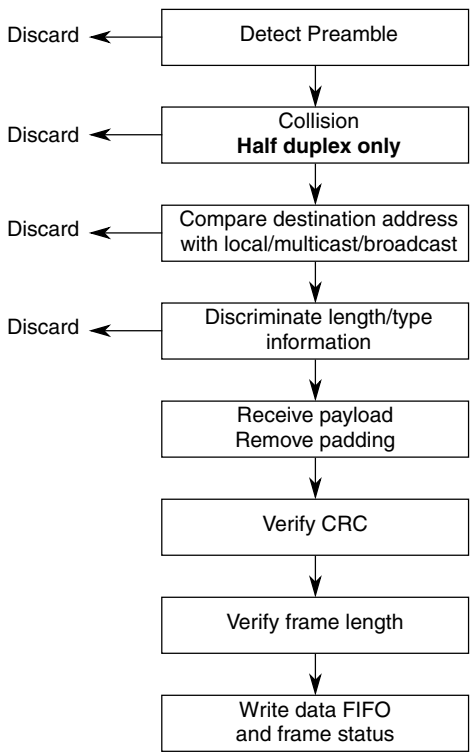


Figure 60-5. MAC receive flow

60.3.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

60.3.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

60.3.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

60.3.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

60.3.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five least significant bits of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

60.3.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 60-22. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

60.3.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set

- A unicast is received that does not match either:
  - Node address (PALR[PADDR1] and PAUR[PADDR2])
  - Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])
- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

60.3.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word RCR[NLC] and interrupt bit EIR[PLR].

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

60.3.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

60.3.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[VLAN]) indicates that the current frame is VLAN tagged.

60.3.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Valid pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

60.3.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

60.3.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the

padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

60.3.4.10 Frame classification (AVB)

To support protocols such as Audio Video Bridging (AVB, IEEE 802.1Qav), normal traffic is separated from time-sensitive traffic immediately at the application interface to allow storing them in different queues for further processing. For every frame received, a classification based on VLAN priority can be performed. When a frame is received, and it contains a VLAN tag, the priority is compared against the values set in the classification match registers (see RCMR register and the following figure.

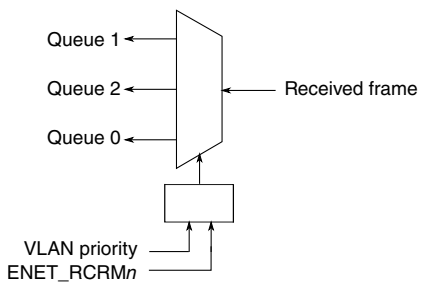


Figure 60-6. AVB frame classification

60.3.4.11 Receive flushing

RX flushing prevents frames in the RX FIFO from being blocked. Blocking can occur if the frame at the head of the RX FIFO cannot be forwarded because the ring it is associated with cannot accept it. This situation occurs when either the ring's RxBD[EMPTY] is not set or ENET_RDAR*n* is not set. When RX flushing is enabled, via QOS[RX_FLUSHn], the blocking frame will be flushed (discarded).

60.3.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission

- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

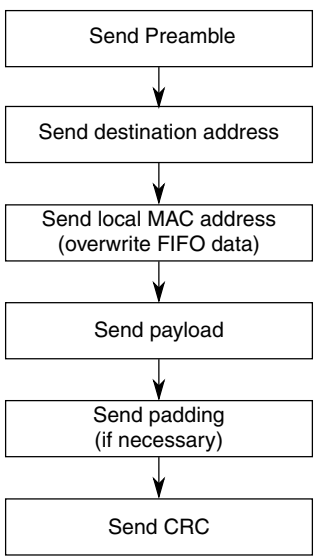


Figure 60-7. Frame transmit overview

60.3.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

60.3.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

60.3.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

60.3.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS/GMII_CRS de-assertion.

60.3.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

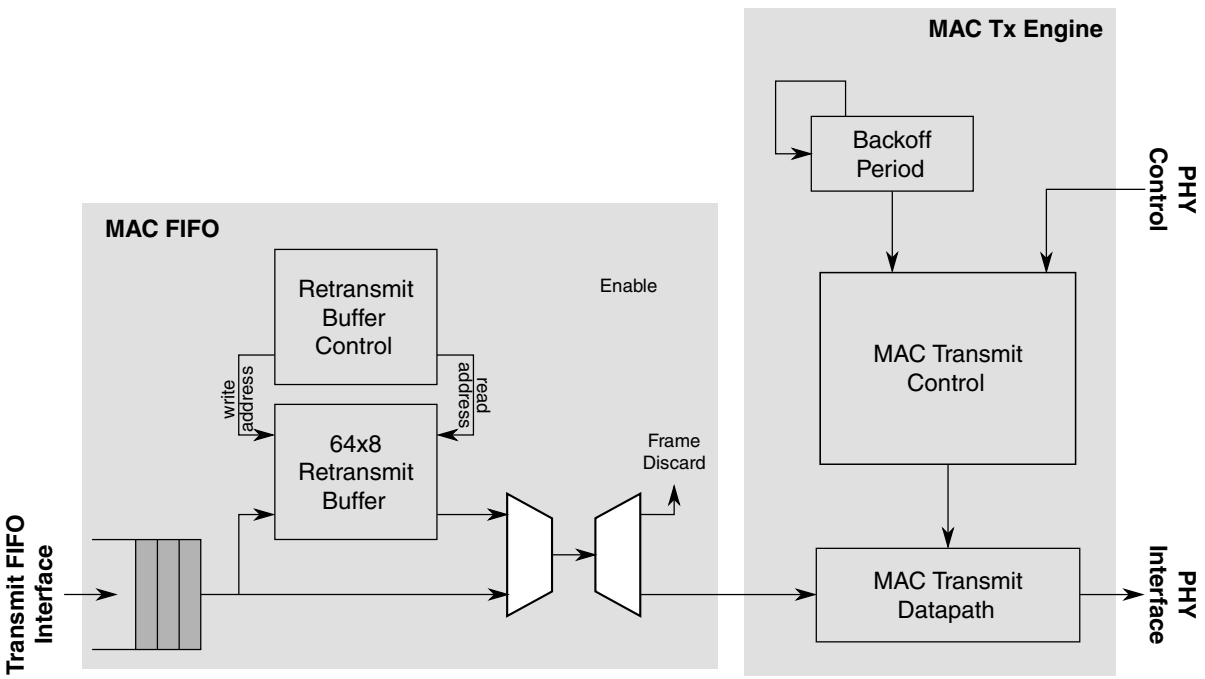


Figure 60-8. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- $k = \min(n, N)$; where n is the number of retransmissions and $N = 10$

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat," feature must disable this feature. When this feature is enabled, the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

60.3.5.6 Rate limiting / traffic shaping support

The MAC-NET supports two methods to optimize frame traffic for either time-sensitive AVB frames (Class A and Class B) or best-effort non-AVB frames:

- Round-robin scheme
- Credit-based traffic shaping

To ensure that sufficient bandwidth is allocated for the AVB frames, the credit-based shaper must be used. Either method can be combined with a time-based shaper to ensure that a frame is always transmitted in its correct time slot.

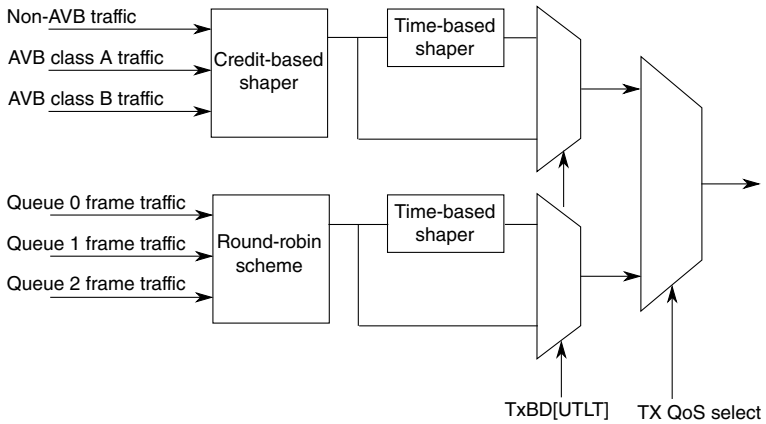


Figure 60-9. Transmit arbitration schemes

60.3.5.6.1 Round-robin policy

In the round-robin QoS scheme, each queue is given an equal opportunity to transmit one frame. For example, if queue *n* has a frame to transmit, the queue transmits its frame. After queue *n* has transmitted its frame, or if queue *n* does not have a frame to transmit, queue *n*+1 is then allowed to transmit its frame, and so on.

60.3.5.6.2 Credit-based shaper

The AVB credit based shaper acts independently, per class, to control the bandwidth distribution between normal traffic and time-sensitive traffic with respect to the total link bandwidth available. As per the IEEE 802.1Q, the maximum bandwidth distribution that can be allocated for the time-sensitive frames is 75%. The following example uses 70% for AVB frames and 30% for non-AVB frames:

Table 60-23. Bandwidth allocation example

AVB Class A	50%
AVB Class B	20%
non-AVB frames	30%
Total	100%

See DMACFG register for information on how to allocate bandwidth. The following figure shows how frame traffic is controlled within the MAC-NET core.

60.3.5.6.3 Time-based shaper

The time-based shaper enables the user to specify when a frame can be transmitted. It is always used in combination with either the round-robin scheme or the credit-based shaper. Use of the time-based shaper can ensure that frames are always transmitted in the correct time slot.

60.3.5.6.3.1 Time-based shaper example

Time-based shaping involves the following procedure:

1. Read the current value of the timer. See ATVR register and ATCR register.
2. Calculate the frame launch time and write this value to the TLT field of the frame's TxBD. See Enhanced transmit buffer descriptor.
3. Instruct the ENET to use the TLT by setting TxBD[UTLT].

The frame will be fetched and transmitted only if TxBD[TLT] is less than the current value of the timer. In other words, the timer must be past, that is, be greater than, the transmit launch time before the frame will be fetched and transmitted.

The transmit launch time must not be greater than the current value of ENET_ATVR + (0.5 × ENET_ATPER). This means the application cannot prepare frames with launch times beyond ENET_ATVR + (0.5 × ENET_ATPER). Because ENET_ATVR wraps, calculate TLT using the equation:

(ENET_ATVR + (0.5 × ENET_ATPER)) mod ENET_ATPER

For example, if ENET_ATPER is set to the recommended value of 1,000,000,000 ns (one second), then the user must not prepare future frames with launch times greater than (500,000,000 + ENET_ATVR).

As a specific example, if the current value of ENET_ATVR is 100,000, then you can prepare the following frames:

1. Frame 1 TxBD[TLT] = 100,000ns + 125,000ns
2. Frame 2 TxBD[TLT] = 100,000ns + 250,000ns
3. And so on.

60.3.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.

- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

60.3.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.
- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

60.3.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.

- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

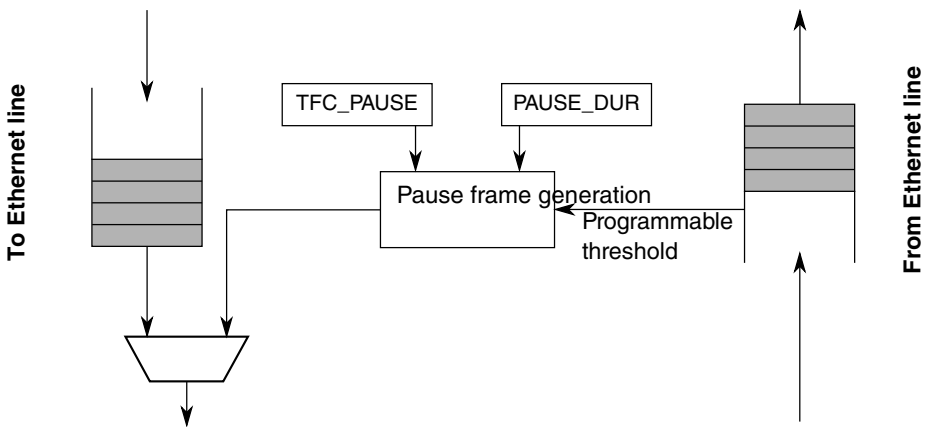


Figure 60-10. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

60.3.7 IP accelerator functions

The following sections describe the IP accelerator functions.

60.3.7.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 60-24. IPv4 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
Zero		Protocol		TCP/UDP length			

Table 60-25. IPv6 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
TCP/UDP length							
Zero						Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

60.3.7.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

60.3.7.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 60-26.

Table 60-26. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

60.3.7.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

60.3.7.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

60.3.7.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)
- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

60.3.7.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

60.3.7.6 IPv6 support

The following sections describe the IPv6 support.

60.3.7.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

60.3.7.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

60.3.8 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

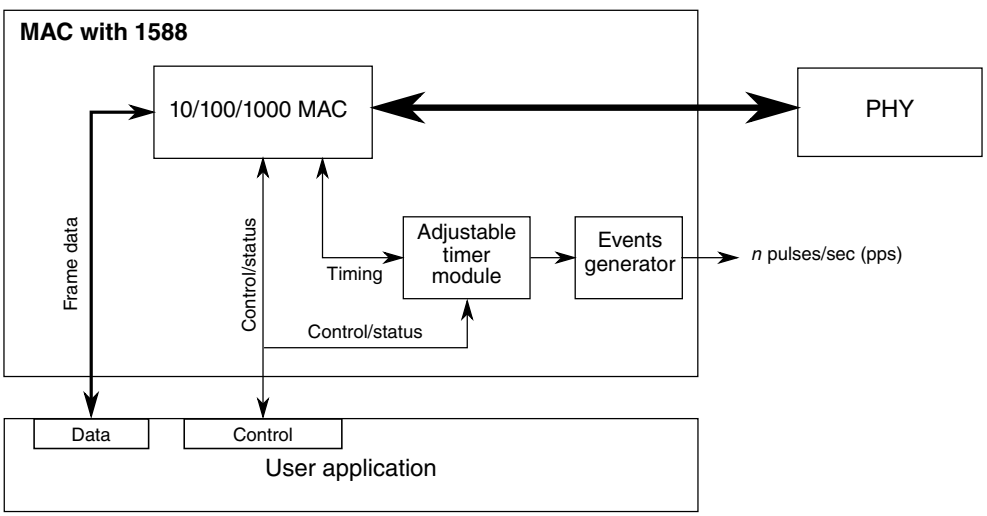


Figure 60-11. IEEE 1588 functions overview

60.3.8.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

60.3.8.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

See Timer Synchronization for Multi-Port Implementations, for external clock input options.

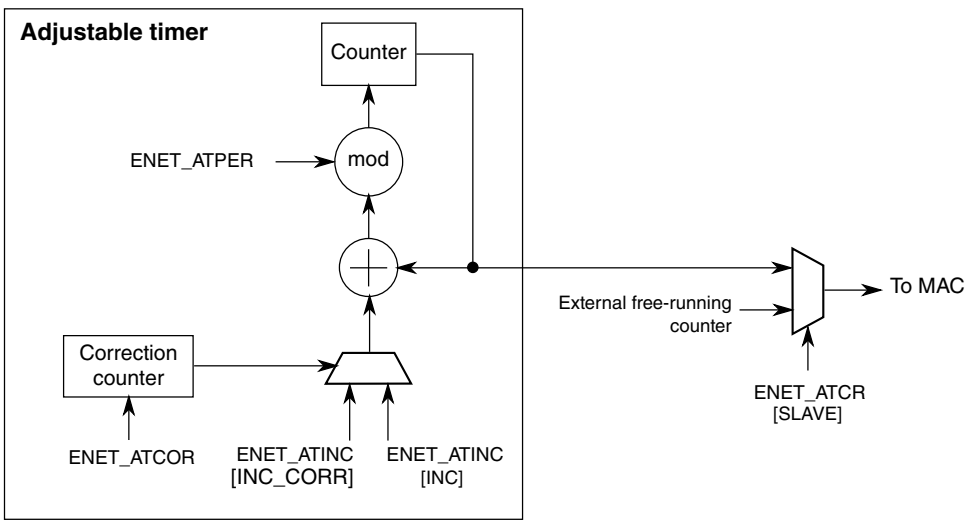


Figure 60-12. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules within the system must operate from a single timer (see Timer Synchronization for Multi-Port Implementations). When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

60.3.8.2 Timer Synchronization for Multi-Port Implementations

Additional inputs are available to provide a timer value for all time-stamping functions. This is necessary to synchronize the two MACs to a single reference timer. See the Chip Configuration details for available clock inputs to the time-stampimg functions. To operate the MAC in slave mode, ENET*n*_ATCR[SLAVE] disables the internal adjustable timer and uses the externally provided timer.

60.3.8.3 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

60.3.8.4 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

60.3.8.5 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

60.3.8.6 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

60.3.8.6.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

60.3.8.6.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TCSR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

60.3.8.6.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

60.3.9 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 1024 words.

60.3.9.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 60-27. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again.

Table continues on the next page...

Table 60-27. Receive FIFO thresholds definition (continued)

Register	Description
	If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of four.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

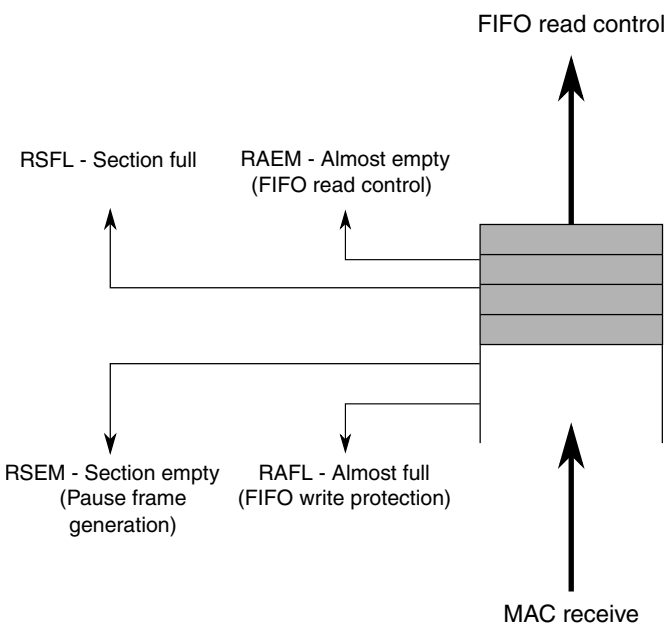


Figure 60-13. Receive FIFO overview

60.3.9.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 60-28. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an GMII/MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation). If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

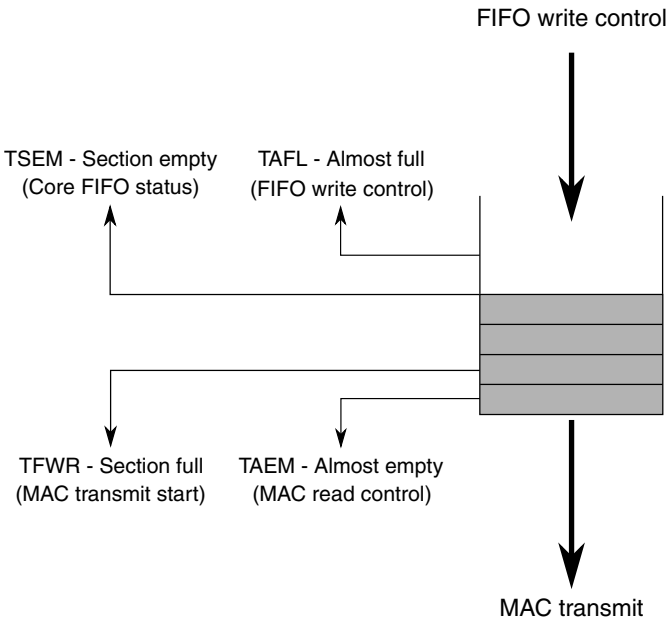


Figure 60-14. Transmit FIFO overview

60.3.10 Internal Loopback option

The core implements internal loopback option. To use internal loopback mode, the RX clock must be provided externally at the RGMII_RCK pin of the PHY interface. The loopback mode is controlled by the following ENET*n*_RCR register field:

Table 60-29. Loopback option

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces. In MII internal loopback, MII_TXCLK and MII_RXCLK must be provided with a clock signal (2.5 MHz for 10 Mbit/s, and 25 MHz for 100 Mbit/s))

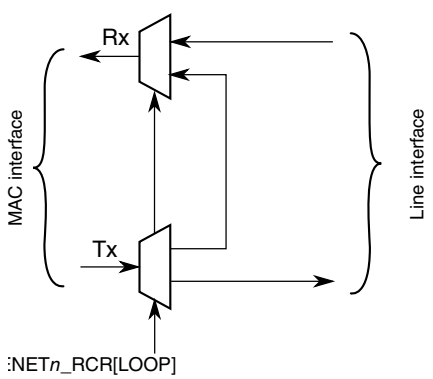


Figure 60-15. Loopback option

60.3.11 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- Legacy buffer descriptors are used in single-ring mode, that is, when DMA*n*CFG[DMA_CLASS_EN] are zero. In multi-ring mode, legacy TxBDs are used only with the round-robin scheme.
- The legacy buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

60.3.11.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 60-33 contains the descriptions for each field.

Table 60-30. Legacy FEC receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer — low halfword															
Offset + 6	Rx data buffer pointer — high halfword															

60.3.11.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 60-35 contains the descriptions for each field.

Table 60-31. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data Length															
Offset + 2	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—

Table continues on the next page...

Table 60-31. Legacy FEC transmit buffer descriptor (TxBD) (continued)

Offset + 4	Tx Data Buffer Pointer — low halfword
Offset + 6	Tx Data Buffer Pointer — high halfword

1. This field is not supported by the uDMA.

60.3.12 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for little-endian chips. ECR[DBSWP] must be set to 1 after reset to enable little-endian mode.

60.3.12.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 60-33 contains the descriptions for each field.

Table 60-32. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer – low halfword															
Offset + 6	Rx data buffer pointer – high halfword															
Offset + 8	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + A	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + C	Payload checksum															
Offset + E	Header length					—	—	—	Protocol type							
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table continues on the next page...

Table 60-32. Enhanced uDMA receive buffer descriptor (RxBD) (continued)

Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 60-33. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
Offset + 2	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 2	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 2	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 2	10–9	Reserved, must be cleared.
Offset + 2	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM bits are set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode. The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 2	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 2	6 MC	Set if the DA is multicast and not BC.
Offset + 2	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.

Table continues on the next page...

Table 60-33. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 2	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 2	3	Reserved, must be cleared.
Offset + 2	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 2	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 2	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 4	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
0ffset + 6	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword[1]
Offset + 8	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.
Offset + 8	12–6	Reserved, must be cleared.
Offset + 8	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + 8	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + 8	3	Reserved, must be cleared.
Offset + 8	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + 8	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + 8	0 FRAG	IPv4 Fragment. This is an accelerator option. This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + A	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + A	14–11	Reserved, must be cleared.
Offset + A	10 PE	PHY Error. This field is written by the uDMA. Set to "1" when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.

Table continues on the next page...

Table 60-33. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + A	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + A	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + A	6–0	Reserved, must be cleared.
Offset + C	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + E	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + E	10–8	Reserved, must be cleared.
Offset + E	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16		
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 64. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

60.3.12.2 Enhanced transmit buffer descriptor

Table 60-34. Enhanced transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer – low halfword															
Offset + 6	Tx Data Buffer Pointer – high halfword															
Offset + 8	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + A	—	INT	TS	PINS	IINS	—	—	UTLT	FTYPE				—	—	—	—
Offset + C	TLT – low halfword															
Offset + E	TLT – high halfword															
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 60-35. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 2	15 R	Ready. Written by the MAC and you. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 2	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.
Offset + 2	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	11	Last in frame. Written by user.

Table continues on the next page...

Table 60-35. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
	L	0 The buffer is not the last in the transmit frame 1 The buffer is the last in the transmit frame
Offset + 2	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 2	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 2	8–0	Reserved, must be cleared.
Offset + 4	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 6	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller. **NOTE:** For optimal performance, make the transmit buffer pointer evenly divisible by 64.
Offset + 8	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + 8	14	Reserved, must be cleared.
Offset + 8	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + 8	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + 8	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + 8	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + 8	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + 8	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + 8	7–0	Reserved, must be cleared.
Offset + A	15	Reserved, must be cleared.
Offset + A	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.

Table continues on the next page...

Table 60-35. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + A	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	10–9	Reserved, must be cleared.
Offset + A	8 UTLT	Use transmit launch time. If set to 1, TLT high and low values are used to determine if frame can be transmitted. This field must only be set in the *first* BD of a frame. TLT is always used in combination with either the round-robin scheme or the credit-based shaper. TLT can be used with a single BD queue. However, at least one DMA class, DMA*n*CFG[DMA_CLASS_ EN] must be enabled. For example, to use TLT with only queue 0, enable one of the DMA*n*CFG[DMA_CLASS_EN] fields and clear both TDAR1 and TDAR2. Although this enables multi-queue, only the single queue 0 is used.
Offset + A	7–4 FTYPE	Type of frame to be transmitted. If a credit-based scheme is used, this field must match the BD ring queue. 0x0 – Non-AVB. Corresponds to ENET_TDSR. 0x1 – AVB Class A. Corresponds to ENET_TDSR1. 0x2 – AVB Class B. Corresponds to ENET_TDSR2. All other values are reserved.
Offset + A	3–0	Reserved, must be cleared.
Offset + C	15–0 TLT low	Transmit launch time low. Low two bytes of time value that specifies when frame can be transmitted.
Offset + E	15–0 TLT high	Transmit launch time high. High two bytes of time value that specifies when frame can be transmitted.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16		
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

60.3.13 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

60.3.13.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 60-36. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 60-37. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 60-38. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

60.3.13.2 Data structure examples

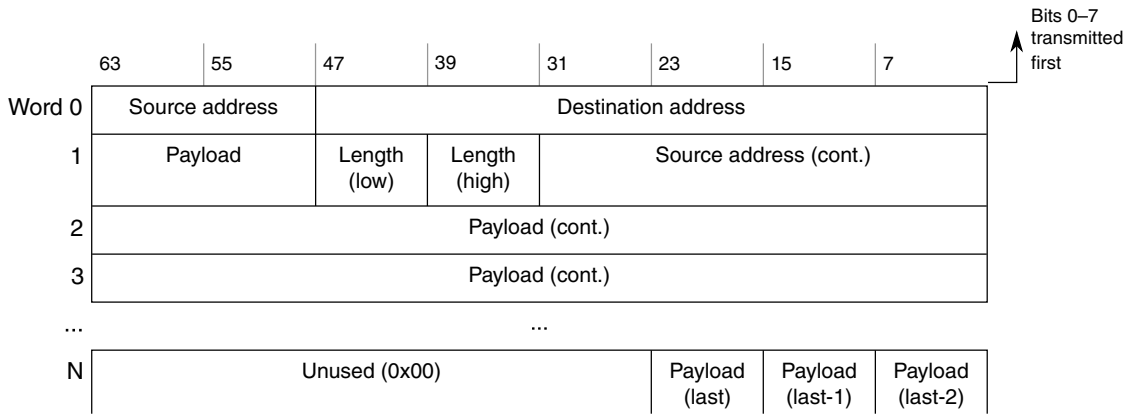


Figure 60-16. Normal Ethernet frame 64-bit mapping example

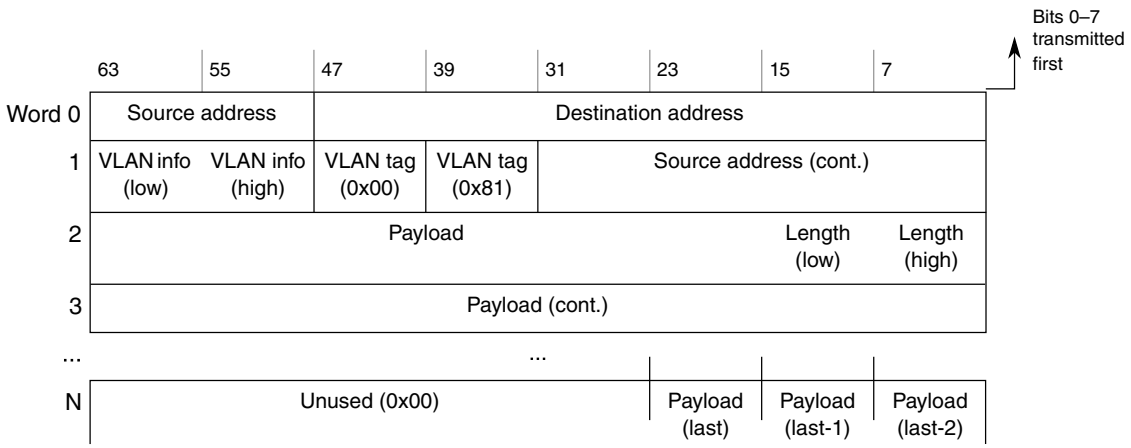


Figure 60-17. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

60.3.13.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

60.3.14 FIFO protection

The following sections describe the FIFO protection mechanisms.

60.3.14.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 60-18) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

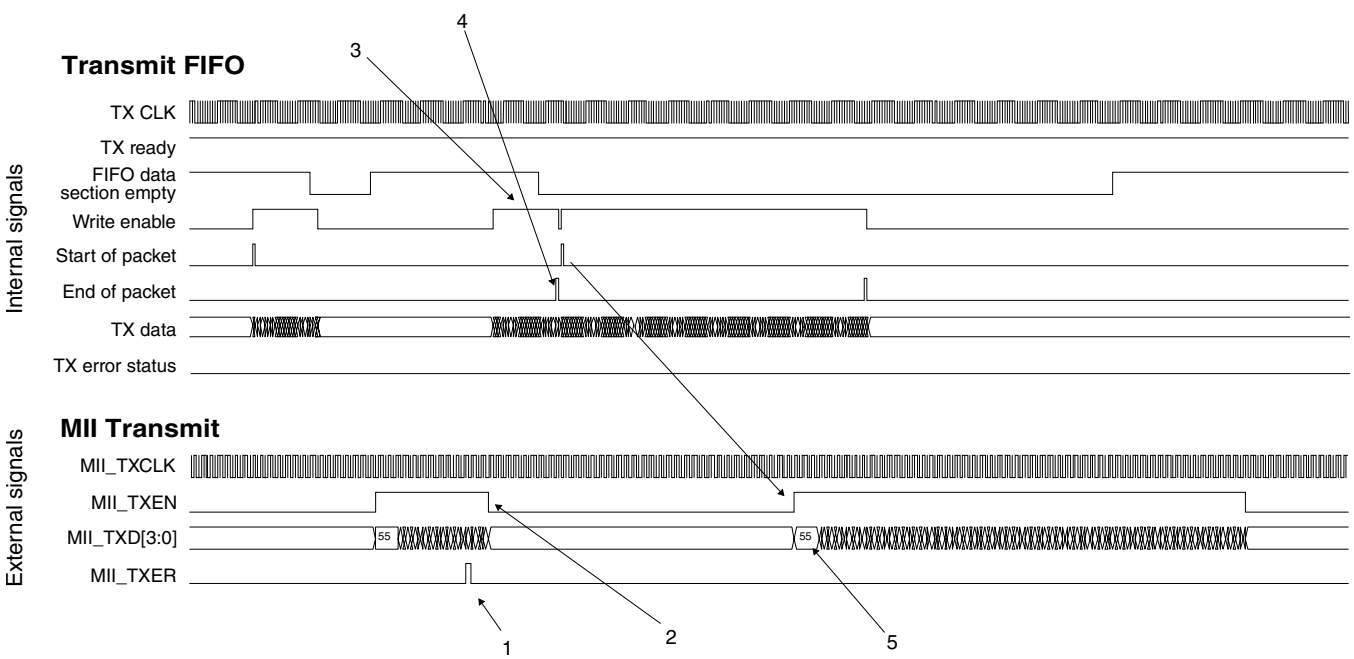


Figure 60-18. Transmit FIFO underflow protection

60.3.14.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

60.3.14.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

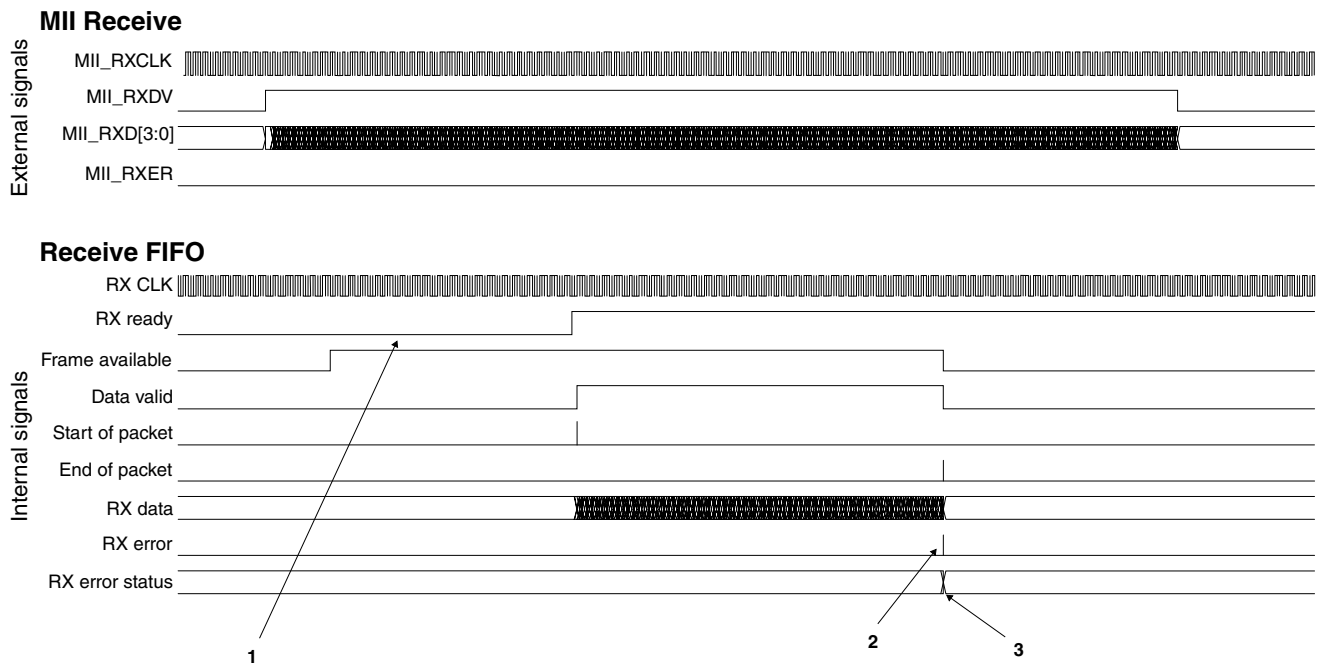


Figure 60-19. Receive FIFO overflow protection

60.3.15 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

60.3.15.1 MDIO clause 22 frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 60-39. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 60-40. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

Table 60-40. MDIO frame field descriptions

Field	Description
(16 bits)	

60.3.15.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 60-41. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC).

The fields and transactions are summarized in the following tables.

Table 60-42. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 60-43 for more information. 00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.

Table continues on the next page...

Table 60-42. MDIO clause 45 frame field descriptions (continued)

Field	Description
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 60-43. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

60.3.15.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

60.3.15.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

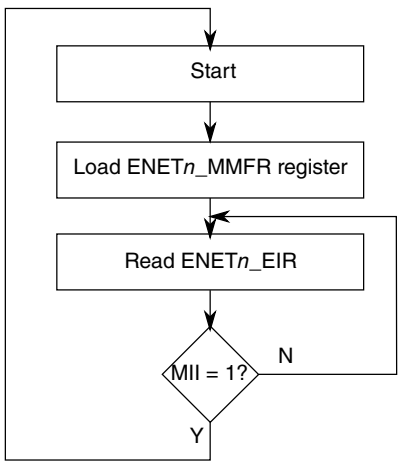


Figure 60-20. MDIO access overview

60.3.16 Ethernet interfaces

The following Ethernet interfaces are implemented:

- Fast Ethernet MII 10/100 (Media Independent Interface)
- RMII 10/100 using interface converters/gaskets
- RGMII 10/100/1000 by way of interface converters/gaskets

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]	RCR[RGMII_EN]
MII - 10 Mbit/s	0	—	0	0
MII - 100 Mbit/s	0	—	0	0
RMII - 10 Mbit/s	0	1	1	0
RMII - 100 Mbit/s	0	0	1	0
RGMII - 10 Mbit/s	0	1	0	1
RGMII - 100 Mbit/s	0	0	0	1
RGMII - 1000 Mbit/s	1	—	0	1

60.3.16.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

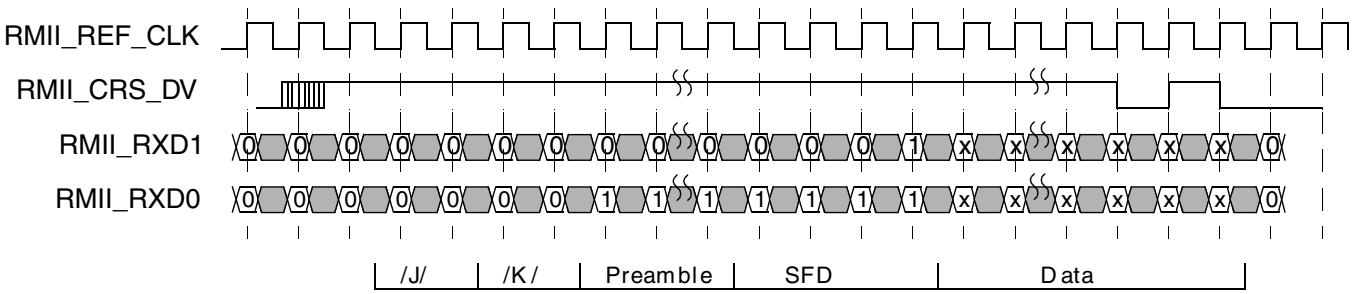


Figure 60-21. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

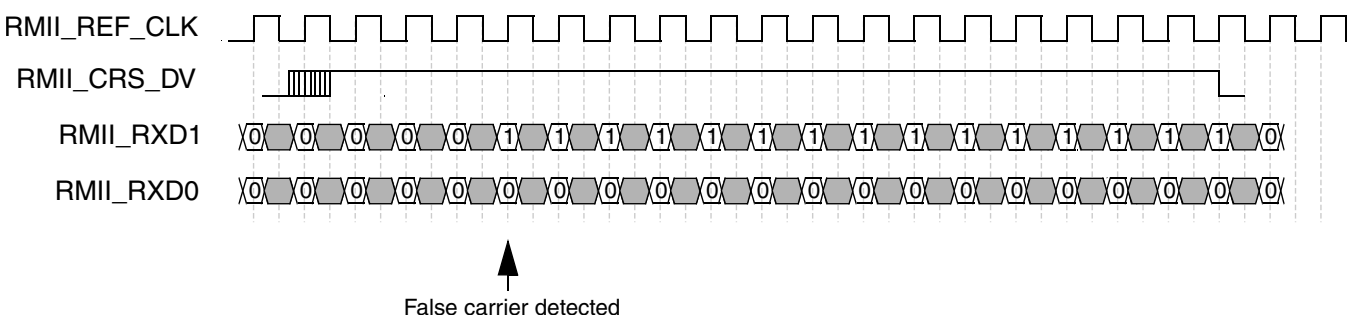


Figure 60-22. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

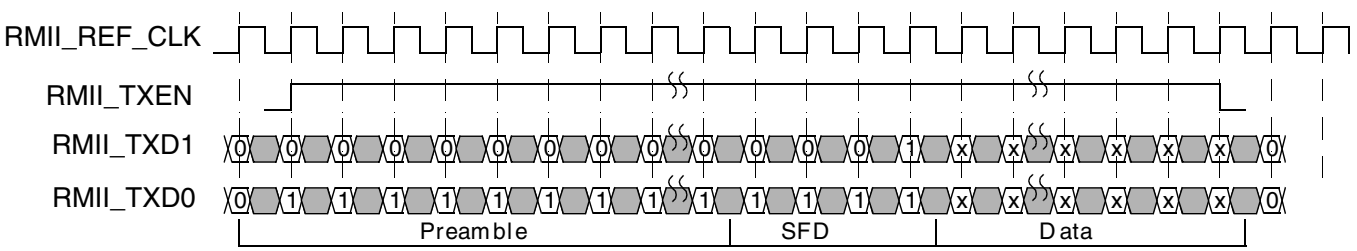


Figure 60-23. RMII transmit operation

60.3.16.2 RGMII interface

In RGMII modes, the data and control information is multiplexed by taking advantage of both edges of the reference clocks.

The data signals contain the lower four data bits on the rising edge and the upper four bits on the falling edge. The control signals are multiplexed into a single clock cycle using the same technique.

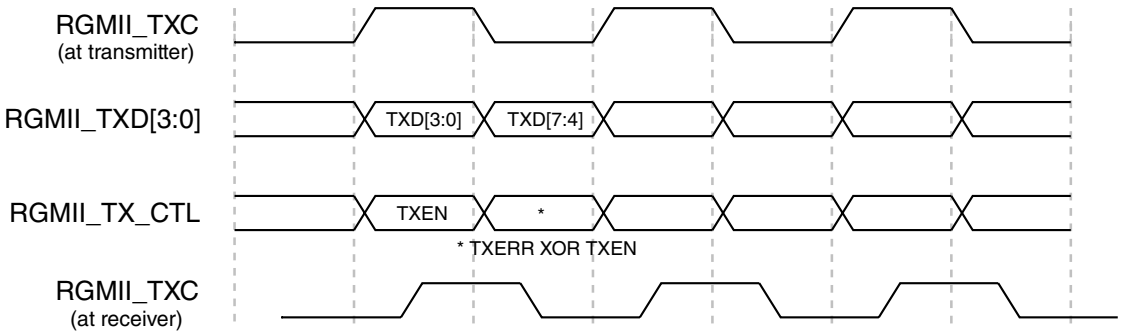


Figure 60-24. RGMII transmit operation

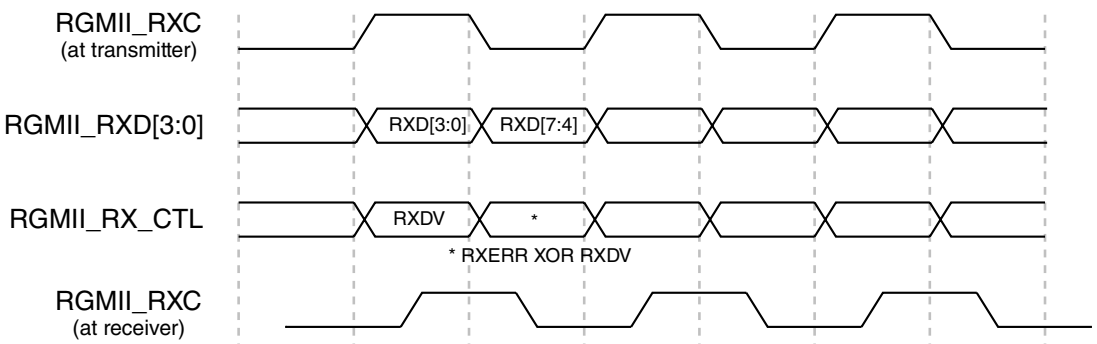


Figure 60-25. RGMII receive operation

60.3.16.3 MII Interface — transmit

On transmit, all data transfers are synchronous to MII_TXCLK rising edge. The MII data enable signal MII_TXEN is asserted to indicate the start of a new frame, and remains asserted until the last byte of the frame is present on the MII_TXD[3:0] bus.

Between frames, MII_TXEN remains deasserted.

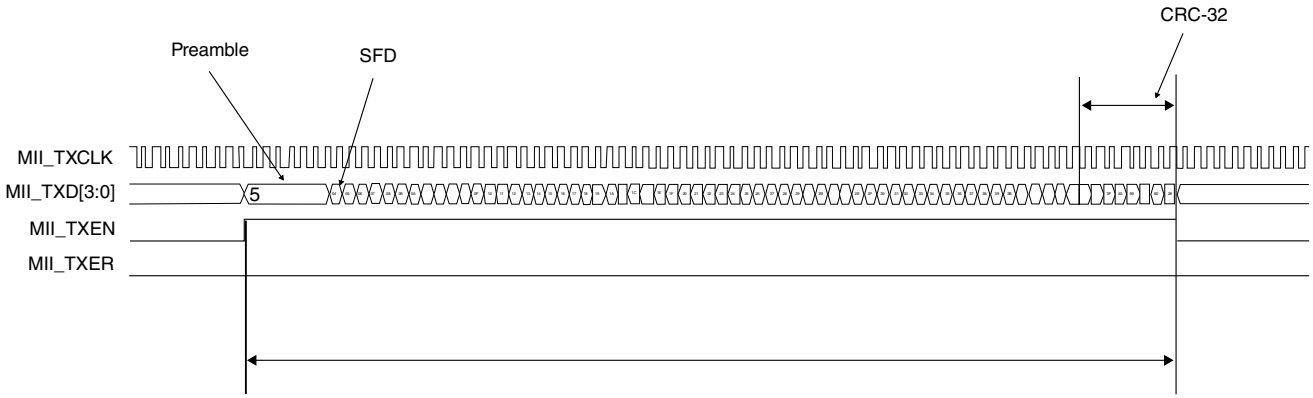


Figure 60-26. MII transmit operation

If a frame is received on the FIFO interface with an error (for example, RxBD[ME] set) the frame is subsequently transmitted with the MII_TXER error signal for one clock cycle at any time during the packet transfer.

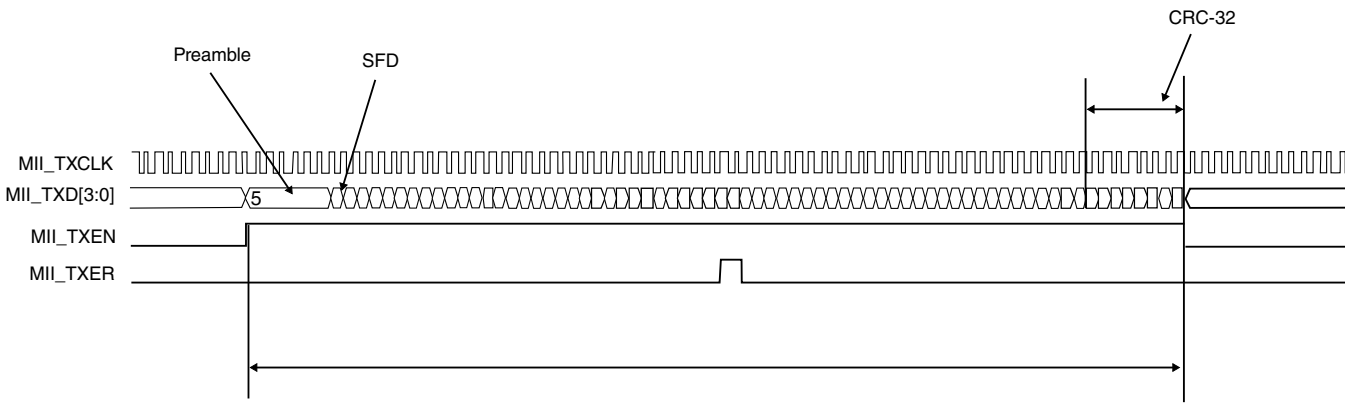


Figure 60-27. MII transmit operation — errored frame

60.3.16.3.1 Transmit with collision — half-duplex

When a collision is detected during a frame transmission (MII_COL asserted), the MAC stops the current transmission, sends a 32-bit jam pattern, and re-transmits the current frame.

(See Collision detection in half-duplex mode for details)

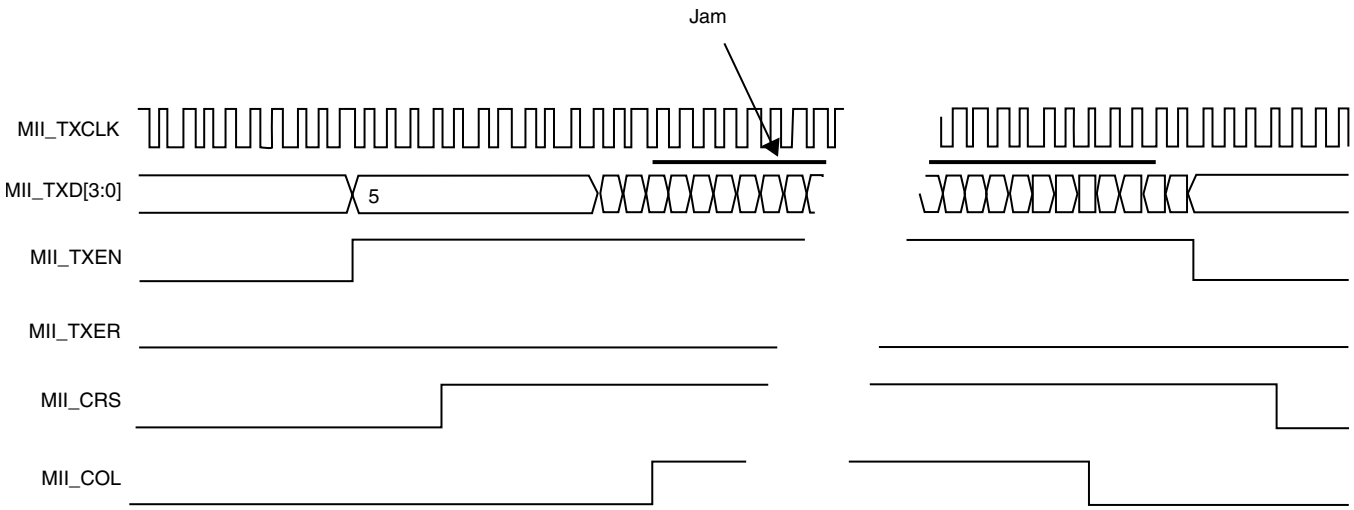


Figure 60-28. MII transmit operation — transmission with collision

60.3.16.4 MII interface — receive

On receive, all signals are sampled on the MII_RXCLK rising edge. The MII data enable signal, MII_RXDV, is asserted by the PHY to indicate the start of a new frame and remains asserted until the last byte of the frame is present on MII_RXD[3:0] bus.

Between frames, MII_RXDV remains deasserted.

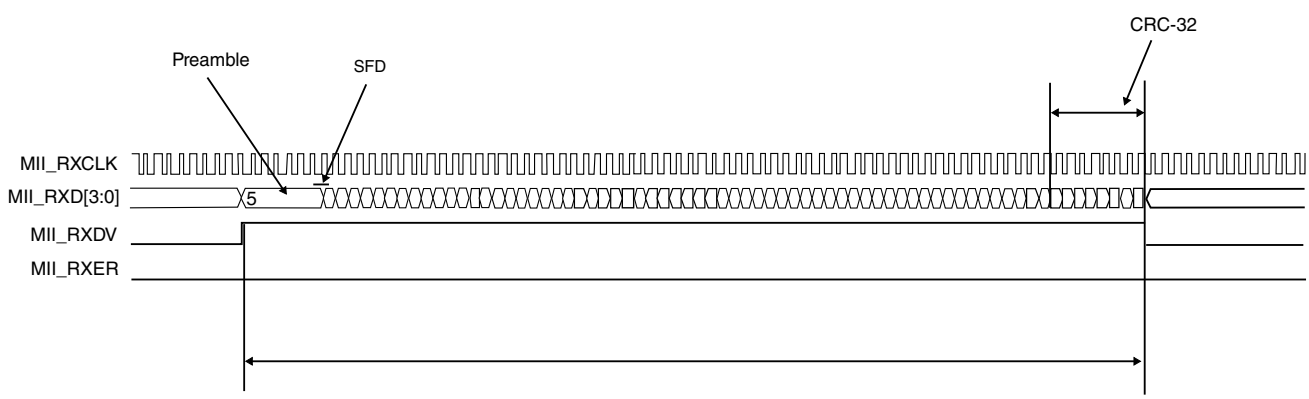


Figure 60-29. MII receive operation

If the PHY detects an error on the frame received from the line, the PHY asserts the MII error signal, MII_RXER, for at least one clock cycle at any time during the packet transfer.

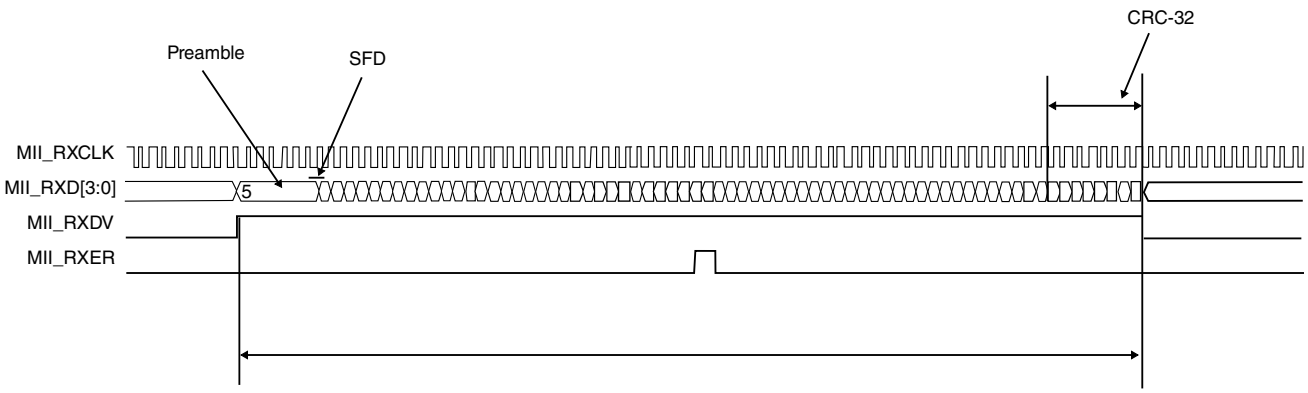


Figure 60-30. MII receive operation — errored frame

A frame received on the MII interface with a PHY error indication is subsequently transferred on the FIFO interface with RxBD[ME] set.

60.3.17 AVB configuration

The following steps give an example of how to initialize the ENET module for AVB.

1. Set up QOS register:
   - Set TX_SCHEME to 000b, credit-based scheme.
   - Set RX_FLUSH0 to 1, enable RX flush for ring 0.
2. Set up TX BD ring and RX BD ring for each queue, 0-2.
   - Program ENET_MRBR, ENET_TDSR, and ENET_RDSR.
   - Program ENET_MRBR1, ENET_TDSR1, and ENET_RDSR1.
   - Program ENET_MRBR2, ENET_TDSR2, and ENET_RDSR2.
   - Program each TX and RX BD ring queue for all classes used in memory.

NOTE

If using credit-based scheme, ensure that enhanced transmit buffer descriptor FTYPE field matches BD ring queue, for example:
- FTYPE = 0h corresponds to ENET_TDSR
- FTYPE = 1h corresponds to ENET_TDSR1
- FTYPE = 2h corresponds to ENET_TDSR2

3. Program ENET_DMA1CFG and ENET_DMA2CFG for class 1 and class 2 corresponding to BD ring queue 1 and 2. DMA_CLASS_EN must be set to one for

that class to be enabled. See DMACFG register for information on how to program IDLE_SLOPE.

4. Program ENET_RCMR1 and ENET_RCMR2 for class 1 and class 2 matching for receive.

NOTE

Even if a match occurs, if ENET_DMA*n*CFG[DMA_CLASS_EN] is zero for the corresponding class, the RX frame will be automatically forwarded to BD ring queue 0.

5. Program the other ENET registers according to application requirements.
6. Program ENET_RDAR, ENET_RDAR1, ENET_RDAR2, ENET_TDAR, ENET_TDAR1, and ENET_TDAR2 to 0100_0000h according to the classes used.
7. Set ENET_ECR[ETHEREN] to one.

60.3.18 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

60.3.18.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

60.3.18.2 Magic packet detection

The core is designed to detect magic packets (See "Magic Packets" topics) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

60.3.18.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

60.3.19 Clocks

The table found here describes the clock sources for ENET. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 60-44. Clocks

Clock name	Description
ipg_clk	Module clock
ipg_clk_rmii	Module clock for RMII
ipg_clk_s	Peripheral access clock
ipg_clk_time (ts_clk)	Peripheral clock
txc_sampling_clk	Transmit sampling clock

60.3.20 Resets and stop controls

The following sections describe the resets and stop controls.

60.3.20.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

60.3.20.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting GMII/TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

60.3.20.3 Debug mode

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the MII/GMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

60.3.20.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

60.3.20.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.
- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped state.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

60.3.20.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the MII/GMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

60.3.20.4.3 Graceful stop interrupt (GRA)

The graceful stop interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

60.3.21 Interrupt coalescence

The purpose of the interrupt coalescing is to reduce the number of interrupts generated by the MAC so as to reduce the CPU loading.

To facilitate this interrupt coalescing for each queue, these registers are available with the same control and configuration fields.

- Transmit Interrupt Coalescing Register (TXICn) where n=0,1,2 for queue/class 0,1,2.
- Receive Interrupt Coalescing Register (RXICn) where n=0,1,2 for queue/class 0,1,2.

When coalescing is enabled by asserting the corresponding ICEN field and such interrupt is also enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt when the threshold number of frames is reached (defined by ICFT) or when the threshold timer expires (defined by ICTT).

When coalescing is disabled by de-asserting ICEN, but interrupt is enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt as they are received without using coalescing. Interrupt coalescing is done for each transmit and receive queue/class independently.

60.3.21.1 Interrupt coalescence setup

Interrupt coalescence supports both legacy and enhanced BDs. The following guidelines are recommended when setting up interrupt coalescence.

- When the MAC is configured for enhanced (IEEE 1588) mode, that is, enhanced BDs:
  - Set the INT bit in the enhanced received buffer descriptor to one.
  - Set the INT bit in the enhanced transmit buffer descriptor(s) to one.
- Clear the RXB, RXB1, and RXB2 fields in the EIMR register.
- Clear the TXB, TXB1, and TXB2 fields in the EIMR register.

60.3.21.2 Updating the frame count threshold on-the-fly

To update the ICFT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new threshold value to the ICFT field.

3. Set ICEN to one.

NOTE

The ICFT field can be updated on-the-fly without disabling the ICEN field. The hardware interrupt will continue and there is a possibility that an interrupt will occur depending on the state of the hardware counter and the previous ICFT value.

60.3.21.3 Updating the timer threshold on-the-fly

To update the ICTT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new timer value to the ICTT field.
3. Set ICEN to one.

60.4 External Signals

The table found here describes the external signals of ENET.

Table 60-45. ENET External Signals

Signal	Description	Mode	Direction
1588_EVENT_IN/ 1588_EVENT_OUT	Capture/compare block input/output event bus signal. • When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. • When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in the register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	MII/RMII	IO
ENET_1588_CLKIN	Alternate IEEE 1588 Ethernet clock input; Clock period should be an integer number in nanoseconds.	MII/RMII	I
MII_COL	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full-duplex mode.	MII	I

Table continues on the next page...

Table 60-45. ENET External Signals (continued)

Signal	Description	Mode	Direction
MII_CRS	Carrier sense. Asserted if receive or transmit equals 1 and negated if both transmit and receive equals 0. **NOTE:** In RMII mode, this signal is present on the RMII_CRS_DV pin.	MII	I
MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	MII/RGMII/RMII	O
MDIO	Transfers control information between the external PHY and the media access controller. Data is synchronous to MDC. This signal is an input after reset.	MII/RGMII/RMII	IO
MII_RXCLK	In MII mode, provides a timing reference for RXDV, RXD[3:0], and RXER. **NOTE:** In RGMII mode, this pin is present as RGMII_RXC and provides a timing reference for RGMII_RXD[3:0] and RGMII_RX_CTL.	MII	I
RXER	When asserted with RXDV, indicates an error detection from the PHY in the current frame.	MII/RMII	I
RX_EN	Asserting this input indicates the PHY has valid nibbles present on the MII. RX_EN must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RX_EN must start no later than the SFD and exclude any EOF. **NOTE:** In RMII mode, this pin also generates the CRS signal.	MII/RMII	I
MII_TXCLK	Input clock, which provides a timing reference for TXEN, TXD[3:0], and TXER.	MII	I
MII_TXER	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	MII	O
TXEN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is de-asserted before the first TXCLK following the final nibble of the frame. **NOTE:** In RGMII mode, it is present as RGMII_TX_CTL, and contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXER (TXEN XOR TXER) on the falling edge of RGMII_TXC.	MII/RMII	O
RXD[3:0]	Contains the Ethernet input data transferred from the PHY to the media-access controller when RXEN is asserted. **NOTE:** In RMII mode, bits [1:0] provide the Ethernet input data.	MII/RMII/RGMII	I
TXD[3:0]	Serial output Ethernet data. Only valid during TX_EN assertion. **NOTE:** In RMII mode, bits [1:0] provide the Serial output Ethernet data.	MII/RMII/RGMII	O
RGMII_RX_CTL	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF. **NOTE:** • In RMII mode, this signal is used as RMII_CRS_DV. • In RGMII mode, contains RXDV on the rising edge of RGMII_RXC, and a logical derivative of RXDV	RGMII	I

Table continues on the next page...

Table 60-45. ENET External Signals (continued)

Signal	Description	Mode	Direction
	and RXER (RXDV XOR RXER) on the falling edge of RGMII_RXC. • In MII mode, this signal is used as MII_RXDV.		
RGMII_RXC	Timing reference for RGMII_RXD[3:0] and RGMII_RX_CTL	RGMII	I
RGMII_TX_CTL	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TXCLK following the final nibble of the frame. **NOTE:** In RGMII mode, contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXER (TXEN XOR TXER) on the falling edge of RGMII_TXC.	RGMII	O
RGMII_TXC	Timing reference for RGMII_TXD[3:0] and RGMII_TX_CTL.	RGMII	O
RMII_REF_CLK	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	RMII	IO
RGMII_REF_CLK	In RGMII mode, this signal provides 125Mhz external reference clock input.	RGMII	I

60.5 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 60-46. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

60.5.1 ENET register descriptions

60.5.1.1 ENET memory map

ENET base address: 4042_4000h

ENET_1G base address: 4042_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt Event Register (EIR)	32	RW	0000_0000h
8h	Interrupt Mask Register (EIMR)	32	RW	0000_0000h
10h	Receive Descriptor Active Register - Ring 0 (RDAR)	32	RW	0000_0000h
14h	Transmit Descriptor Active Register - Ring 0 (TDAR)	32	RW	0000_0000h
24h	Ethernet Control Register (ECR)	32	RW	7000_0000h
40h	MII Management Frame Register (MMFR)	32	RW	0000_0000h
44h	MII Speed Control Register (MSCR)	32	RW	0000_0000h
64h	MIB Control Register (MIBC)	32	RW	C000_0000h
84h	Receive Control Register (RCR)	32	RW	05EE_0001h
C4h	Transmit Control Register (TCR)	32	RW	0000_0000h
E4h	Physical Address Lower Register (PALR)	32	RW	0000_0000h
E8h	Physical Address Upper Register (PAUR)	32	RW	0000_8808h
ECh	Opcode/Pause Duration Register (OPD)	32	RW	0001_0000h
F0h - F8h	Transmit Interrupt Coalescing Register (TXIC0 - TXIC2)	32	RW	0000_0000h
100h - 108h	Receive Interrupt Coalescing Register (RXIC0 - RXIC2)	32	RW	0000_0000h
118h	Descriptor Individual Upper Address Register (IAUR)	32	RW	0000_0000h
11Ch	Descriptor Individual Lower Address Register (IALR)	32	RW	0000_0000h
120h	Descriptor Group Upper Address Register (GAUR)	32	RW	0000_0000h
124h	Descriptor Group Lower Address Register (GALR)	32	RW	0000_0000h
144h	Transmit FIFO Watermark Register (TFWR)	32	RW	0000_0000h
160h	Receive Descriptor Ring 1 Start Register (RDSR1)	32	RW	0000_0000h
164h	Transmit Buffer Descriptor Ring 1 Start Register (TDSR1)	32	RW	0000_0000h
168h	Maximum Receive Buffer Size Register - Ring 1 (MRBR1)	32	RW	0000_0000h
16Ch	Receive Descriptor Ring 2 Start Register (RDSR2)	32	RW	0000_0000h
170h	Transmit Buffer Descriptor Ring 2 Start Register (TDSR2)	32	RW	0000_0000h
174h	Maximum Receive Buffer Size Register - Ring 2 (MRBR2)	32	RW	0000_0000h
180h	Receive Descriptor Ring 0 Start Register (RDSR)	32	RW	0000_0000h
184h	Transmit Buffer Descriptor Ring 0 Start Register (TDSR)	32	RW	0000_0000h
188h	Maximum Receive Buffer Size Register - Ring 0 (MRBR)	32	RW	0000_0000h
190h	Receive FIFO Section Full Threshold (RSFL)	32	RW	0000_0000h
194h	Receive FIFO Section Empty Threshold (RSEM)	32	RW	0000_0000h
198h	Receive FIFO Almost Empty Threshold (RAEM)	32	RW	0000_0004h
19Ch	Receive FIFO Almost Full Threshold (RAFL)	32	RW	0000_0004h
1A0h	Transmit FIFO Section Empty Threshold (TSEM)	32	RW	0000_0000h
1A4h	Transmit FIFO Almost Empty Threshold (TAEM)	32	RW	0000_0004h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1A8h	Transmit FIFO Almost Full Threshold (TAFL)	32	RW	0000_0008h
1ACh	Transmit Inter-Packet Gap (TIPG)	32	RW	0000_000Ch
1B0h	Frame Truncation Length (FTRL)	32	RW	0000_07FFh
1C0h	Transmit Accelerator Function Configuration (TACC)	32	RW	0000_0000h
1C4h	Receive Accelerator Function Configuration (RACC)	32	RW	0000_0000h
1C8h - 1CCh	Receive Classification Match Register for Class n (RCMR1 - RCMR2)	32	RW	0000_0000h
1D8h - 1DCh	DMA Class Based Configuration (DMA1CFG - DMA2CFG)	32	RW	0000_0000h
1E0h	Receive Descriptor Active Register - Ring 1 (RDAR1)	32	RW	0000_0000h
1E4h	Transmit Descriptor Active Register - Ring 1 (TDAR1)	32	RW	0000_0000h
1E8h	Receive Descriptor Active Register - Ring 2 (RDAR2)	32	RW	0000_0000h
1ECh	Transmit Descriptor Active Register - Ring 2 (TDAR2)	32	RW	0000_0000h
1F0h	QOS Scheme (QOS)	32	RW	0000_0000h
204h	Tx Packet Count Statistic Register (RMON_T_PACKETS)	32	R	0000_0000h
208h	Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)	32	R	0000_0000h
20Ch	Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)	32	R	0000_0000h
210h	Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)	32	R	0000_0000h
214h	Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)	32	R	0000_0000h
218h	Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)	32	R	0000_0000h
21Ch	Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)	32	R	0000_0000h
220h	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)	32	R	0000_0000h
224h	Tx Collision Count Statistic Register (RMON_T_COL)	32	R	0000_0000h
228h	Tx 64-Byte Packets Statistic Register (RMON_T_P64)	32	R	0000_0000h
22Ch	Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)	32	R	0000_0000h
230h	Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)	32	R	0000_0000h
234h	Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)	32	R	0000_0000h
238h	Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)	32	R	0000_0000h
23Ch	Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)	32	R	0000_0000h
240h	Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)	32	R	0000_0000h
244h	Tx Octets Statistic Register (RMON_T_OCTETS)	32	R	0000_0000h
24Ch	Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)	32	R	0000_0000h
250h	Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
254h	Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)	32	R	0000_0000h
258h	Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)	32	R	0000_0000h
25Ch	Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)	32	R	0000_0000h
260h	Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)	32	R	0000_0000h
264h	Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)	32	R	0000_0000h
268h	Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)	32	R	0000_0000h
26Ch	Reserved Statistic Register (IEEE_T_SQE)	32	R	0000_0000h
270h	Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)	32	R	0000_0000h
274h	Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)	32	R	0000_0000h
284h	Rx Packet Count Statistic Register (RMON_R_PACKETS)	32	R	0000_0000h
288h	Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)	32	R	0000_0000h
28Ch	Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)	32	R	0000_0000h
290h	Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)	32	R	0000_0000h
294h	Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)	32	R	0000_0000h
298h	Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)	32	R	0000_0000h
29Ch	Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)	32	R	0000_0000h
2A0h	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)	32	R	0000_0000h
2A8h	Rx 64-Byte Packets Statistic Register (RMON_R_P64)	32	R	0000_0000h
2ACh	Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)	32	R	0000_0000h
2B0h	Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)	32	R	0000_0000h
2B4h	Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)	32	R	0000_0000h
2B8h	Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)	32	R	0000_0000h
2BCh	Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)	32	R	0000_0000h
2C0h	Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)	32	R	0000_0000h
2C4h	Rx Octets Statistic Register (RMON_R_OCTETS)	32	R	0000_0000h
2C8h	Frames not Counted Correctly Statistic Register (IEEE_R_DROP)	32	R	0000_0000h
2CCh	Frames Received OK Statistic Register (IEEE_R_FRAME_OK)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2D0h	Frames Received with CRC Error Statistic Register (IEEE_R_CRC)	32	R	0000_0000h
2D4h	Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)	32	R	0000_0000h
2D8h	Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)	32	R	0000_0000h
2DCh	Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)	32	R	0000_0000h
2E0h	Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)	32	R	0000_0000h
400h	Adjustable Timer Control Register (ATCR)	32	RW	0000_0000h
404h	Timer Value Register (ATVR)	32	RW	0000_0000h
408h	Timer Offset Register (ATOFF)	32	RW	0000_0000h
40Ch	Timer Period Register (ATPER)	32	RW	3B9A_CA00h
410h	Timer Correction Register (ATCOR)	32	RW	0000_0000h
414h	Time-Stamping Clock Period Register (ATINC)	32	RW	0000_0000h
418h	Timestamp of Last Transmitted Frame (ATSTMP)	32	R	0000_0000h
604h	Timer Global Status Register (TGSR)	32	RW	0000_0000h
608h	Timer Control Status Register (TCSR0)	32	RW	0000_0000h
60Ch	Timer Compare Capture Register (TCCR0)	32	RW	0000_0000h
610h	Timer Control Status Register (TCSR1)	32	RW	0000_0000h
614h	Timer Compare Capture Register (TCCR1)	32	RW	0000_0000h
618h	Timer Control Status Register (TCSR2)	32	RW	0000_0000h
61Ch	Timer Compare Capture Register (TCCR2)	32	RW	0000_0000h
620h	Timer Control Status Register (TCSR3)	32	RW	0000_0000h
624h	Timer Compare Capture Register (TCCR3)	32	RW	0000_0000h

60.5.1.2 Interrupt Event Register (EIR)

60.5.1.2.1 Offset

Register	Offset
EIR	4h

60.5.1.2.2 Function

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

60.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	RXFLUSH_2	RXFLUSH_1	RXFLUSH_0					TXF2	TXB2	RXF2	RXB2	TXF1	TXB1	RXF1	RXB1
W	W1C	W1C	W1C	W1C	0			0	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.2.4 Fields

Field	Function
31 —	Reserved
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27	Transmit Frame Interrupt

Table continues on the next page...

Field	Function
TXF	Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE:** In situations where the device has various masters generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled in ENET_TFWR[STRFWD]. See Transmit FIFO Watermark Register (TFWR). Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14 RXFLUSH_2	RX DMA Ring 2 flush indication. This ring's RX frame has been flushed due to either RDAR2[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_2] is enabled. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET_1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>13
RXFLUSH_1</td><td>RX DMA Ring 1 flush indication. This ring's RX frame has been flushed due to either RDAR1[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_1] is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET_1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>12
RXFLUSH_0</td><td>RX DMA Ring 0 flush indication. . This ring's RX frame has been flushed due to either RDAR[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_0] is enabled.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET_1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>11-9
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>8
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>7
TXF2</td><td>Transmit frame interrupt, class 2
This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 2).
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIR</td></tr>
<tr><td>ENET_1G</td><td>EIR</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>6
TXB2</td><td>Transmit buffer interrupt, class 2
This field indicates that a transmit buffer descriptor has been updated (ring/class 2).
NOTE: This field is not supported in every instance. The following table includes only supported registers.</td></tr>
</table>

Table continues on the next page...

Field	Function
	Instance / **Field supported in** / **Field not supported in** ENET — EIR ENET_1G EIR —
5 RXF2	Receive frame interrupt, class 2 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 2). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET — EIR ENET_1G EIR —
4 RXB2	Receive buffer interrupt, class 2 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 2). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET — EIR ENET_1G EIR —
3 TXF1	Transmit frame interrupt, class 1 This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 1). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET — EIR ENET_1G EIR —
2 TXB1	Transmit buffer interrupt, class 1 This field indicates that a transmit buffer descriptor has been updated (ring/class 1). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET — EIR

Table continues on the next page...

Field	Function
	Instance / **Field supported in** / **Field not supported in** ENET_1G / EIR / —
1 RXF1	Receive frame interrupt, class 1 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 1). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / EIR ENET_1G / EIR / —
0 RXB1	Receive buffer interrupt, class 1 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 1). **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / EIR ENET_1G / EIR / —

60.5.1.3 Interrupt Mask Register (EIMR)

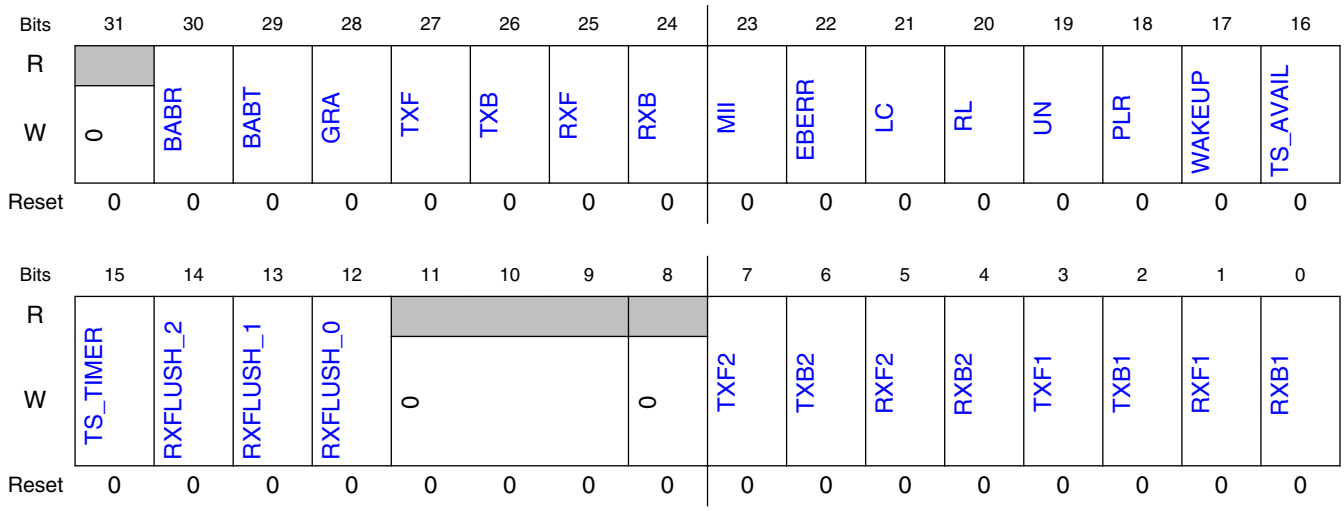
60.5.1.3.1 Offset

Register	Offset
EIMR	8h

60.5.1.3.2 Function

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

60.5.1.3.3 Diagram



60.5.1.3.4 Fields

Field	Function
31 —	This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The

Table continues on the next page...

Field	Function
	corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

Field	Function
	0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.
14 RXFLUSH_2	Corresponds to interrupt source EIR[RXFLUSH_2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / EIMR ENET_1G / EIMR / — 0b - The corresponding interrupt source is masked.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>13
RXFLUSH_1</td><td>Corresponds to interrupt source EIR[RXFLUSH_1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIMR</td></tr><tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr></table>
0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>12
RXFLUSH_0</td><td>Corresponds to interrupt source EIR[RXFLUSH_0] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_0] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIMR</td></tr><tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr></table>
0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>11-9
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>8
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>7
TXF2</td><td>Transmit frame interrupt, class 2

Corresponds to interrupt source EIR[TXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>ENET</td><td>—</td><td>EIMR</td></tr><tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr></table>
0b - The corresponding interrupt source is masked.</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>6
TXB2</td><td>Transmit buffer interrupt, class 2
Corresponds to interrupt source EIR[TXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>
0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>5
RXF2</td><td>Receive frame interrupt, class 2
Corresponds to interrupt source EIR[RXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>
0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>4
RXB2</td><td>Receive buffer interrupt, class 2
Corresponds to interrupt source EIR[RXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>
0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>3
TXF1</td><td>Transmit frame interrupt, class 1</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>Corresponds to interrupt source EIR[TXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>

0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>2

TXB1</td><td>Transmit buffer interrupt, class 1

Corresponds to interrupt source EIR[TXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>

0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>1

RXF1</td><td>Receive frame interrupt, class 1

Corresponds to interrupt source EIR[RXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>EIMR</td></tr>
<tr><td>ENET_1G</td><td>EIMR</td><td>—</td></tr>
</table>

0b - The corresponding interrupt source is masked.
1b - The corresponding interrupt source is not masked.</td></tr>
<tr><td>0

RXB1</td><td>Receive buffer interrupt, class 1

Corresponds to interrupt source EIR[RXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.</td></tr>
</table>

Field	Function
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / — / EIMR ENET_1G / EIMR / — 0b - The corresponding interrupt source is masked. 1b - The corresponding interrupt source is not masked.

60.5.1.4 Receive Descriptor Active Register - Ring 0 (RDAR)

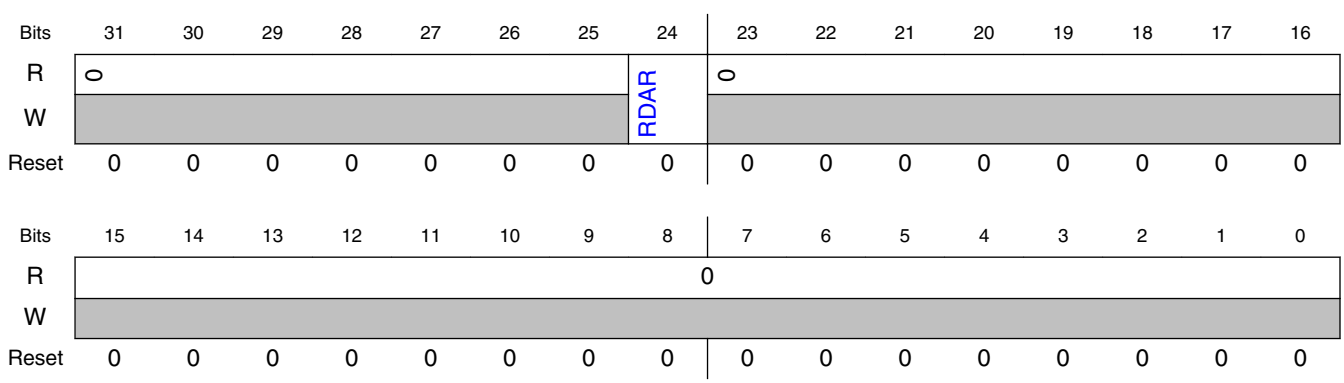
60.5.1.4.1 Offset

Register	Offset
RDAR	10h

60.5.1.4.2 Function

RDAR is a command register, written by the user, to indicate that the receive descriptor ring 0 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

60.5.1.4.3 Diagram



60.5.1.4.4 Fields

Field	Function
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.5 Transmit Descriptor Active Register - Ring 0 (TDAR)

60.5.1.5.1 Offset

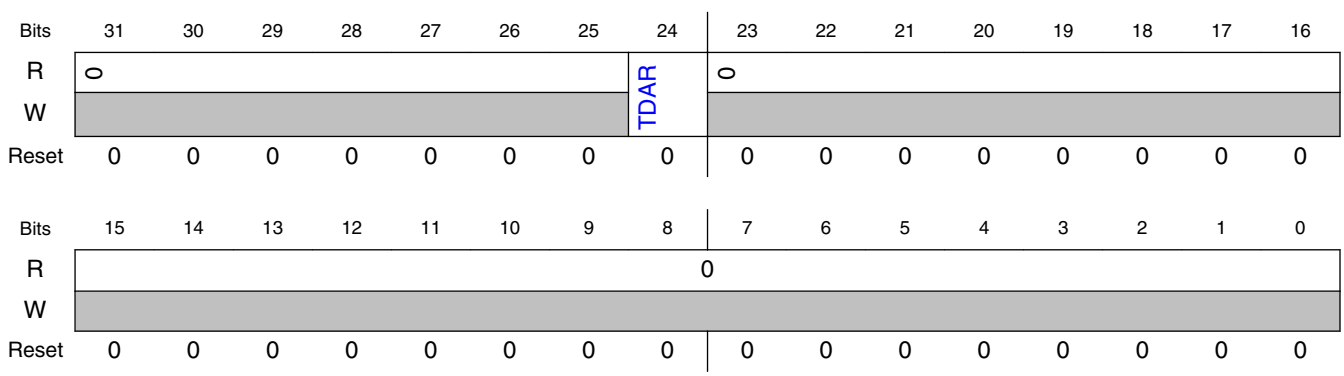
Register	Offset
TDAR	14h

60.5.1.5.2 Function

The TDAR is a command register that the user writes to indicate that the transmit descriptor ring 0 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

60.5.1.5.3 Diagram



60.5.1.5.4 Fields

Field	Function
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.6 Ethernet Control Register (ECR)

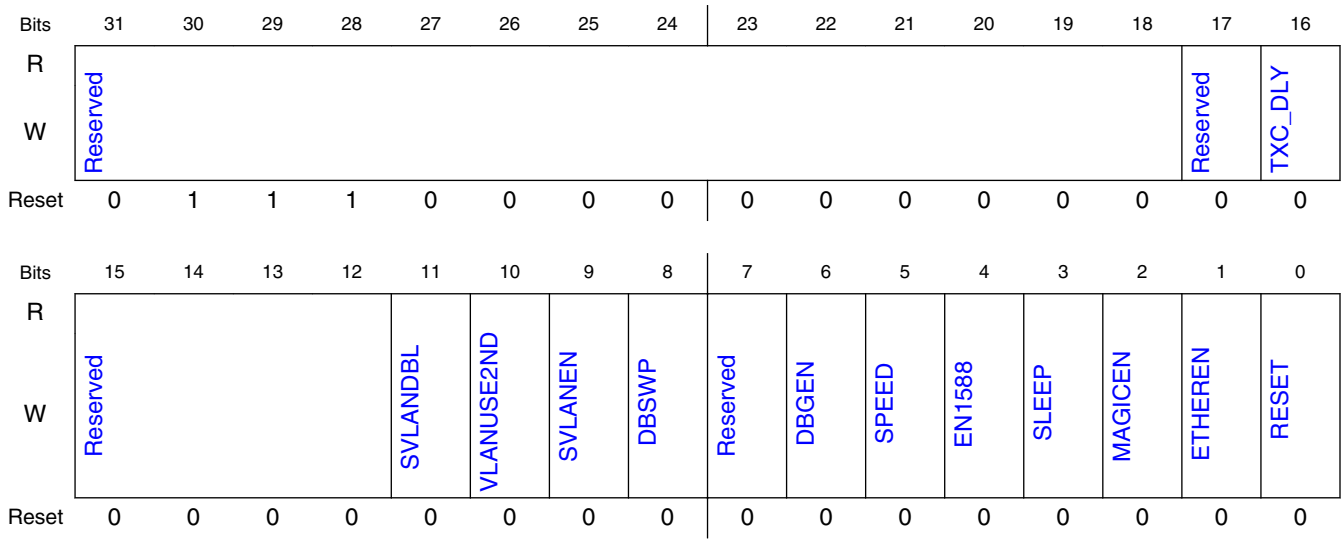
60.5.1.6.1 Offset

Register	Offset
ECR	24h

60.5.1.6.2 Function

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

60.5.1.6.3 Diagram



60.5.1.6.4 Fields

Field	Function
31-18 —	Always write 01110000000000b to this field.
17 —	Always write 0 to this field.
16 TXC_DLY	Transmit clock delay This field allows you to generate a delayed version of the RGMII_TXC clock. **NOTE:** This field only support RGMII - 100/10M mode. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. 0b - RGMII_TXC is not delayed. 1b - Generate delayed version of RGMII_TXC.
15-12 —	Always write 0 to this field.
11 SVLANDBL	S-VLAN double tag If enabled, S-VLAN detection requires a double-tagged frame to define a frame as being a VLAN frame. The following rules apply: • If the first tag is the S-VLAN type, it must be followed by a second tag with the C-VLAN type to declare the frame as VLAN frame.

Instance	Field supported in	Field not supported in
ENET	—	ECR
ENET_1G	ECR	—

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
• If the first tag is the S-VLAN type but no 2nd tag follows, it is not considered a VLAN frame but instead treated as an untagged frame.
• If the first tag is the C-VLAN type, it is considered a VLAN frame as normal.

NOTE: VLANUSE2ND can be used to determine from which tag the data should be extracted. This applies only if SVLAN_EN = 1, ignored otherwise.
NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET_1G</td><td>ECR</td><td>—</td></tr>
</table>

0b - Disable S-VLAN double tag
1b - Enable S-VLAN double tag
</td></tr>
<tr><td>10
VLANUSE2ND</td><td>
VLAN use second tag

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET_1G</td><td>ECR</td><td>—</td></tr>
</table>

0b - Always extract data from the first VLAN tag if it exists.
1b - When a double-tagged frame is detected, the data of the second tag is extracted for further processing. A double-tagged frame is defined as: The first tag can be a C-VLAN or a S-VLAN (if SVLAN_ENA = 1) The second tag must be a C-VLAN
</td></tr>
<tr><td>9
SVLANEN</td><td>
S-VLAN enable

Enable additional detection of S-VLAN tag according to IEEE802.1Q.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET_1G</td><td>ECR</td><td>—</td></tr>
</table>

0b - Only the EtherType 0x8100 will be considered for VLAN detection.
1b - The EtherType 0x88a8 will be considered in addition to 0x8100 (C-VLAN) to identify a VLAN frame in receive. When a VLAN frame is identified, the two bytes following the VLAN type are extracted and used by the classification match comparators, RCMRn.
</td></tr>
<tr><td>8
DBSWP</td><td>
Descriptor Byte Swapping Enable

Swaps the byte locations of the buffer descriptors.

NOTE: This field must be written to 1 after reset.
0b - The buffer descriptor bytes are not swapped to support big-endian devices.
1b - The buffer descriptor bytes are swapped to support little-endian devices.
</td></tr>
</table>

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>7
—</td><td>Always write 0 to this field.</td></tr>
<tr><td>6
DBGEN</td><td>Debug Enable
Enables the MAC to enter hardware freeze mode when the device enters debug mode.
0b - MAC continues operation in debug mode.
1b - MAC enters hardware freeze mode when the processor is in debug mode.</td></tr>
<tr><td>5
SPEED</td><td>Selects between 10/100-Mbit/s and 1000-Mbit/s modes of operation.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>ECR</td></tr>
<tr><td>ENET_1G</td><td>ECR</td><td>—</td></tr>
</table>
0b - 10/100-Mbit/s mode
1b - 1000-Mbit/s mode</td></tr>
<tr><td>4
EN1588</td><td>EN1588 Enable
Enables enhanced functionality of the MAC.
0b - Legacy FEC buffer descriptors and functions enabled.
1b - Enhanced frame time-stamping functions enabled. Has no effect within the MAC besides controlling the DMA control bit ena_1588.</td></tr>
<tr><td>3
SLEEP</td><td>Sleep Mode Enable
0b - Normal operating mode.
1b - Sleep mode.</td></tr>
<tr><td>2
MAGICEN</td><td>Magic Packet Detection Enable
Enables/disables magic packet detection.
NOTE: MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection.
NOTE: EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode.
0b - Magic detection logic disabled.
1b - The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.</td></tr>
<tr><td>1
ETHEREN</td><td>Ethernet Enable
Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
Hardware clears this field under the following conditions:
• RESET is set by software
• An error condition causes the EBERR field to set.
NOTE:
• ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only after all other ENET-related registers have been configured.
• If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts.
0b - Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame.</td></tr>
</table>

Table continues on the next page...

Field	Function
	1b - MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

60.5.1.7 MII Management Frame Register (MMFR)

60.5.1.7.1 Offset

Register	Offset
MMFR	40h

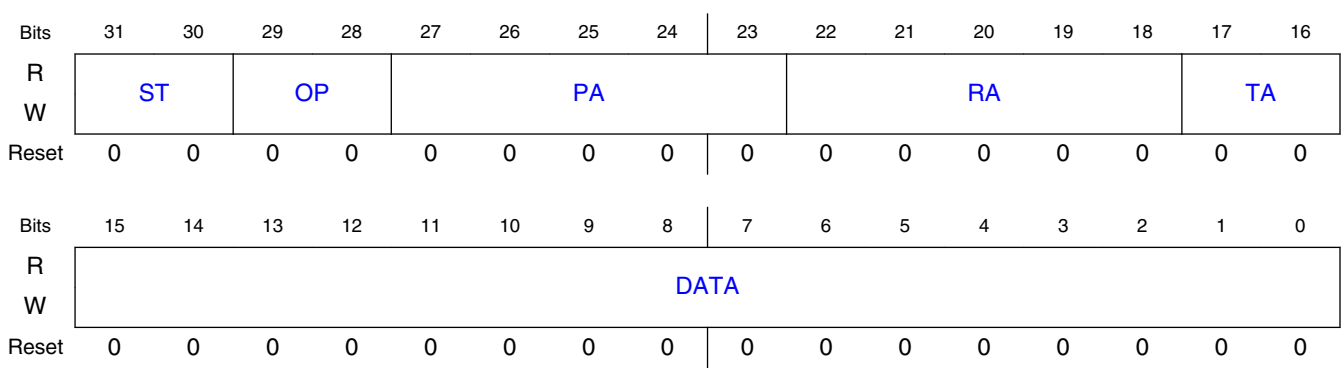
60.5.1.7.2 Function

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

60.5.1.7.3 Diagram



60.5.1.7.4 Fields

Field	Function
31-30 ST	Start Of Frame Delimiter See Table 60-39 (Clause 22) or Table 60-41 (Clause 45) for correct value.
29-28 OP	Operation Code See Table 60-39 (Clause 22) or Table 60-41 (Clause 45) for correct value.
27-23 PA	PHY Address See Table 60-39 (Clause 22) or Table 60-41 (Clause 45) for correct value.
22-18 RA	Register Address See Table 60-39 (Clause 22) or Table 60-41 (Clause 45) for correct value.
17-16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
15-0 DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

60.5.1.8 MII Speed Control Register (MSCR)

60.5.1.8.1 Offset

Register	Offset
MSCR	44h

60.5.1.8.2 Function

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

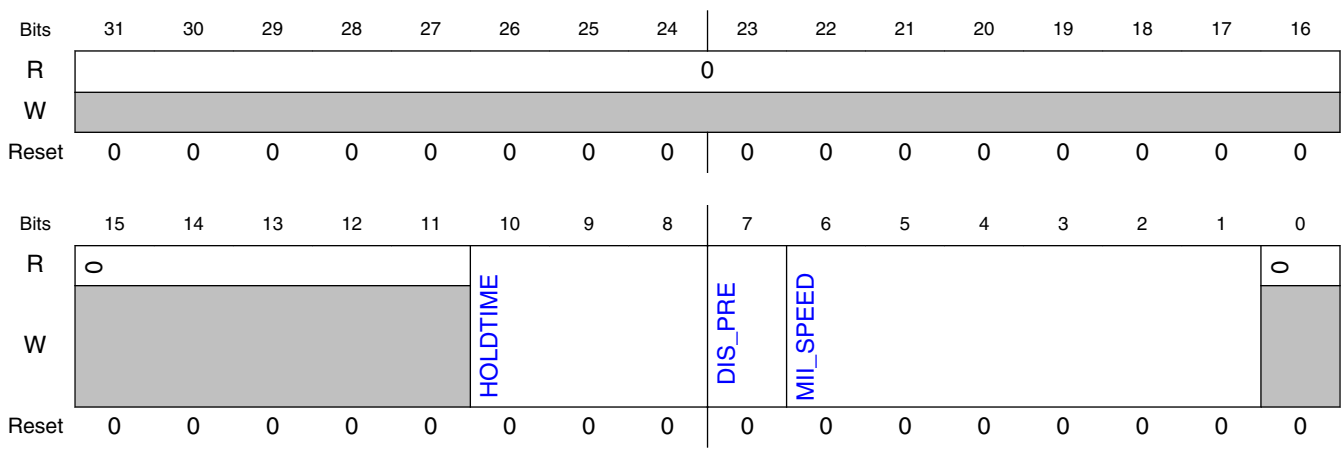
MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

Table 60-47. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

60.5.1.8.3 Diagram



60.5.1.8.4 Fields

Field	Function
31-11 —	Reserved
10-8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000b - 1 internal module clock cycle 001b - 2 internal module clock cycles 010b - 3 internal module clock cycles 111b - 8 internal module clock cycles
7 DIS_PRE	Disable Preamble

Table continues on the next page...

Field	Function
	Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0b - Preamble enabled. 1b - Preamble (32 ones) is not prepended to the MII management frame.
6-1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 —	Reserved

60.5.1.9 MIB Control Register (MIBC)

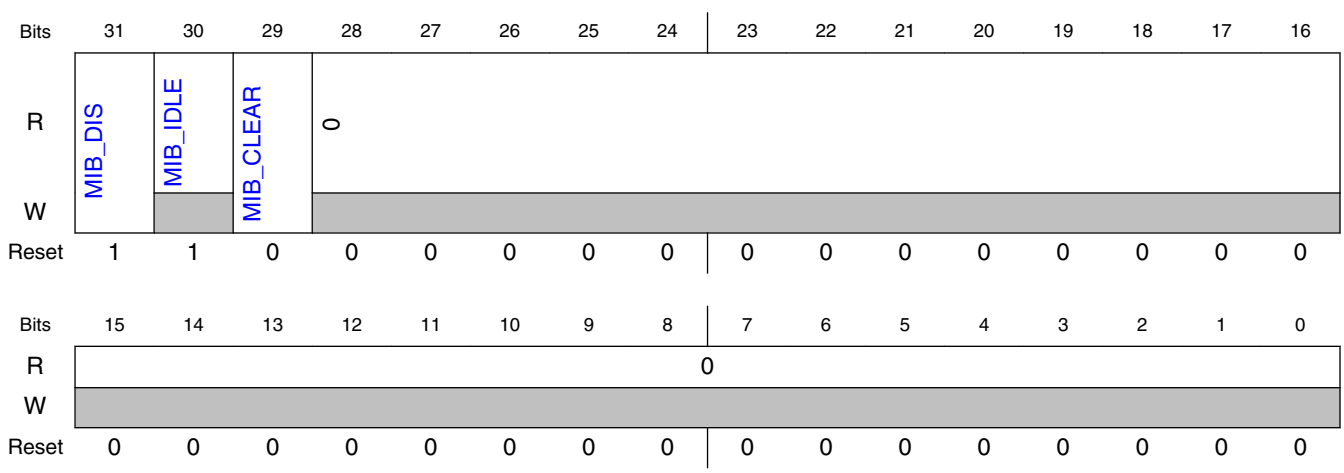
60.5.1.9.1 Offset

Register	Offset
MIBC	64h

60.5.1.9.2 Function

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

60.5.1.9.3 Diagram



60.5.1.9.4 Fields

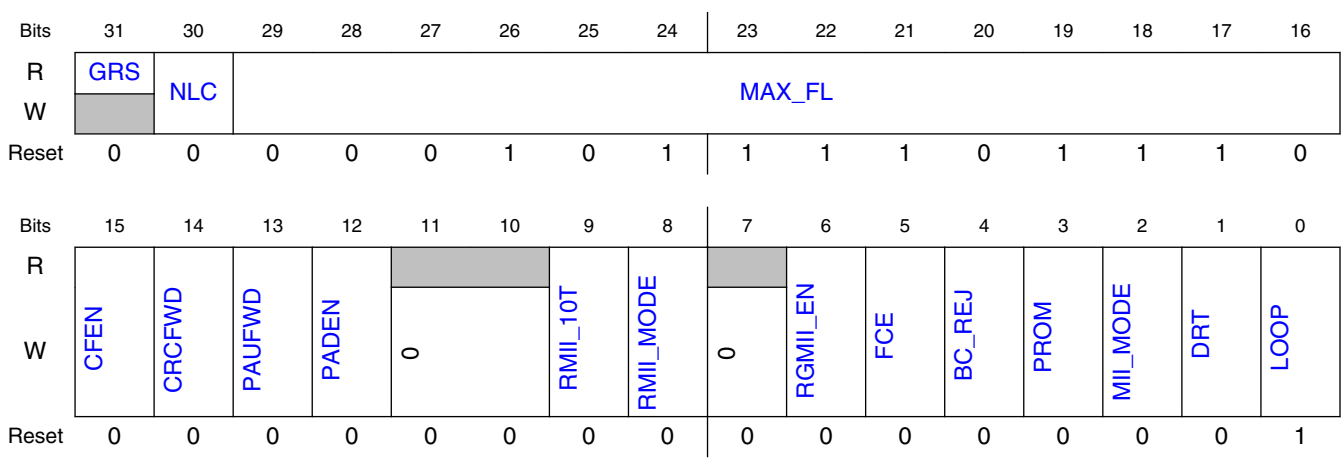
Field	Function
31 MIB_DIS	Disable MIB Logic If this control field is set, 0b - MIB logic is enabled. 1b - MIB logic is disabled. The MIB logic halts and does not update any MIB counters.
30 MIB_IDLE	MIB Idle 0b - The MIB block is updating MIB counters. 1b - The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0b - See note above. 1b - All statistics counters are reset to 0.
28-0 —	Reserved

60.5.1.10 Receive Control Register (RCR)

60.5.1.10.1 Offset

Register	Offset
RCR	84h

60.5.1.10.2 Diagram



60.5.1.10.3 Fields

Field	Function
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped. 0b - Receive not stopped 1b - Receive stopped
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0b - The payload length check is disabled. 1b - The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29-16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0b - MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1b - MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0b - The CRC field of received frames is transmitted to the user application. 1b - The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0b - Pause frames are terminated and discarded in the MAC. 1b - Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0b - No padding is removed on receive by the MAC. 1b - Padding is removed from received frames.
11-10 —	This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII or RGMII . **NOTE:** The descriptions of the field settings vary by module instance.

Instance	Field value and description
ENET	0b - 100-Mbit/s operation. 1b - 10-Mbit/s operation.
ENET_1G	0b - 100-Mbit/s or 1-Gbit/s operation. 1b - 10-Mbit/s operation.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>8
RMII_MODE</td><td>RMII Mode Enable
Specifies whether the MAC is configured for MII mode or RMII operation , when ECR[SPEED] is cleared .
NOTE: Do not set both RCR[RGMII_EN] and RCR[RMII_MODE].
0b - MAC configured for MII mode.
1b - MAC configured for RMII operation.</td></tr>
<tr><td>7
—</td><td>This write-only field is reserved. It must always be written with the value 0.</td></tr>
<tr><td>6
RGMII_EN</td><td>RGMII Mode Enable
NOTE: Do not set both RCR[RGMII_EN] and RCR[RMII_MODE].
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>ENET</td><td>—</td><td>RCR</td></tr>
<tr><td>ENET_1G</td><td>RCR</td><td>—</td></tr>
</table>
0b - MAC configured for non-RGMII operation
1b - MAC configured for RGMII operation. If ECR[SPEED] is set, the MAC is in RGMII 1000-Mbit/s mode. If ECR[SPEED] is cleared, the MAC is in RGMII 10/100-Mbit/s mode.</td></tr>
<tr><td>5
FCE</td><td>Flow Control Enable
If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.
0b - Disable flow control
1b - Enable flow control</td></tr>
<tr><td>4
BC_REJ</td><td>Broadcast Frame Reject
If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.
0b - Will not reject frames as described above
1b - Will reject frames as described above</td></tr>
<tr><td>3
PROM</td><td>Promiscuous Mode
All frames are accepted regardless of address matching.
0b - Disabled.
1b - Enabled.</td></tr>
<tr><td>2
MII_MODE</td><td>Media Independent Interface Mode
This field must always be set.
0b - Reserved.
1b - MII or RMII mode, as indicated by the RMII_MODE field.</td></tr>
<tr><td>1
DRT</td><td>Disable Receive On Transmit
0b - Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode.
1b - Disable reception of frames while transmitting. (Normally used for half-duplex mode.)</td></tr>
<tr><td>0
LOOP</td><td>Internal Loopback
This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0.
0b - Loopback disabled.</td></tr>
</table>

Field	Function
	1b - Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

60.5.1.11 Transmit Control Register (TCR)

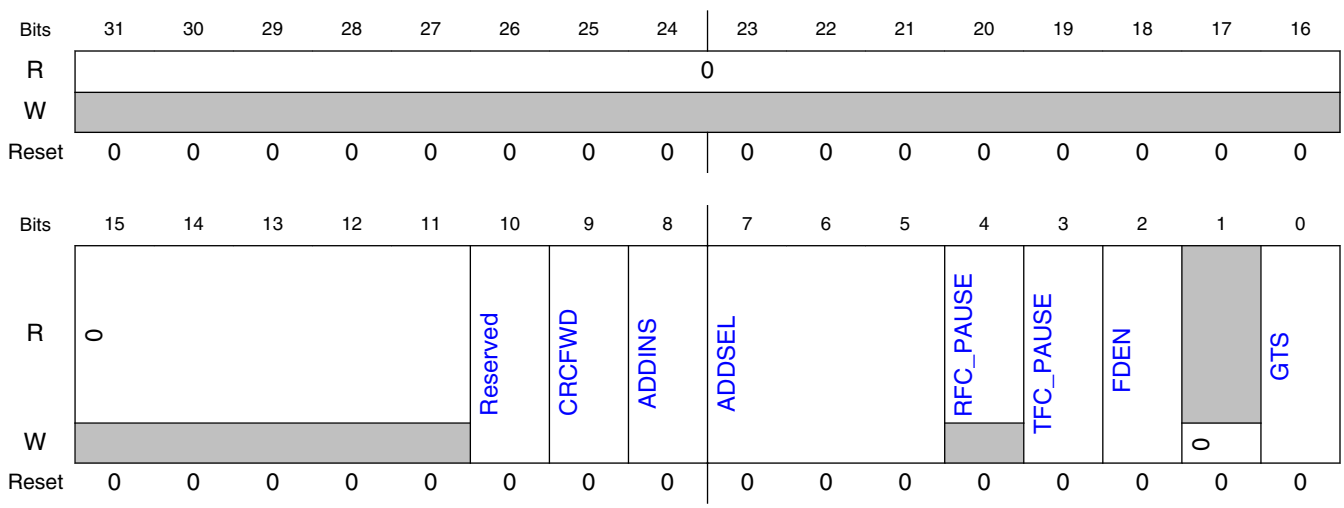
60.5.1.11.1 Offset

Register	Offset
TCR	C4h

60.5.1.11.2 Function

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

60.5.1.11.3 Diagram



60.5.1.11.4 Fields

Field	Function
31-11 —	Reserved
10	This field is read/write and must be set to 0.

Table continues on the next page...

Field	Function
—	
9 CRCFWD	Forward Frame From Application With CRC 0b - TxBD[TC] controls whether the frame has a CRC from the application. 1b - The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0b - The source MAC address is not modified by the MAC. 1b - The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7-5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000b - Node MAC address programmed on PADDR1/2 registers. 100b - Reserved. 101b - Reserved. 110b - Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0b - No PAUSE frame transmitted. 1b - The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared. 0b - Disable full-duplex 1b - Enable full-duplex
1 —	This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt. 0b - Disable graceful transmit stop 1b - Enable graceful transmit stop

60.5.1.12 Physical Address Lower Register (PALR)

60.5.1.12.1 Offset

Register	Offset
PALR	E4h

60.5.1.12.2 Function

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

60.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.12.4 Fields

Field	Function
31-0 PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

60.5.1.13 Physical Address Upper Register (PAUR)

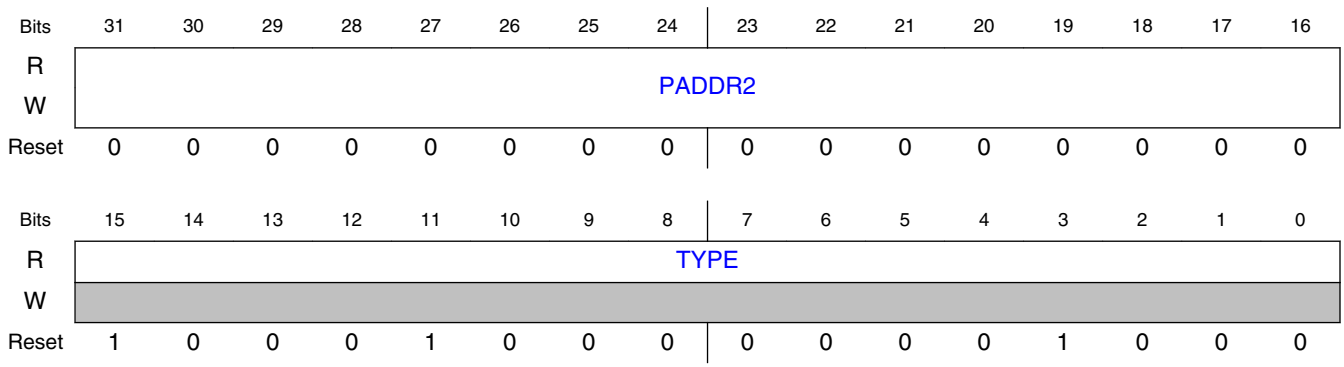
60.5.1.13.1 Offset

Register	Offset
PAUR	E8h

60.5.1.13.2 Function

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

60.5.1.13.3 Diagram



60.5.1.13.4 Fields

Field	Function
31-16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
15-0 TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

60.5.1.14 Opcode/Pause Duration Register (OPD)

60.5.1.14.1 Offset

Register	Offset
OPD	ECh

60.5.1.14.2 Function

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

60.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OPCODE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAUSE_DUR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.14.4 Fields

Field	Function
31-16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
15-0 PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

60.5.1.15 Transmit Interrupt Coalescing Register (TXIC0 - TXIC2)

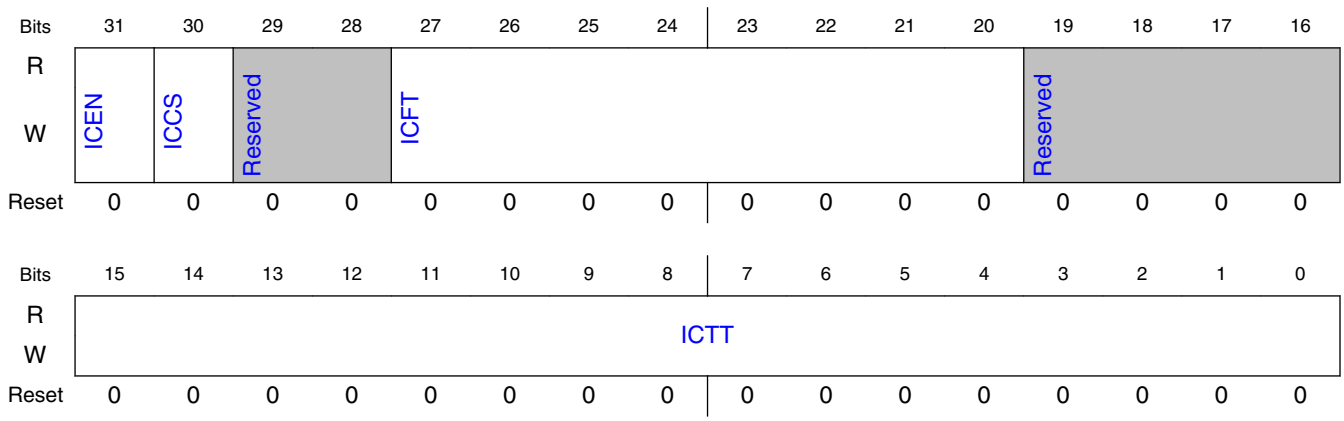
60.5.1.15.1 Offset

Register	Offset
TXIC0	F0h
TXIC1	F4h
TXIC2	F8h

60.5.1.15.2 Function

See Interrupt coalescence for more information.

60.5.1.15.3 Diagram



60.5.1.15.4 Fields

Field	Function
31 ICEN	Interrupt Coalescing Enable 0b - Disable Interrupt coalescing. 1b - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0b - Use MII/GMII TX clocks. 1b - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be transmitted for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 TX clock periods. This value determines the maximum amount of time after transmitting a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame transmission defined by ICFT and starts again upon transmission of the next first frame. Must be greater than zero to avoid unpredictable behavior.

60.5.1.16 Receive Interrupt Coalescing Register (RXIC0 - RXIC2)

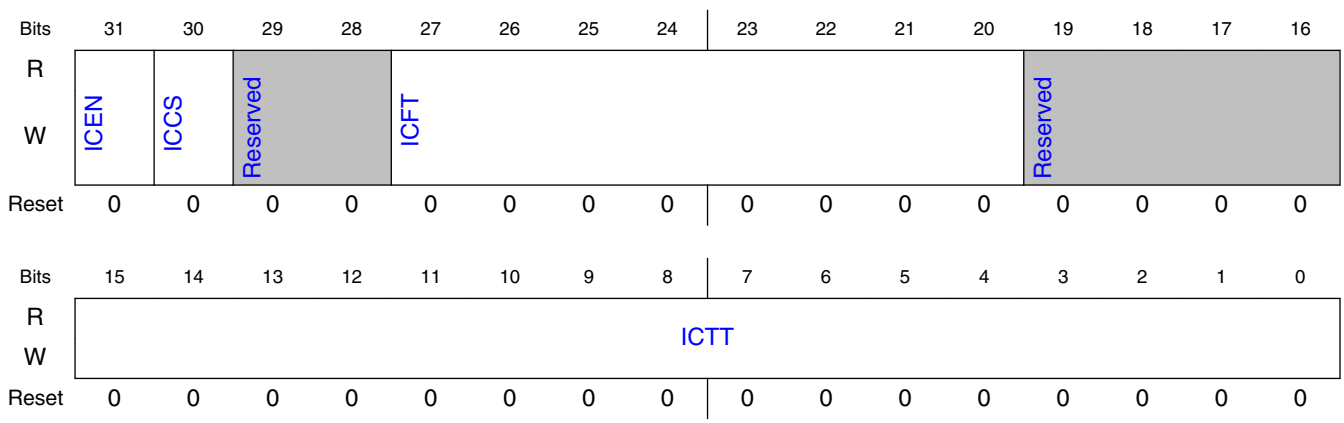
60.5.1.16.1 Offset

Register	Offset
RXIC0	100h
RXIC1	104h
RXIC2	108h

60.5.1.16.2 Function

See Interrupt coalescence for more information.

60.5.1.16.3 Diagram



60.5.1.16.4 Fields

Field	Function
31 ICEN	Interrupt Coalescing Enable 0b - Disable Interrupt coalescing. 1b - Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0b - Use MII/GMII TX clocks. 1b - Use ENET system clock.
29-28 —	This field must be set to 0.
27-20 ICFT	Interrupt coalescing frame count threshold

Table continues on the next page...

Field	Function
	This value determines the number of frames needed to be received for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19-16 —	This field must be set to 0.
15-0 ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 RX clock periods. This value determines the maximum amount of time after receiving a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame reception defined by ICFT and starts again upon reception of the next first frame. Must be greater than zero to avoid unpredictable behavior.

60.5.1.17 Descriptor Individual Upper Address Register (IAUR)

60.5.1.17.1 Offset

Register	Offset
IAUR	118h

60.5.1.17.2 Function

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

60.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.17.4 Fields

Field	Function
31-0 IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

60.5.1.18 Descriptor Individual Lower Address Register (IALR)

60.5.1.18.1 Offset

Register	Offset
IALR	11Ch

60.5.1.18.2 Function

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

60.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.18.4 Fields

Field	Function
31-0 IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

60.5.1.19 Descriptor Group Upper Address Register (GAUR)

60.5.1.19.1 Offset

Register	Offset
GAUR	120h

60.5.1.19.2 Function

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

60.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.19.4 Fields

Field	Function
31-0 GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

60.5.1.20 Descriptor Group Lower Address Register (GALR)

60.5.1.20.1 Offset

Register	Offset
GALR	124h

60.5.1.20.2 Function

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

60.5.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.20.4 Fields

Field	Function
31-0 GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

60.5.1.21 Transmit FIFO Watermark Register (TFWR)

60.5.1.21.1 Offset

Register	Offset
TFWR	144h

60.5.1.21.2 Function

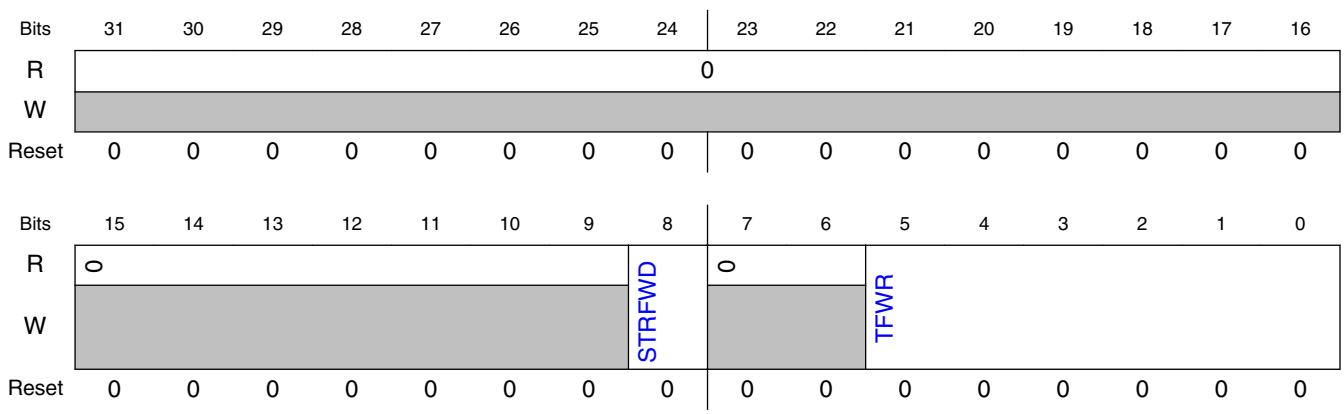
If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to '0', the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to '1'. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

60.5.1.21.3 Diagram



60.5.1.21.4 Fields

Field	Function
31-9 —	Reserved
8 STRFWD	Store And Forward Enable 0b - Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1b - Enabled.

Table continues on the next page...

Field	Function
7-6 —	Reserved
5-0 TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 00_0000b - 64 bytes written. 00_0001b - 64 bytes written. 00_0010b - 128 bytes written. 00_0011b - 192 bytes written. 01_1111b - 1984 bytes written. 11_1111b - 4032 bytes written.

60.5.1.22 Receive Descriptor Ring 1 Start Register (RDSR1)

60.5.1.22.1 Offset

Register	Offset
RDSR1	160h

60.5.1.22.2 Function

RDSR1 points to the beginning of circular receive buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDSR1
ENET_1G	RDSR1	—

60.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.22.4 Fields

Field	Function
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.23 Transmit Buffer Descriptor Ring 1 Start Register (TDSR1)

60.5.1.23.1 Offset

Register	Offset
TDSR1	164h

60.5.1.23.2 Function

TDSR1 provides a pointer to the beginning of the circular transmit buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDSR1
ENET_1G	TDSR1	—

60.5.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.23.4 Fields

Field	Function
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 1.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.24 Maximum Receive Buffer Size Register - Ring 1 (MRBR1)

60.5.1.24.1 Offset

Register	Offset
MRBR1	168h

60.5.1.24.2 Function

MRBR1 is a user-programmable register that dictates the maximum size of all ring-1 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR1 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR1 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_ZIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR1 greater than or equal to 256 bytes.

NOTE

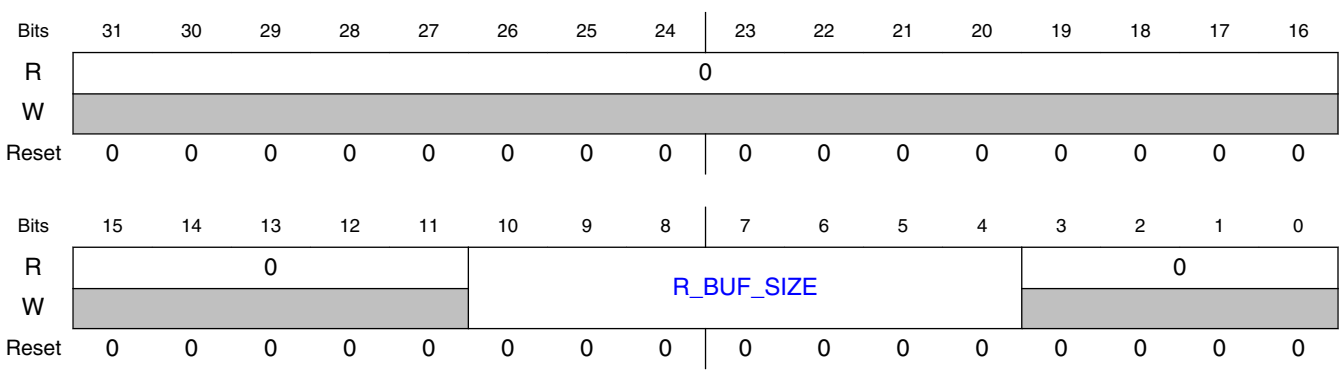
This register must be initialized before operation.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	MRBR1
ENET_1G	MRBR1	—

60.5.1.24.3 Diagram



60.5.1.24.4 Fields

Field	Function
31-11 —	Reserved
10-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

60.5.1.25 Receive Descriptor Ring 2 Start Register (RDSR2)

60.5.1.25.1 Offset

Register	Offset
RDSR2	16Ch

60.5.1.25.2 Function

RDSR points to the beginning of circular receive buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDSR2
ENET_1G	RDSR2	—

60.5.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.25.4 Fields

Field	Function
31-3 R_DES_START	Pointer to the beginning of receive buffer descriptor queue 2.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.26 Transmit Buffer Descriptor Ring 2 Start Register (TDSR2)

60.5.1.26.1 Offset

Register	Offset
TDSR2	170h

60.5.1.26.2 Function

TDSR2 provides a pointer to the beginning of circular transmit buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

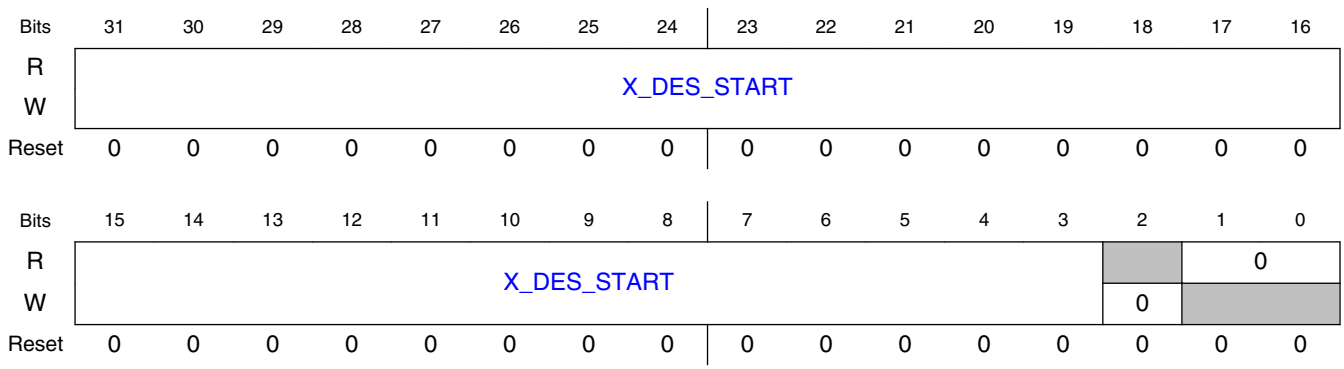
This register must be initialized prior to operation

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDSR2
ENET_1G	TDSR2	—

60.5.1.26.3 Diagram



60.5.1.26.4 Fields

Field	Function
31-3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 2.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.27 Maximum Receive Buffer Size Register - Ring 2 (MRBR2)

60.5.1.27.1 Offset

Register	Offset
MRBR2	174h

60.5.1.27.2 Function

MRBR2 is a user-programmable register that dictates the maximum size of all ring-2 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR2 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR2 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR2 greater than or equal to 256 bytes.

NOTE

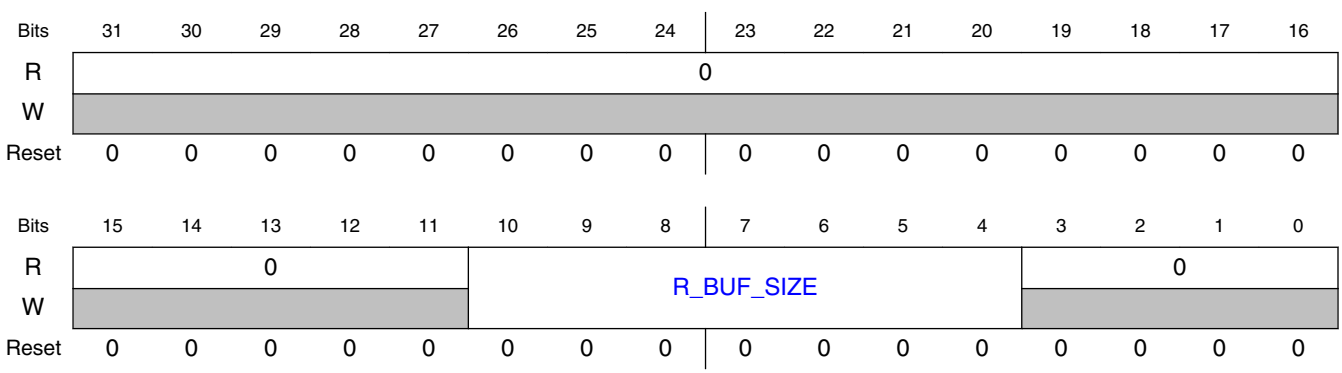
This register must be initialized prior to operation

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	MRBR2
ENET_1G	MRBR2	—

60.5.1.27.3 Diagram



60.5.1.27.4 Fields

Field	Function
31-11 —	Reserved
10-4 R_BUF_SIZE	Receive buffer size (in bytes)
3-0 —	Reserved

60.5.1.28 Receive Descriptor Ring 0 Start Register (RDSR)

60.5.1.28.1 Offset

Register	Offset
RDSR	180h

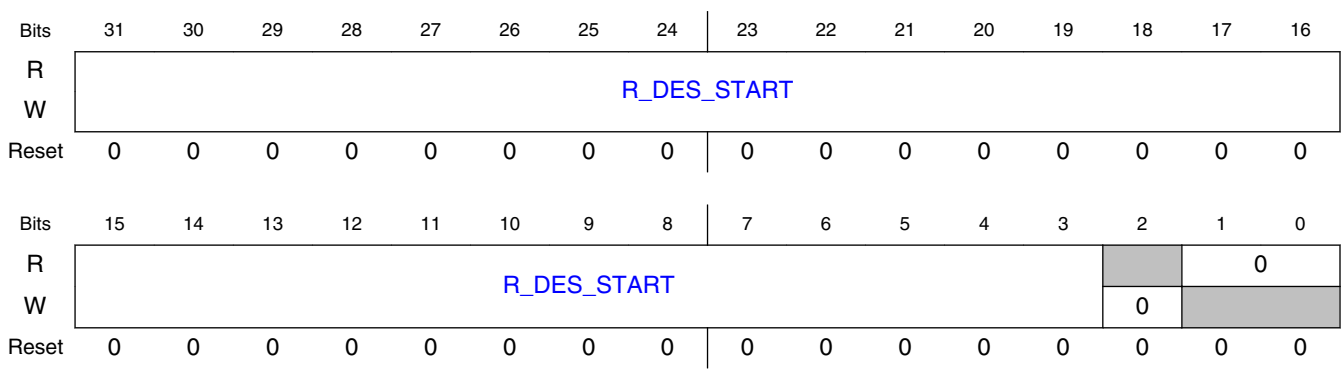
60.5.1.28.2 Function

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

60.5.1.28.3 Diagram



60.5.1.28.4 Fields

Field	Function
31-3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue. 0
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.29 Transmit Buffer Descriptor Ring 0 Start Register (TDSR)

60.5.1.29.1 Offset

Register	Offset
TDSR	184h

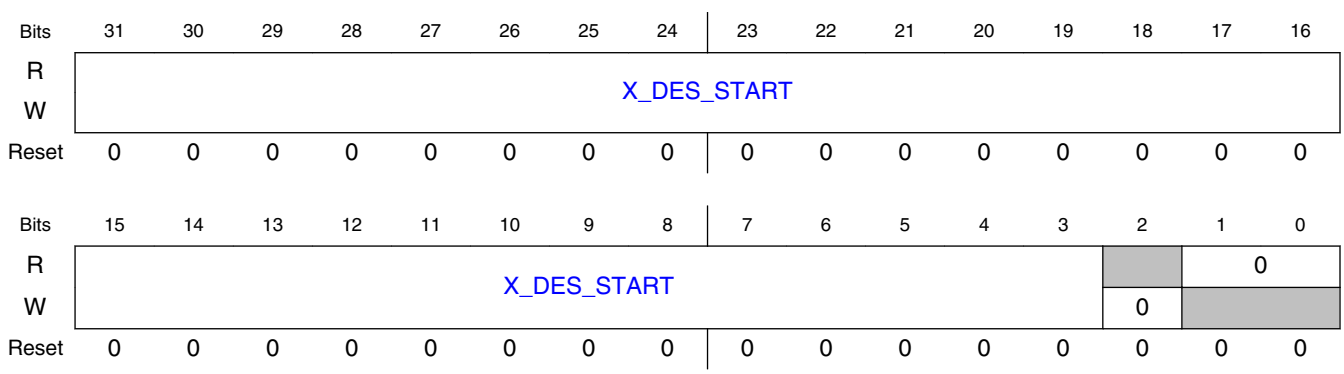
60.5.1.29.2 Function

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue 0 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

60.5.1.29.3 Diagram



60.5.1.29.4 Fields

Field	Function
31-3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2 —	This write-only field is reserved. It must always be written with the value 0.
1-0 —	Reserved

60.5.1.30 Maximum Receive Buffer Size Register - Ring 0 (MRBR)

60.5.1.30.1 Offset

Register	Offset
MRBR	188h

60.5.1.30.2 Function

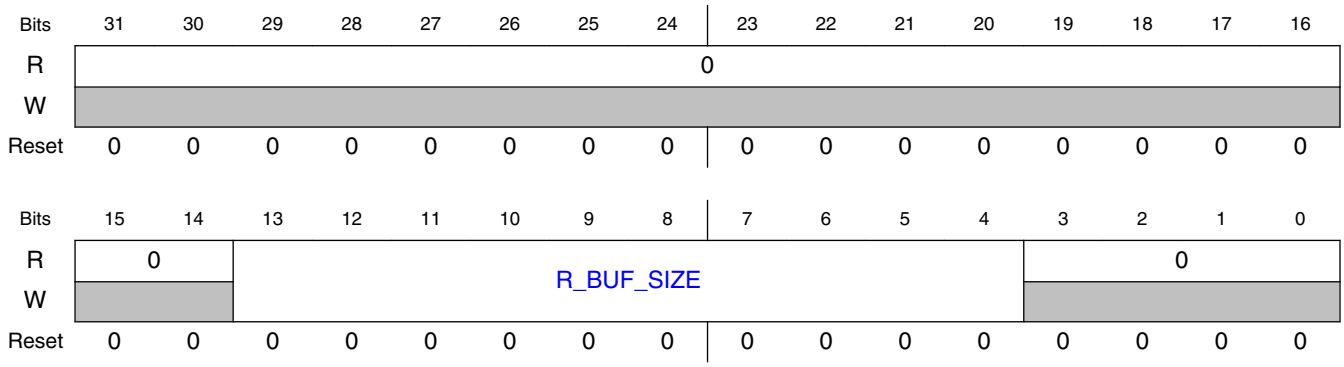
The MRBR is a user-programmable register that dictates the maximum size of all ring-0 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- R_BUF_SIZE is concatenated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits of this register are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

60.5.1.30.3 Diagram



60.5.1.30.4 Fields

Field	Function
31-14 —	Reserved
13-4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.
3-0 —	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size.

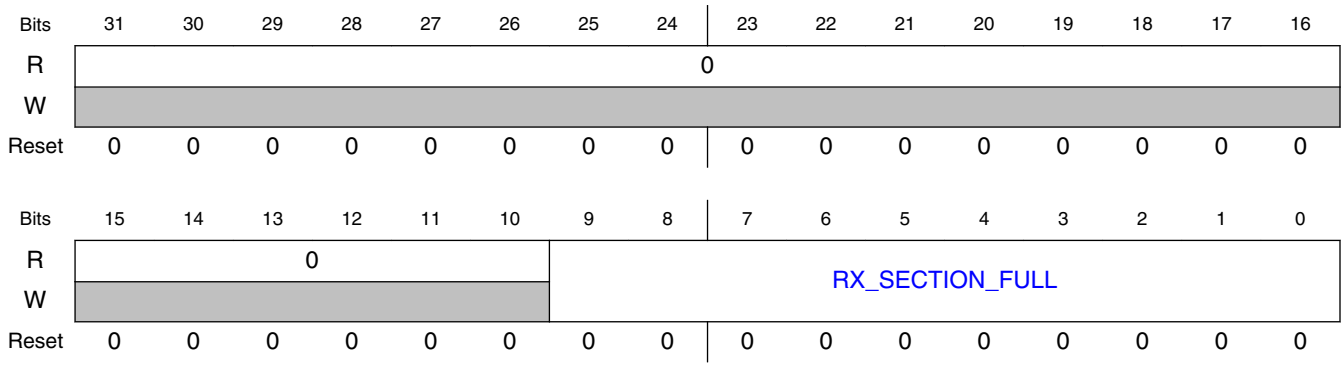
Instance	Field supported in	Field not supported in
ENET	MRBR	—
ENET_1G	MRBR[10–4]	MRBR[13–11]

60.5.1.31 Receive FIFO Section Full Threshold (RSFL)

60.5.1.31.1 Offset

Register	Offset
RSFL	190h

60.5.1.31.2 Diagram



60.5.1.31.3 Fields

Field	Function
31-10 —	Reserved
9-0 RX_SECTION_FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation). **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

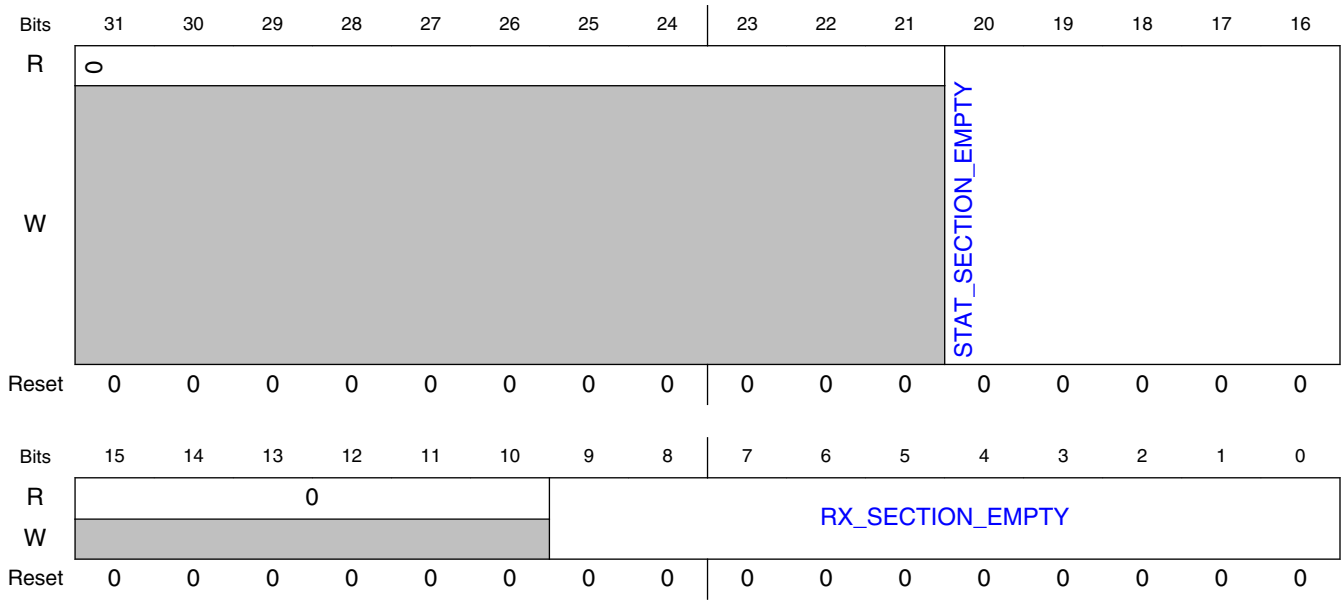
Instance	Field supported in	Field not supported in
ENET	RSFL[7–0]	RSFL[9–8]
ENET_1G	RSFL	—

60.5.1.32 Receive FIFO Section Empty Threshold (RSEM)

60.5.1.32.1 Offset

Register	Offset
RSEM	194h

60.5.1.32.2 Diagram



60.5.1.32.3 Fields

Field	Function
31-21 —	Reserved
20-16 STAT_SECTION_EMPTY	RX Status FIFO Section Empty Threshold Defines number of frames in the receive FIFO, independent of its size, that can be accepted. If the limit is reached, reception will continue normally, however a pause frame will be triggered to indicate a possible congestion to the remote device to avoid FIFO overflow. A value of 0 disables automatic pause frame generation
15-10 —	Reserved
9-0 RX_SECTION_EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client. **NOTE:** The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
ENET	RSEM[7–0]	RSEM[9–8]

Field	Function		
	Instance	Field supported in	Field not supported in
	ENET_1G	RSEM	—

60.5.1.33 Receive FIFO Almost Empty Threshold (RAEM)

60.5.1.33.1 Offset

Register	Offset
RAEM	198h

60.5.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						RX_ALMOST_EMPTY									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

60.5.1.33.3 Fields

Field	Function
31-10 —	Reserved
9-0 RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

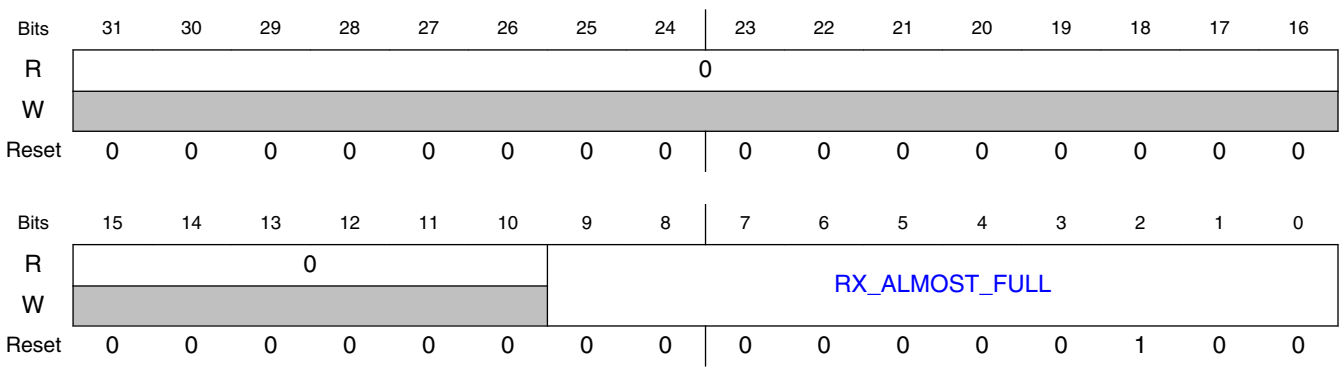
Field	Function		
	Instance	Field supported in	Field not supported in
	ENET	RAEM[7–0]	RAEM[9–8]
	ENET_1G	RAEM	—

60.5.1.34 Receive FIFO Almost Full Threshold (RAFL)

60.5.1.34.1 Offset

Register	Offset
RAFL	19Ch

60.5.1.34.2 Diagram



60.5.1.34.3 Fields

Field	Function
31-10 —	Reserved
9-0 RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

Field	Function
	NOTE: This field is not supported in every instance. The following table includes only supported registers.

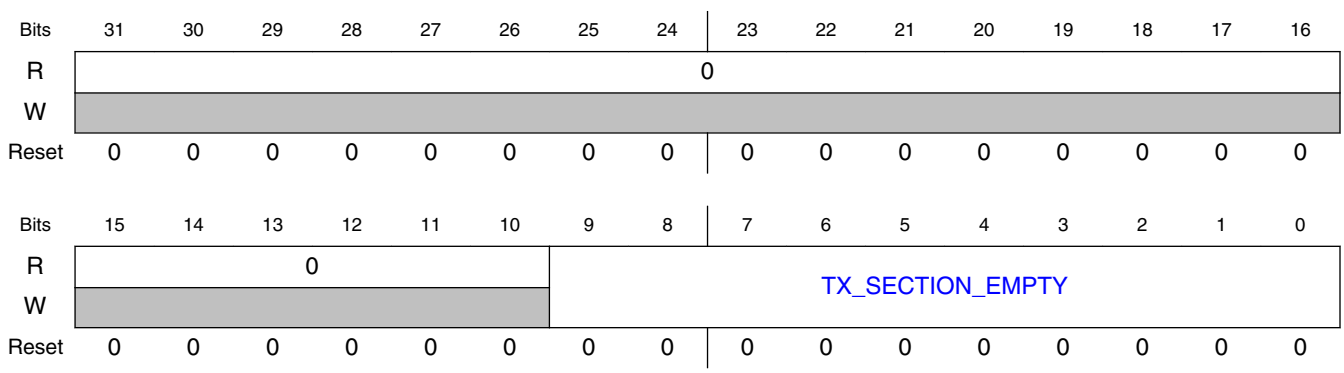
Instance	Field supported in	Field not supported in
ENET	RAFL[7–0]	RAFL[9–8]
ENET_1G	RAFL	—

60.5.1.35 Transmit FIFO Section Empty Threshold (TSEM)

60.5.1.35.1 Offset

Register	Offset
TSEM	1A0h

60.5.1.35.2 Diagram



60.5.1.35.3 Fields

Field	Function
31-10 —	Reserved
9-0 TX_SECTION_EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

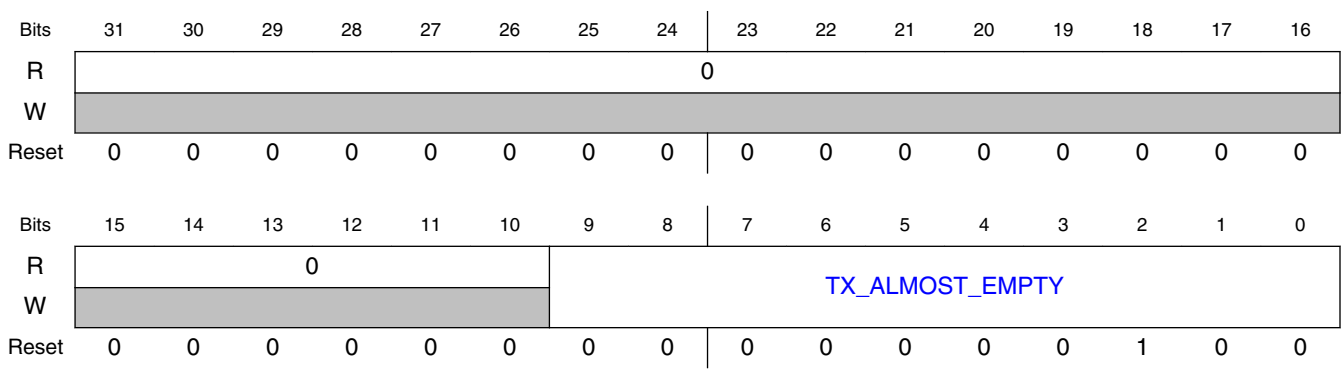
Field	Function
	NOTE: This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / TSEM[7–0] / TSEM[9–8] ENET_1G / TSEM / —

60.5.1.36 Transmit FIFO Almost Empty Threshold (TAEM)

60.5.1.36.1 Offset

Register	Offset
TAEM	1A4h

60.5.1.36.2 Diagram



60.5.1.36.3 Fields

Field	Function
31-10 —	Reserved
9-0 TX_ALMOST_EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold.

Field	Function
	When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

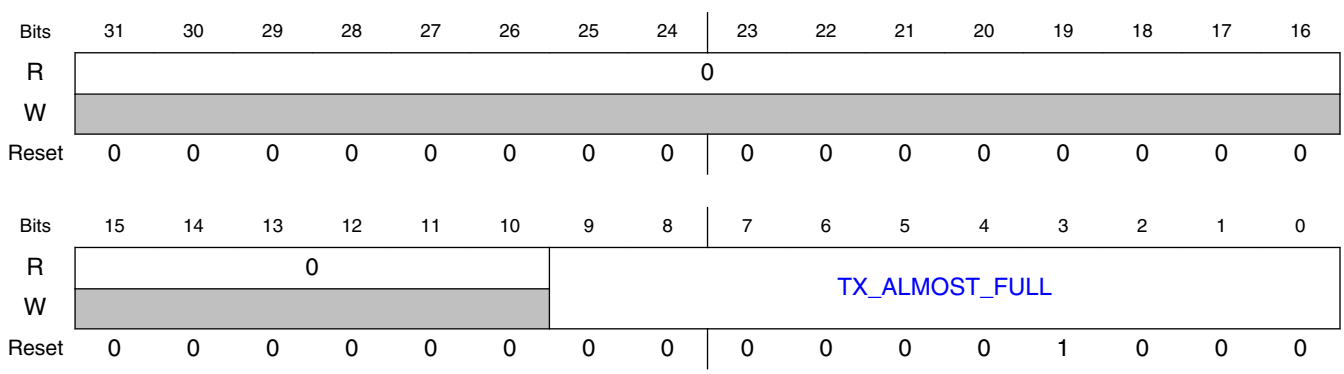
Instance	Field supported in	Field not supported in
ENET	TAEM[7–0]	TAEM[9–8]
ENET_1G	TAEM	—

60.5.1.37 Transmit FIFO Almost Full Threshold (TAFL)

60.5.1.37.1 Offset

Register	Offset
TAFL	1A8h

60.5.1.37.2 Diagram



60.5.1.37.3 Fields

Field	Function
31-10 —	Reserved

Table continues on the next page...

Field	Function
9-0 TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
ENET	TAFL[7–0]	TAFL[9–8]
ENET_1G	TAFL	—

60.5.1.38 Transmit Inter-Packet Gap (TIPG)

60.5.1.38.1 Offset

Register	Offset
TIPG	1ACh

60.5.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											IPG				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

60.5.1.38.3 Fields

Field	Function
31-5 —	Reserved
4-0 IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE:** The IPG value read will be the value that was written, even if it is out of range.

60.5.1.39 Frame Truncation Length (FTRL)

60.5.1.39.1 Offset

Register	Offset
FTRL	1B0h

60.5.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TRUNC_FL													
W																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

60.5.1.39.3 Fields

Field	Function
31-14 —	Reserved
13-0 TRUNC_FL	Frame Truncation Length Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL].

Field	Function
	NOTE: Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

60.5.1.40 Transmit Accelerator Function Configuration (TACC)

60.5.1.40.1 Offset

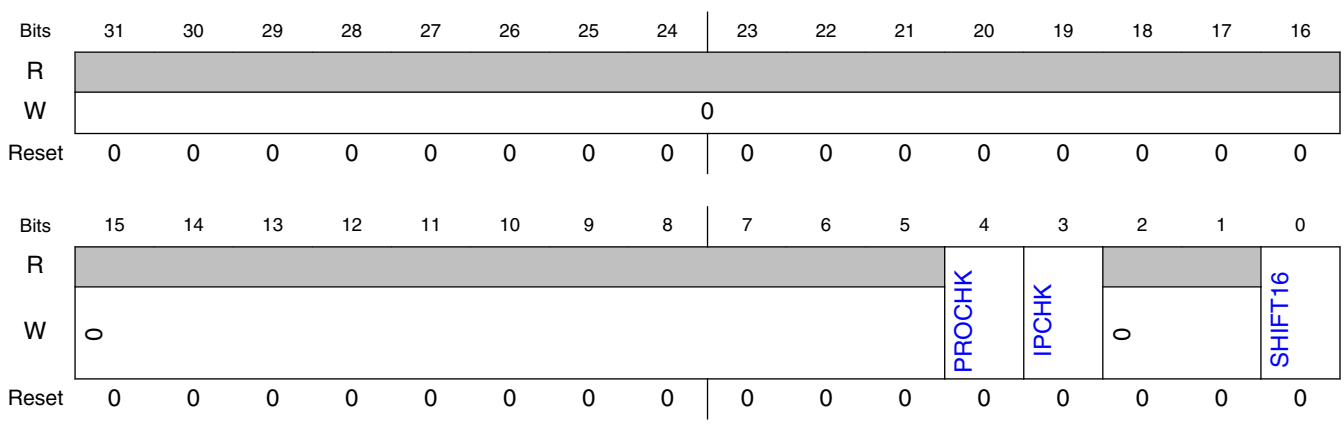
Register	Offset
TACC	1C0h

60.5.1.40.2 Function

TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

60.5.1.40.3 Diagram



60.5.1.40.4 Fields

Field	Function
31-5 —	This write-only field is reserved. It must always be written with the value 0.

Table continues on the next page...

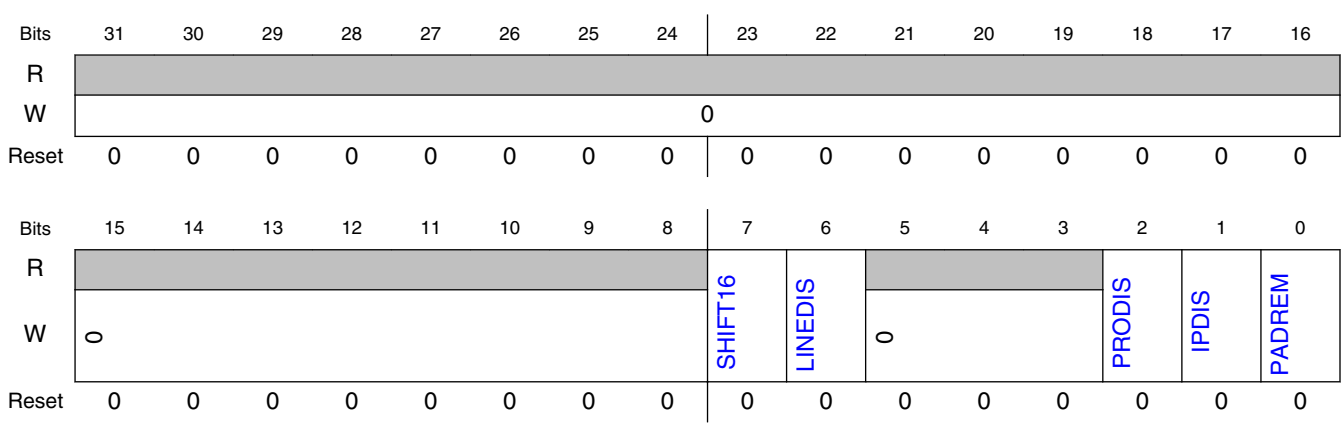
Field	Function
4 PROCHK	Enables insertion of protocol checksum. 0b - Checksum not inserted. 1b - If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0b - Checksum is not inserted. 1b - If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2-1 —	This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0b - Disabled. 1b - Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

60.5.1.41 Receive Accelerator Function Configuration (RACC)

60.5.1.41.1 Offset

Register	Offset
RACC	1C4h

60.5.1.41.2 Diagram



60.5.1.41.3 Fields

Field	Function
31-8 —	This write-only field is reserved. It must always be written with the value 0.
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0b - Disabled. 1b - Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0b - Frames with errors are not discarded. 1b - Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5-3 —	This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0b - Frames with wrong checksum are not discarded. 1b - If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0b - Frames with wrong IPv4 header checksum are not discarded. 1b - If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0b - Padding not removed. 1b - Any bytes following the IP payload section of the frame are removed from the frame.

60.5.1.42 Receive Classification Match Register for Class n (RCMR1 - RCMR2)

60.5.1.42.1 Offset

Register	Offset
RCMR1	1C8h
RCMR2	1CCh

60.5.1.42.2 Function

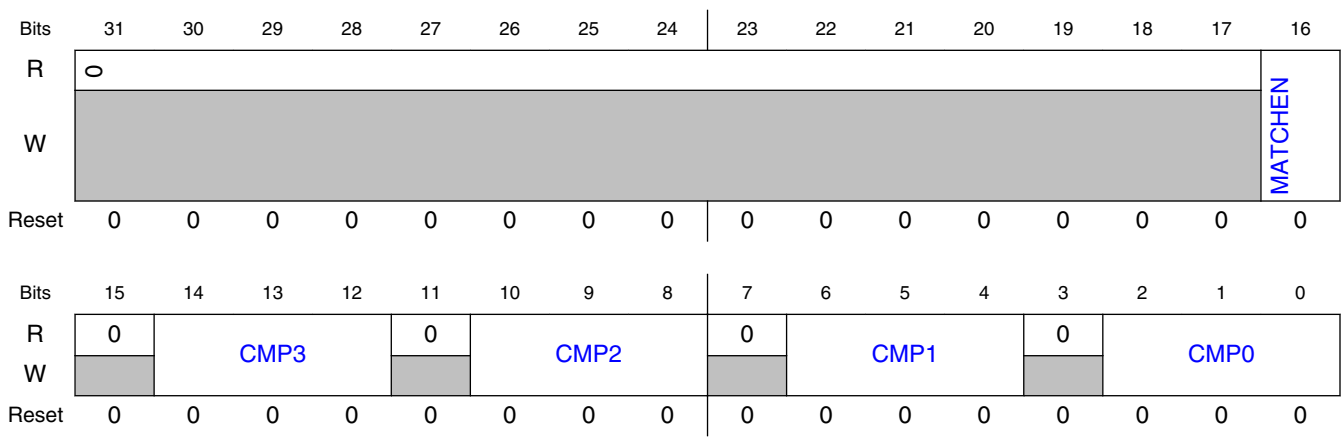
This match register allows specifying up to four priorities, which are tested (OR'ed) simultaneously. The match detection uses the extracted VLAN field according to the rules for VLAN detection configured through the ECR register. If both match registers, RCMR1 and RCMR2, report a match at the same time, only the class 1 match is indicated as the final result.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RCMR1–RCMR2
ENET_1G	RCMR1–RCMR2	—

60.5.1.42.3 Diagram



60.5.1.42.4 Fields

Field	Function
31-17 —	This write-only field is reserved. It must always be written with the value 0.
16 MATCHEN	Match Enable **NOTE:** A comparison is done only on incoming VLAN frames. If no VLAN frame is received no match will occur. If both match registers have overlapping compare values and hence can match both on the same frame, only class 1 will be indicated and the class 2 match is ignored.

Table continues on the next page...

Field	Function
	0b - Disabled (default): no compares will occur and the classification indicator for this class will never assert. 1b - The register contents are valid and a comparison with all compare values is done when a VLAN frame is received.
15 —	This write-only field is reserved. It must always be written with the value 0.
14-12 CMP3	Compare 3 Fourth value to compare against. If unused it must be set to the same value as CMP0.
11 —	This write-only field is reserved. It must always be written with the value 0.
10-8 CMP2	Compare 2 Third value to compare against. If unused it must be set to the same value as CMP0.
7 —	This write-only field is reserved. It must always be written with the value 0.
6-4 CMP1	Compare 1 Second value to compare against. If unused it must be set to the same value as CMP0.
3 —	This write-only field is reserved. It must always be written with the value 0.
2-0 CMP0	Compare 0 A three-bit value that will be compared with the frame's VLAN priority field (if a VLAN frame is received). All four compare values, CMP0..3, will be used in parallel. If any of the values match, a match for the class is reported (if MATCHEN is 1). **NOTE:** To implement a single priority match, all four compare values must be set to the same value.

60.5.1.43 DMA Class Based Configuration (DMA1CFG - DMA2CFG)

60.5.1.43.1 Offset

Register	Offset
DMA1CFG	1D8h
DMA2CFG	1DCh

60.5.1.43.2 Function

The DMA class based configuration registers are used to configure the DMA controller interface to support the additional class 1 (buffer descriptor ring 1) and class 2 (buffer descriptor ring 2) traffic and define configuration options such as bandwidth allocation as needed.

NOTE

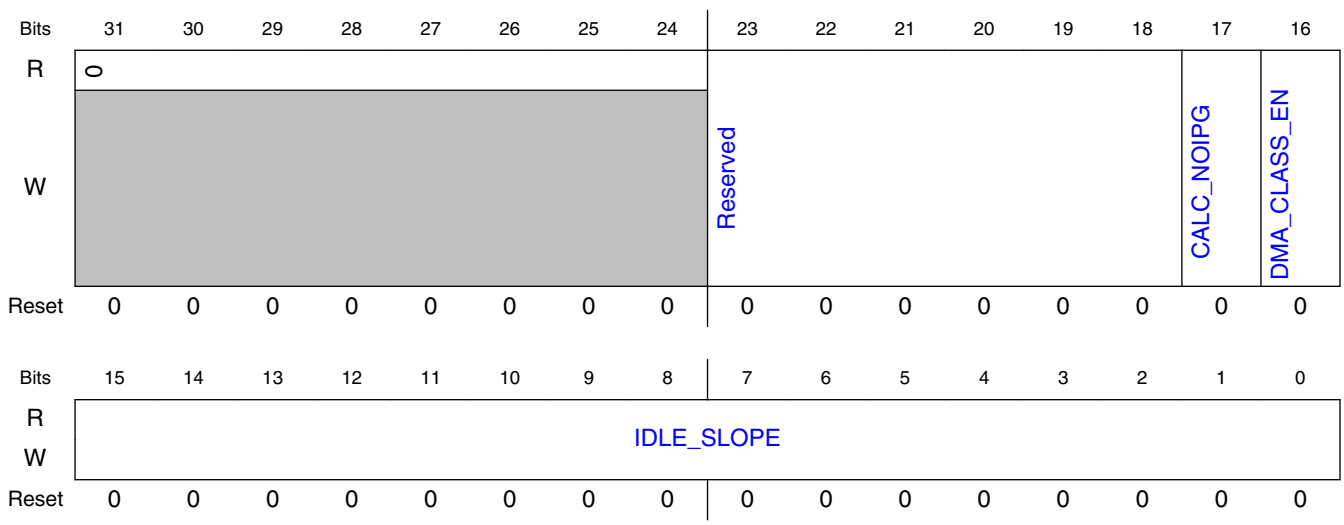
The registers are cleared when ECR[ETHEREN] becomes 0.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	DMA1CFG–DMA2CFG
ENET_1G	DMA1CFG–DMA2CFG	—

60.5.1.43.3 Diagram



60.5.1.43.4 Fields

Field	Function
31-24 —	Reserved
23-18 —	Write only zeroes to this field **NOTE:** Write only zeroes to this field.
17 CALC_NOIPG	Calculate no IPG Disable inclusion of IPG bytes for bandwidth calculations. 0b - The traffic shaper function should consider 12 octets of IPG in addition to the frame data transferred for a frame when doing bandwidth calculations. This is the default. 1b - Addition of 12 bytes for the IPG should be omitted when calculating the bandwidth (for traffic shaping, when writing a frame into the transmit FIFO, the shaper will usually consider 12 bytes of IPG for every frame as part of the bandwidth allocated by the frame. This addition can be

Table continues on the next page...

Field	Function
	suppressed, meaning short frames will become more bandwidth than large frames due to the relation of data to IPG overhead).
16 DMA_CLASS_EN	DMA class enable 0b - The DMA controller's channel for the class is not used. Disabling the DMA controller of a class also requires disabling the class match comparator for the class (see registers RCMRn). When class 1 and class 2 queues are disabled then their frames will be placed in queue 0. 1b - Enable the DMA controller to support the corresponding descriptor ring for this class of traffic.
15-0 IDLE_SLOPE	Idle slope 16-bit value to define the per class idle slope setting used by the credit based shaper defining allocated bandwidth for the class. This value is used to calculate the BW (bandwidth) fraction, given by the equation, BW fraction = 1/(1+512/IDLE_SLOPE). Idle slope is restricted to certain values. For values less than 128, idle slope = 2^n, where n = 0, 1, 2, ...6. For values equal to or greater than 128, idle slope = 128×m, where m = 1, 2, 3, ...12. Example 1. BW fraction = 0.20 = 1/(1+(512/128)); therefore idleslope = 128. Example 2. BW fraction = 0.33 = 1/(1+(512/256)); therefore idleslope = 256. Example 3. BW fraction = 0.75 = 1/(1+(512/1536)); therefore idleslope = 1536. **NOTE:** For AVB applications, the BW fraction of class 1 and class 2 combined must not exceed .75.

60.5.1.44 Receive Descriptor Active Register - Ring 1 (RDAR1)

60.5.1.44.1 Offset

Register	Offset
RDAR1	1E0h

60.5.1.44.2 Function

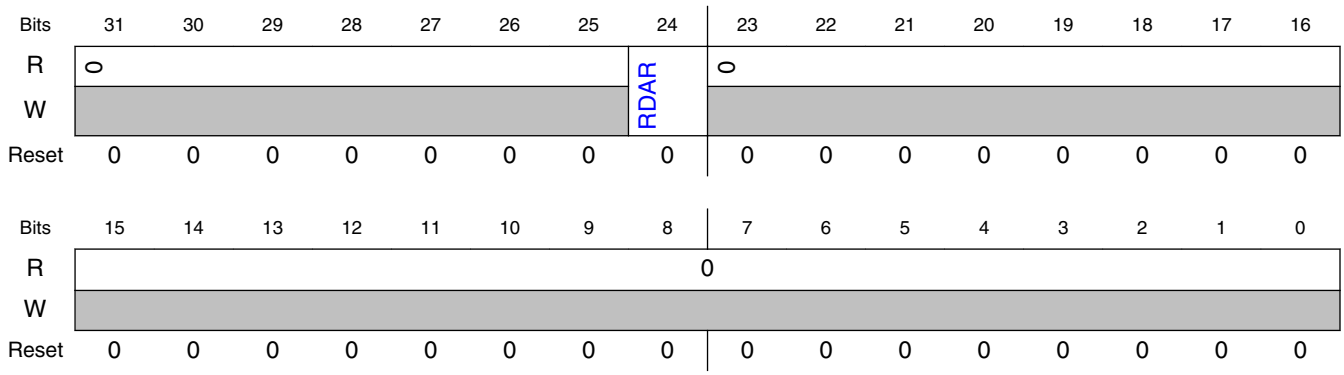
RDAR1 is a command register, written by the user, to indicate that the receive descriptor ring 1 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDAR1
ENET_1G	RDAR1	—

60.5.1.44.3 Diagram



60.5.1.44.4 Fields

Field	Function
31-25 —	Reserved
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.45 Transmit Descriptor Active Register - Ring 1 (TDAR1)

60.5.1.45.1 Offset

Register	Offset
TDAR1	1E4h

60.5.1.45.2 Function

TDAR1 is a command register that the user writes to indicate that transmit descriptor ring 1 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

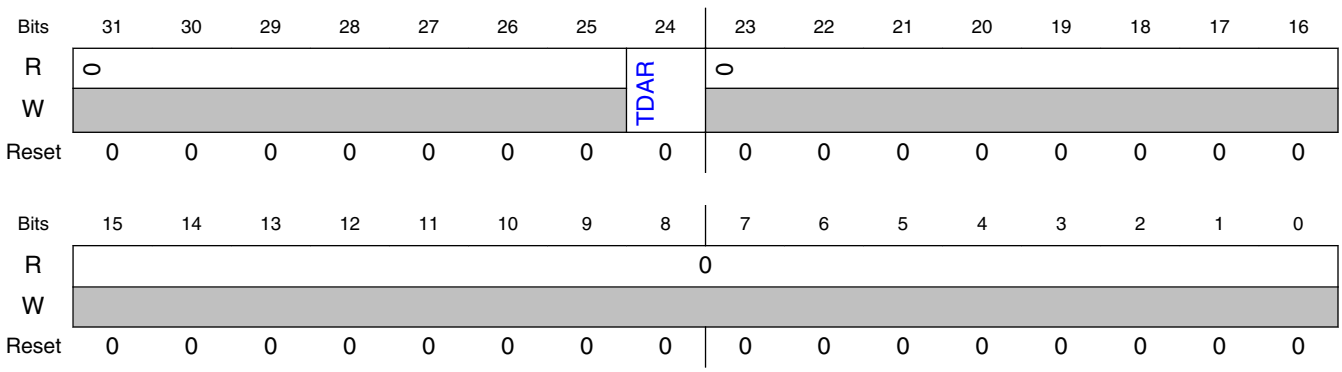
The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDAR1
ENET_1G	TDAR1	—

60.5.1.45.3 Diagram



60.5.1.45.4 Fields

Field	Function
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.46 Receive Descriptor Active Register - Ring 2 (RDAR2)

60.5.1.46.1 Offset

Register	Offset
RDAR2	1E8h

60.5.1.46.2 Function

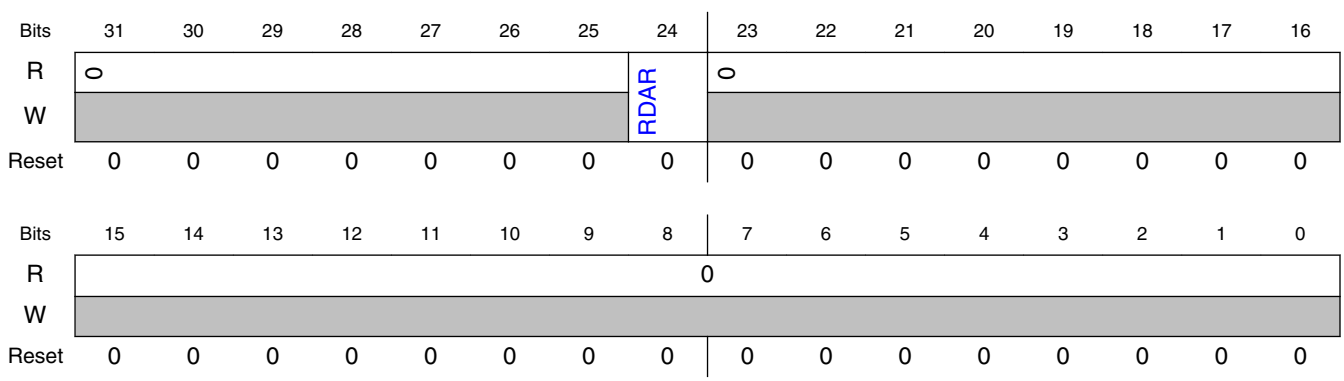
RDAR2 is a command register, written by the user, to indicate that the receive descriptor ring 2 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	RDAR2
ENET_1G	RDAR2	—

60.5.1.46.3 Diagram



60.5.1.46.4 Fields

Field	Function
31-25 —	Reserved
24 RDAR	Receive Descriptor Active

Table continues on the next page...

Field	Function
	Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.47 Transmit Descriptor Active Register - Ring 2 (TDAR2)

60.5.1.47.1 Offset

Register	Offset
TDAR2	1ECh

60.5.1.47.2 Function

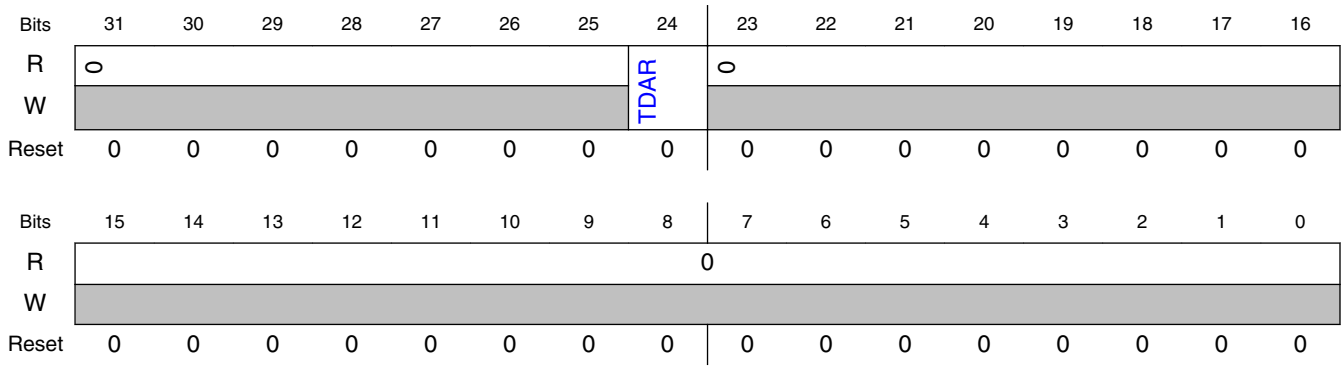
TDAR2 is a command register that the user writes to indicate that transmit descriptor ring 2 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	TDAR2
ENET_1G	TDAR2	—

60.5.1.47.3 Diagram



60.5.1.47.4 Fields

Field	Function
31-25 —	Reserved
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23-0 —	Reserved

60.5.1.48 QOS Scheme (QOS)

60.5.1.48.1 Offset

Register	Offset
QOS	1F0h

60.5.1.48.2 Function

This register sets the QOS scheme.

NOTE

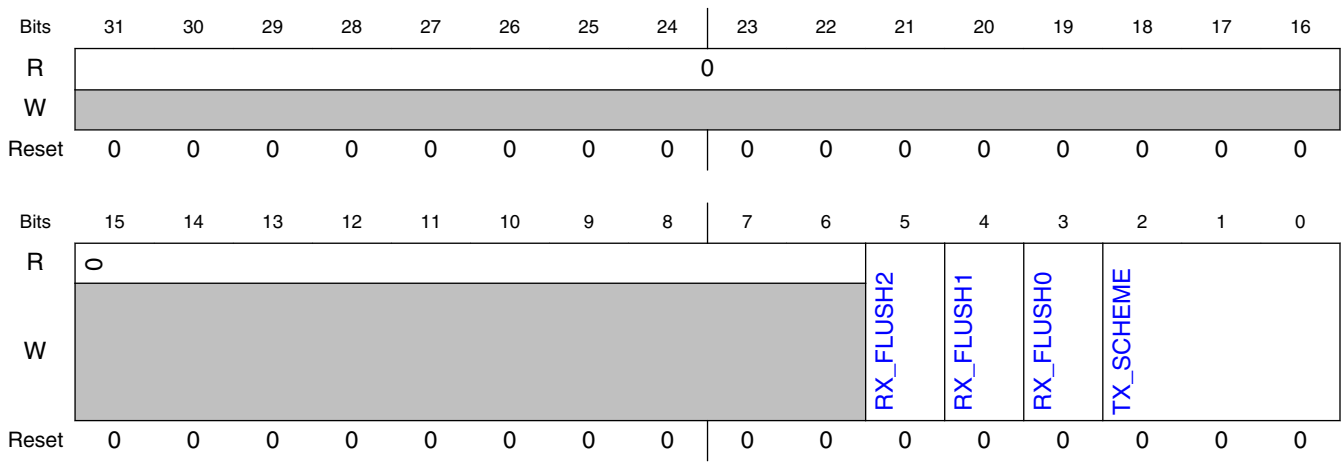
When both class 1 and class 2 are disabled, RX flushing for these rings must also be disabled.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
ENET	—	QOS
ENET_1G	QOS	—

60.5.1.48.3 Diagram



60.5.1.48.4 Fields

Field	Function
31-6 —	Reserved
5 RX_FLUSH2	RX Flush Ring 2 Enable or disable RX Flush for ring 2. See Receive flushing. 0b - Disable 1b - Enable
4 RX_FLUSH1	RX Flush Ring 1 Enable or disable RX Flush for ring 1. See Receive flushing. 0b - Disable 1b - Enable
3 RX_FLUSH0	RX Flush Ring 0 Enable or disable RX Flush for ring 0. See Receive flushing. 0b - Disable 1b - Enable
2-0	TX scheme configuration

Field	Function
TX_SCHEME	Configuration information for DMA to select transmitter queue selection/arbitration scheme. 000b - Credit-based scheme 001b - Round-robin scheme 010b-111b - Reserved

60.5.1.49 Tx Packet Count Statistic Register (RMON_T_PACKETS)

60.5.1.49.1 Offset

Register	Offset
RMON_T_PACKETS	204h

60.5.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.49.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Packet count Transmit packet count

60.5.1.50 Tx Broadcast Packets Statistic Register (RMON_T_BC_PKT)

60.5.1.50.1 Offset

Register	Offset
RMON_T_BC_PKT	208h

60.5.1.50.2 Function

RMON Tx Broadcast Packets

60.5.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.50.4 Fields

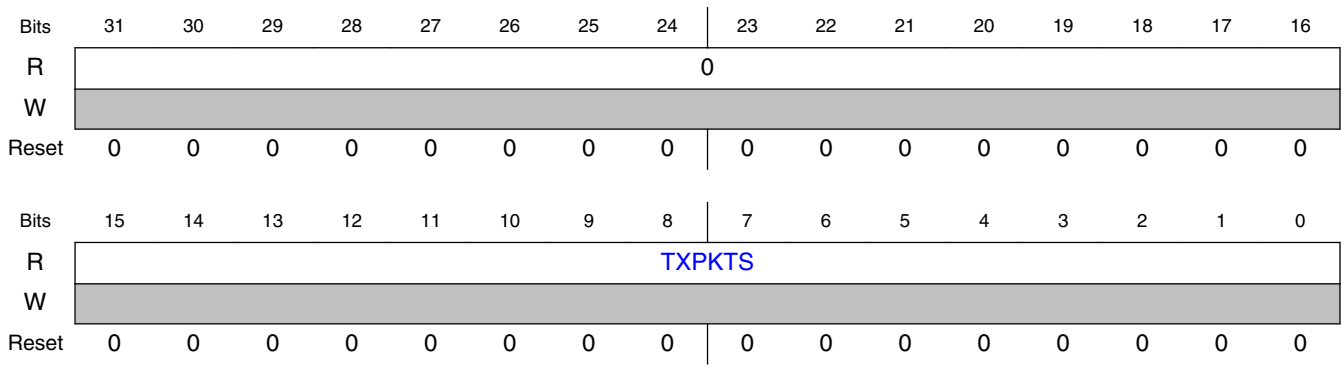
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of broadcast packets

60.5.1.51 Tx Multicast Packets Statistic Register (RMON_T_MC_PKT)

60.5.1.51.1 Offset

Register	Offset
RMON_T_MC_PKT	20Ch

60.5.1.51.2 Diagram



60.5.1.51.3 Fields

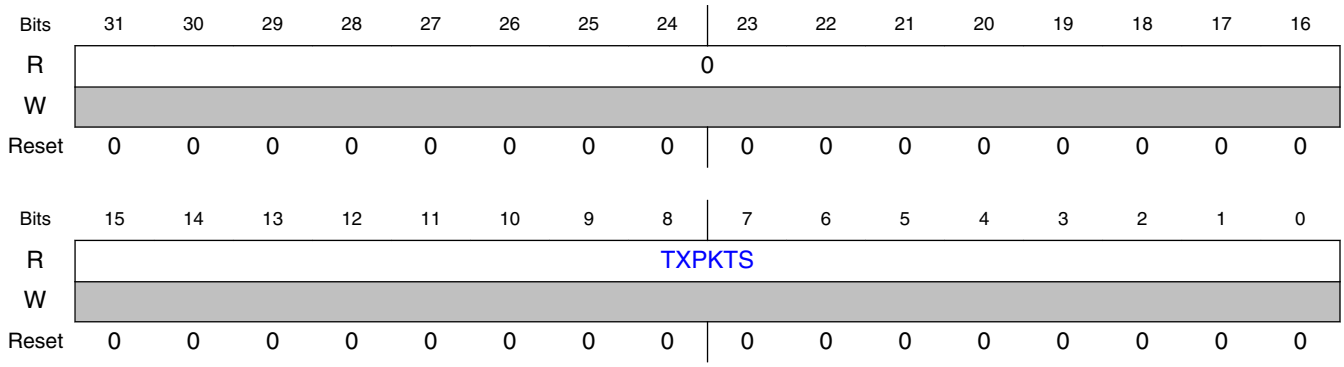
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of multicast packets

60.5.1.52 Tx Packets with CRC/Align Error Statistic Register (RMON_T_CRC_ALIGN)

60.5.1.52.1 Offset

Register	Offset
RMON_T_CRC_ALIGN	210h

60.5.1.52.2 Diagram



60.5.1.52.3 Fields

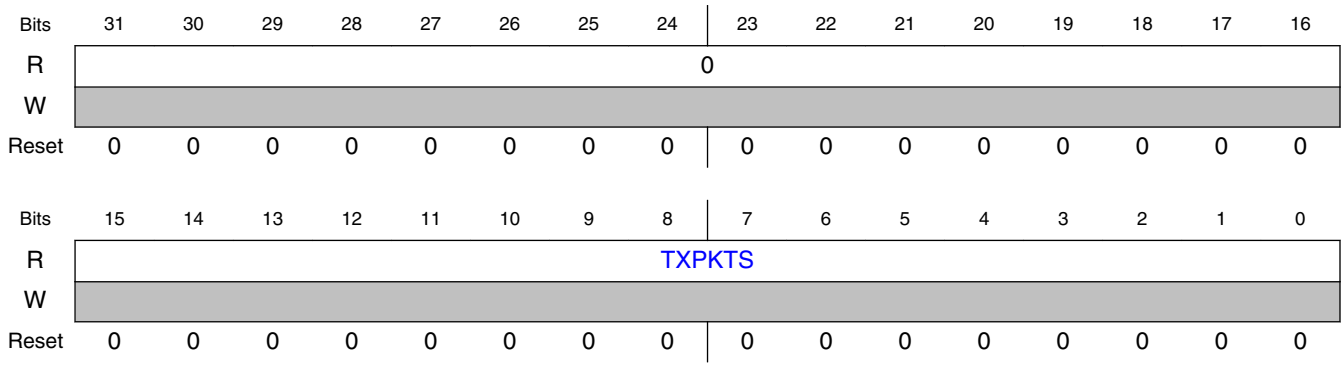
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of packets with CRC/align error

60.5.1.53 Tx Packets Less Than Bytes and Good CRC Statistic Register (RMON_T_UNDERSIZE)

60.5.1.53.1 Offset

Register	Offset
RMON_T_UNDERSIZE	214h

60.5.1.53.2 Diagram



60.5.1.53.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets less than 64 bytes with good CRC

60.5.1.54 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (RMON_T_OVERSIZE)

60.5.1.54.1 Offset

Register	Offset
RMON_T_OVERSIZE	218h

60.5.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.54.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

60.5.1.55 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_T_FRAG)

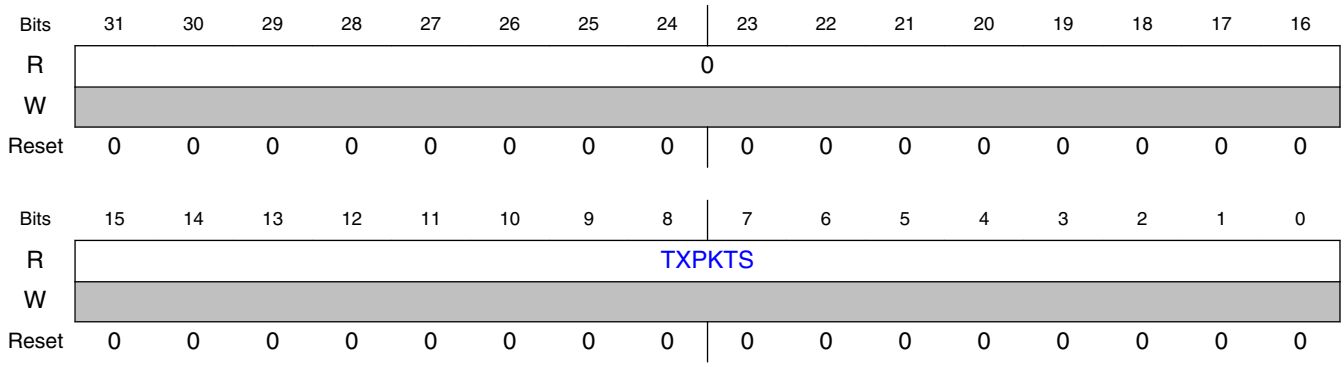
60.5.1.55.1 Offset

Register	Offset
RMON_T_FRAG	21Ch

60.5.1.55.2 Function

.

60.5.1.55.3 Diagram



60.5.1.55.4 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of packets less than 64 bytes with bad CRC

60.5.1.56 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (RMON_T_JAB)

60.5.1.56.1 Offset

Register	Offset
RMON_T_JAB	220h

60.5.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.56.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

60.5.1.57 Tx Collision Count Statistic Register (RMON_T_COL)

60.5.1.57.1 Offset

Register	Offset
RMON_T_COL	224h

60.5.1.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.57.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit collisions

60.5.1.58 Tx 64-Byte Packets Statistic Register (RMON_T_P64)

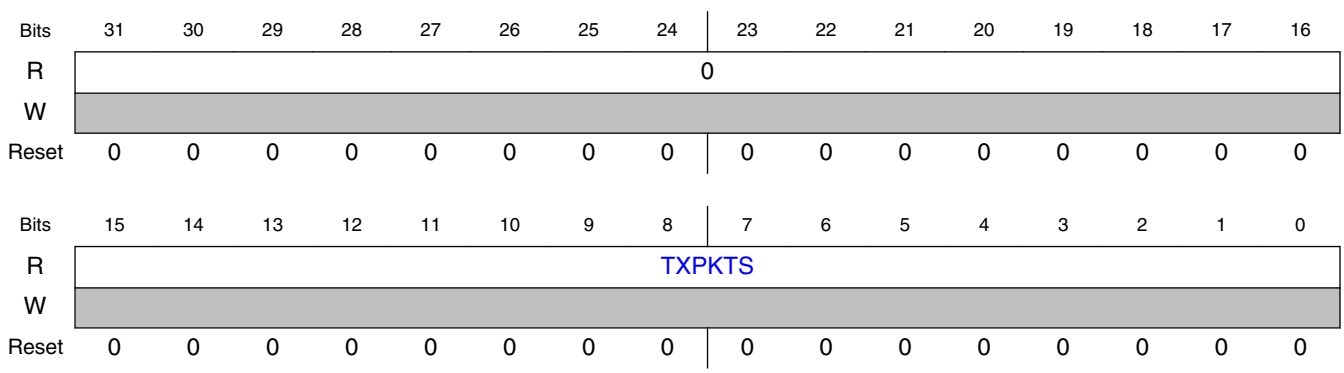
60.5.1.58.1 Offset

Register	Offset
RMON_T_P64	228h

60.5.1.58.2 Function

.

60.5.1.58.3 Diagram



60.5.1.58.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 TXPKTS	Number of 64-byte transmit packets

60.5.1.59 Tx 65- to 127-byte Packets Statistic Register (RMON_T_P65TO127)

60.5.1.59.1 Offset

Register	Offset
RMON_T_P65TO127	22Ch

60.5.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.59.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 65- to 127-byte transmit packets

60.5.1.60 Tx 128- to 255-byte Packets Statistic Register (RMON_T_P128TO255)

60.5.1.60.1 Offset

Register	Offset
RMON_T_P128TO255	230h

60.5.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.60.3 Fields

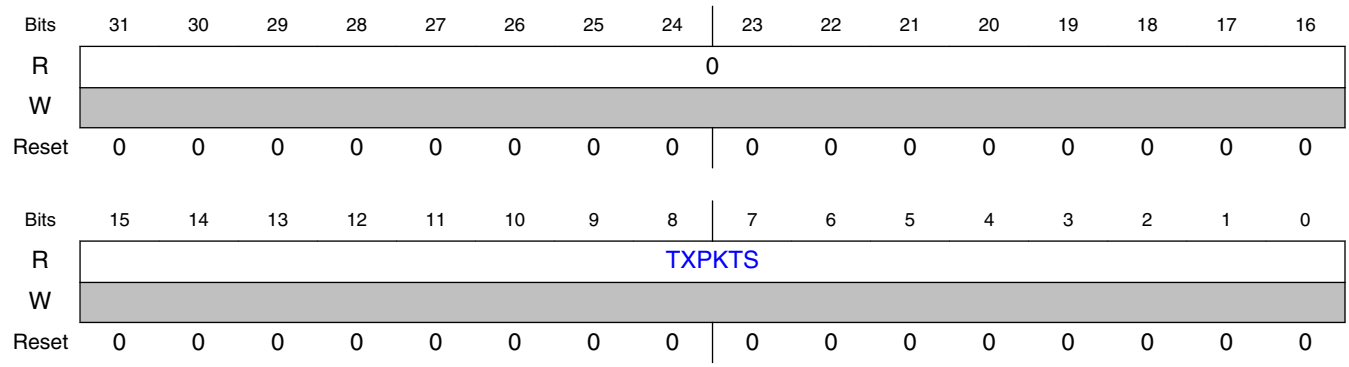
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 128- to 255-byte transmit packets

60.5.1.61 Tx 256- to 511-byte Packets Statistic Register (RMON_T_P256TO511)

60.5.1.61.1 Offset

Register	Offset
RMON_T_P256TO511	234h

60.5.1.61.2 Diagram



60.5.1.61.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 256- to 511-byte transmit packets

60.5.1.62 Tx 512- to 1023-byte Packets Statistic Register (RMON_T_P512TO1023)

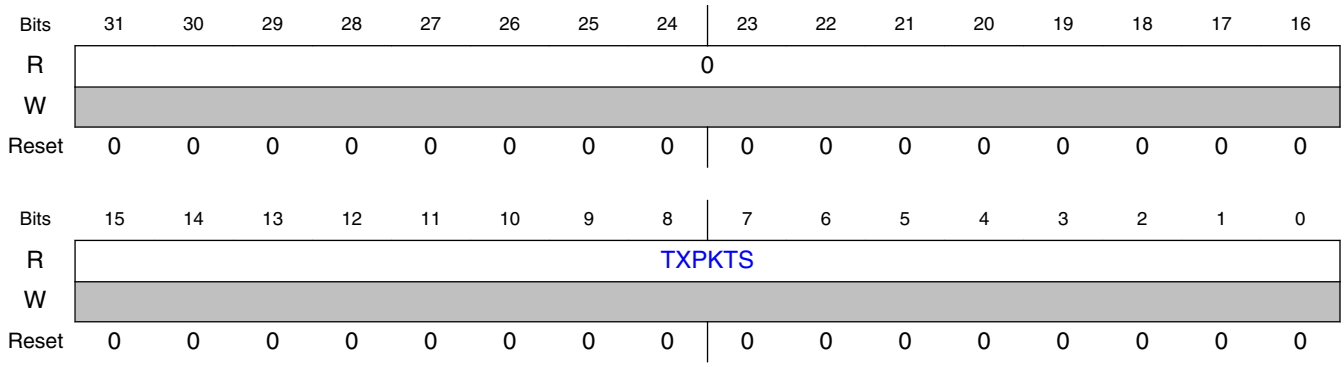
60.5.1.62.1 Offset

Register	Offset
RMON_T_P512TO1023	238h

60.5.1.62.2 Function

.

60.5.1.62.3 Diagram



60.5.1.62.4 Fields

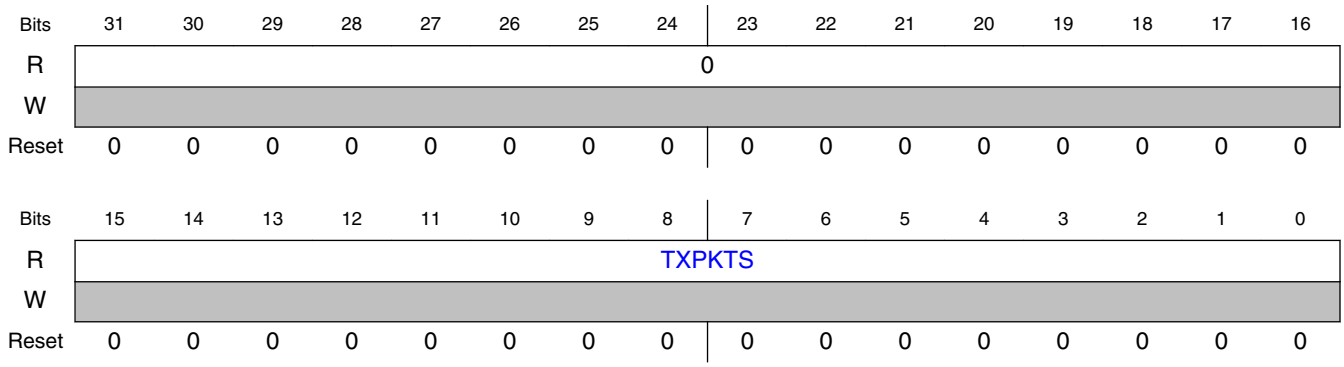
Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 512- to 1023-byte transmit packets

60.5.1.63 Tx 1024- to 2047-byte Packets Statistic Register (RMON_T_P1024TO2047)

60.5.1.63.1 Offset

Register	Offset
RMON_T_P1024TO2047	23Ch

60.5.1.63.2 Diagram



60.5.1.63.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of 1024- to 2047-byte transmit packets

60.5.1.64 Tx Packets Greater Than 2048 Bytes Statistic Register (RMON_T_P_GTE2048)

60.5.1.64.1 Offset

Register	Offset
RMON_T_P_GTE2048	240h

60.5.1.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.64.3 Fields

Field	Function
31-16 —	Reserved
15-0 TXPKTS	Number of transmit packets greater than 2048 bytes

60.5.1.65 Tx Octets Statistic Register (RMON_T_OCTETS)

60.5.1.65.1 Offset

Register	Offset
RMON_T_OCTETS	244h

60.5.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.65.3 Fields

Field	Function
31-0 TXOCTS	Number of transmit octets

60.5.1.66 Frames Transmitted OK Statistic Register (IEEE_T_FRAME_OK)

60.5.1.66.1 Offset

Register	Offset
IEEE_T_FRAME_OK	24Ch

60.5.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.66.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted OK **NOTE:** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

60.5.1.67 Frames Transmitted with Single Collision Statistic Register (IEEE_T_1COL)

60.5.1.67.1 Offset

Register	Offset
IEEE_T_1COL	250h

60.5.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.67.3 Fields

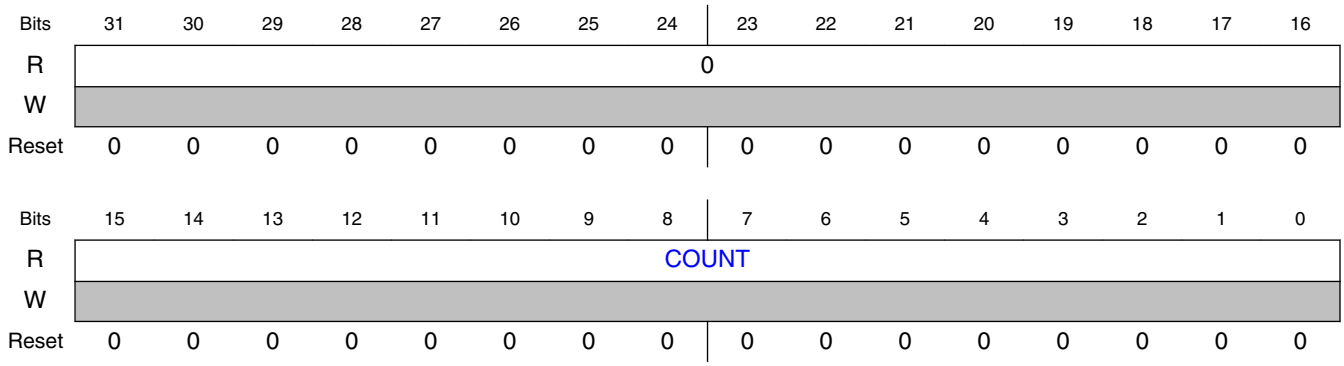
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with one collision

60.5.1.68 Frames Transmitted with Multiple Collisions Statistic Register (IEEE_T_MCOL)

60.5.1.68.1 Offset

Register	Offset
IEEE_T_MCOL	254h

60.5.1.68.2 Diagram



60.5.1.68.3 Fields

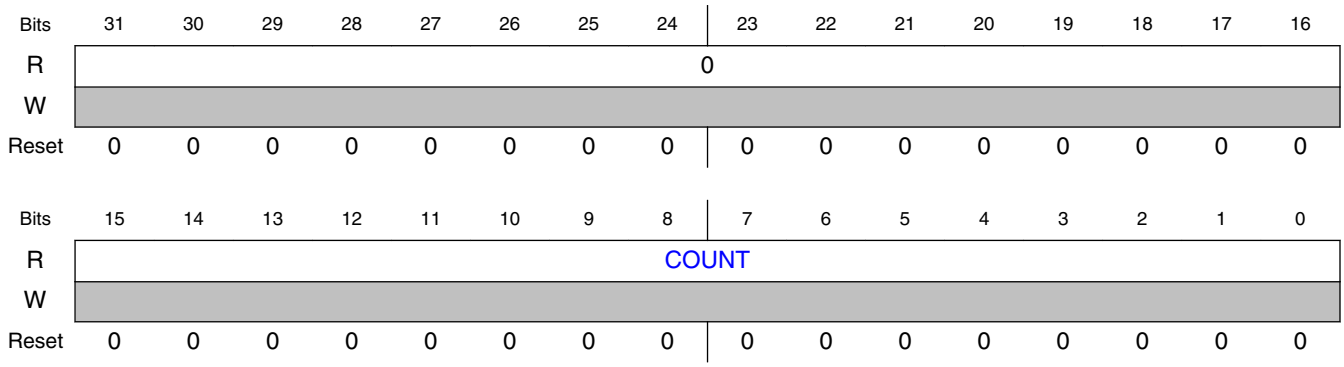
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with multiple collisions

60.5.1.69 Frames Transmitted after Deferral Delay Statistic Register (IEEE_T_DEF)

60.5.1.69.1 Offset

Register	Offset
IEEE_T_DEF	258h

60.5.1.69.2 Diagram



60.5.1.69.3 Fields

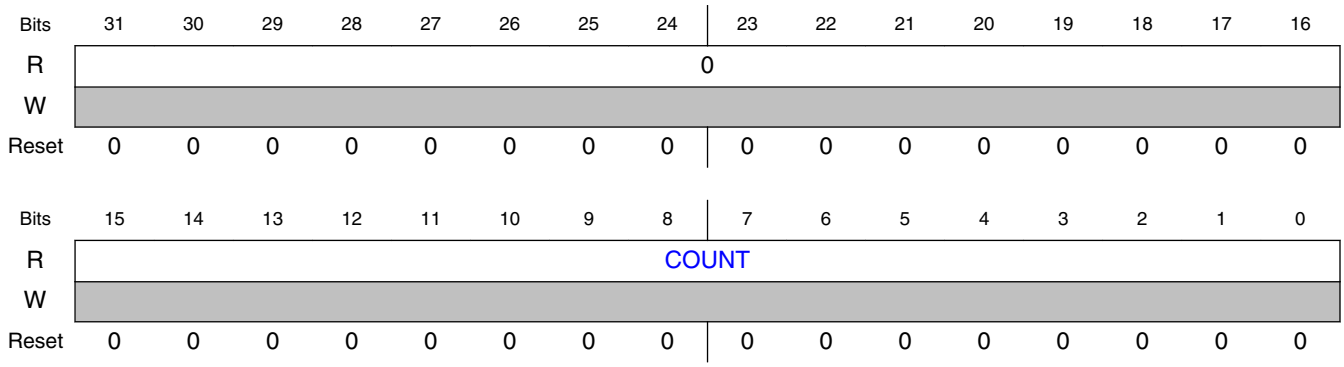
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with deferral delay

60.5.1.70 Frames Transmitted with Late Collision Statistic Register (IEEE_T_LCOL)

60.5.1.70.1 Offset

Register	Offset
IEEE_T_LCOL	25Ch

60.5.1.70.2 Diagram



60.5.1.70.3 Fields

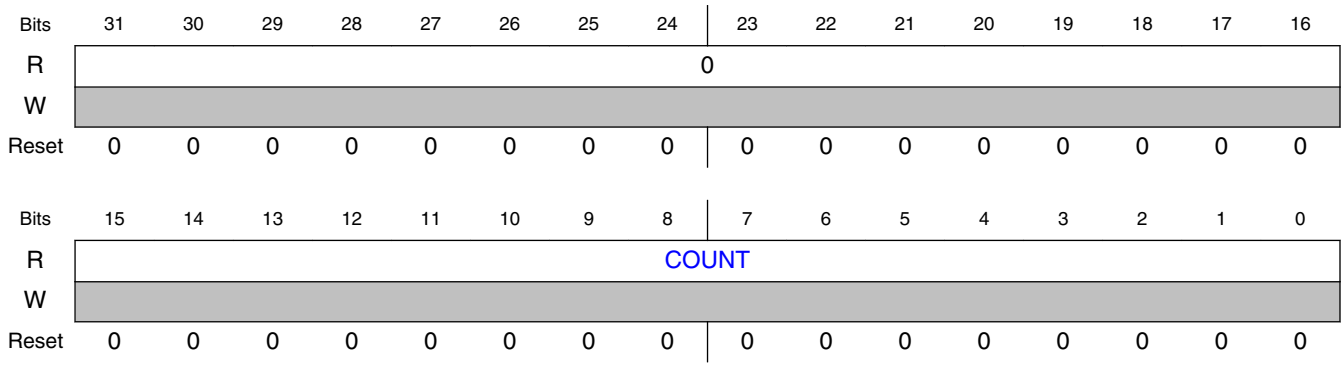
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with late collision

60.5.1.71 Frames Transmitted with Excessive Collisions Statistic Register (IEEE_T_EXCOL)

60.5.1.71.1 Offset

Register	Offset
IEEE_T_EXCOL	260h

60.5.1.71.2 Diagram



60.5.1.71.3 Fields

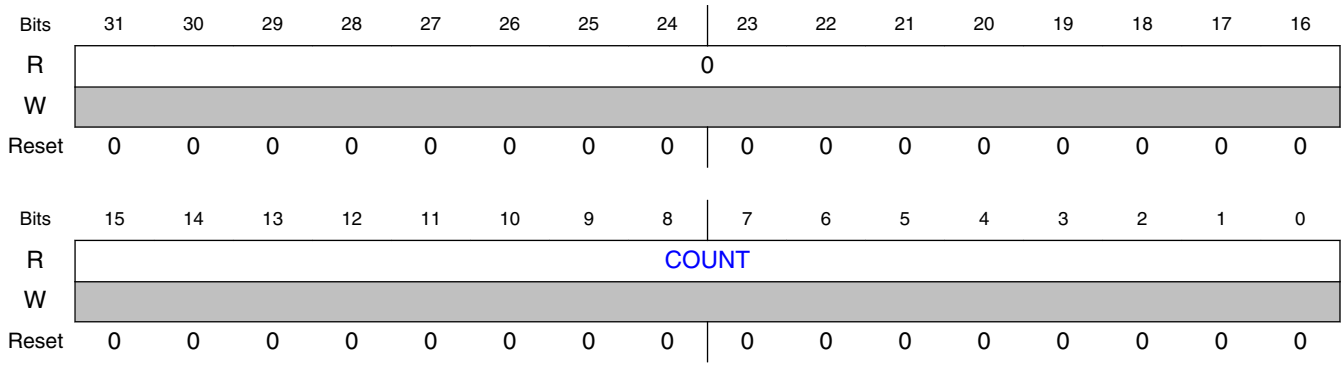
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with excessive collisions

60.5.1.72 Frames Transmitted with Tx FIFO Underrun Statistic Register (IEEE_T_MACERR)

60.5.1.72.1 Offset

Register	Offset
IEEE_T_MACERR	264h

60.5.1.72.2 Diagram



60.5.1.72.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with transmit FIFO underrun

60.5.1.73 Frames Transmitted with Carrier Sense Error Statistic Register (IEEE_T_CSERR)

60.5.1.73.1 Offset

Register	Offset
IEEE_T_CSERR	268h

60.5.1.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.73.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames transmitted with carrier sense error

60.5.1.74 Reserved Statistic Register (IEEE_T_SQE)

60.5.1.74.1 Offset

Register	Offset
IEEE_T_SQE	26Ch

60.5.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.74.3 Fields

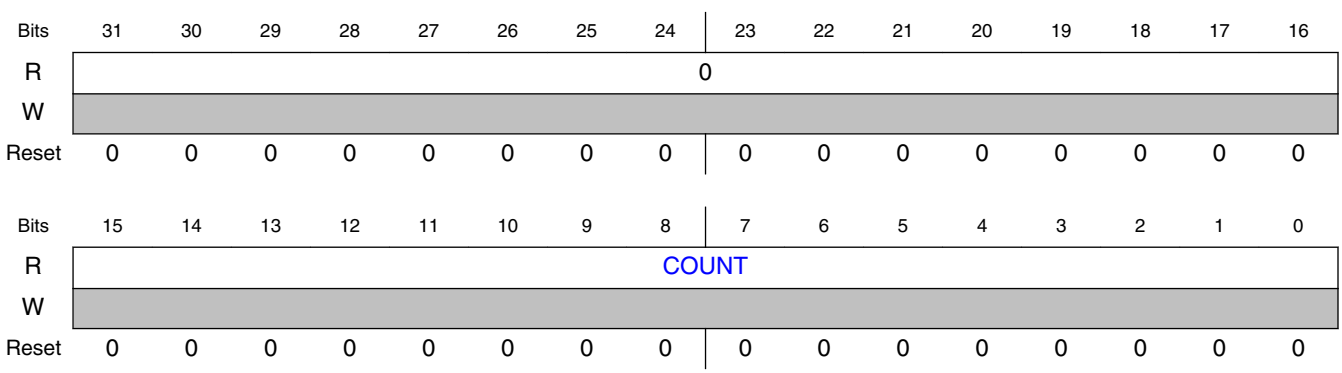
Field	Function
31-16 —	Reserved
15-0 COUNT	This read-only field is reserved and always has the value 0 This read-only field is reserved and always has the value 0. **NOTE:** Counter not implemented as no SQE information is available.

60.5.1.75 Flow Control Pause Frames Transmitted Statistic Register (IEEE_T_FDXFC)

60.5.1.75.1 Offset

Register	Offset
IEEE_T_FDXFC	270h

60.5.1.75.2 Diagram



60.5.1.75.3 Fields

Field	Function
31-16 —	Reserved

Table continues on the next page...

Field	Function
15-0 COUNT	Number of flow-control pause frames transmitted

60.5.1.76 Octet Count for Frames Transmitted w/o Error Statistic Register (IEEE_T_OCTETS_OK)

60.5.1.76.1 Offset

Register	Offset
IEEE_T_OCTETS_OK	274h

60.5.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.76.3 Fields

Field	Function
31-0 COUNT	Octet count for frames transmitted without error Counts total octets (includes header and FCS fields).

60.5.1.77 Rx Packet Count Statistic Register (RMON_R_PACKETS)

60.5.1.77.1 Offset

Register	Offset
RMON_R_PACKETS	284h

60.5.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.77.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of packets received

60.5.1.78 Rx Broadcast Packets Statistic Register (RMON_R_BC_PKT)

60.5.1.78.1 Offset

Register	Offset
RMON_R_BC_PKT	288h

60.5.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.78.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive broadcast packets

60.5.1.79 Rx Multicast Packets Statistic Register (RMON_R_MC_PKT)

60.5.1.79.1 Offset

Register	Offset
RMON_R_MC_PKT	28Ch

60.5.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.79.3 Fields

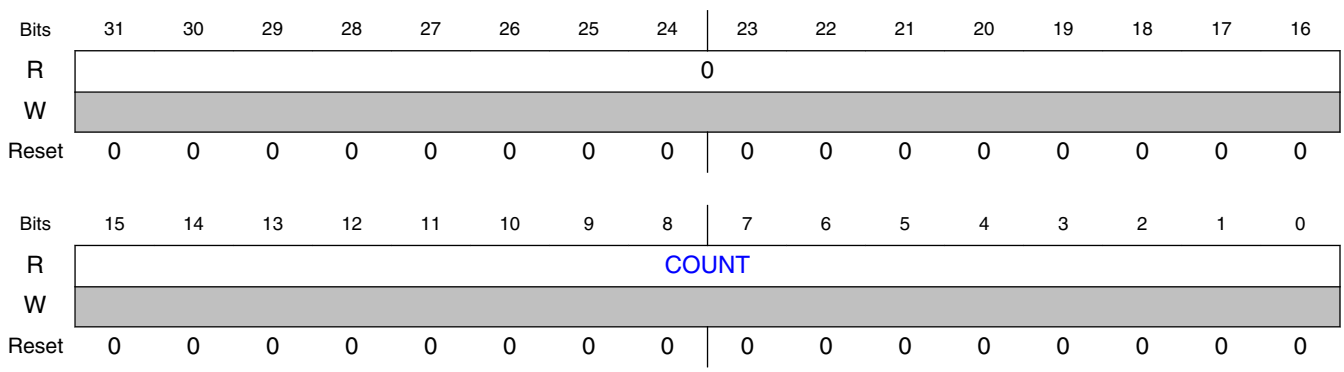
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive multicast packets

60.5.1.80 Rx Packets with CRC/Align Error Statistic Register (RMON_R_CRC_ALIGN)

60.5.1.80.1 Offset

Register	Offset
RMON_R_CRC_ALIGN	290h

60.5.1.80.2 Diagram



60.5.1.80.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with CRC or align error

60.5.1.81 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (RMON_R_UNDERSIZE)

60.5.1.81.1 Offset

Register	Offset
RMON_R_UNDERSIZE	294h

60.5.1.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.81.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and good CRC

60.5.1.82 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (RMON_R_OVERSIZE)

60.5.1.82.1 Offset

Register	Offset
RMON_R_OVERSIZE	298h

60.5.1.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.82.3 Fields

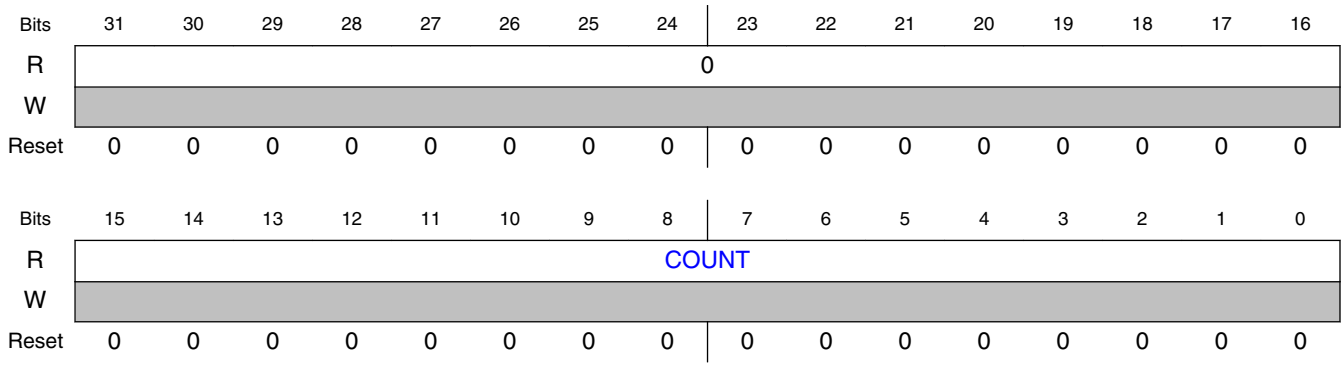
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and good CRC

60.5.1.83 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (RMON_R_FRAG)

60.5.1.83.1 Offset

Register	Offset
RMON_R_FRAG	29Ch

60.5.1.83.2 Diagram



60.5.1.83.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets with less than 64 bytes and bad CRC

60.5.1.84 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (RMON_R_JAB)

60.5.1.84.1 Offset

Register	Offset
RMON_R_JAB	2A0h

60.5.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.84.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of receive packets greater than MAX_FL and bad CRC

60.5.1.85 Rx 64-Byte Packets Statistic Register (RMON_R_P64)

60.5.1.85.1 Offset

Register	Offset
RMON_R_P64	2A8h

60.5.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.85.3 Fields

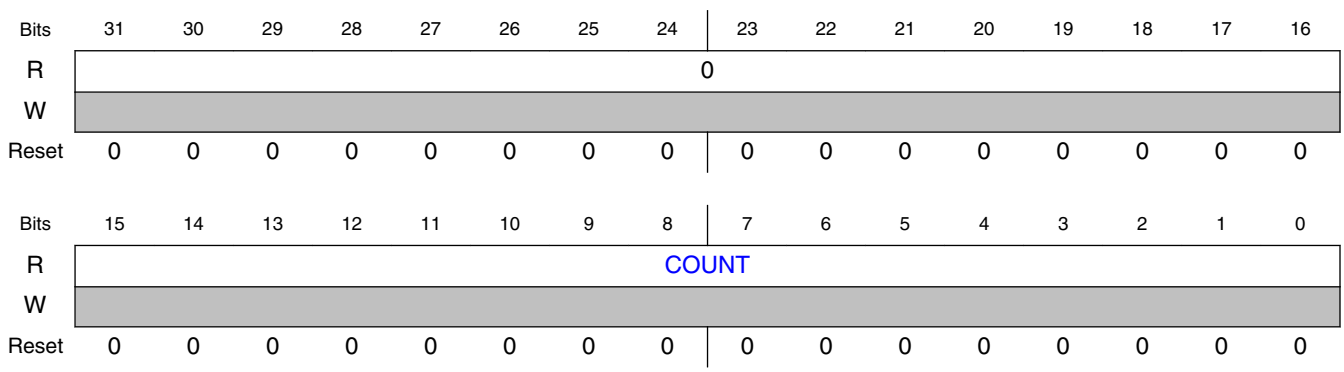
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 64-byte receive packets

60.5.1.86 Rx 65- to 127-Byte Packets Statistic Register (RMON_R_P65TO127)

60.5.1.86.1 Offset

Register	Offset
RMON_R_P65TO127	2ACh

60.5.1.86.2 Diagram



60.5.1.86.3 Fields

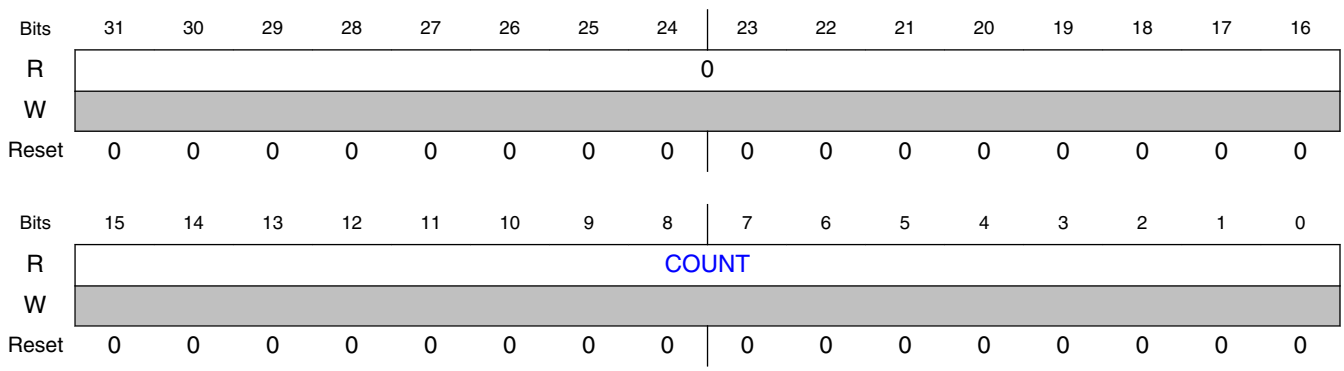
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 65- to 127-byte receive packets

60.5.1.87 Rx 128- to 255-Byte Packets Statistic Register (RMON_R_P128TO255)

60.5.1.87.1 Offset

Register	Offset
RMON_R_P128TO255	2B0h

60.5.1.87.2 Diagram



60.5.1.87.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 128- to 255-byte receive packets

60.5.1.88 Rx 256- to 511-Byte Packets Statistic Register (RMON_R_P256TO511)

60.5.1.88.1 Offset

Register	Offset
RMON_R_P256TO511	2B4h

60.5.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.88.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 256- to 511-byte receive packets

60.5.1.89 Rx 512- to 1023-Byte Packets Statistic Register (RMON_R_P512TO1023)

60.5.1.89.1 Offset

Register	Offset
RMON_R_P512TO1023	2B8h

60.5.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.89.3 Fields

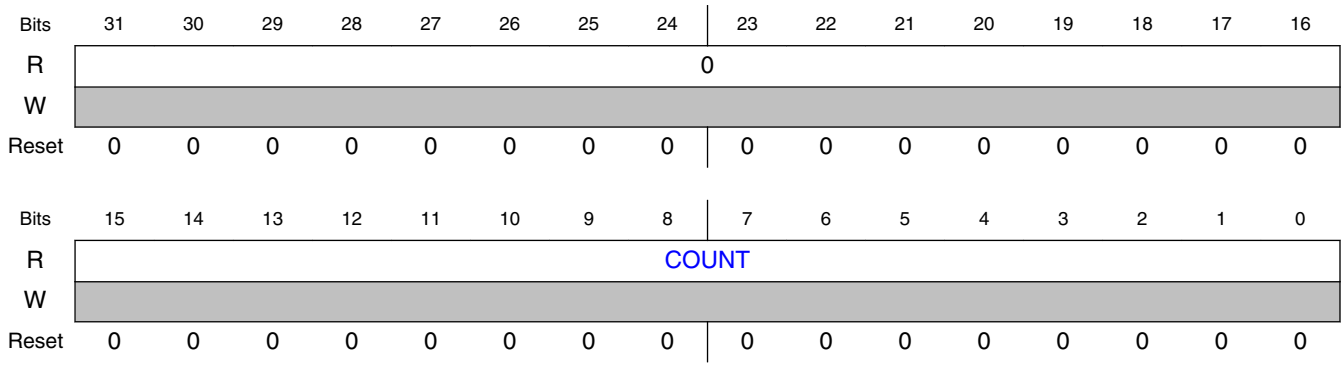
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 512- to 1023-byte receive packets

60.5.1.90 Rx 1024- to 2047-Byte Packets Statistic Register (RMON_R_P1024TO2047)

60.5.1.90.1 Offset

Register	Offset
RMON_R_P1024TO2047	2BCh

60.5.1.90.2 Diagram



60.5.1.90.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of 1024- to 2047-byte receive packets

60.5.1.91 Rx Packets Greater than 2048 Bytes Statistic Register (RMON_R_P_GTE2048)

60.5.1.91.1 Offset

Register	Offset
RMON_R_P_GTE2048	2C0h

60.5.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.91.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of greater-than-2048-byte receive packets

60.5.1.92 Rx Octets Statistic Register (RMON_R_OCTETS)

60.5.1.92.1 Offset

Register	Offset
RMON_R_OCTETS	2C4h

60.5.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.92.3 Fields

Field	Function
31-0 COUNT	Number of receive octets

60.5.1.93 Frames not Counted Correctly Statistic Register (IEEE_R_DROP)

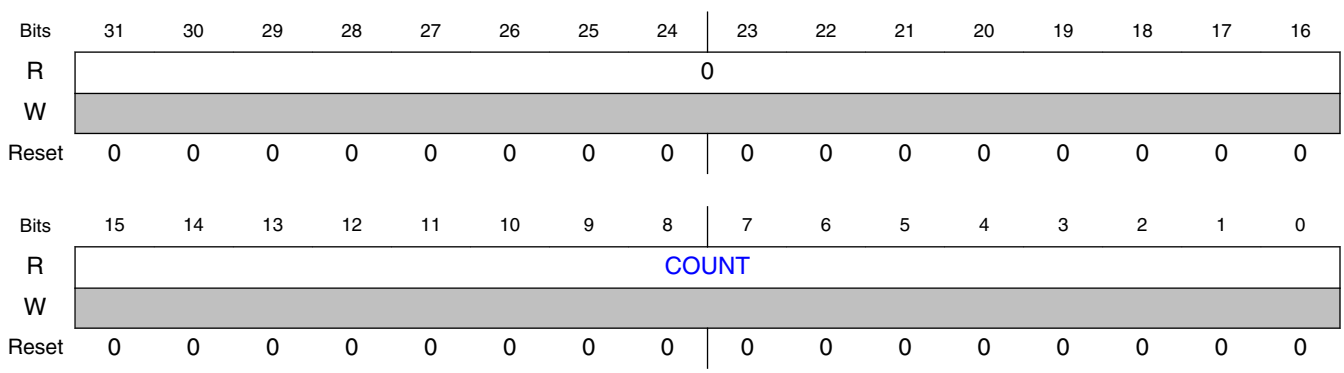
60.5.1.93.1 Offset

Register	Offset
IEEE_R_DROP	2C8h

60.5.1.93.2 Function

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

60.5.1.93.3 Diagram



60.5.1.93.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 COUNT	Frame count

60.5.1.94 Frames Received OK Statistic Register (IEEE_R_FRAME_OK)

60.5.1.94.1 Offset

Register	Offset
IEEE_R_FRAME_OK	2CCh

60.5.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.94.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received OK

60.5.1.95 Frames Received with CRC Error Statistic Register (IEEE_R_CRC)

60.5.1.95.1 Offset

Register	Offset
IEEE_R_CRC	2D0h

60.5.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.95.3 Fields

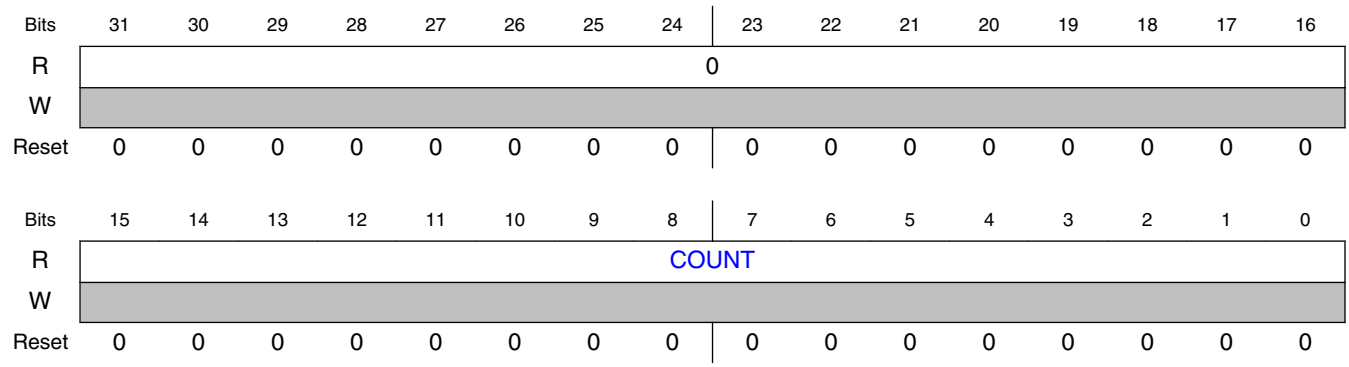
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received with CRC error

60.5.1.96 Frames Received with Alignment Error Statistic Register (IEEE_R_ALIGN)

60.5.1.96.1 Offset

Register	Offset
IEEE_R_ALIGN	2D4h

60.5.1.96.2 Diagram



60.5.1.96.3 Fields

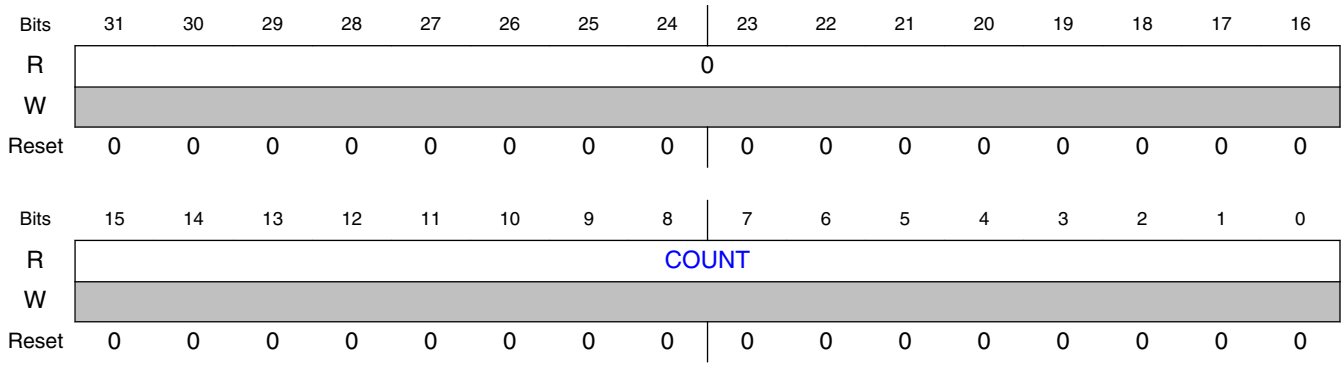
Field	Function
31-16 —	Reserved
15-0 COUNT	Number of frames received with alignment error

60.5.1.97 Receive FIFO Overflow Count Statistic Register (IEEE_R_MACERR)

60.5.1.97.1 Offset

Register	Offset
IEEE_R_MACERR	2D8h

60.5.1.97.2 Diagram



60.5.1.97.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Receive FIFO overflow count

60.5.1.98 Flow Control Pause Frames Received Statistic Register (IEEE_R_FDXFC)

60.5.1.98.1 Offset

Register	Offset
IEEE_R_FDXFC	2DCh

60.5.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.98.3 Fields

Field	Function
31-16 —	Reserved
15-0 COUNT	Number of flow-control pause frames received

60.5.1.99 Octet Count for Frames Received without Error Statistic Register (IEEE_R_OCTETS_OK)

60.5.1.99.1 Offset

Register	Offset
IEEE_R_OCTETS_OK	2E0h

60.5.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.99.3 Fields

Field	Function
31-0 COUNT	Number of octets for frames received without error **NOTE:** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

60.5.1.100 Adjustable Timer Control Register (ATCR)

60.5.1.100.1 Offset

Register	Offset
ATCR	400h

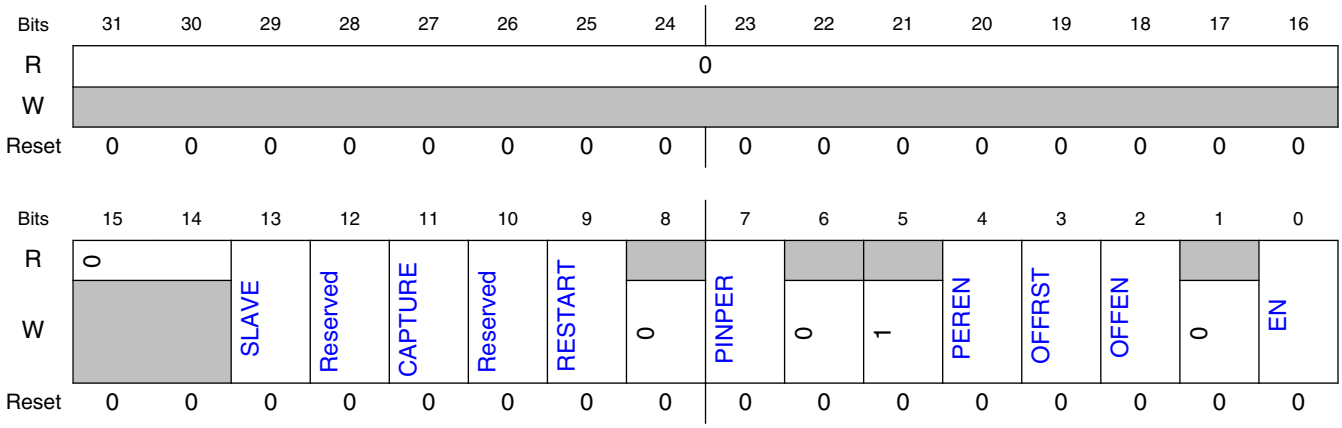
60.5.1.100.2 Function

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

60.5.1.100.3 Diagram



60.5.1.100.4 Fields

Field	Function
31-14 —	Reserved
13 SLAVE	Enable Timer Slave Mode 0b - The timer is active and all configuration fields in this register are relevant. 1b - The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 —	Always write 0 to this field Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. **NOTE:** **WARNING:** To ensure that the correct time value is read from the ATVR register, a minimum amount of time must elapse from issuing this command to reading the ATVR register. This minimum time is defined by the greater of either six register clock cycles or six 1588/timestamp clock cycles. 0b - No effect. 1b - The current time is captured and can be read from the ATVR register.
10 —	Always write 0 to this field Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled. **NOTE:** **WARNING:** The Reset Timer command requires at least 6 clock cycles of either the register clock or the 1588/timestamp clock, whichever is greater, to complete.
8	Reserved

Table continues on the next page...

Field	Function
—	
7 PINPER	Enables event signal output external pin frc_evt_period assertion on period event Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0b - Disable. 1b - Enable.
6 —	Reserved
5 —	This field must be written always with one **NOTE:** This field must be written always with one.
4 PEREN	Enable Periodical Event 0b - Disable. 1b - A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0b - The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1b - If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0b - Disable. 1b - The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 —	Reserved
0 EN	Enable Timer 0b - The timer stops at the current value. 1b - The timer starts incrementing.

60.5.1.101 Timer Value Register (ATVR)

60.5.1.101.1 Offset

Register	Offset
ATVR	404h

60.5.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.101.3 Fields

Field	Function
31-0 ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

60.5.1.102 Timer Offset Register (ATOFF)

60.5.1.102.1 Offset

Register	Offset
ATOFF	408h

60.5.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.102.3 Fields

Field	Function
31-0 OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

60.5.1.103 Timer Period Register (ATPER)

60.5.1.103.1 Offset

Register	Offset
ATPER	40Ch

60.5.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PERIOD															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD															
Reset	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

60.5.1.103.3 Fields

Field	Function
31-0 PERIOD	Value for generating periodic events Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE:** The value of PERIOD has the following constraint: 2^{32} – ENET_ATINC[INC_COR] – 3×ENET_ATINC[INC] ≥ PERIOD > 0.

60.5.1.104 Timer Correction Register (ATCOR)

60.5.1.104.1 Offset

Register	Offset
ATCOR	410h

60.5.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	COR														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.104.3 Fields

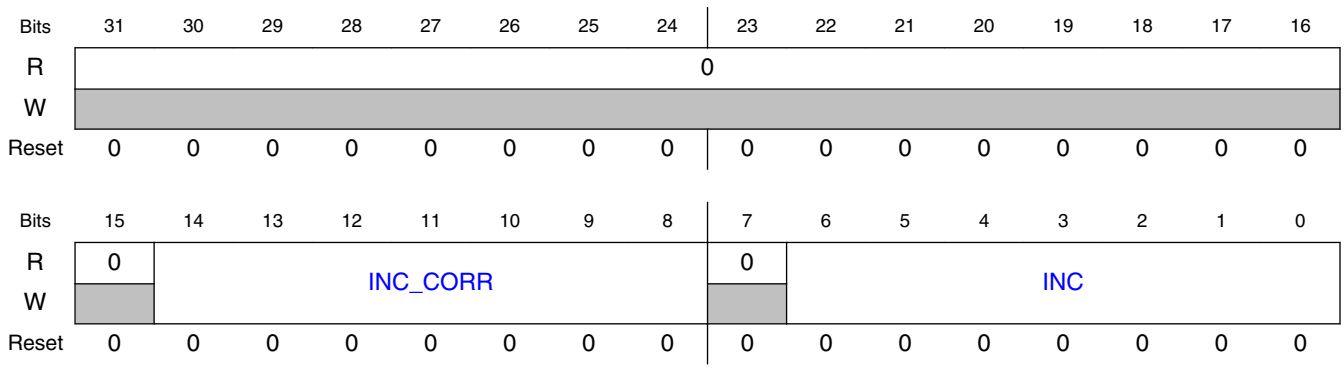
Field	Function
31 —	Reserved
30-0 COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

60.5.1.105 Time-Stamping Clock Period Register (ATINC)

60.5.1.105.1 Offset

Register	Offset
ATINC	414h

60.5.1.105.2 Diagram



60.5.1.105.3 Fields

Field	Function
31-15 —	Reserved
14-8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 —	Reserved
6-0 INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

60.5.1.106 Timestamp of Last Transmitted Frame (ATSTMP)

60.5.1.106.1 Offset

Register	Offset
ATSTMP	418h

60.5.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TIMESTAMP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMESTAMP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.106.3 Fields

Field	Function
31-0 TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set the ff_tx_ts_frm signal asserted from the user application Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

60.5.1.107 Timer Global Status Register (TGSR)

60.5.1.107.1 Offset

Register	Offset
TGSR	604h

60.5.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TF3	TF2	TF1	TF0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.107.3 Fields

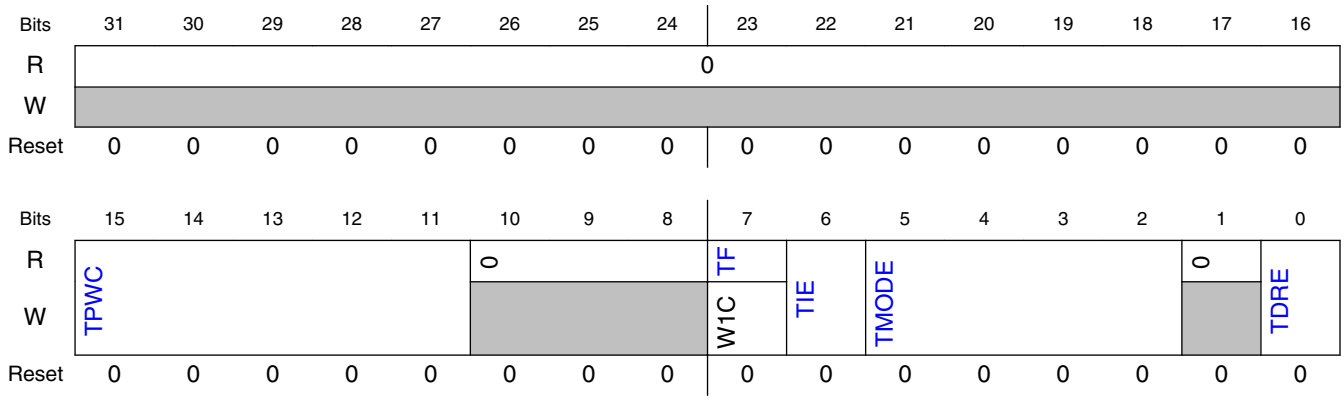
Field	Function
31-4 —	Reserved
3 TF3	Copy Of Timer Flag For Channel 3 0b - Timer Flag for Channel 3 is clear 1b - Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0b - Timer Flag for Channel 2 is clear 1b - Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0b - Timer Flag for Channel 1 is clear 1b - Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0b - Timer Flag for Channel 0 is clear 1b - Timer Flag for Channel 0 is set

60.5.1.108 Timer Control Status Register (TCSR0 - TCSR3)

60.5.1.108.1 Offset

Register	Offset
TCSR0	608h
TCSR1	610h
TCSR2	618h
TCSR3	620h

60.5.1.108.2 Diagram



60.5.1.108.3 Fields

Field	Function
31-16 —	Reserved
15-11 TPWC	Timer Pulse Width Control Specifies the pulse width associated with TMODE values of 1110 or 11X1. Updating this field takes a few cycles to register because it is synchronized to the 1588 clock. When changing this field: 1. Always disable the channel and read the TMODE field to verify that the channel is disabled. 2. Set TPWC to the desired value. 3. Re enable the channel. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Instance** / **Field supported in** / **Field not supported in** ENET / TCSR0–TCSR3 / — ENET_1G / — / TCSR0–TCSR3 0_0000b - Pulse width is one 1588-clock cycle. 0_0001b - Pulse width is two 1588-clock cycles. 0_0010b - Pulse width is three 1588-clock cycles. 0_0011b - Pulse width is four 1588-clock cycles. 1_1111b - Pulse width is 32 1588-clock cycles.
10-8 —	Reserved
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it. 0b - Input Capture or Output Compare has not occurred. 1b - Input Capture or Output Compare has occurred.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>6
TIE</td><td>Timer Interrupt Enable
0b - Interrupt is disabled
1b - Interrupt is enabled</td></tr>
<tr><td>5-2
TMODE</td><td>Timer Mode
Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first.
NOTE: The descriptions of the field settings vary by module instance.
<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>ENET</td><td>0000b - Timer Channel is disabled.
0001b - Timer Channel is configured for Input Capture on rising edge.
0010b - Timer Channel is configured for Input Capture on falling edge.
0011b - Timer Channel is configured for Input Capture on both edges.
0100b - Timer Channel is configured for Output Compare - software only.
0101b - Timer Channel is configured for Output Compare - toggle output on compare.
0110b - Timer Channel is configured for Output Compare - clear output on compare.
0111b - Timer Channel is configured for Output Compare - set output on compare.
1000b - Reserved
1010b - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow.
10x1b - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow.
110xb - Reserved
1110b - Timer Channel is configured for Output Compare - pulse output low on compare for 1 to 32 1588-clock cycles as specified by TPWC.
1111b - Timer Channel is configured for Output Compare - pulse output high on compare for 1 to 32 1588-clock cycles as specified by TPWC.</td></tr><tr><td>ENET_1G</td><td>0000b - Timer Channel is disabled.
0001b - Timer Channel is configured for Input Capture on rising edge.
0010b - Timer Channel is configured for Input Capture on falling edge.
0011b - Timer Channel is configured for Input Capture on both edges.
0100b - Timer Channel is configured for Output Compare - software only.
0101b - Timer Channel is configured for Output Compare - toggle output on compare.
0110b - Timer Channel is configured for Output Compare - clear output on compare.
0111b - Timer Channel is configured for Output Compare - set output on compare.
1000b - Reserved
1010b - Timer Channel is configured for Output Compare - clear output on compare, set output on overflow.
10x1b - Timer Channel is configured for Output Compare - set output on compare, clear output on overflow.
110xb - Reserved
1110b - Timer Channel is configured for Output Compare - pulse output low on compare for one 1588-clock cycle.
1111b - Timer Channel is configured for Output Compare - pulse output high on compare for one 1588-clock cycle.</td></tr></table></td></tr>
</table>

Table continues on the next page...

Field	Function
1 —	Reserved
0 TDRE	Timer DMA Request Enable 0b - DMA request is disabled 1b - DMA request is enabled

60.5.1.109 Timer Compare Capture Register (TCCR0 - TCCR3)

60.5.1.109.1 Offset

Register	Offset
TCCR0	60Ch
TCCR1	614h
TCCR2	61Ch
TCCR3	624h

60.5.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

60.5.1.109.3 Fields

Field	Function
31-0 TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the

Field	Function
	second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

Chapter 61
Ethernet Quality Of Service (ENET_QOS)

61.1 Chip-specific ENET QOS information

Table 61-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

61.2 Overview

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

The EQoS-TSN module supports the 10/100/1000Mbps applications in compliance with IEEE802.3-2015 specifications. It can support advanced applications such as Time-Sensitive Networking (TSN) and Audio Video Bridge (AVB). Other prominent features of the MAC core include a flexible receive parser, media clock recovery and generation, and safety.

61.2.1 Block diagram

The following figure shows the block diagram of this module. The module has four main blocks to perform all functions. The AHB interface is connected to all DMA channels. The DMA arbiter helps in arbitration of all paths (transmit and receive) in channels. Each channel has a separate set of control and status registers (CSR) for managing the transmit and receive functions, descriptor handling, and interrupt handling.

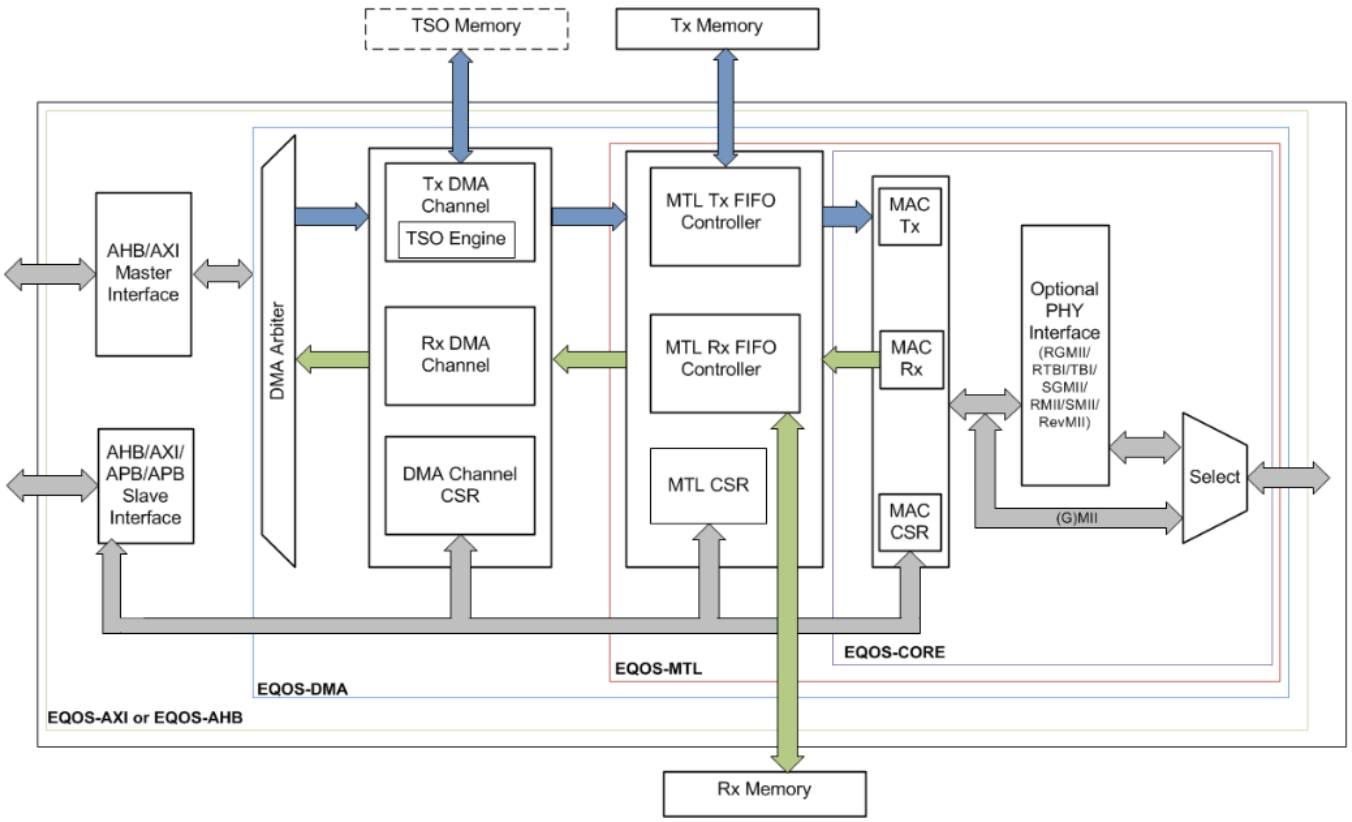


Figure 61-1. System-Level Block Diagram

61.2.2 Features

This module supports the following features in addition to the default features defined in IEEE802.3 specification

- MII (10/100Mbps), RMII (10/100Mbps), RGMII (10/100/1000Mbps)
- Time aware shaper (IEEE802.1Qbv), Time synchronization (IEEE1588-2008), and Frame pre-emption (IEEE802.1Qbu) for Time-Sensitive Networking (TSN)
- Media clock recovery and generation for AVB
- RMII specification version 1.2 from RMII consortium
- RGMII specification version 2.6 from HP/Marvell
- Separate interface for Transmit, Receive and Control path

- Broadcast/Multicast Packet duplication
- Full duplex flow control operations (IEEE 802.3x pause packet and priority flow control)
- Network statistics with MAC management (RMON) counters
- Flexibility to control the Pulse per second (PPS) output signal
- MDIO (clause 22 and Clause 45) interface for configuration and management of PHY device
- Preamble and start of packet data (SFD) insertion/deletion
- Automatic CRC and pad generation/stripping option, option to disable CRC checking
- Programmable inter-packet gap
- Source address insertion/replacement
- VLAN insertion, replacement and deletion in transmitted packets with per-packet or static-global control
- Insertion, replacement and deletion of up to two VLAN tags
- Insertion, replacement and deletion of queue/channel based VLAN tags
- IEEE802.1Q VLAN tag detection and option to delete from received packets
- Programmable safety watchdog timeout limits
- Flexible address filtering modes:
  - Up to 2 additional 48-bit perfect (DA) address filters with masks for each byte
  - Up to 2 48-bit SA address comparison check with masks for each byte
  - 64-bit Hash filter for multicast and unicast (DA) addresses
  - Option to pass all multi-cast addressed packets
  - Promiscuous mode to pass all packets without any filtering for network monitoring
  - Pass all incoming packets (as per filter) with a status report
- Additional packet filtering:
  - VLAN tag-based: Perfect match and Hash-based filtering. Filtering based on either outer or inner VLAN tag is possible
  - Layer 3 and Layer 4-based: TCP or UDP over IPv4 or IPv6
  - Extended VLAN tag based filtering 4 filter selections

61.3 Functional description

NOTE

This section and all its sub-sections are Synopsys Proprietary.
Used with permission.

The following sections describe functional details of the ENET module.

61.3.1 Architecture

This section describes the interfaces, protocols, and functionality. Following are the main blocks of the ENET QOS block.

- DMA Controller
- MAC Transaction Layer (MTL)
- MAC
- Interrupts

61.3.1.1 DMA controller

The DMA has independent Transmit (Tx) and Receive (Rx) engines, and a control and status register (CSR) space. The Tx engine transfers data from the system memory to the MTL interface, whereas the Rx engine transfers data from the MTL interface to the system memory. The DMA engine uses descriptors to efficiently move data from source to destination with minimal Host intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the Host for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and Host communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA supports up to 5 Tx and 5 Rx Descriptor lists (or DMA channels). The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_Ch[n]_Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. Once the DMA processes the last descriptor in the list, it automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when end of packet (EOP) is detected.

This module supports the ring structure for the DMA descriptor. For more information on Descriptor, see the Descriptors.

Following sections describes the operation of DMA Controller.

61.3.1.1.1 DMA Controller Bus Burst Access

The DMA engines attempt to transfer data in a burst of maximum size as programmed in the PBL bit fields of Transmit Control and Receive Control registers of the respective DMA. The Rx and Tx descriptors are always accessed in the maximum possible (limited by PBL or 16 * 8/bus width) burst length for 16 bytes to be read. The burst transfers initiated by the DMA can be split into multiple burst transfers as per the AHB requirements and the settings of DMA_SysBus_Mode register.

The Tx DMA initiates a data transfer only when sufficient space is available in the MTL Tx Queue to accommodate either of the following:

- Bytes corresponding to the configured burst (PBL * bus_width/8)
- Remaining bytes in Tx Buffer without EOP
- Number of bytes till EOP

The Rx DMA initiates a data transfer in the following conditions:

- Sufficient data is available in MTL Rx Queue to accommodate the configured burst
- EOP (when it is less than the configured burst length) is detected in the Rx Queue

61.3.1.1.2 DMA application data buffer alignment

The Tx and Rx data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for buffers can be aligned to any of the four bytes. However, the DMA always initiates write transfers with address aligned to the bus width and dummy data (old data) in the invalid byte lanes. This typically happens during the transfer of the beginning or end of an Ethernet packet. For the receiver, the software driver should discard the dummy bytes based on the start address of the buffer and size of the packet.

Table 61-2. Data buffer alignment examples

Examples	
Buffer Read	If the Tx buffer address is 32'h00000FF2 (for 32-bit data bus), and 15 bytes need to be transferred, the DMA reads five full words from address 32'h00000FF0, but when transferring data to the MTL Tx queue, the extra bytes (the first two bytes) are dropped or ignored. Similarly, the last 3 bytes of the last transfer are also ignored. The DMA always ensures that it transfers a full 32-bit data to the MTL Tx queue, unless it is the end of packet.

Table continues on the next page...

Table 61-2. Data buffer alignment examples (continued)

Buffer Write	If the Rx buffer address is 32'h0000FF2 (for 64-bit data bus) and 16 bytes of a received packet need to be transferred, the DMA writes 3 full words from address 32'h00000FF0. However, the first 2 bytes of the first transfer and the last 6 bytes of the third transfer have dummy data. The DMA considers the offset address only if it is the first Rx buffer of the packet. The DMA ignores the offset address and performs full word writes for the middle and the last Rx buffer of the packet.

61.3.1.1.3 DMA buffer size calculations

The DMA does not update the size fields in the Tx and Rx descriptors. The DMA updates only the status fields (RDES and TDES) of the descriptors. The driver has to perform the size calculations.

The Tx DMA transfers the exact number of bytes (indicated by buffer size field of TDES2) towards the MAC. If a descriptor is marked as first (FD bit of TDES3 is set), the DMA marks the first transfer from the buffer as SOP. If a descriptor is marked as last (LD bit of TDES3), the DMA marks the last transfer from that data buffer as EOP to the MTL.

The Rx DMA transfers data to a buffer until the buffer is full or the end of packet is received from the MTL. When the FD bit of a descriptor is set, the amount of valid data in a buffer is accurately indicated by the buffer size field (programmed in DMA Channel Receive Control register) minus the data buffer pointer offset. The offset is zero when the data buffer pointer is aligned to the data bus width. If a descriptor is marked as last, the buffer may not be full (as indicated by the buffer size in Bits [14:1] of Receive Control register). To compute the amount of valid data in this final buffer, the driver must read the packet length (PL bits of RDES3[14:0]) and subtract the sum of the buffer sizes of the preceding buffers in this packet. The Rx DMA always transfers the start of next packet with a new descriptor.

61.3.1.2 MAC transaction layer

The MAC Transaction Layer (MTL) provides the FIFO memory Interface to buffer and regulate the packets between system memory and the MAC. It also enables the data to be transferred between the system clock and MAC clock domains. The MTL layer has two data paths: Transmit path and Receive Path. The size of the FIFO is 8 Kb for Tx and Rx. The MTL communicates with the host through Application Transmit Interface (ATI) on the Transmit path and Application Receive Interface (ARI) on the receive path.

61.3.1.2.1 Transmit path

Internal DMA handles all transactions for the transmit path through ATI. Internal DMA pushes the Ethernet packets read from system memory into the corresponding queue. The packet is then popped out and transferred to the MAC when the queue threshold is reached (threshold mode) or complete packet is in the queue (store-and-forward mode). When EOP is transferred, the status of the transmission is taken from the MAC and transferred back to the internal DMA.

61.3.1.2.1.1 Transmit control word

The following control information related to packet transmission is provided as a part of the Control Word through ATI interface:

- Packet Length (if DCB is enabled with WFQ scheduling algorithm)
- CRC Pad Control
- Source Address Insertion Control
- VLAN Insertion and Replacement Control and VLAN Tag for Outer and Inner VLAN Tags
- TCP/IP Checksum Insertion Control
- One-step Timestamping Control Correction
- Transmit Timestamp Enable

61.3.1.2.1.2 Transmit operation

The following two modes of operation triggers reading of the data towards the MAC:

- Threshold mode: In Threshold (or cut-through) mode, as soon as the number of bytes in the Queue cross the configured threshold level (or when the end of packet is written before the threshold is crossed), the data is ready to be popped out and forwarded to the MAC. The threshold level is configured by using the TTC bits of MTL_TxQ0_Operation_Mode register corresponding to an MTL queue.
- Store-and-forward mode:

In store-and-forward mode, the MTL pops out the packet towards the MAC only when one or more of the following conditions are true:

- A complete packet is stored in the Queue
- The Tx FIFO becomes almost full
- The ATI watermark becomes low

The watermark becomes low when the requested queue does not have space to accommodate the requested burst length on the ATI. Therefore, the MTL when operating in the store-and-forward mode allows packet transmission even if the packet length is bigger than the Tx queue size.

The application can flush complete content of the Tx queue by setting Bit 0 (FTQ) of Transmit Operation Mode register corresponding to an MTL queue. This bit is self-clearing and initializes the Queue pointers to the default state. If the FTQ bit is set during a packet transfer from the MTL to the MAC, the MTL stops further transfer because the queue is considered to be empty. Therefore, an underflow event occurs at the MAC transmitter.

61.3.1.2.1.3 Initialization flow

Upon reset, the MTL is ready to manage the flow of data to and from the DMA, and MAC. For configurations with a single Transmit Queue, there are no initialization requirements for enabling the MTL. For configurations with multiple Transmit Queues, initialize the queue size for each of the queues by programming the TQS bits of MTL_TxQ0_Operation_Mode register corresponding to a Transmit queue. Also, initialize the MAC block. Internal DMA controllers must be individually enabled through their respective CSRs.

61.3.1.2.2 Receive path

The MTL Rx module receives the packets from the MAC and pushes them into the Rx Queue. The status (fill level) of the queue is indicated to DMA when it crosses the configured Receive threshold (RTC bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue), or the complete packet is received. The MTL also indicates the fill level of the queue so that the DMA can initiate preconfigured burst transfers towards the AHB interface.

For more information on queues, see Multiple Channels and Queues Support.

61.3.1.2.2.1 Receive Operation

During a Receive operation, the MTL is a slave for the MAC. The general sequence of events is as follows:

1. When the MAC receives a packet, it indicates the availability of receive data.
2. The MAC indicates the SOP and EOP delimiters.
3. The MTL accepts the data and pushes it into corresponding Rx queue.
4. After the EOP is transferred, the MAC drives the status word which is also pushed into the corresponding Rx queue by the MTL.
5. If IEEE 1588 timestamp feature is enabled and the 64-bit timestamp is available along with the packet status, it is pushed into the Rx queue as a part of the status word. Therefore, two additional locations are taken per packet to store the timestamp in the Rx queue.

6. The MTL takes the data out of the queue and sends it to the DMA depending on the following modes:

61.3.1.2.2.1.1 Threshold mode

In the (default) Threshold mode, the MTL reads the data and indicates its availability to the application or DMA when one of the following occurs:

- Data bytes equal to the threshold amount are written to the Rx queue (RTC bits[1:0] of MTL_RxQ0_Op-eration_Moderegister of corresponding MTL queue)
- A full packet of data is received into the queue

61.3.1.2.2.1.2 Store and forward mode

In the store-and-forward mode (when Bit 5 of MTL_RxQ0_Operation_Mode register of a queue is set to 1), the initial Rx queue locations are reserved for the status words before writing the SOP. A packet is read out only after it is completely written into the Rx queue. In this mode, all error packets are dropped (if configured through Bit 4 of MTL_RxQ0_Operation_Mode register of a queue) such that only valid packets are read and forwarded to the application.

61.3.1.2.2.2 Multi packet receive operation

In Threshold mode, the packet status is available immediately after the packet data. In store-and-forward mode, the packet data is available after the packet status. The MTL is capable of storing any number of packets into the queue as long as it is not full.

If the MAC receives a packet when the corresponding Rx queue is full, the MTL ignores that packet and overflow pulse is generated on the corresponding Rx queue. In addition, the MTL increments the overflow counter in the MTL_RxQ0_Missed_Packet_Overflow_Cnt of corresponding queue.

61.3.1.2.2.3 Error handling in receive operation

Following things would happen, If the MTL Rx queue is full before it receives the EOP data from the MAC.

1. An overflow is declared
2. The whole packet (including the status word) is dropped
3. The overflow counter in the DMA (Overflow Counter register of corresponding MTL queue) is incremented.

This is true even if Bit 4 (FEP) of MTL_RxQ0_Operation_Mode register of corresponding MTL queue is set.

If the start address of such a packet has already been transferred to the Read Controller, the rest of the packet is dropped and a dummy EOP is written to the queue along with the status word with overflow status. The status indicates a partial packet because of overflow. In such packets, the Packet Length field is invalid. If the MTL Receive Queue is configured to operate in the store-and-forward mode and the length of the received packet is more than the queue size, overflow occurs and all such packets are dropped.

The MTL Rx Control logic can filter error and undersized packets, if enabled by using the FEP and FUP bits of MTL_RxQ0_Operation_Mode register of corresponding MTL queue. If the start address of such a packet has already been transferred to the Rx Queue Read Controller, that packet is not filtered. The start address of the packet is transferred to the Read Controller after the packet crosses the receive threshold set by Bits[1:0] of MTL_RxQ0_Operation_Mode register of corresponding MTL queue.

If the MTL Receive Queue is configured to operate in the store-and-forward mode, all error packets can be filtered and dropped. For the application or DMA to flush the error packet being read from the queue, it must assert the flush signal. The MTL then stops transferring data to the application (DMA). It internally reads the rest of the packet and drops it. The MTL then starts the transfer of next packet (if available).

61.3.1.3 MAC

MAC can support MII, RMII or RGMII Phy interface. MAC consists of Mac Transmit interface (MTI), MAC Receive Interface (MRI) and MAC Control Interface (MCI).

61.3.1.3.1 MAC transmission

The MAC transmission process is as follows:

1. Transmission is initiated when the MAC Transaction Layer (MTL) pushes in data with the SOP signal asserted.
2. When the SOP signal is detected, the MAC accepts the data and begins transmitting to the MII, RMII or RGMII.
3. After the EOP is transferred to the MAC, the MAC does one of the following:
    - The MAC completes the normal transmission and gives the transmission status to the MTL.
    - If a normal collision (in half-duplex mode) occurs during transmission, the MAC gives the Transmit Status with retry bit set to the MTL. The MAC gives the Retry request till one of the following is true:
        - Packet is successfully transmitted

        - Maximum retry requests expire
        - When maximum retry requests expire, the MAC aborts the packet transmission with Excessive Collision Transmit Status. The MAC accepts and drops all further data until the next SOP is received. The MTL block should retransmit the same packet from SOP on observing a Retry request (in the Status) from the MAC.
    - If any one of the following happens, the MAC aborts the packet transmission:
        - No carrier (half-duplex mode)
        - Loss of carrier (half-duplex mode)
        - Excessive deferral (half-duplex mode)
        - Late collisions (half-duplex mode)
        - Jabber
        - The MAC accepts and drops all further data until the next SOP is received
4. The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. The MAC accepts and drops all further data until the next SOP is received.
5. During the normal transfer of a packet from MTL, if the MAC receives a SOP without getting an EOP for the previous packet, it ignores the SOP and considers the new packet as continuation of the previous packet.

61.3.1.3.2 MAC reception

A receive operation is initiated when the MAC detects an SFD on RMII, RGMII, or MII. The MAC strips the preamble and SFD before proceeding to process the packet. The header fields are checked for filtering and the FCS field used to verify the CRC for the packet. The received packet is stored in a shallow buffer until the address filtering is performed. The packet is dropped in the MAC if it fails the address filter.

61.3.2 Using PHY interfaces

This module support the following modes:

- RMII 10/100 Mbps interface
- MII 10/100 Mbps interface
- RGMII 10/100/1000 Mbps interface

The internal Phy_intf_sel input signal decides which mode has been selected. Phy_intf_sel signal gets sampled at reset. PS and FES bit of MAC configuration register selects the speed of mode.

The module supports the IEEE802.3-2015 specification for MII, RMII (RMII specification version 1.2 from the RMII consortium) and RGMII specification.

NOTE

RMII reference clock (50 MHz) can be fed to IP from an external source or internally from PLL on SoC.

The module supports can access the phy registers through Station Management Agent (SMA) which is a two wire Station management interface (MIM). One wire is management data clock (MDC) and other wire is management data input/output (MDIO). As per IEEE 802.3 specification, maximum operating frequency of MDC is 2.5 Mhz which can be derived from system clock using a divider. The CR bit of MAC_MDIO_Address register can be programmed to generate different MDC clock frequency.

SMA supports both MDIO clause 45 and clause 22 frame structure as per IEEE802.3 specification. The C45E bit of MAC_MDIO_Address register can be set to enable the clause 45 frame structure.

If the MII interface is overclocked to run at 200 Mbps, it reduces the accuracy of the 1588 timestamp. This is because the MAC logic uses the MII interface speed, as specified in MAC_Configuration[FES], to adjust or compensate for the timestamps taken at MII, as compared to the time generated in the PTP clock domain.

61.3.3 VLAN and double VLAN insertion, deletion, replacement and tagging

These sections describes the VLAN and Double VLAN features.

61.3.3.1 Double VLAN processing

The module can support processing of two VLAN tags in double VLAN tagging processing. These two VLAN tags can also be named as inner (C-VLAN) and outer (S-VLAN) VLAN tag. If packet is having only one tag then it would be considered as outer tag.

It supports the following function in Double VLAN processing feature using the MAC_VLAN_Incl, MAC_Inner_VLAN_Incl and MAC_VLAN_Tag registers.

- Insertion, replacement, or deletion of up to two VLAN tags in the Transmit path.
- Packet filtering and stripping based on any one of the two VLAN Tags in the Receive path. Stripping and providing up to two VLAN Tags in the Receive path as a part of the Receive status.

61.3.3.1.1 Transmit path

Following table describes the features supported by the MAC on the Transmit side.

Table 61-3. Double VLAN Processing Features in Transmit Path

Feature	Description
Support for C-VLAN and S-VLAN Tag types	The inner or outer VLAN tag can be of C-VLAN and S-VLAN type. The VLAN type is specified through the CSVL bit of MAC_VLAN_INCL and MAC_INNER_VLAN_INCL registers. The module supports processing of any sequence of outer and inner VLAN tags. **NOTE:** The module does not support the C-VLAN S-VLAN sequence. The MAC does not check whether the packet provided by the application has a valid sequence of the VLAN Tag types or the insertion or replacement operation results in invalid sequence of VLAN Tag type. Therefore, the application must provide correct sequence of VLAN Tag types and program the MAC in such a way that it results in correct sequence of VLAN Tag types in the transmitted packet. The application must ensure the following: • The inner tag should not be S-VLAN when outer C-VLAN Tag insertion is enabled. • The outer tag should not be C-VLAN when inner S-VLAN Tag insertion is enabled. • The inner tag should not be S-VLAN when outer tag should be replaced with C-VLAN. • The outer tag should not be C-VLAN when inner tag should be replaced with S-VLAN.
VLAN Tag deletion	VLAN tag deletion for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN deletion is enabled, the MAC deletes the tag present at the corresponding position. When a packet has only one tag, it is considered as the outer tag. If inner tag deletion is enabled and the packet has only one tag, the MAC does not delete the inner tag.
VLAN Tag Insertion or Replacement	VLAN tag insertion or replacement for outer or inner tag can be enabled through VLC field in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register, respectively. When VLAN tag insertion or replacement is enabled, the VLTI bit in the MAC_VLAN_Incl or MAC_Inner_VLAN_Incl register is used to determine whether the VLAN tag should be taken from the register or the Control Word.

61.3.3.1.2 Receive path

Following table describes the features supported by the MAC on the receive side and the corresponding bits in the MAC_VLAN_Tag register.

Table 61-4. Double VLAN Processing in Receive Path

Feature	Description
Outer or inner VLAN tag-based filtering	The MAC can filter packets based on the outer or inner VLAN tag through the ERIVLT bit.
C-VLAN or S-VLAN tag-based filtering	The MAC can filter packets based on the C-VLAN or S-VLAN type based on the ERSVLM bit.
Outer and Inner VLAN Tag stripping	The MAC can strip the outer and inner VLAN Tags from received frame based on the EVLS and EIVLS bits.
16-bit outer and inner VLAN Tag and Type in Rx status	The MAC can provide the 16-bit outer and inner VLAN Tag and Type in the Rx status based on the EVLRXS and EIVLRXS bits, respectively.
Disabling or skipping checking of outer VLAN Tag type	The MAC can disable or skip checking of outer VLAN Tag type to match C-VLAN or SVLAN based on the DOVLTC bit.

61.3.3.1.3 Source address and VLAN insertion, replacement, or deletion

The module supports the SA and VLAN insertion, replacement and deletion. The source address (SA) and VLAN fields are Tx packet-related control information that are provided as part of the control word through interfaces. IP supports the feature to insert or replace the source address based on the information in the MAC Address Registers, and also the feature to insert, replace or delete the VLAN fields (VLAN Type and VLAN Tag) based on the setting of the VLTI bit in the MAC_VLAN_Incl register. SA insertion or replacement feature can be enabled for all Transmit packets or selective packets. Similarly, VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets or selective packets.

61.3.3.1.4 Programming source address insertion or replacement

The software can use the SA insertion or replacement feature to instruct the MAC to do the following for Tx packets:

- Insert the content of the MAC Address Registers in the SA field
- Replace the content of the SA field with the content of the MAC Address Registers

When SA insertion is enabled, the application must ensure that the packets sent to the MAC do not have the SA field. The MAC does not check whether the SA field is present in the Transmit packet and it inserts the content of MAC Address Registers in the SA field. Similarly, when SA replacement is enabled, the application must ensure that the SA field is present in the packets sent to the MAC.

The MAC replaces the six bytes following the Destination Address field in the Transmit packet with the content of the MAC Address Registers.

- SARC bit of MAC_Configuration register can be program to enable the SA insertion or replacement for all Tx packets.
- SA insertion for selective packet can be enabled by programming the SA Insertion Control field (bits [25:23] of TDES3) in the first Transmit descriptor of the packet. When Bit 25 of TDES3 is set, the SA Insertion Control field indicates insertion or replacement by MAC Address1 registers. When Bit 25 of TDES3 is reset, it indicates insertion or replacement by MAC Address 0 registers.

If MAC Address1 Registers are not enabled, the MAC Address0 registers are used for insertion or replacement irrespective of the value of the most-significant bit of the SA Insertion Control field.

61.3.3.1.5 Programming VLAN insertion, replacement, or deletion

The software can use the VLAN insertion, replacement, or deletion feature to instruct the MAC to do the following for Tx packets:

- Delete the VLAN Type and VLAN Tag fields
- Insert or replace the VLAN Type and VLAN Tag fields

Insertion or replacement is done based on the setting of VLTI bit in the MAC_VLAN_Incl register as described in the following table:

Table 61-5. VLAN Insertion or Replacement Based on VLTI Bit

Condition	Description
VLTI bit is set	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field depending on the Content of the VT field of Transmit context descriptor of the packet
VLTI bit is reset	The MAC inserts or replaces the following: • VLAN Type field (C-VLAN or S-VLAN as indicated by the CSVL bit of MAC_VLAN_Incl register) • VLAN Tag field with the VLT field of MAC_VLAN_Incl register

When VLAN replacement or deletion is enabled, the MAC checks if the VLAN Type field (0x8100 or 0x88a8) is present after the DA and SA fields in the Transmit packet. The replace or delete operation does not occur if the VLAN Type field is not detected in two bytes following the DA and SA fields. However, when VLAN insertion is enabled, the MAC does not check the presence of VLAN Type field in the Transmit packet and just inserts the VLAN Type and VLAN Tag fields.

VLAN insertion, replacement, or deletion feature can be enabled for all Tx packets by programming the VLC and VLP fields of MAC_VLAN_Incl register.

VLAN insertion, replacement, or deletion for selective packets can be enabled by programming the VTIR field of TDES2 Normal Descriptor.

In addition, the VLP (VLAN Priority control) bit must be reset in MAC_VLAN_Incl register (for outer VLAN) and MAC_Inner_VLAN_Incl register (in inner VLAN) for the MAC to take the control inputs from the host, depending on the configuration.

61.3.3.1.6 Queue/Channel Based VLAN Tag Insertion on Tx

This module supports channel/queue based VLAN tag insertion on all transmitted packets.

61.3.3.1.6.1 Accessing Queue/Channel Specific VLAN Tag Registers

Queue/Channel specific VLAN tag registers are accessed using indirect addressing through the MAC_VLAN_Incl register. VLAN type and tag value can be independently programmed for each queue/channel.

61.3.4 Packet Filtering

There are various packets filtering scheme available in MAC receiver. Filtering sequence in their precedence order of received packet is shown in the following figure.

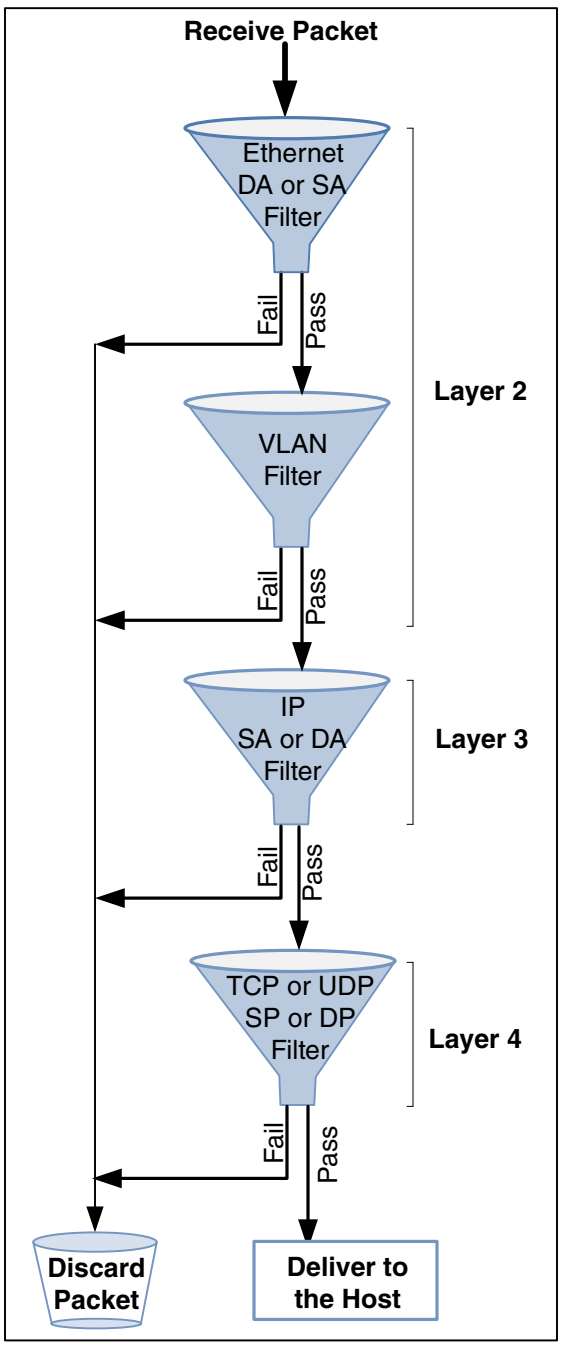


Figure 61-2. Packet Filtering Sequence

If any of the Layer filters are not enabled, that filter is bypassed and the subsequent filter is applied. A packet that fails any of the filters, is discarded. However, depending on the entry address and the number of parsed entries of the frame offset, discarded packets can be routed to RX_DMA_CH0 and a parser interrupt can be generated.

Even if no parser matched, the frame can be routed into the DMA channel indicated in DMA_CH_NO[2:0].

61.3.4.1 Source address or destination address filtering

The Address Filtering Module of the MAC checks the source address (SA) and destination address (DA) fields of each incoming packet. Programming of different type of address filtering is described in the following sections.

61.3.4.1.1 Unicast destination address filtering

MAC supports up to 64 MAC addresses for unicast perfect filtering. If perfect filtering is selected (HUC bit of MAC_Packet_Filter register is reset), the MAC compares all 48 bits of received unicast address with the programmed MAC address for any match. Default MacAddr0 is always enabled. The MacAddr1 to MacAddr63 addresses are selected with an individual enable bit. Each byte of MacAddr1 to MacAddr31 can be masked during comparison with corresponding received DA byte by setting the corresponding Mask Byte Control bit in the register. This enables group address filtering for the DA.

In hash filtering mode (when HUC bit is set), the MAC performs imperfect filtering for unicast addresses using a 64-bit Hash table. For hash filtering, the MAC uses the upper 6 bits CRC(by CRC32) of the received destination address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register, and a value of 111111 selects Bit 63 of Hash Table register. If the corresponding bit (indicated by the 6-bit CRC) is set to 1, the unicast packet is considered to have passed the Hash filter; otherwise, the packet is considered to have failed the Hash filter.

61.3.4.1.2 Multicast destination address filtering

PM bit of MAC_Packet_Filter register can be enabled to pass all multicast packets or disabled to do multicast addresses filtering based on HMC bit of MAC_Packet_Filter register. The multicast address is compared with the programmed MAC Destination Address registers (1–31). Also, Group address filtering is supported.

In Hash filtering mode, the MAC performs imperfect filtering using a 64-bit Hash table. The MAC uses the upper 6-bits CRC(by CRC32) of received multicast address to index the content of the Hash table. A value of 000000 selects Bit 0 of selected register and a value of 111111 selects Bit 63 of the Hash Table register. If the corresponding bit is set to 1, the multicast packet is considered to have passed the Hash filter. Otherwise, the packet is considered to have failed the Hash filter.

61.3.4.1.3 Hash or perfect address filtering

HPF bit of MAC_Packet_Filter register must be configured for enabling or disabling the destination address filtering by either Hash filter or Perfect filter. HUC and HMC bit of MAC_Packet_Filter register must be configured to select the Hash filtering or Perfect filtering for Unicast/Multicast packet.

61.3.4.1.4 Broadcast address filtering

The MAC does not filter any broadcast packets by default. DBF bit of MAC_Packet_filter register must be set to reject the broadcast packets.

61.3.4.1.5 Unicast source address filtering

By default, MAC compares the source address field with the value programmed in the source address registers. Bit 30 of MAC address register [1-2] can be configure to use it for source address instead of destination address comparison. The MAC also supports group filtering with source address(SA) . You can filter a group of addresses by masking one or more bytes of the address. The MAC drops the packets that fail the SA filter if the SAF bit is set in MAC_Packet_Filter register. Otherwise, the result of the SA filter is given as a status bit in the Receive Status word. When the SAF bit is set, the SA filter and DA filter result is AND'ed to decide whether the packet needs to be forwarded. This means that the packet is dropped if either filter fails. The packet is forwarded to the application only if the packet passes both filters in-order.

61.3.4.1.6 Inverse filtering

For DA and SA filtering, you can invert the filter-match result at the final output by setting the DAIF and SAIF bits of MAC_Packet_Filter register. The DAIF bit is applicable for both Unicast and Multicast DA packets. The result of the unicast or multicast destination address filter is inverted in this mode. Similarly, when the SAIF bit is set, the result of unicast SA filter is reversed.

Following two tables summarize the DA and SA filtering based on the type of packets received.

Table 61-6. Destination Address Filtering

Packet Type	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA Filter Operation
Broadcast	1	X	X	X	X	X	X	Pass
	0	X	X	X	X	X	0	Pass
	0	X	X	X	X	X	1	Fail

Table continues on the next page...

Table 61-6. Destination Address Filtering (continued)

Unicast	1	X	X	X	X	X	X	Pass all packets
	0	X	0	0	X	X	X	Pass on Perfect/ Group filter match
	0	X	0	1	X	X	X	Fail on Perfect/ Group filter match
	0	0	1	0	X	X	X	Pass on Hash filter match
	0	0	1	1	X	X	X	Fail on Hash filter match
	0	1	1	0	X	X	X	Pass on Hash or Perfect/ Group filter match
	0	1	1	1	X	X	X	Fail on Hash or Perfect/ Group filter match
Multicast	1	X	X	X	X	X	X	Pass all packets
	X	X	X	X	X	1	X	Pass all packets
	0	X	X	0	0	0	X	Pass on Perfect/ Group filter match and drop Pause packets if PCF = 0x
	0	0	X	0	1	0	X	Pass on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	0	1	0	X	Pass on Hash or Perfect/ Group filter match and drop Pause packets if PCF = 0x
	0	X	X	1	0	0	X	Fail on Perfect/ Group filter

Table continues on the next page...

Table 61-6. Destination Address Filtering (continued)

								match and drop Pause packets if PCF = 0x
	0	0	X	1	1	0	X	Fail on Hash filter match and drop Pause packets if PCF = 0x
	0	1	X	1	1	0	X	Fail on Hash or Perfect/ Group filter match and drop Pause packets if PCF = 0x

Table 61-7. Source Address Filtering

Packet Type	PR	SAIF	SAF	SA Filter Operation
Unicast	1	X	X	Pass all packets.
	0	0	0	Pass status on Perfect or Group filter match but do not drop packets that fail
	0	1	0	Fail status on Perfect or Group filter match but do not drop packet
	0	0	1	Pass on Perfect or Group filter match and drop packets that fail
	0	1	1	Fail on Perfect or Group filter match and drop packets that fail

NOTE

When the RA bit of MAC_Packet_Filter register is set, all packets are forwarded to the system along with the correct result of the address filtering in the Rx Status.

61.3.4.2 VLAN Filtering

VLAN tag filter can be done by either perfect match or hash table. Depending upon the configuration, MAC can compare either lower 12 bits or all 16 bits of received VLAN tag for perfect match. MAC forwarded the VLAN tagged packet with match status while it drops all the packets which don't matches.

MAC uses 16-bit VLAN hash table for group address filtering based on the VLAN tag. If hash filtering is enabled then most significant 4-bit of CRC-32 of VLAN tag are used to index the content of the VLAN Hash Table Register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the VLAN tag of the packet matched and the packet should be forwarded. A value of 0 indicates that VLAN-tagged packet should be dropped.

The MAC also supports the inverse matching for VLAN packets. In the inverse matching mode, when the VLAN tag of a packet matches the perfect or hash filter, the packet should be dropped. If the VLAN perfect and VLAN hash match are enabled, a packet is considered as matched if either the VLAN hash or the VLAN perfect filter matches. When inverse match is set, a packet is forwarded only when both perfect and hash filters indicate mismatch.

Table below shows the different possibilities for VLAN matching and the final VLAN match status. When the RA bit of MAC_Packet_Filter register is set, all packets are received and the VLAN match status is indicated in the VF bit of RDES2 Normal Descriptor (Write-Back Format). When the RA bit is not set and the VTFE bit is set in MAC_Packet_Filter register, the packet is dropped if the final VLAN match status is Fail.

When VLAN VID is programmed to 0 in the VL field of MAC_VLAN_Tag register, all VLAN-tagged packets are considered as perfect matched but the status of the VLAN hash match depends on the VTHM and VTIM bits in MAC_VLAN_Tag register.

Table 61-8. VLAN Match Status

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
VID = 0	Pass	0	X	X	Pass
	Pass	1	X	0	Pass
	Pass	1	Fail	1	Pass
	Pass	1	Pass	1	Fail
VID!= 0	Pass	X	X	0	Pass
	Fail	0	X	0	Fail
	Fail	1	Fail	0	Fail
	Fail	1	Pass	0	Pass

Table continues on the next page...

Table 61-8. VLAN Match Status (continued)

VID	VLAN Perfect Filter Match Result	VTHM Bit	VLAN Hash Filter Match Result	VTIM Bit	Final VLAN Match Status
	Fail	0	X	1	Pass
	Pass	X	X	1	Fail
	Fail	1	Pass	1	Fail
	Fail	1	Fail	1	Pass

In this table, X represents any value.

61.3.4.3 VLAN Filter Fail Packets Queue

When VLAN filtering is enabled, the VLAN Filter Fail Packets can be routed to a programmable Queue (VFFQ) when Bit RA = 1 or VTFE = 0 of MAC_Packet_Filter register and the enable bit (VFFQE of MAC_RXQ_CTRL4) for the queue is set.

The packets that passes the VLAN filtering, are routed to Rx queues based on the VLAN TAG priority field by programming the PSRQ in the corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers. The packets that fail the VLAN filter are discarded if RA=0 or VTFE=1. However, when RA=1 or VTFE=0, the VLAN filter fail packets are still forwarded to the application. In such a scenario, when the VLAN Filter Fail Queue Enable (VFFQE) bit is set, the VLAN filter fail packets are forwarded to the Rx queue number programmed in the VFFQ bit. If VFFQE=0, the Rx queue number is determined by the VLAN priority mapping as per the PSRQ fields.

Table below shows the Rx Queue routing table for Unicast tagged packets, with DA/SA Filter enabled.

Table 61-9. Rx Queue Routing table for Unicast Tagged Packets

RA	VFTE	SA/DA Filter Result	VLAN Filter Result	VFFQE	Queue Routing
X	X	PASS	PASS	X	PSRQ
0	0	PASS	FAIL	0	PSRQ
0	0	PASS	FAIL	1	VFFQ
0	X	FAIL	X	X	DROPPED
0	1	PASS	FAIL	X	DROPPED
1	X	FAIL	X	0	UFFQ*/PSRQ
1	X	FAIL	X	1	UFFQ*/VFFQ
1	X	PASS	FAIL	0	PSRQ
1	X	PASS	FAIL	1	VFFQ

X : Don't care condition * : When UFFQE is enabled else PSRQ

61.3.4.4 Extended Receive VLAN Filtering and Routing

When Extended Rx VLAN filtering and routing is enabled then both perfect filtering and Hash filtering can be enabled. The overall VLAN filter result is based on the Perfect filter result and Hash filter result (if enabled). Filter result is passed to application/software as part of the status bit. Extended Routing will happen only if VLAN filter has passed. Routing is only based on Perfect filter result. Each Perfect filter has a DMA channel enable and DMA channel number filed which should be programmed for routing.

For more details about routing, see details on 'Extended VLAN Based DMA Selection' in Dynamic (Per Packet) Mapping.

61.3.4.4.1 Comparison mode

Software/Application has following comparison option for each VLAN tag filter

- It can program the MAC to compare the inner or outer VLAN tag with either 12 bits or 16 bits programmed VID.
- It can choose that VID comparison is for SVLAN or CVLAN type frames, if type check is enabled for a filter.

61.3.4.4.2 Filtering

Perfect Filtering is done based on the MAC_VLAN_Tag_Filter registers. For each VLAN Tag Filter, the MAC will compare the relevant VLAN Tag ID and gives a result. If any one of the VLAN Tag Filters gives a match then the frame is considered to have passed the VLAN Tag Filters. If the frame mismatches all the filters, then the frame is considered to have failed the VLAN filter. This behavior is applicable only when the Inverse Filtering is not enabled in MAC_VLAN_Tag_Ctrl Register.

If inverse Filtering is enabled and the frame has mis-matched all the relevant filters then it is considered to have passed the VLAN filter. If the frame matches any one of the relevant filters then it is considered as a fail. If none of the enabled filters can perform a comparison or if none of the filters are enabled, then the frame is bypassed to the application.

The overall filter result and the programming on the VTFE and RA bits of the MAC Filter Register determine if the frame will be dropped or forwarded to the application. If RA = 1 or VTFE = 0, it does not matter if the filter result is a pass or fail. The frame is

always forwarded. If RA = 0 and VTFE = 1, only then, if the VLAN Tag Filter result is a pass does the MAC forward the frame. If the frame is forwarded to the application, then the relevant filter result is indicated through the Status bits.

61.3.4.4.3 Filter Status

The Extended Receive VLAN Filtering and Routing feature provides two status bits to indicate the comparison result of the VLAN tags.

By default, the MAC indicates the VLAN Filter Status through one bit in the status – VF field in RDES2. When Extended RX VLAN Filtering and Routing is enabled, two status bits are used to indicate the comparison result of VLAN tags. The Outer VLAN Tag Filter Pass and Inner VLAN Tag Filter Pass bits are defined in the following positions. The status indicated through these bits is highly dependent on the programming as explained below.

In RDES2:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In ARI status: MAC Filter Status:

- Bit 15 – Outer VLAN Tag Filter Status
- Bit 14 – Inner VLAN Tag Filter Status

In MRI Status:

- Bit 47 – Outer VLAN Tag Filter Status
- Bit 46 – Inner VLAN Tag Filter Status

Outer VLAN Tag Filter Status (OTS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Outer VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Outer VLAN Tag has either failed the relevant Outer VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Outer VLAN Tag Comparison, then this bit is reset.

- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Outer VLAN Tag Comparison.
- This bit is valid for both Single and Double VLAN Tagged frames.

Inner VLAN Tag Filter Status (ITS)

- In perfect Filtering, without inverse filtering enabled, if this bit is set, it indicates that the frame's Inner VLAN Tag has matched one of the VLAN Tag Filters.
- If this bit is reset, it indicates that the frame's Inner VLAN Tag has either failed the relevant Inner VLAN Tag Filters or bypassed them.
- If none of the filters are enabled for Inner VLAN Tag Comparison, then this bit is reset.
- If Inverse Filtering is enabled and this bit is set, then the frame's VLAN Tag has passed all the relevant VLAN Tag Filters. If it is reset, then it has failed at least one of or bypassed all the Filters programmed for Inner VLAN Tag Comparison.
- This bit is valid for only Double VLAN Tagged frames, when Double VLAN Processing is enabled.

The application must look at the status bits and the programming to determine if the Frame has passed or failed the VLAN Filter.

Following two tables 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled' and 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' show the possible Filter combinations and the corresponding filter results. These tables explain the scenarios when Double VLAN Processing and Hash VLAN filter are enabled in the design.

Legend for the table 'OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled'

- VTIM: VLAN Tag Inverse Match Enable – bit 17 in VLAN_Tag_Ctrl Register.
- HFO: Hash Filter enabled for Outer VLAN Tag Comparison - bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- HFI: Hash Filter enabled for Inner VLAN Tag Comparison – bit 25 and bit 27 in VLAN_Tag_Ctrl Register.
- PFO – Perfect Filter comparison enabled for Outer VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Outer VLAN Tag comparison (bit 20 is set to 0).

- PFI – Perfect Filter comparison enabled for Inner VLAN Tag - Any of the MAC_VLAN_Tag_Filter Registers is enabled (bit 16 is set) and programmed for Inner VLAN Tag comparison (bit 20 is set to 1).
- OTS – Outer VLAN Tag Filter Status
- ITS – Inner VLAN Tag Filter Status

Following table shows the possible values of status bits (OTS and ITS) when at least one Perfect filter is enabled.

Table 61-10. OTS and ITS Bit Values with At Least 1 Perfect Filter Enabled

VTIM	HFO	HFI	PFO	PFI	OTS	ITS
0	0	0	0	1	0	1/0
0	0	0	1	0	1/0	0
0	0	0	1	1	1/0	1/0
0	1	0	1	1	1/0	1/0
0	1	0	1	0	1/0	0
0	1	0	0	1	1/0	1/0
0	0	1	1	1	1/0	1/0
0	0	1	1	0	1/0	1/0
0	0	1	0	1	0	1/0
1	0	0	0	1	0	1/0
1	0	0	1	0	1/0	0
1	0	0	1	1	1/0	1/0
1	1	0	1	1	1/0	1/0
1	1	0	1	0	1/0	0
1	1	0	0	1	1/0	1/0
1	0	1	1	1	1/0	1/0
1	0	1	1	0	1/0	1/0
1	0	1	0	1	0	1/0

Following table shows the possible values of status bits (OTS and ITS) when none of the perfect filters are enabled and only the VLAN Hash Filter is enabled.

Table 61-11. OTS and ITS Bit Values with Only VLAN Hash Filter Enabled

VTIM	HFO	HFI	OTS	ITS
0	0	0	0	0
0	1	0	1/0	0
0	0	1	1/0	1/0
1	0	0	1/0	0
1	1	0	1/0	0

Table continues on the next page...

Table 61-11. OTS and ITS Bit Values with Only VLAN Hash Filter Enabled (continued)

VTIM	HFO	HFI	OTS	ITS
1	0	1	1/0	1/0

With no perfect filters enabled, any VLAN packet is considered to have bypassed the perfect filter. If Hash Filter is enabled for one of the Tags, then the respective Status bit depends on the Filter's result. The Status bits are set to 0 if VLAN Hash Filter is also not enabled.

If the value of ITS/OTS is shown as 1/0; then it indicates that the final result is dependent on the enabled relevant filter's result.

Example 1: The second row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that at least one Perfect Filter is enabled for Outer VLAN tag comparison and none of the filters are enabled for Inner VLAN tag comparison. Inverse VLAN Filtering is not enabled. The bit OTS is given as 1/0. If the received frame passes at least one of the enabled Outer VLAN Tag filters then the bit is set to 1. If the frame doesn't pass any of the enabled Outer VLAN Tag filters, then the bit is set to 0.

Example 2: Last Row of table 'OTS and ITS Bit Values with Only VLAN Hash Filter Enabled' indicates that Inverse Filtering is enabled, Hash Filter and at least one perfect filter is enabled for Inner VLAN Tag comparison, then if the received frame's Inner VLAN tag mismatches with both the Hash Filter and all the enabled Perfect filters, then the frame will have the ITS bit set to 1 else it is set to 0. OTS will be set to 0 as no comparison is performed.

61.3.4.4.4 Stripping

Each of the VLAN Tags has individual control over stripping. The programming options of always strip, never strip, strip on pass and strip on fail are available. Inner or Outer VLAN Tag Stripping is based on the pass or fail results of the individual tag. If a tag is bypassed by all the relevant filters, stripping is not applicable for the tag.

- If strip on Pass is enabled for the outer VLAN Tag, then the stripping occurs only if the Outer VLAN tag has passed the relevant filters. The Outer VLAN Tag Filter Result bit will be set.
- If strip on Fail is enabled for the outer VLAN Tag, then stripping occurs only if the Outer VLAN Tag has failed relevant filters. The Outer VLAN Tag Filter Result Bit will be reset.

- If the Outer VLAN tag of the received frame is bypassed by the entire filter (no comparison has been made), then the tag is not stripped, though the Status Bit is still 0.
- As multiple filters are enabled, it is possible that the received VLAN frame could have matched any one or more of the filters. The VLAN Tag's value is not always deterministic from the filter status bits.

If the software/application strips the VLAN Tag based on the filter result, it might lose the VID. So the suggested use is, if stripping is enabled for any of the tags, the tag can be put in the status. For this the software/application will have to enable the respective "VLAN Tag in Status" bit - 24 or 31 in the MAC VLAN Tag Control Register.

See section Programming Guidelines for Extended VLAN Filtering and Routing on Receive

61.3.4.5 Layer 3 and Layer 4 Filtering

IP supports the Layer 3 based packet filtering which is IP Source or Destination Address filtering in the IPv4 or IPv6 packets, and Layer 4 based packet filtering which is Source or Destination Port number filtering in TCP or UDP. When layer 3 and layer 4 packet filtering is enabled then packets are filtered in the following manner.

- Matched Packets: The MAC forwards the packets that match all enabled fields to the application along with the status. The MAC gives the matched field status only if the IPC bit of MAC_Configuration register is set and one of the following conditions is true:
  - All enabled Layer 3 and Layer 4 fields match
  - At least one of the enabled field matches and other fields are bypassed or disabled

When multiple Layer 3 and Layer 4 filters are enabled, any filter match is considered as a match. If more than one filter matches, the MAC provides the status of the lowest filter where Filter 0 is the lowest filter and Filter 3 is the highest filter.

- Unmatched Packets: The MAC drops the packets that do not match any of the enabled fields. You can use the inverse match feature to block or drop a packet with specific TCP or UDP over IP fields and forward all other packets.

- Non-TCP or UDP IP Packets: By default, all non-TCP or UDP IP packets are bypassed from the Layer 3 and Layer 4 filters. MAC can optionally be programmed to drop all non-TCP or UDP over IP packets.
- IP has a control register “MAC_L3_L4_Control” to control the Layer L3 and L4 packet filtering along with address registers to program the Layer L3 and L4 fields to be matched.

61.3.4.6 Flexible Receive Parser

When this function is enabled, all incoming packets are parsed as per the programmable instruction stored in memory. It does the following function

- Packet Filter (Accept or Reject)
- DMA Channel Selection

Flexible receive parser block operates over the first 128 bytes data of receive packet based on 64 bytes of instruction on each packet. Following figure shows the functional block diagram of Flexible Receive Parser.

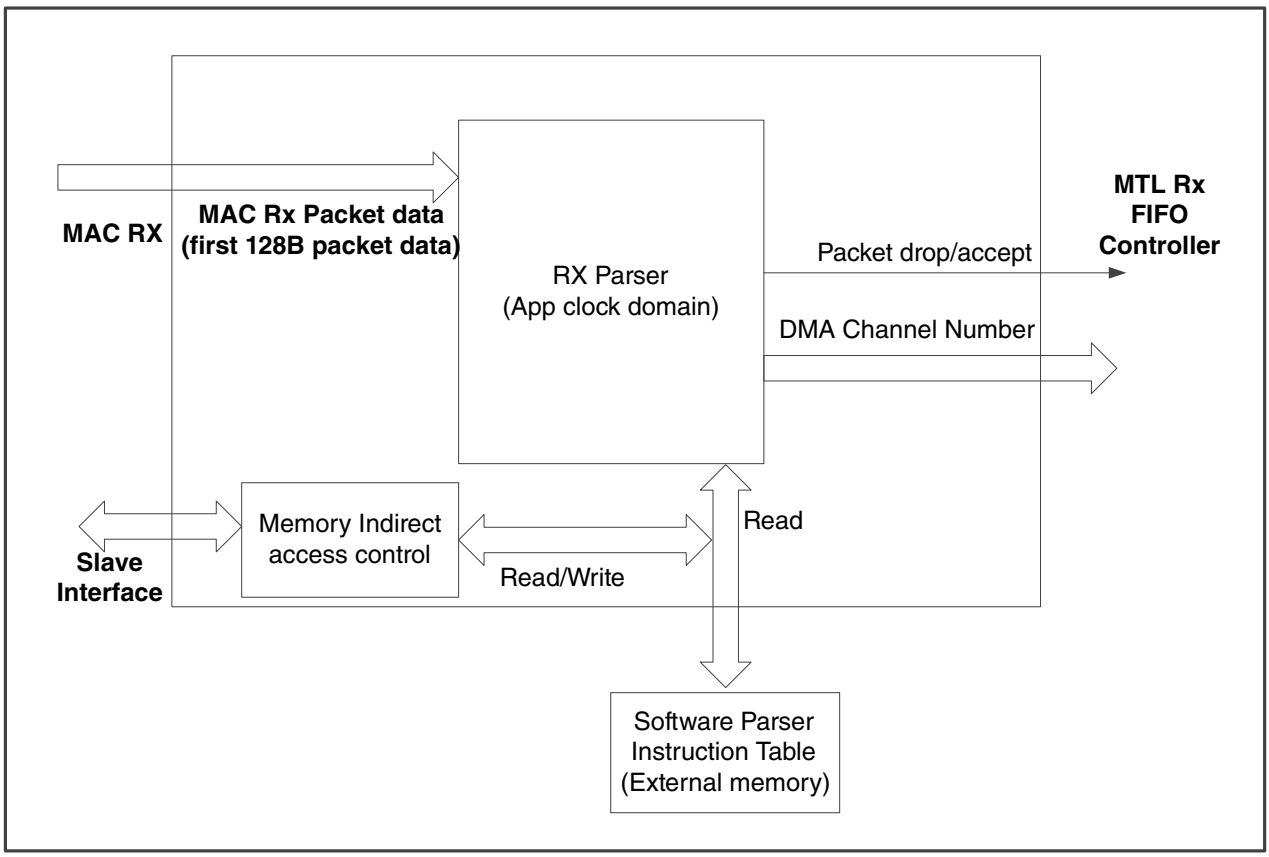


Figure 61-3. Functional block diagram of flexible recieve parser

61.3.4.6.1 Instruction Table Format of the Flexible Receive Parser

Software must build the 96-bit wide instruction before enabling the receive parser feature. Each instruction is programmed as an entry in the instruction memory.

The format of each entry in the Instruction table is shown in the following table.

Table 61-12. Rx Parser Instruction Entry Format

Field	Field Name	Description
31:0	MATCH_DATA	The 4-byte data. This data is used for comparing with incoming packet data starts at the frame offset as defined in [79:72] of the entry. The comparison is done only on those bits whose corresponding mask bits are set to 1 (MATCH_EN bits of [63:32]). See the Notes (below this table) on the byte order relative to Ethernet arrival data

Table continues on the next page...

Table 61-12. Rx Parser Instruction Entry Format (continued)

Field	Field Name	Description
		and endianness supported by QoS IP on the slave interface. The programming of this instruction table should adhere to the same.
62:32	MATCH_EN	When the MATCH_EN is set to 1, the corresponding Packet data bit is used for comparing. Otherwise, corresponding data bit is don't care.
64	AF	Accept frame
65	RF	Reject frame
66	IM	Inverse match
67	NIC	Next instruction control
71:68	Reserved	-
79:72	Frame Offset	[77:72] indicates the frame offset in terms of 4 bytes. (Here [79:78] always 0)Frame offset in the packet data to be compared for Match.This is in terms of 4 bytes. Value Actual Frame offset in bytes 0 01 42 8… ..63 252 The max value programmed in this Instruction Table is 128 B. The 'Actual Frame Offset' should not cross this limit.
87:80	OK Index	If (NIC==0) Memory Index to be used next when ENTRY_MATCH==1 and neither AF or RF is set to 1. If (NIC==1) Memory Index to be used next when ENTRY_MATCH==0 Note: ENTRY_MATCH is a calculated value based on MATCH_DATA, MATCH EN, and IM. It is not a field.
92:88	DMA CH NO	Indicates the DMA Channel Number (1-bit for each). This will be used when ENTRY_MATCH==1 and AF==1 (Accept frame). The encoding is as follows: • bit[0]- DMA Channel Number 0 • bit[1]- DMA Channel Number 1 • bit[2]- DMA Channel Number 2 • bit[3]- DMA Channel Number 3 • bit[4]- DMA Channel Number 4. Note: The DMA Channel number is encoded using bit wise as QoS IP support multicasting/broadcast across DMA Channels. See section 3.3.12 for more details and proper usage.
128:93	Reserved	0 (This field is only for software view and reserved for future enhancements. The memory width remains as 93-bits for now)

The parser begins parsing from the 0th entry, for each received packet. The subsequent parsing entry location (next entry or OK_INDEX[]) depends on the parser result of the current entry. Following figure shows the detailed flow.

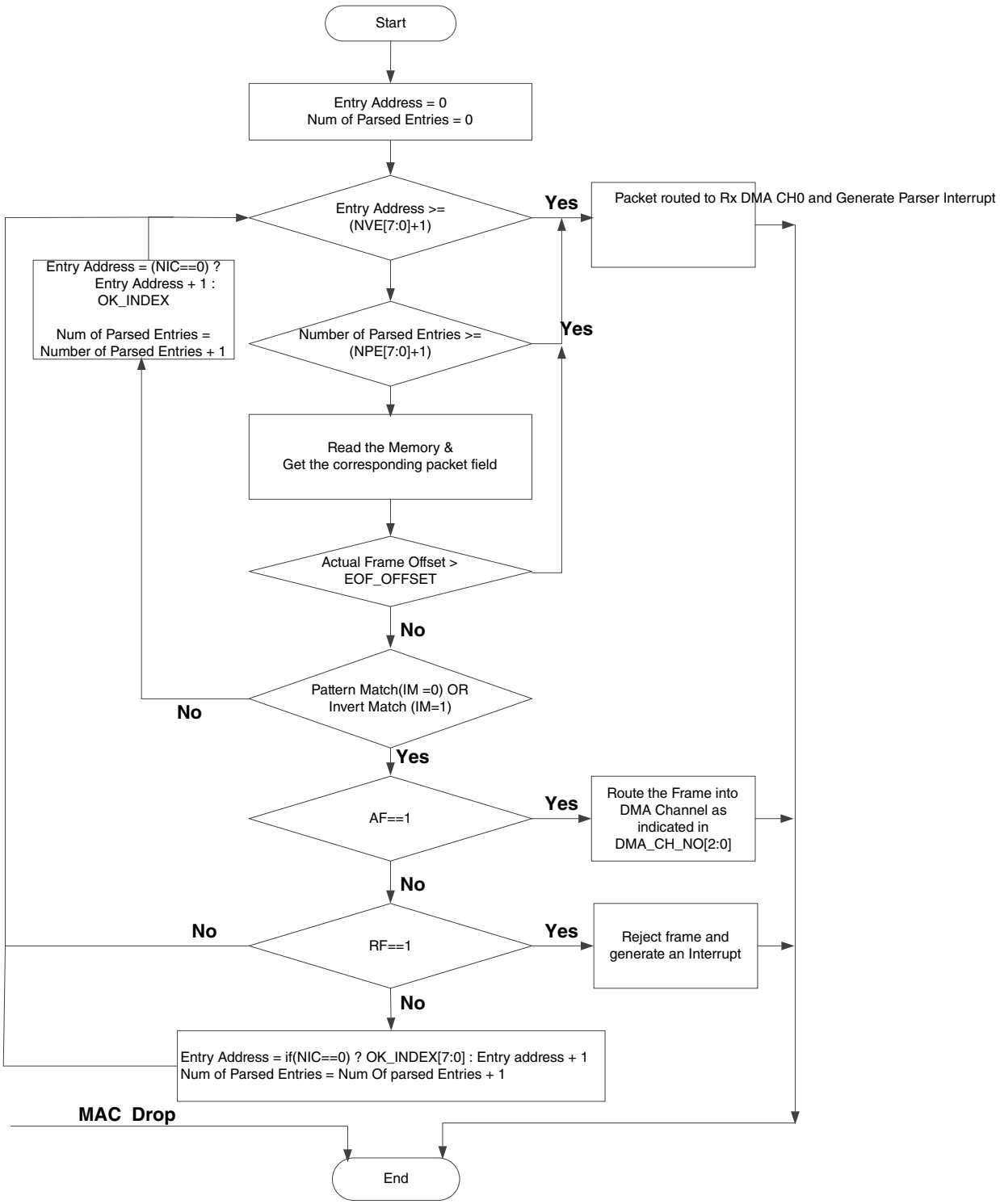


Figure 61-4. Flexible recieve parser flow chart

In the above figure, MAC Drop means the packet is dropped by the MAC due to another filter condition (for example, DA filter) and will not reach the Flexible Receive Parser.

The EOF_OFFSET shown in the figure is equal to the least among the following:

- Actual Packet size
- 128 Byte when frame pre-emption feature is not enabled/configured
- 128 Byte when frame pre-emption feature is enabled and packet is pre-emptible.
- 64 Byte when frame pre-emption feature is enabled and packet is Express Packet

61.3.4.6.2 Number of Valid Entries (NVE)

NVE indicates the number of valid entries in the instruction table. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NVE[7:0].

While parsing, if the Entry address is found to be is greater than NVE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[0], NVEOVIS (Number of Valid Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status)).

61.3.4.6.3 Number of Parsable Entries (NPE)

NPE indicates the number of entries that can be parsed on an incoming packet. The default value is 64. Software can change this value by writing into MTL_RXP_Control_Status registers field NPE[7:0] (Number of Parsable Entries) bits [21:16].

While parsing, if the number of parsed entries is more than NPE[7:0]+1, an error is flagged in the MTL_RXP_Interrupt_Control_Status register bit[1], NPEOVIS (Number of Parsable Entries overflow status). IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

While NVE[] indicates is the number of valid entries in the Rx parser memory built by software, NPE[] indicates the worstcase parsable entries considering the various possible paths that the parser can take. (NVE[] >= NPE[]).

61.3.4.6.4 Frame Offset

It indicates the frame offset, in terms of 4 bytes which is used while comparing against the current table entry.

In an entry, an error is flagged in the TL_RXP_Interrupt_Control_Status register bit[2], FOOVIS (Frame Offset Overflow Interrupt Status) in any of the following cases:

When Frame Offset is more than

- Packet size
- 128B if pre-emption is disabled
- 128B if pre-emption is enabled and packet is pre-empitble
- 64B if pre-emption is enabled and packet is an Express packet

IP generates an interrupt status (MTL_Interrupt_Status bit[23]) and MTLPIS (MTL Parser Interrupt Status).

NOTE

If the frame ends after the (frame_offset *4) bytes but before ((frame_offset *4)+4) bytes, the Receive parser declares Frame Offset Error if comparison is enabled (via MASK bits) for the non-received bytes; from (frame_offset *4) to ((frame_offset *4)+4).

61.3.4.6.5 Frame Reject

In cases where an Entry matches (considering the Inverse match, IM bit in the instruction table) and Frame Accept bit is not set, and Frame Reject bit (RF) is set, the frame is dropped.

When a frame is dropped due to RF = 1, IP generates an interrupt status (MTL_Interrupt_Status, bit[23], MTLPIS (MTL Parser Interrupt Status).

61.3.4.6.6 Out of Order Processing

The Rx parser flow can be out of order. The decision tree need not be based on the order in which packet arrives. That is, the decision tree can be out of order; the decision tree need not be based on the same order as the packet data arrival (data within one packet). This implies that the next entry and next frame offset (OK_INDEX[]/ FRAME_OFFSET[]) can be less than the current entry address and current Frame Offset.

61.3.4.6.7 DMA Channel Selection

When the Rx parser is enabled (FRPE bit[16] is set to 1 in MTL_Operation_Mode register) the Rx parser overrides the DMA selection criteria of the MAC.

When the Rx parser is disabled (FRPE bit[16] is set to 0 in MTL_Operation_Mode register) the MAC/MTL determines the DMA channel number.

61.3.4.6.8 Receive Queue Selection

The Rx Queue is determined by the existing MAC functionality, irrespective of the enabling/disabling of the Rx Parser. This is because the Rx parser is mainly designed for selecting the DMA channel and Rx queuing is decided by the MAC to enable the proper functioning of the Pause/PFC feature.

NOTE

When there are multiple queues and multiple DMA Channels are enabled, a particular DMA channel might receive out of order packets. This is because queue selection is based on MAC criteria (VLAN Priority and other criteria) and Rx DMA channel selection is based on Rx parser.

61.3.4.6.9 MAC Packet Filtering/Drop/Error Handling

The MAC does the packet filtering based on the received packet fields. Disable the Packet Filtering features inside the MAC when the Rx parser is enabled. To disable the Packet Filtering features inside the MAC,

1. Set Promiscuous Mode (PR, bit[0] of MAC_Packet_Filter register)
2. Set all other bits in the MAC_Packet_Filter register to its default values

When MAC decides to drop the packet (due to Rx MAC dependent Error such as GMII Error, Receive Watch Dog Error, CRC Error and Giant Frame Error) the packet is dropped irrespective of the Rx parser decision.

61.3.4.6.10 PAD Strip/CRC Strip Handling

The pad strip and CRC strip are controlled by the software and applicable to all the received packets. When these features are enabled, software must build the Rx parser instructions accordingly.

NOTE

Enable PAD and CRC stripping in real use case.

61.3.4.6.11 VLAN Strip Handling

The MAC supports stripping the VLANs, outer as well as inner VLAN. When MAC is programmed to strip the VLAN, the Rx parser considers those VLAN tags to be part of the incoming packet.

61.3.4.6.12 Multicast and Broadcast Support

IP supports multicast and broadcast packet and the packet is routed to the highest queue number.

When Rx Parser is enabled, the packet routing is decided by the DMA Channel Numbers (DMA CH NO, Bits [95:88]) in the Instruction Table.

When Rx Parser is disabled, the packet routing is determined by DCS (DMA Channel Select) field of the MAC_AddressX_High register. For more details, see Broadcast/ Multicast Packet Duplication.

NOTE

DCS/DMA CH NO field is per DMA Channel control and therefore, multiple DMA channels can be selected for routing the packet.

61.3.4.6.13 Pre-emption Support

The MAC supports the frame pre-emption feature when the Flexible Rx parser feature is enabled. The minimum packet size for Frame Preemption is 64 bytes. This means that:

- The Rx parser can operate on 64 bytes of Express traffic.
- The Rx parser can operate on 128 bytes of pre-emptible traffic, as a 64 bytes packet of pre-emptible traffic might be interrupted by Express traffic before that packet is fully received, so the parser needs to have 64 bytes on either side of the express traffic in order to capture the full 64-bytes of a pre-empted packet.

61.3.4.6.14 Software Access to the Flexible Receive Parser Memory

Software can read/write into the Rx parser memory, using the following registers via indirect addressing.

- MTL_RXP_Indirect_Acc_Control_Status
- MTL_RXP_Indirect_Acc_data

For more details, see ENET_QOS register descriptions.

61.3.4.6.15 Statistical Counters

IP supports the following statistics counters for flexible receive parser.

1. MTL_RXP_Drop_Cnt
2. DMA_RXP_Error_Cnt

DMA_CH(#i)_Rxp_Accept_cnt (i=0; i<2)

For more details, see ENET_QOS register descriptions.

61.3.4.6.16 Changing the Instruction Table by Software

Software can update the instruction table in the memory in the following ways

- Disable the MAC Receiver and Receive Parser by setting the RE field of the MAC_Configuration register to 0 and FRPE bit of MTL_Operation_mode register to 0. Wait for Rx Parser to become inactive (RXPIA, MTL_RXP_Control_Status register bit[31]).
- (Optional) Program in the Entry address 0 to unconditionally (all MATCH_EN bit 0) skip to OK_INDEX[] (to certain location in the memory) where it is programmed to reject or accept all packets.

61.3.4.6.17 Receive Cut-Through Functionality

In the case of cut-through mode, the Rx controller attempts to transfer the received packets to DMA just after the cut-through threshold amount of packet data is received (RTC field in the MTL_RxQx_Operation_Mode register).

However due to Rx parser's result worst case arrival time can be more than the programmed cut-through threshold. So, the Rx controller delays the packet transfer to DMA accordingly. So, the cut-through functionality is re-defined as follows

The Rx controller attempts to transfer packet to DMA after at least RTC number of bytes have been received and Rx parser results are available.

61.3.4.6.18 Receive Packet Drop Indication

In most cases, the packet drops internally in the Rx Queue. However, back-to-back packets can arrive for the same queue and first packet's Rx parser result is not available when the second packet arrives. In such cases, the packet cannot be flushed internally and

forwarded to DMA or the Application Interface and status is indicated accordingly. For more details, see RDES2, bit 16(RXPD) in Receive Normal Descriptor (Write-Back Format).

61.3.4.6.19 Runt Packet Handling

The Rx parser can perform the operation with 64B minimum size packet. If the runt packet is received (should be indicated by MAC), the Rx parser assumes that it gets dropped in the MAC. This is because, back-to-back runt packets (< 64B) cannot be handled in the Rx parser.

61.3.4.6.20 Uncorrectable ECC Error Handling

When the parser detects an uncorrectable ECC error while parsing the parser memory, the packet is not dropped. Instead, the packet is sent to the application with an Error Indication. The RXPI (RX Parser Incomplete) is set in the packet status along with Error Summary (ES) bit.

For more details, see Receive Normal Descriptor (Write-Back Format).

61.3.5 IEEE 1588 Timestamp Support

The MAC supports the IEEE 1588 Precision Time Protocol (PTP). The MAC is able to do precise time stamping and capturing of incoming and outgoing frame to enable support of PTP with its associated software. The MAC supports the all clock types defined in IEEE 1588-2008. Typical frequency of PTP timestamp clock is 125 MHz. The MAC supports:

- Taking a timestamp snapshot of all packets or only PTP type packets.
- Taking a timestamp snapshot of only event messages.
- Identifying the PTP message type, version, and PTP payload in packets sent directly over Ethernet and sending the PTP message type status given by the MAC. The PTP status is finally written back to the Receive Descriptor: RDES1 Normal Descriptor (Write-Back Format).
- Measuring sub-second time in decimal or binary format.

61.3.5.1 Delay Request-Response Mechanism

The system or network is classified into the master and slave nodes for distributing the timing and clock information. Following figure shows the process that PTP uses for synchronizing a slave node to a master node by exchanging PTP messages.

Figure 61-5. Networked Time Synchronization

As shown in the figure, PTP uses the following process:

1. The master broadcasts the PTP Sync messages to all its nodes. The Sync message contains the reference time information of the master. This message leaves the system of the master at t1. This time must be captured for Ethernet ports at the internal GMII or MII interface.
2. The slave receives the Sync message and also captures the exact time, t2, using its timing reference.
3. The master sends a Follow_up message to the slave, which contains t1 information for later use.
4. The slave sends a Delay_Req message to the master and notes the exact time, t3, at which this packet leaves the internal GMII or MII interface.
5. The master receives the message, capturing the exact time t4, at which the message enters its system.
6. The master sends the t4 information to the slave in the Delay_Resp message.
7. The slave uses the four values of t1, t2, t3, and t4 to synchronize its local timing reference to the timing reference of the master.

Most of the PTP implementation is done in the software above the Ethernet layer. However, the hardware support is required to capture the exact time when specific PTP packets enter or leave the Ethernet port at the internal GMII or MII interface. This timing information must be captured and returned to the software for proper implementation of PTP with high accuracy.

61.3.5.2 Peer-to-Peer PTP Transparent Clock (P2P TC) Message Support

The IEEE 1588-2008 supports peer-to-peer PTP (Pdelay) message in addition to the SYNC, Delay Request, Follow-up, and Delay Response messages. Following figure shows the method to calculate the propagation delay in clocks supporting peer-to-peer path correction.

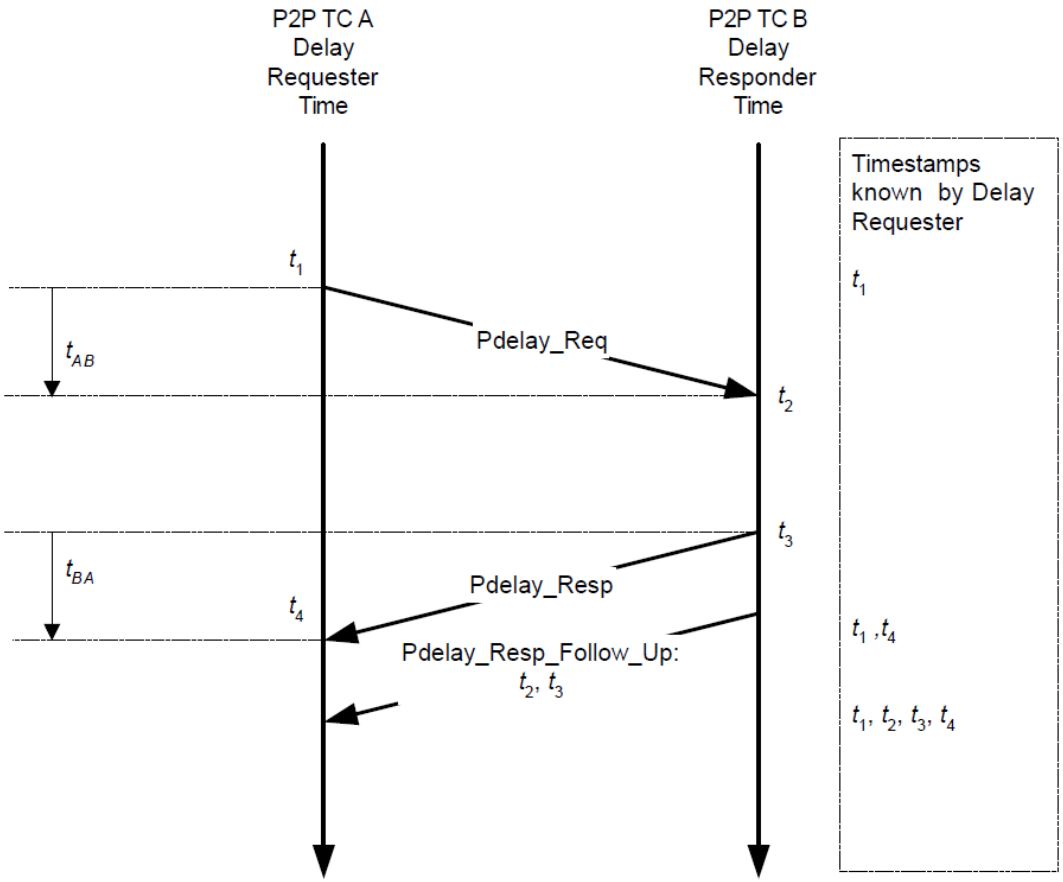


Figure 61-6. Propagation Delay Calculation in Clocks Supporting Peer-to-Peer Path Correction

As shown in the above figure, the propagation delay is calculated in the following way:

1. Port 1 issues a Pdelay_Req message and generates a timestamp (t1) for the Pdelay_Req message.
2. Port 2 receives the Pdelay_Req message and generates a timestamp (t2) for this message.
3. Port 2 returns a Pdelay_Resp message and generates a timestamp (t3) for this message. To minimize errors because of any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message. Port 2 returns any one of the following:
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp message
   - Difference between the timestamps t2 and t3 in the Pdelay_Resp_Follow_Up message
   - Timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages, respectively
4. Port 1 generates a timestamp (t4) on receiving the Pdelay_Resp message.

5. Port 1 uses all four timestamps to compute the mean link delay.

61.3.5.3 Timestamp Correction

According to the IEEE 1588 specification, a timestamp must be captured when the PTP message timestamp point (leading edge of the first bit of the octet immediately following the Start Frame Delimiter octet) crosses the boundary between the node and the network. As the MAC takes the timestamp at an internal point far from the actual boundary of the node and network, this captured timestamp is corrected/updated for the ingress/egress path latency (including the delay in the PHY layers). Further correction is done for the inaccuracies/errors introduced due to the clock (internal GMII or MII Tx, Rx clock) being different at the capture point as compared to the PTP clock (clk_ptp_ref_i) that is used to generate the time. The resultant CDC (Clock Domain Crossing) circuits add error depending on the clock period of the internal GMII or MII and PTP clocks.

61.3.5.3.1 Ingress Correction

In the Receive side the timestamp captured at the internal snapshot point is delayed (later in time) as compared to the time at which that packet's SFD bit is received at the port's boundary. Therefore, the captured timestamp must be reduced by the ingress latency and the errors in CDC sampling. This correction value must be determined/calculated by the software and written into the MAC_Timestamp_Ingress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between boundary point and the input of the core.

   If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum ingress latency. The software can read these registers and determine the average ingress latency in the PHY. Alternatively (if the PHY does not support these registers), the ingress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the input of the core to the internal capture point.

   The internal ingress latency can be read from the MAC_Ingress_Timestamp_Latency register. This is a read-only register and gives the latency in scaled Nanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency differs based on the active PHY interface (for example, RMII) and the operating speed. Therefore, the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC Synchronization

The CDC synchronization error is almost equal to 2 times the clock-period of the PTP clock (clk_ptp_ref_i).

The values determined from these 3 components should be added by the software and must be written into the TSIC and TSICSNS fields of the MAC_Timestamp_Ingress_Corr_* registers.

NOTE

The value written to the register must be negative (two's complement as explained below), because it has to be subtracted from the captured timestamp. The MAC Receiver adds the value in this register to the captured timestamp and then gives the resultant value as the timestamp of the received packet.

When TSCTRLSSR bit in MAC_Timestamp_Control register is set, the nanoseconds field of the captured timestamp is in decimal format with a granularity of 1ns. So the Bit31 of TSIC must be set to 1 (for negative value) and bits[30:0] must be written with "10^9 - total ingress_correction_value[nanosecond part]" represented in binary. For example, if the required correction value is -5 ns, then the value is 0xBB9A_C9FB.

When TSCTRLSSR bit in MAC_Timestamp_Control register is reset, the nanoseconds field of the captured timestamp is in binary format with a granularity of ~0.466ns. Therefore, bits[30:0] must be written with "2^31 - total ingress_correction_value" represented in binary with bit[31] = 1.

61.3.5.3.2 Egress Correction

In the Transmit side the timestamp captured at the internal snapshot point is earlier (advanced in time) as compared to the time at which that packet's SFD bit is output at the port's boundary. Therefore, the captured timestamp must be compensated by the egress latency and the errors in CDC sampling. This correction value must be determined/ calculated by the software and written into the MAC_Timestamp_Egress_Corr_* registers.

The correction value consists of the following 3 components:

1. External latency in the PHY layer between the output of the core and the boundary of the port and the network

If the PHY is compliant to the IEEE 802.3 Clause 45 MMD registers, it has a register indicating the maximum and minimum egress latency. The software can read these registers and determine the average egress latency in the PHY. Alternatively (if the PHY does not support these registers), the egress latency must be determined from its datasheet or timing characteristics.

2. Internal latency from the internal capture point and the output of the core

   This internal egress latency can be read from the MAC_Egress_Timestamp_Latency register. This is a read-only register and gives the latency in scaledNanoseconds format defined in IEEE 1588 Clause 5.3.2. The latency will differ based on the active PHY interface (for example, RMII) and the operating speed. Hence the software must read this register after any speed change in the MAC to determine the current internal latency.

3. CDC synchronization error

   The CDC synchronization error value for one-step timestamping would be = (1 * period of clk_ptp_ref_i + 4 * period of clk_tx_i). Otherwise (Two-step timestamping mode), the value = -(2 * period of clk_ptp_ref_i).

61.3.5.3.3 Frequency Range of Reference Timing Clock

The timestamp information is transferred across asynchronous clock domains, from the MAC clock domain to the application/system clock domain. Therefore, a minimum delay is required between two consecutive timestamp captures. This delay is 4 clock cycles of MII and 3 clock cycles of PTP clocks. If the delay between two timestamp captures is less than this delay, the MAC does not take a timestamp snapshot for the second packet.

61.3.5.3.4 PTP Processing and Control

Following table shows the common message header for the PTP messages. This format is taken from the IEEE 1588-2008.

Table 61-13. Message Format Defined in IEEE 1588-2008

Bits								Octet	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
Reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
Reserved								1	5

Table continues on the next page...

Table 61-13. Message Format Defined in IEEE 1588-2008 (continued)

Bits								Octet	Offset
7	6	5	4	3	2	1	0		
flagField								2	6
correctionField								8	8
Reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField (*)								1	32
logMessageInterval								1	33

(*) – control Field is used in version 1. In version 2, message Type field is used for detecting different message types.

There are some fields in the Ethernet payload that are being use to detect the PTP packet type and control the snapshot to be taken. These fields are given below in table for PTP packets over Ethernet.

Following table provides information about the fields that are matched to control the snapshots for the PTP packets sent over Ethernet for IEEE 1588 version 1 and 2. The octet positions for the tagged packets are offset by 4. This is based on the IEEE 1588-2008, Annex D and the message format.

Table 61-14. Ethernet PTP Packet Fields Required for Control And Status

Field Matched	Octet Position	Matched Value	Description
MAC Destination Multicast Address[1]	0–5	01-1B-19-00-00-00 01-80-C2-00-00-0E	All PTP messages can use any of the following multicast addresses[2]: • 01-1B-19-00-00-00 • 01-80-C2-00-00-0E[3]
MAC Packet Type	12, 13	0x88F7	PTP Ethernet packet
PTP Control Field (IEEE 1588 version 1)	46	0x00, 0x01, 0x02, 0x03, or 0x04	• 0x00: SYNC • 0x01: Delay_Req • 0x02: Follow_Up • 0x03: Delay_Resp • 0x04: Management
PTP Message Type Field (IEEE 1588 version 2)	14 (nibble)	0x0, 0x1, 0x2, 0x3, 0x8, 0x9, 0xB, 0xC, or 0xD	• 0x0: SYNC • 0x1: Delay_Req • 0x2: Pdelay_Req • 0x3: Pdelay_Resp

Table continues on the next page...

Table 61-14. Ethernet PTP Packet Fields Required for Control And Status (continued)

Field Matched	Octet Position	Matched Value	Description
			• 0x8: Follow_Up • 0x9: Delay_Resp • 0xA: Pdelay_Resp_Follow_Up • 0xB: Announce • 0xC: Signaling • 0xD: Management
PTP Version	15 (nibble)	0x1 or 0x2	• 0x1: Supports PTP version 1 • 0x2: Supports PTP version 2

1. The unicast address match of destination addresses (DA), programmed in MAC address 0 to 31, is used if the TSENMACADDR bit of MAC_Timestamp_Control register is set.
2. IEEE 1588-2008, Annex F
3. The MAC does not consider the PTP version 1 messages with Peer delay multicast address (01-80-C2-00-00-0E) as valid PTP messages.

61.3.5.4 Transmit Path Functions

The MAC captures a timestamp when the Start Packet Delimiter (SFD) of a packet is sent on the internal GMII or MII interface. The packets, for which the timestamps has to be captured can be controlled on per-packet basis. Each Transmit packet can be marked to indicate whether a timestamp should be captured for it. The MAC does not process the transmitted packets to identify the PTP packets. You can specify the packets by using the control bits in the Transmit descriptor. The MAC returns the timestamp to the software inside the corresponding Transmit descriptor, thus connecting the time-stamp automatically to the specific PTP packet. The 64-bit timestamp information is written to the TDES0 and TDES1 fields. The TDES0 field holds the 32 least significant bits of the timestamp.

61.3.5.5 Receive Path Functions

The MAC can be programmed to capture the timestamp of all packets received on the internal GMII or MII interface or to process packets to identify the valid PTP messages. Use the following options of the MAC_Timestamp_Control register to control the snapshot of the time to be sent to the application:

- Enable snapshot for all packets
- Enable snapshot for IEEE 1588 version 1 or version 2 timestamp
- Enable snapshot for PTP packets transmitted directly over Ethernet
- Enable timestamp snapshot for the received packet for IPv4 or IPv6
- Enable timestamp snapshot only for EVENT messages (SYNC, DELAY_REQ, PDELAY_REQ, or PDELAY_RESP)
- Enable the node to be a master or slave and select the snapshot type

Table 61-15. Timestamp Snapshot Dependency on Register Bits

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP Messages
00	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp,Pdelay_Req, Pdelay_Resp,Pdelay_Resp_Follow_Up
01	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10	x	x	SYNC, Delay_Req
11	x	x	Pdelay_Req, Pdelay_Resp

The DMA returns the timestamp to the software inside the corresponding Receive Descriptor. The extended status, containing the timestamp message status and the IPC status, is written in normal descriptor RDES1 and the snapshot of the timestamp is written in RDES0 and RDES1 fields of context descriptor. The RDES0 field holds the 32 least significant bits of the timestamp.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (System Time Correction), see System Time Correction

61.3.5.6 IEEE 1588 System Time Source

The ENET QOS block supports both external and internal time source for reference timestamping.

- External Time source uses the 64-bit external time reference and clock as input in IP. The clock input is used to synchronize the external timing reference into MAC clock domain. Upper 32-bit indicates the time in seconds while lower 32-bit indicates the time in nanoseconds.
- Internal time source takes only the reference clock input and uses it to internally generate the Reference time (also called the system time) and capture timestamps.
  - UInteger48 seconds Field: The seconds field is the integer portion of the timestamp in units of seconds. It is 48-bits wide.

    For example, 2.000000001 seconds are represented as seconds Field = 0x0000_0000_0002.

  - UInteger32 nano secondsField The nanoseconds field is the fractional portion of the timestamp in units of nanoseconds.

    For example, 2.000000001 seconds are represented as nanoSeconds = 0x0000_0001. The nanoseconds field supports the following two modes:

    - Digital rollover mode: In this mode, the maximum value in the nanoseconds field is 0x3B9A_C9FF, that is, (10e9-1) nanoseconds.
    - Binary rollover mode: In this mode, the nanoseconds field rolls over and increments the seconds field after value 0x7FFF_FFFF. Accuracy is ~0.466 ns per bit.

There is a system time register module which is being used when internal time reference is being used. The 80-bit time is maintained in this module and updated using the input reference clock (clk_ptp_ref_i). This time is the source for taking snapshots (timestamps) of Ethernet packets being transmitted or received at the internal GMII or MII interface.

The system time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register. For initialization, the system time counter is written with the value in the Timestamp Update register. For system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, the frequency offset and/or frequency drift of a slave clock (clk_ptp_ref_i) with respect to the master clock (as defined in IEEE 1588-2002) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals.

NOTE

For Programming Guidelines for IEEE 1588 Timestamping (For Internal Time-stamp Source Configuration). see Initialization Guidelines for System Time Generation

61.3.5.7 IEEE 1588 Higher Word Register

If system time source is internal, then higher word register can be invoked. MAC timestamp is 64-bit wide. The values of the upper 16-bits of the seconds field can be read from the CSR register.

61.3.5.8 Flexible Pulse-Per-Second Output

The ENET QOS block supports the flexibility of programming the start or stop time, width and interval of pulse generated on pulse per second (PPS) output. IP supports 4 such output signals. By default, the block is in fixed pulse per second output mode with an interval of 1 second.

Initially. the start time should be programmed in target time registers "MAC_PPSx_Target_Time_Seconds and MAC_PPSx_Target_Time_Nanoseconds" for all desired/enabled PPS output. Reprogramming the start or stop time must be done after the synchronization of the earlier programmed value. Bit 31 of the MAC_PPS#_Target_Time_Nanoseconds register indicates that the synchronization is complete. If the application programs a start or stop time that has already elapsed, the MAC sets an error status bit indicating the programming error. If enabled, the MAC also sets the Target Time Reached interrupt event. The application can cancel the start or stop request only if the corresponding start or stop time has not elapsed. If the time has elapsed, the cancel command has no effect.

The PPS width and interval are programmed in terms of granularity of system time, that is, number of the units of sub-second increment value. For example, to have a PPS pulse width of 40 ns and interval of 100 ns with PTP reference clock of 50 MHz, you should program the width and interval to values 2 and 5, respectively. You can achieve smaller granularity by using a faster PTP reference clock. Before giving the command to trigger a pulse or pulse train on the PPS output, you should program or update the interval and width of the PPS signal output.

NOTE

See Programming Guidelines for Flexible Pulse-Per-Second Output

61.3.5.9 Media Clock Generation and Recovery

Media clock recovery and generation needs a dedicated hardware and software to complete the operation. IP generates a reference clock with the help of dedicated hardware which is then multiplied to get the desired media clock. Software extracts the presentation time from the incoming PTP 1722 frames and programs these value into CSR registers so IP can read them for media clock recovery.

As shown in the following figure, the timestamp array and the DMA module are external to IP. IP supports a 32-bit presentation time counter in nanoseconds that wraps at the full32 bit value to match up with 1722 presentation time format and an handshake mechanism with external application(such as DMA) to program the presentation time into CSR register.

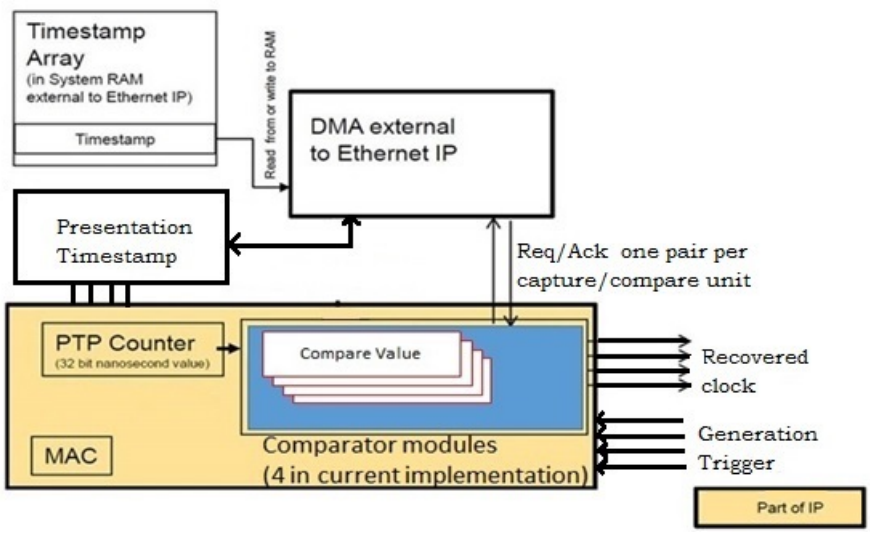


Figure 61-7. Block Diagram of Media Clock Generation and Recovery

61.3.5.9.1 Presentation Time Counter

Presentation time counter is another view of the PTP System time (1588 timer). For a given PTP System time (PTP[63:32] represents time in seconds and PTP[31:0] represents time in nanoseconds), there exists a corresponding Presentation Time (32-bit value in nanosecond). The Presentation time computed is referred to as Current Presentation Time (CPT).

The Current Presentation Time (CPT) can be viewed as derived from 64bit PTP System Time. PTPNS[63:0] = (PTP[63:32] * 32'd1,000,000,000) + PTP[31:0] where, PTPNS is the PTP system time converted into a 64bit nanosecond value.

Current Presentation Time[31:0] = PTPNS[31:0]

Media Clock Recovery is possible when the System Time is internally generated in IP and CPT is computed in the same way as the Internal PTP System Time generation. Because CPT and System Time values are in nanoseconds, and must be synchronous, the increment cycle of both the timers are same. The timer is updated at the same instance, but the updated value could be different. The software must compute a separate 32-bit update value in nanoseconds for the CPT update. CPT sampled at different edges of a triggering input results in the generation of Media Clock timestamps which are inserted in 1722 based AVBTP packets, to recover the clock at the destination.

61.3.5.9.2 Comparator Modules

As shown in the figure 'Block Diagram of Media Clock Generation and Recovery' the module supports four comparator modules to handle multiple media clocks that might be required. The Comparator modules hand shake with the application to

Program the timestamps into the MAC_PPS(#i)_Target_Time_Seconds register when PTGE field of MAC_Timestamp_Control register is set.

Set the Presentation Time Control bits of the particular instance for recovery mode.

The timestamps received from the application are referred as Target Presentation Time (TPT).

When MCGREN#i field of MAC_PPS_Control register is set to 1, the IP operates in MCGR mode. The comparator module sends up to two requests to the application for TPT write, when PTGE field of MAC_Timestamp_Control register is set and the Presentation time control bits of the particular instance is set for MCGR mode. Subsequent requests are generated each time a Presentation Time match occurs. (CPT transitions from a value less than TPT, to a value greater than or equal to CPT). The first request is asserted when a specific comparator instance is set to MCGR mode with a non-zero Presentation Time Control and an additional request is made when the first data is received into the Comparator. This allows the application to write the next TPT value while IP processes the previous TPT value for a match.

The TPT read from the MAC_PPS(#i)_Target_Time_Second is considered as a future time. A toggle/pulse is generated in the next cycle when a match is detected. When a match is detected, the mcgr_dma_req_o#i request for that comparator is asserted (to obtain next TPT) until the corresponding application acknowledgment is set.

The presentation control bits (PPSCMD#i field of MAC_PPS_Control register) determine the shape of the generated waveform. Also, these bits can be programmed to either toggle or generate a high/low pulse for one PTP clock cycle, upon match.

Media Clock Recovery: A match is when the free-running CPT value matches the received TPT value. A output signal EMAC_TMR signal as shown in the IO mux sheet is asserted (toggle, low pulse, or high pulse) based on the presentation control value programmed.

Media Clock Generation: Based on the presentation control value programmed in MAC_PPS_Control register, the particular comparator captures the presentation time and programs it into MAC_PPS(#i)_Target_Time_Second register. A request is raised to the application to read the captured timestamp. Until the acknowledgment is received from the application, acknowledging that the read operation is complete, no new timestamps are captured.

61.3.5.9.3 Media Clock Generation and Recovery Flow

Following figure shows the media clock generation and recovery flow. For both media clock generation and recovery, set the PTGE field of MAC_Timestamp register for CPT generation and set the MCGREN#i field of MAC_PPS_Control Register for the corresponding instance to be enabled in the MCGR mode.

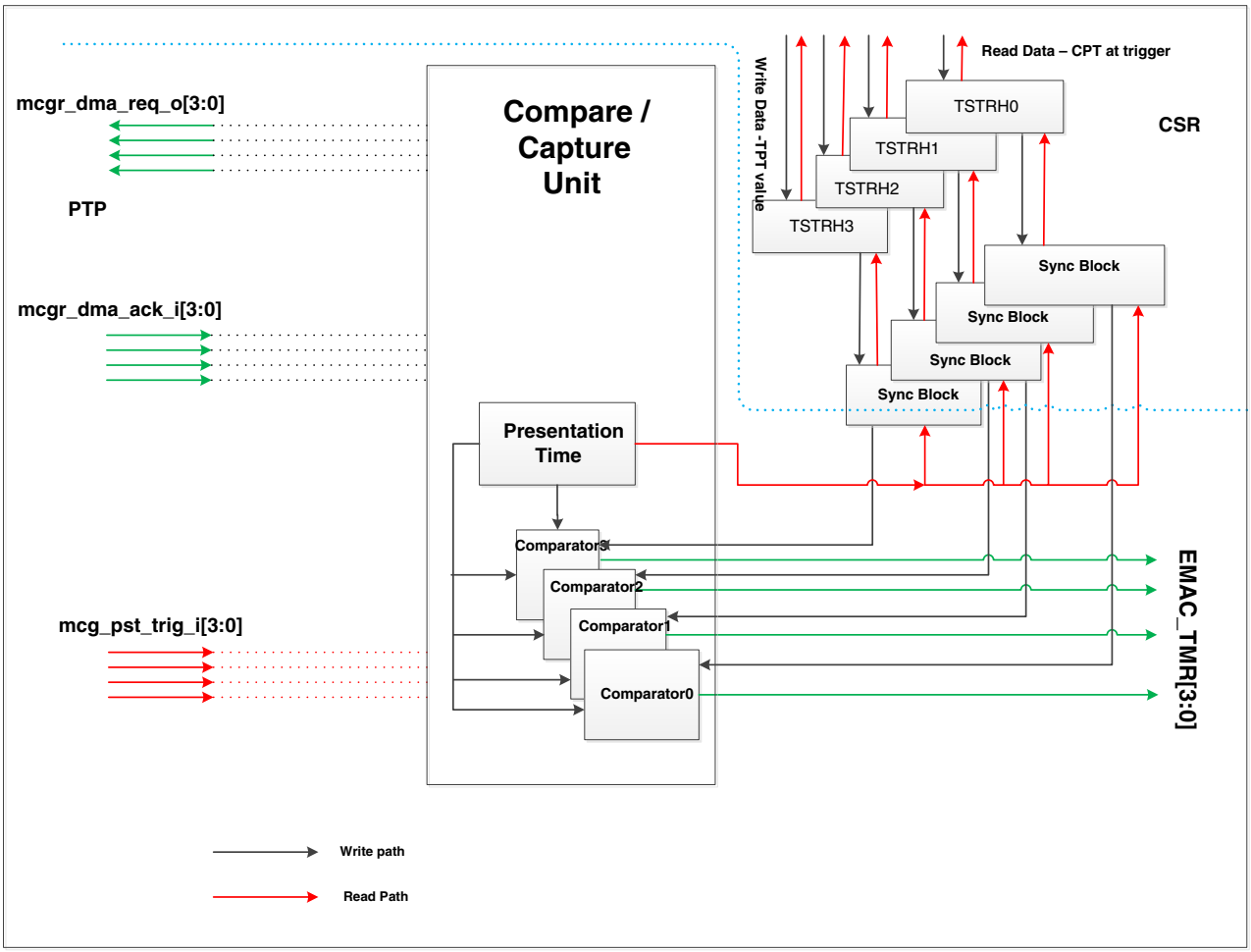


Figure 61-8. Media Clock Generation and Recovery Flow

NOTE

1. The consecutive triggers to sample the presentation time should always be asserted after a few PTP clock cycles to allow for synchronization delays. (This is not an issue when the input trigger maximum frequency is 8KHz and the PTP clock is running at least at 1MHz)
2. DMA acknowledgment can be both posted as well as non-posted, in the Media Clock Recovery Mode.
3. See "Programming Guidelines for Media Clock Generation and Recovery"

61.3.5.10 One step timestamp

IP supports the one step time stamp feature. It can be enabled for a packet by setting the Bit 20 (OSTC) in the control word.

NOTE

See Programming Guidelines for IEEE 1588 Timestamping.

61.3.5.11 IEEE 1588 Sub Nanoseconds Timestamp

The ENET QOS block supports the ingress and egress correction accuracy in sub-nanoseconds. It is programmed in the TSICSNS and TSECSNS fields of the MAC Timestamp Ingress Correction Subnanosecond and MAC Timestamp Egress Correction Subnanosecond registers, respectively. In this case, the correction has an unit of nanosecond, multiplied by 2^8. The least significant 8 bits of the value should be programmed in the sub-nanoseconds register.

For example, if the required egress correction is 3.6 nS and TSCTRLSSR bit is set, then TSEC must be 0x3 and TSECSNS must be 0x99 (0.6 * 256). Similarly, if the required ingress correction is -3.6 ns and TSCTRLSSR bit is set, then TSIC must be 0xBB9A_C9FC and TSICSNS must be 0x66 (fractional part of (10^9 - 3.6) * 256).

61.3.6 Multiple Channels and Queues Support

IP supports the five queues and channel on Tx and Rx paths.

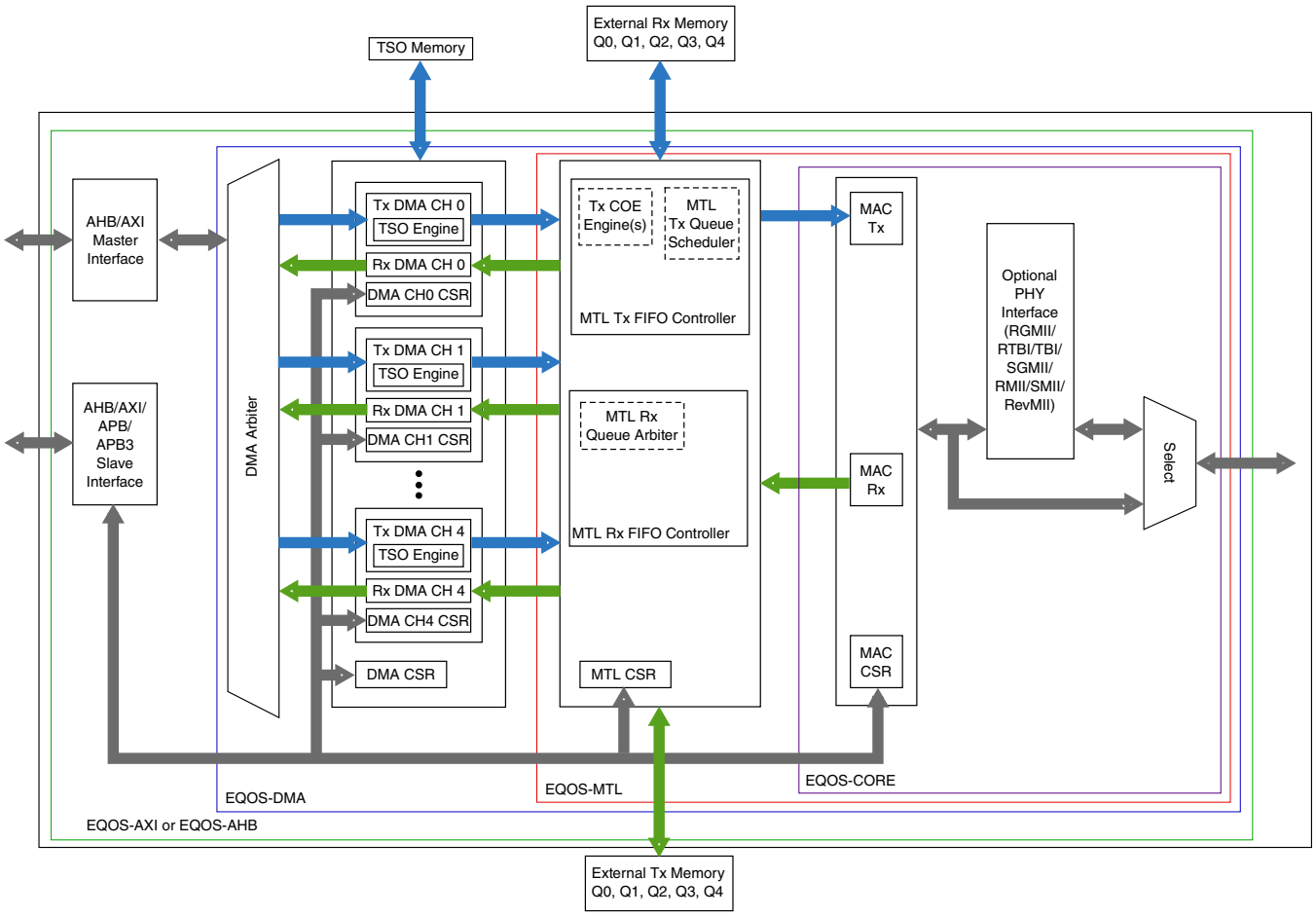


Figure 61-9. Module with Multiple Channels and Queues Block Diagram

DMA arbiter of IP supports two types of arbitration scheme, fixed priority and weighted round-robin. The DMA arbiter performs the arbitration between the Tx and Rx paths of DMA channels for accessing descriptors and data buffers. The DA bit of the DMA_Mode register specifies the arbitration scheme (fixed or weighted round-robin) between the Tx and Rx DMA of a channel.

Priority between the corresponding Tx DMA and Rx DMA can be set by using TXPR field of the DMA_Mode register. PR field of the DMA_Mode register can be set to specify the weighted priority between the Tx DMA and Rx DMA in round robin arbitration scheme.

Following table provides information about the priority scheme between Tx DMA and Rx DMA.

Table 61-16. Priority Scheme for Tx DMA and Rx DMA

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
x	x	x	0	1	Rx always has priority over Tx

Table continues on the next page...

Table 61-16. Priority Scheme for Tx DMA and Rx DMA (continued)

Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Priority Schemes
0	0	0	0	0	Tx and Rx have equal priority. Rx gets the access first on simultaneous requests
0	0	1	0	0	Rx has priority over Tx in ratio 2:1
0	1	0	0	0	Rx has priority over Tx in ratio 3:1
0	1	1	0	0	Rx has priority over Tx in ratio 4:1
1	0	0	0	0	Rx has priority over Tx in ratio 5:1
1	0	1	0	0	Rx has priority over Tx in ratio 6:1
1	1	0	0	0	Rx has priority over Tx in ratio 7:1
1	1	1	0	0	Rx has priority over Tx in ratio 8:1
x	x	x	1	1	Tx always has priority over Rx
0	0	0	1	0	Tx and Rx have equal priority. Tx gets the access first on simultaneous requests
0	0	1	1	0	Tx has priority over Rx in ratio 2:1
0	1	0	1	0	Tx has priority over Rx in ratio 3:1
0	1	1	1	0	Tx has priority over Rx in ratio 4:1
1	0	0	1	0	Tx has priority over Rx in ratio 5:1
1	0	1	1	0	Tx has priority over Rx in ratio 6:1
1	1	0	1	0	Tx has priority over Rx in ratio 7:1
1	1	1	1	0	Tx has priority over Rx in ratio 8:1

61.3.6.1 Multiple Queues and Channels Support in the Transmit Path

IP supports the five transmit queues and five channels.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing

61.3.6.2 Multiple Queues and Channels Support in the Receive Path

IP supports five receive channels and five queues.

In the Receive direction, the MTL Rx Controller selects the Rx DMA for which it is transferring or reading the data from the Rx FIFO memory. This scheduling is based on the programming done in the respective MTL_RxQ[x]_Control register.

Each Rx DMA indicates when it is ready to transfer data and the size of the burst-length (number of beats) that it has to transfer. The scheduler checks whether sufficient data (of requested burst length) is available to be transferred to these DMAs and then selects the Rx DMA that gets serviced using the programmed priorities.

NOTE

See Programming Guidelines for Multi-Channel Multi-Queuing.

61.3.6.3 Rx Queue to DMA Mapping

The packets in the MTL Rx Queues can be routed to any one of the multiple DMA channels by programming the register MTL_RxQ_DMA_Map0 Register for queues 0 through 3, and MTL_RxQ_DMA_Map1 for queue 4.

The following types of Rx Queue to DMA mapping are possible through programming.

61.3.6.3.1 Static Mapping

In this mode, all of the packets of an Rx Queue are connected to a specific DMA channel. For example, all the packets from Rx Queue 0 can be routed to a DMA channel by programming Q0MDMACH (bit[3:0]) and Q0DDMACH (bit[7] = 0) of the MTL_RxQ_DMA_Map0 Register.

Similarly, packets from other Rx Queues can be routed to any DMA channel by programming register fields corresponding to each Queue.

61.3.6.3.2 Dynamic (Per Packet) Mapping

In this mode, the destination DMA channel of a packet being read from a Rx Queue is not constant but decided independently for each packet. For example, if you set the Q1DDMACH bit of the MTL_RxQ_DMA_Map0 register, the static mapping is disabled for Rx Queue 1 and the value in Q1MDMACH is ignored. The destination DMA channel is decided by the MAC receiver for each packet, depending on the following in decreasing order of priority:

1. L3-L4 filter Based DMA Selection: The TCP/UDP and IP header fields of the received packet are matched against the corresponding values programmed and enabled for comparison in the MAC_L3_L4_Address_Control register. If the match is successful, the DMA channel number programmed in the DMCHN field of the MAC_L3_L4_Ad-dress_Control register is selected as the destination DMA channel number provided DMCHEN bit of the same register is set. If none of the L3-L4 Registers give a comparison match, then the module proceeds to the next step below.
2. Extended VLAN Based DMA Selection: Extended routing is applicable only if the VLAN Filter has passed. Routing is only based on Perfect Filter Result. Each perfect filter has a DMA Channel Enable and a DMA Channel Number field which can be programmed. Routing is applicable for a filter, only if the DMA Channel Enable bit is set. The frame is routed to the smallest Matching Filter's DMA Channel provided it is enabled. If that filter's DMA Channel number is not enabled, the frame gets routed to Channel 0. For example, if a frame's VLAN tag matches filters 7, 3, and 1. Then the MAC checks if Filter 1's DMA Channel Number is enabled through programming. If yes, the frame gets routed to the programmed value; otherwise it gets routed to DMA. When the inverse filter is enabled; is routed to the least mismatched filter's DMA channel number provided it is enabled. If the DMA Channel enable bit is not set, then the frame is routed based on DA based addressing or to Channel 0.

   If Hash Filter is also enabled, it is used to determine the Filter result only. Routing will still depend on the Perfect Filters enabled. If none of the perfect filters are enabled or if all of them are bypassed, then the VLAN Filter based routing will not take place. The frame will be routed via DA Based addressing or to Channel 0. If all the perfect filters give a fail result and the Hash Filter has passed, then the VLAN Filter result is a pass but routing will be based on DA Based Addressing or to Channel 0. Similar behavior is seen when inverse Filtering is enabled as well.

3. Ethernet DA-Based DMA Selection: The DA address of the received packet is compared against the programmed DA values in MAC Address Registers. If the address matches any of the programmed values, the corresponding DCS field (when enabled) determines the destination DMA channel number.

If none of the previous operations is able to make a successful match/decision, then the packet is routed to DMA Channel 0 by default.

61.3.6.3.2.1 Broadcast/Multicast Packet Duplication

Broadcast/Multicast Packet Duplication feature provides a mechanism to send the received Broadcast/Multicast packets to multiple DMA channels.

The DCS field of the of MAC_Address0_High and other optional MAC_Address(#i)_High (i=1 to 2) registers determines the DMA channel number to which the received packet (that matches the MAC Address present in that register) must be routed.

Packet duplication is not supported in other DMA channel routing mechanisms such as Extended VLAN, or L3-L4 based routing.

The Packet Duplication feature is supported only on the highest MTL Queue configured. Therefore, packet duplication for Broadcast/Multicast packets requires that the MCBCQ field of MAC_RxQ_Ctrl1 register is programmed to the highest RxQ present in the configuration.

61.3.6.4 Selection of Tag Priorities Assigned to Rx Queues

The VLAN Tag priorities can be assigned to Rx Queues by programming the PSRQ field in the corresponding MAC_RxQ_Ctrl2 registers. The bit corresponding to the VLAN Tag priority can be set in the PSRQ field for assigning that priority to the Rx Queue. For example, if you want to assign VLAN Tag priority of 3 to Rx Queue 0, set bit [3] in PSRQ field of MAC_RxQ_Ctrl2 register. The VLAN Tag priority assigned to particular Rx Queue must be unique, that is, more than one Rx Queue cannot be assigned the same VLAN Tag priority. However, more than one VLAN Tag priorities can be assigned to same Rx Queue.

The settings in the PSRQ field is used for VLAN Tagged Rx packet routing to Rx Queues as well as for PFC based Tx flow control. The received VLAN Tagged Rx packet is routed to Rx Queue that has the VLAN Tag priority match. In PFC based Tx flow control, PSRQ field corresponding to a particular Rx Queue is used for enabling VLAN Tag priorities in the PFC packet transmitted when corresponding Rx Queue threshold levels are reached.

61.3.6.5 Rx Side Routing from MAC to Queues

IP supports Rx Side Routing from the MAC to Queues. The MAC routes the Rx packets to the Rx Queues based on following packet types in that order

- Unicast/Multicast destination address packets that fail the destination address filter
- Multicast/Broadcast destination address packets that pass the destination address filter
- VLAN Tag Priority field in VLAN Tagged AV data packets
- AV Control packets
- VLAN Tagged IEEE 1588 PTP over Ethernet packets
- Untagged IEEE1588 PTP over Ethernet packets
- VLAN Tag Priority field in VLAN Tagged packets
- Untagged packets

The outer C-VLAN Tagged AV data Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The AV Control Rx packets can be routed based on the Rx Queue number specified in the AVCPQ field in MAC_RxQ_Ctrl1 register and corresponding Rx Queue is enabled for AV through RXQ#EN field in MAC_RxQ_Ctrl0 register. These packets may be single VLAN Tagged with C-VLAN type or Double VLAN Tagged with outer VLAN Tag of C-VLAN type when Double VLAN feature is enabled (EDVLP bit in MAC_VLAN_Tag register is set) with inner C-VLAN Tagged or inner S-VLAN Tagged when SVLAN processing is enabled (ESVL bit in MAC_VLAN_Tag register is set). This type of Rx packet routing is available when AV feature and Multiple Rx Queues are selected in the configuration.

The VLAN Tagged Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers and the corresponding Rx Queue is enabled for DCB/generic through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register.

The VLAN tagged IEEE 1588 PTP over Ethernet Rx packets can be routed based on the priorities assigned to Rx Queues through PSRQ field in corresponding MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers or the Rx Queue number specified in the PTPQ field in MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This is determined by programming in TPQC field of MAC_RxQ_Ctrl1 register. This type of Rx packet routing is available when IEEE 1588 Timestamp feature support and Multiple Rx Queues are selected in the configuration. The VLAN Tagged IEEE 1588 PTP over Ethernet Rx packets are detected only when 802.1AS mode is disabled (AV8021ASMEN bit in MAC_Timestamp_Control register is set to 0), otherwise this type of Rx packets are routed as generic VLAN Tagged Rx packets.

The multicast/broadcast destination address Rx packets that pass the destination address filter can be routed based on the Rx Queue number specified in the MCBCQ field of MAC_RxQ_Ctrl1 register when enabled through MCBCQEN bit of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The untagged Rx packets can be routed based on the Rx Queue number specified in the UPQ field of MAC_RxQ_Ctrl1 register and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The unicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the UFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through UFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

The multicast destination address Rx packets that fail the destination address filter can be routed based on the Rx Queue number specified in the MFFQ field of MAC_RxQ_Routing_Ctrl register when enabled through MFFQE bit of MAC_RxQ_Routing_Ctrl register, RA bit of MAC_Packet_Filter register is set to 1 and the corresponding Rx Queue is enabled through RXQ#EN field in MAC_RxQ_Ctrl0 register. This type of Rx packet routing is available when Multiple Rx Queues are selected in the configuration.

If the Rx packet cannot be classified in any of the defined packet type for routing, it will be routed through Rx Queue 0.

61.3.6.6 Rx Side Arbitration Between DMA and MTL

IP controller supports Rx side arbitration between the DMA and the MTL.

After completing the current packet processing, the DMA Channel Controller fetches the next descriptor. After the descriptor fetching is complete, the DMA Channel Controller evaluates the amount of data to be to be transferred to the Rx Buffer based on the programmed PBL and Rx Buffer Length. Accordingly, it requests the MTL to transfer the data.

After servicing the current request, the MTL Rx Queue arbitration scheme selects Rx Queue based on the arbitration scheme (RAA field of the MTL_Operation_Mode register) and the weights programmed in the Queue <n> Receive Control Register. The arbitration is done among Queues for which DMA is ready to service. After the Rx Queue is selected, PBL (Programmable Burst Length) amount of data is read out from that Queue and is routed to the Rx DMA Channel based on the Rx Channel selection criteria.

The arbitration takes place when every PBL data transfer is completed and descriptors are ready for the processing from at least one DMA Channel.

61.3.6.7 Tx Side Arbitration between DMA and MTL

IP support Tx side arbitration between the DMA and the MTL. Since the number of Tx DMA channels is always equal to the number of Tx Queues in the MTL, they are always mapped directly. For instance, each Tx DMA pushes data into its respective Tx Queue assigned to it.

The data inside each Tx Queue is stored in packets. Hence, if two DMAs are allowed to transfer data into the same queue, when a Tx DMA starts a packet transfer, the other DMA cannot transfer data unless the previous packet is completely pushed-in. This means that the second DMA remains idle until the first packet is transferred. Hence, each DMA is always connected directly to its corresponding Tx Queue.

61.3.6.8 Time-Sensitive Networking (TSN)

The ENET QOS block supports TSN in 100 Mbps or 1000 Mbps mode, which enables the transfer of time-sensitive traffic over a network. The block implements the IEEE 802.1Qbv-2015 (Enhancements to Scheduling Traffic), IEEE 802.3br (Interspersing Traffic) and IEEE 802.1Qbu (Frame Preemption) TSN protocol.

The Transmit path of Queue 0 supports the strict priority algorithm and is used for best-effort traffic while transmit paths of additional queues support traffic management by using the credit-based shaper algorithm. For a queue, the credit-based shaper algorithm determines that a queue is available for transmission if the following conditions are true:

- The queue contains one or more packets.
- The credit for the queue is positive as per the algorithm.
- AV traffic can be received on all queues. The block can also receive untagged PTP packets in addition to AV traffic on any queue.

61.3.6.8.1 Enhancements to Scheduled Traffic (EST)

The IEEE 802.1Qbv-2015 defines the schedule for each of the queues on every egress port which makes the implementation aware of traffic arrival schedule. This information can be used to block the lower priority traffic from transmission in this time window/slot. This ensures that scheduled traffic is forwarded from sender to receiver through all the network nodes with a deterministic delay.

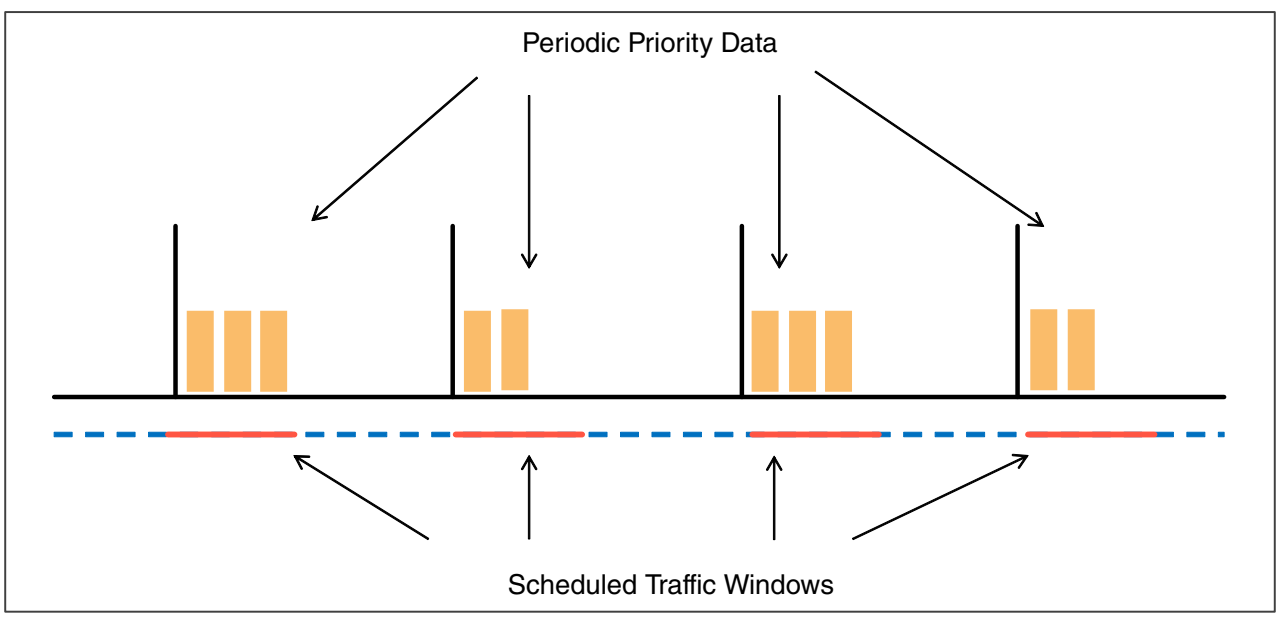


Figure 61-10. Time Aware Shaper Implementation for Traffic Scheduling

One of the important requirements to achieve a low latency is to ensure there are no interfering frames during the scheduled windows that are reserved for high priority traffic. The use of scheduled traffic imposes limitations while starting a transmission.

As shown in the following figure, if an interfering frame begins transmission just before the start of a reserved time period, it can extend transmission into the reserved window, and potentially interfere with higher priority traffic. Therefore, a guard band whose size is equal to the largest possible interfering frame is required before the window starts.

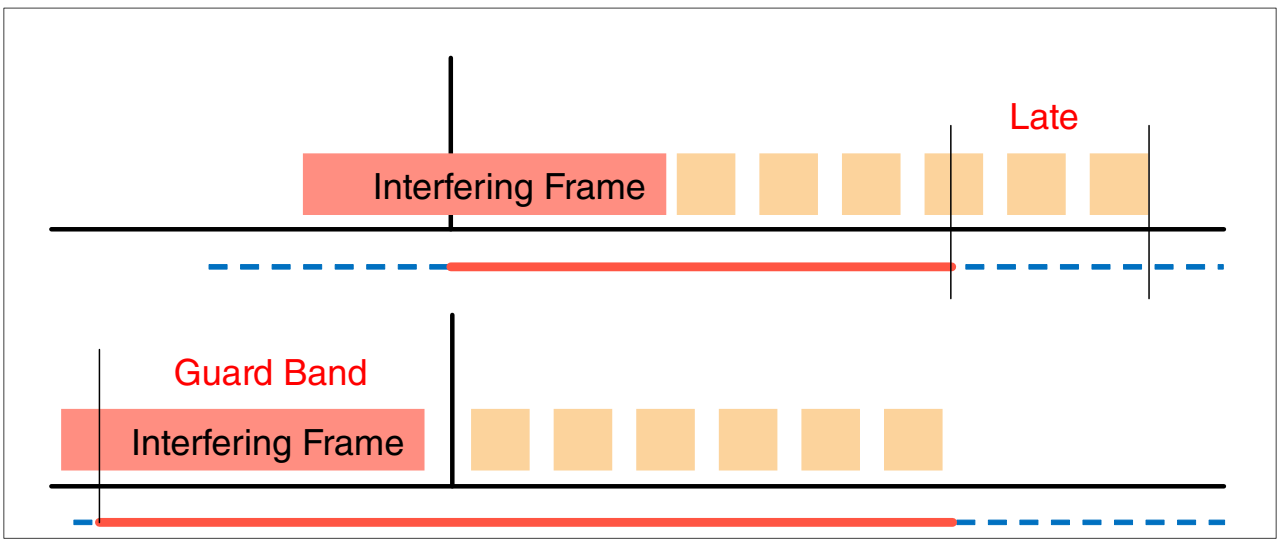


Figure 61-11. Implementing a Guard Band to Avoid Delays Due to Interfering Frames

A larger guard band equates to a less efficient use of network bandwidth. However, with the implementation of IEEE802.1Qbu (frame preemption), this issue is addressed. Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame.

During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

61.3.6.8.1.1 Frequently Used Terms in EST

- Gate Control List: The list in the hardware memory that holds the gate controls and the associated time intervals.
- Gate Controls: For a given schedule (row in gate control list) there is a Gate Open (O) and Gate Close (C) state associated with each traffic class. The set of O or C values, whose width is same as configured TCs is called the Gate Controls. Example: CCOCO means TC4=C, TC3=C, TC2=O and so on.

- Time Interval: Time interval (in nanoseconds) is a 16, 20, or 24-bit configurable field in the Gate Control list that indicates the time for which the associated gate controls are valid.
- Base Time Register: Each Gate Control List is associated with a 64-bit Base Time Register that holds the start time (in PTP format) for the list.

61.3.6.8.1.2 Updates to the Transmit Scheduling to Support EST

To support EST, following updates are required to the transmit scheduling:

- Implementation of Gate Control List (GCL)
- Enforcing gate controls in the scheduler
- Accounting for Gate Open (O) duty cycle in the computation of idleSlope (CBS)

61.3.6.8.1.2.1 Implementation of Gate Control List (GCL)

Following figure shows block diagram from the IEEE 802.1Qbv spec that illustrates how the gate control list governs the gate close(C) and open(O) events based on the schedule provided for each event. GCL has the following two parts:

- Time Interval: Defines the time in nano seconds for which the gate controls are valid and should be applied before reading the next gate controls from the list. Configurable width of 16, 20 or 24 to represent a maximum of 64us, 1ms, or 16ms schedule interval respectively. Supports left shift of the time interval upto 7 bits to be able to apply a multiplication factor from 1 to 128ns (in steps of 2(power)n). The maximum value (post shifting) of this field must be 999,999,999 ns.
- Gate Control: Defines the Open (O) represented by logic 1 or Close (C) represented by logic 0 state for the gate of each TC.

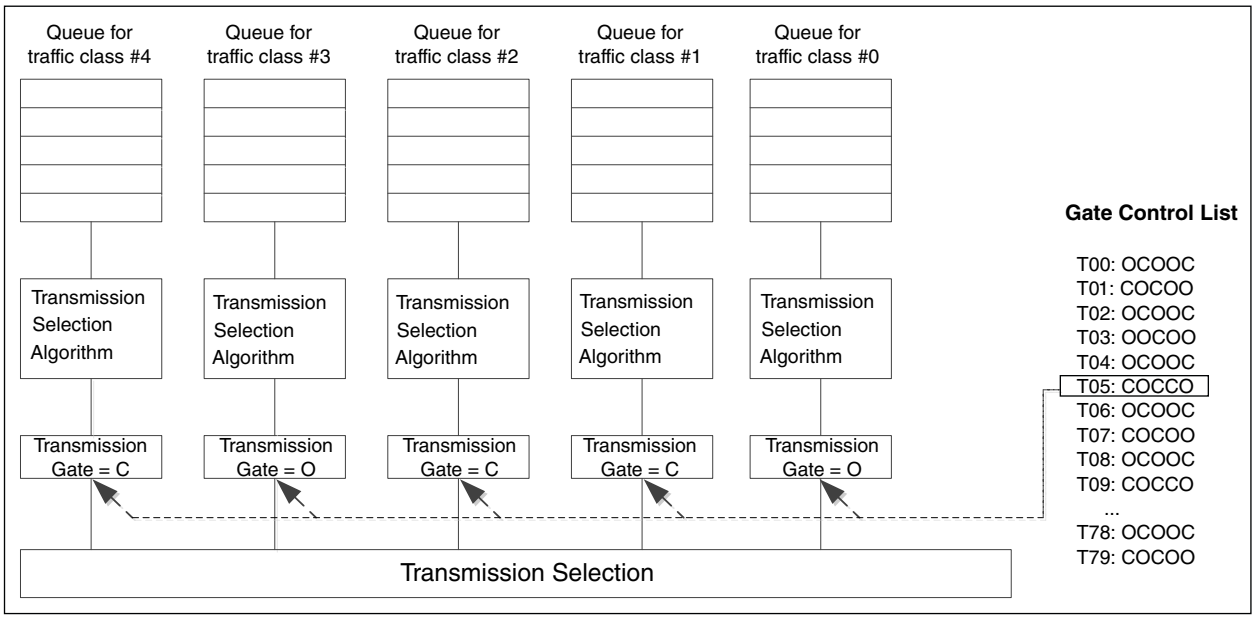


Figure 61-12. Block Diagram from IEEE 802.1 Qbv Specification - GCL Governing Gate Close and Open Events

The implementation of GCL consists of the following two Gate Control Lists:

- HWOL - Hardware Owned List which is a list for hardware access.
- SWOL - Software Owned List which is a list for software access.

The access to these lists is mutually exclusive. The ownership to the list is set by hardware in the SWOL field of MTL_EST_Status register. Following table provides the implementation details of GCL.

Table 61-17. External Memory Used for Holding the Two Gate-Control Lists

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OOCCC	T0	HWOL or SWOL
OOOOC	T1	
CCCCO	T2	
CCCCC	T3	
I	I	
I	I	
OCOCO		
OOOOC	T last	
OOOOC	T0	SWOL or HWOL
OOOOC	T1	
OOCCC	T2	

Table continues on the next page...

Table 61-17. External Memory Used for Holding the Two Gate-Control Lists (continued)

Gate Control (up to 5 bits)	Time Interval (ns) 16, 20, or 24 bits	
OOOOC	T3	
I	I	
I	I	
OCOCC		
OOCCC	T last	

61.3.6.8.1.2.2 Registers Related to Gate Control List

Following are the set of 4 registers (one for each GCL) related to GCL. These registers are implemented through Indirect Addressing using the GCL_Control and GCL_Data registers.

1. 64-bit Base Time Register (BTR)

2. 40-bit Cycle Time (CTR)

3. m-bit Time Extension (TER)

4. n-bit [Gate Control] List Length (LLR) (n depends of the configured GCL depth)

61.3.6.8.1.2.2.1 Base Time Register (BTR)

Base Time Registers is a 64-bit register that holds the start time to execute the GCL. The format of the BTR is same as the PTP format (upper 32-bits holds time in seconds and lower 32 bits hold time in nano seconds). Once the execution of a given list begins, the implementation can update the value in BTR to indicate the next list execution begin time.

61.3.6.8.1.2.2.2 Cycle Time Register (CTR)

Cycle Time Register is a 40-bit register that holds the time at which the execution of the GCL should be repeated. The Cycle Time Register consists of an 8-bit value in seconds, and a 32-bit value in nano seconds (similar to the PTP time format with truncated seconds register). For a given Gate Control List the start time is "Base Time" + N * "Cycle Time" where N is an integer value representing the iteration number starting with 0 for first iteration. If the Gate Control List execution takes longer than the Cycle time, then the list is truncated at the Cycle Time and the subsequent loop begins at Cycle Time.

61.3.6.8.1.2.2.3 Time Extension Register (TER)

Time Extension Register (TER) is a m-bit (where m = Configured Time Interval width + 7) register that holds the amount of time (in nano seconds) the current Gate Control List can be extended before switching to the new Gate Control List. This will be useful in avoiding small fragments of the current list being executed before switching to a new list.

61.3.6.8.1.2.2.4 List Length Register (LLR)

List Length Register (LLR) is an n-bit register (when n is 7, 8, 9, 10, or 11 for a GCL configured depth of 64, 128, 256, 512, or 1024 respectively) that holds the integer value of the length of the GCL (that is the number of valid rows in each GCL). The processing of the GCL stops after the number of rows read equals to the LLR value.

61.3.6.8.1.2.3 Transmission Gating Implementation

A Bridge or an end station can be enhanced to allow transmission from each TC that is yet to be scheduled relative to a known timescale. To achieve this, a transmission gate is associated with each TC; the state of the transmission gate determines if the queued frames can be selected for transmission.

For a TC, the transmission gate can be in one of the following two states:

1. OPEN:Queued frames are selected for transmission, in accordance with the definition of the transmission selection algorithm associated with the TC.
2. CLOSED: Queued frames are not selected for transmission.

A frame on a traffic class queue is not available for transmission if the transmission gate is in the closed state or if there is insufficient time available to transmit the entirety of that frame before the next gate-close event associated with that queue.

The implementation has visibility to the current schedule of gate controls and the immediate next schedule of the gate controls. So the maximum Gate Open period does not exceed the sum of the two Time Intervals. This is because, a frame is selected for transmission only if the gate is currently Open and the duration of gate open interval is greater than the time taken to transfer the entire frame.

The implementation must know the frame size before the transmission, so that the transmission overruns can be avoided and only the frames that can complete are scheduled at all times.

The implementation adequately compensates for the implementation delays in the data transfer from the buffer to the line by offsetting the current time with all the relevant delays (provided by the CTOV field of MTL_EST_Control register). This ensures that the schedule provided is always accurately implemented at the line.

61.3.6.8.1.3 Idle Slope Computation Updates

When EST is enabled, credit is accumulated only when the gate is open; therefore, the effective data rate of the idleSlope must be increased to reflect the duty cycle for the transmission gate associated with the queue.

The idleSlope is computed based on the GateOpenTime and OperCycleTime values. Program the idleSlope registers (implemented one per CBS enabled TC) based on the following equation. The existing MTL register has sufficient bit width to accommodate the new values for idleSlope.

idleSlope = (operIdleSlope(N) * OperCycle/GateOpenTime)

61.3.6.8.1.4 Operational Details of GCL

Set the switch to software List, the SSWL field of the MTL_EST_Control Register, so that hardware can access the programmed gate control list. The first set of gate controls are applied when the current time is equal to the value in the Base Time Register (BTR) and is held until the programmed "Time Interval" value.

To avoid transmission overruns, one additional gate control event is always read ahead from the list. This enables the GCL to determine the next gate close events (if any) for the TCs that are open.

The scheduling based on the Gate Open state and Time Interval of only the current and subsequent schedule. An internal accumulator is used to add the time intervals when gate controls are applied. BTR + Accumulator holds the time at which the next set of gate controls are to be applied.

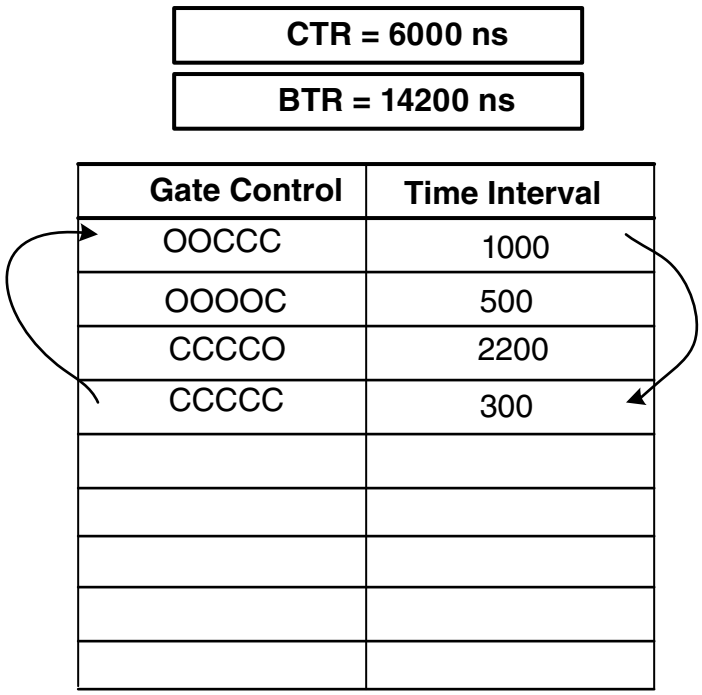


Figure 61-13. GCL and Associated Registers - BaseTime and CycleTime

The GCL is read progressively from the first row adhering to the schedule. The read operations continue until the List Length (from LLR Register) is reached and the execution of the list restarts at BTR + CTR time. At this point the BTR is incremented by value in CTR to mark the beginning of a new cycle. In the absence of any gate controls, all the gates are deemed to be in Open state, during the execution of the list.

In cases where the execution time of the list is greater than the CycleTime, the list is truncated and the next iteration starts when the current time equals BTR + CTR.

Table 61-18. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17900	CCCCC	4000	14200
18200	OOOOO	0	20200
20200	OOCCC	1000	20200
21200	OOOOC	1500	20200

In the example in table, the execution starts at 14200 and the first set of Gate Controls "OOCCC" are applied immediately. As the time interval is 1000ns the next set of gate controls are applied at 14200 (BTR) + 1000 (Accumulator) = 15200 ns . As there are no gate controls available after the execution of the last gate control and before the next iteration of the loop, the gates are deemed to be in open state during that time period as depicted at time 18200 in the table.

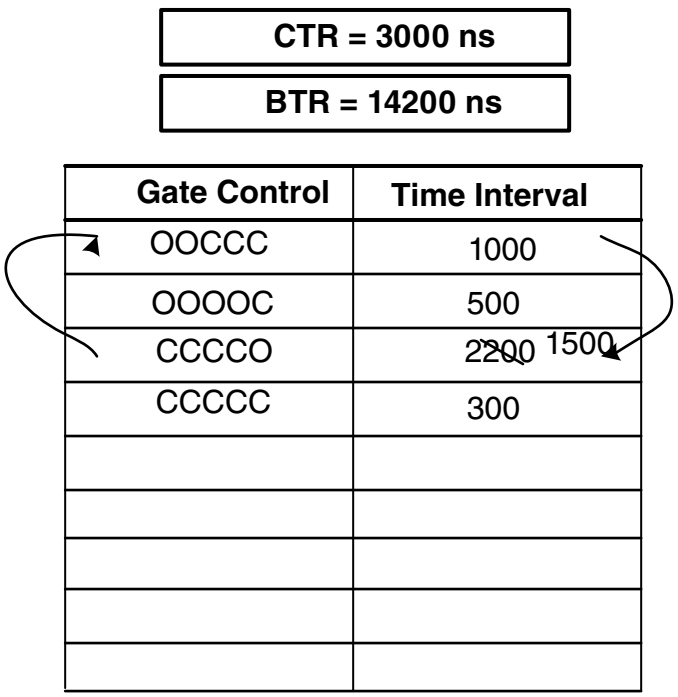


Figure 61-14. GCL and Associated Registers - BaseTime and CycleTime List Execution Time > CycleTime

As the list execution takes longer than the allocated CycleTime, the list is truncated and the list is started from the BTR+CTR as shown in the following table.

Table 61-19. GCL and Associated Registers - BTR and CTR, Execution Time > Cycle Time

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
14200	OOCCC	1000	14200
15200	OOOOC	1500	14200
15700	CCCCO	3700	14200
17200	OOCCC	1000	17200
18200	OOOOC	1500	17200
18700	CCCCO	3700	17200

While applying the third set of gate controls, the BTR + CycleTime (17200) < BTR + Accumulator (17900), so the list is truncated and execution switches to a new iteration at 17200.

61.3.6.8.1.5 Installing a New GCL

When a new software programmed GCL is available and must be executed at the new BTR value, the switch to the new GCL can happen in one of the following ways.

- New Base Time aligned with current schedule
- New Base Time unaligned with current schedule

61.3.6.8.1.5.1 New Base Time Aligned with Current Schedule

If the choice of cycle time for the new gating cycle is unchanged from the cycle time for the current gating cycle, and if the BTR chosen for the new gating cycle (new BTR) is an integer multiple of the current cycle time (+ current BTR), then the new gating cycle exactly lines up with the old gating cycle, that is, the cycle start times for the new gating cycle is same as they would have been for the old configuration. This could be considered to be the ideal case and allows the new gating cycle to be installed and executed with no timing issues. The implementation completes the execution of an iteration of the current list and switches to the new list at the beginning of the BaseTime listed in the new list.

If (New BaseTime > = Current Time)

ConfigChangeTime = New BaseTime

Else If (New BaseTime < Current Time)

1. Set the BTRError

2. ConfigChangeTime = (New BaseTime + N* New CycleTime)

where N is the smallest integer for which the relation ConfigChangeTime >= CurrentTime and (N =< 8) is TRUE.

When N > 8 the hardware cannot auto recover and the loop count value in BTRError reporting is set to 1111 requiring the software to reprogram the New Base Time.

Following figure illustrates the installation of the new GCL along with the timelines.

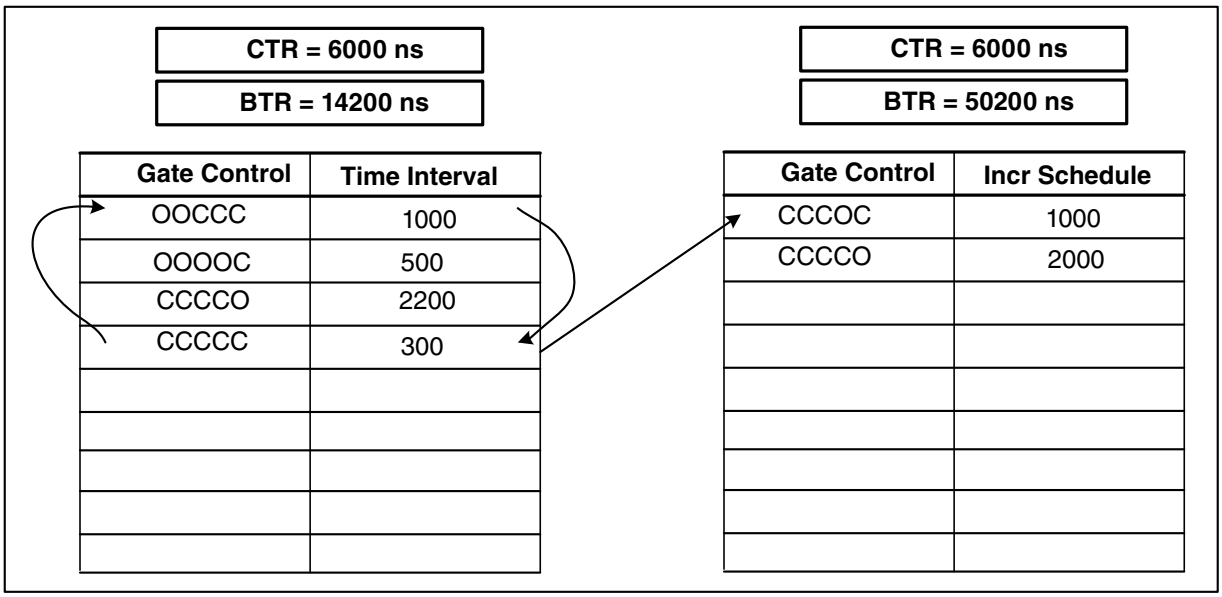


Figure 61-15. Switching to a New Configuration that is Aligned with the Existing Configuration

In the above example, after the sixth iteration of the first GCL, the BTR values of the old and new GCL are equal. At that point the new GCL is processed as a natural extension to the existing GCL.

Table 61-20. GCL and Associated Registers - BTR and CTR

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47900	CCCCC	4000	44200
48200	OOOOO	0	50200
50200	CCCOC	1000	50200
51200	CCCCO	3000	50200
53200	OOOOO	0	56200

61.3.6.8.1.5.2 New Base Time Unaligned with Current Schedule

If new CycleTime differs from current CycleTime or New BaseTime is in the future and is not an integer multiple of current CycleTime, then the old and new cycles do not line up; when new BaseTime is reached (when the new configuration is installed and starts to execute), the last old cycle is normally truncated to start the first new cycle. This could be undesirable if it results in a very short last old cycle; arguably it would be better to simply extend the penultimate old cycle by that small amount, rather than starting a very short

cycle. The Cycle Time Extension Register (related to the current config list) allows this extension of the last old cycle to be done in a defined way; if the last complete old cycle ends normally in less than current Cycle Time Extension (TER) ns before the new base time, then the last complete cycle before new BaseTime is reached is extended so that it ends at new BaseTime.

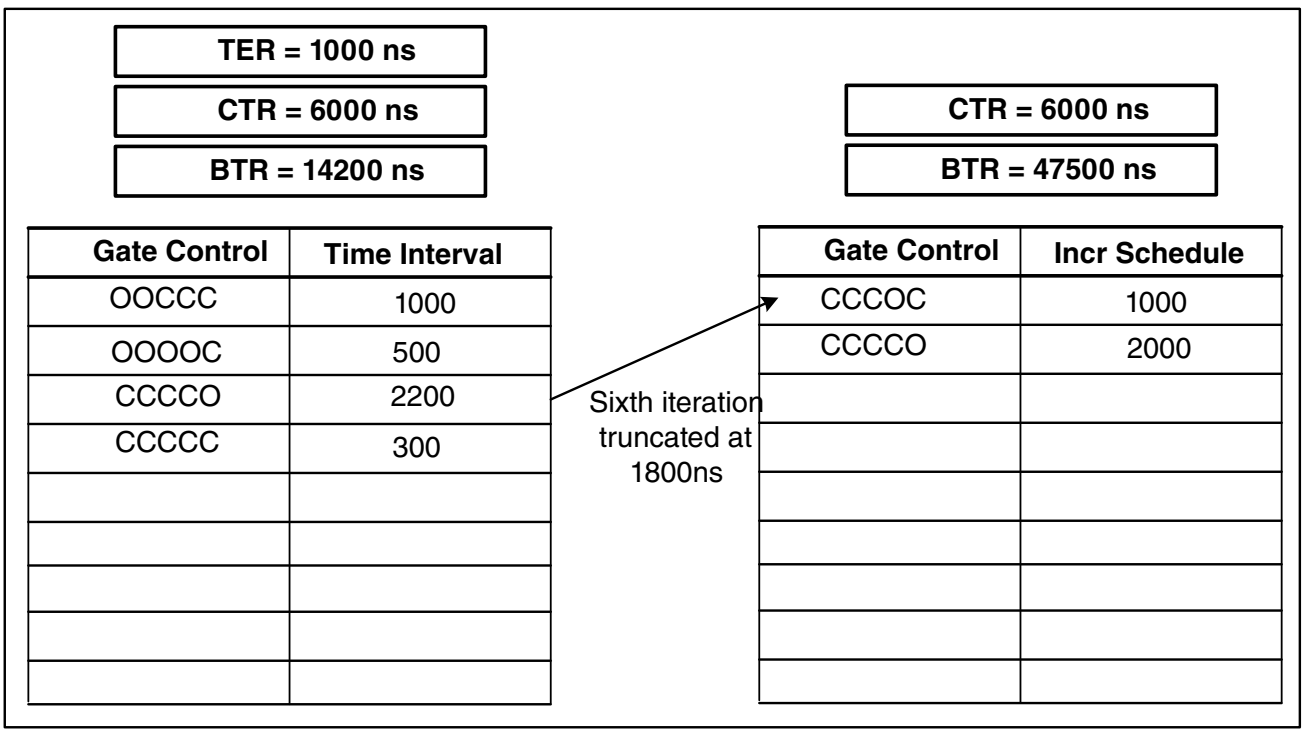


Figure 61-16. Switching to New List Truncating the Current List

At the end of the fifth iteration the Current time + Cycle time extension (TER) < New BTR so the sixth iteration of current configuration is started. During the sixth iteration of the current list when the new BTR value is smaller than the next schedule in the current list, it switches to the new list.

Table 61-21. Extending to New List By Truncating the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
44200	OOCCC	1000	44200
45200	OOOOC	1500	44200
45700	CCCCO	3700	44200
47500	CCCOC	4000	44200
48500	CCCCO	0	50200
50500	OOOOO	1000	50200

Below is an example where the current config list is extended instead of starting a new iteration as the extension time of 800ns is less than the allowed cycle extension time (TER) of 1000ns.

Figure 61-17. Switching to New List By Extending the Current List

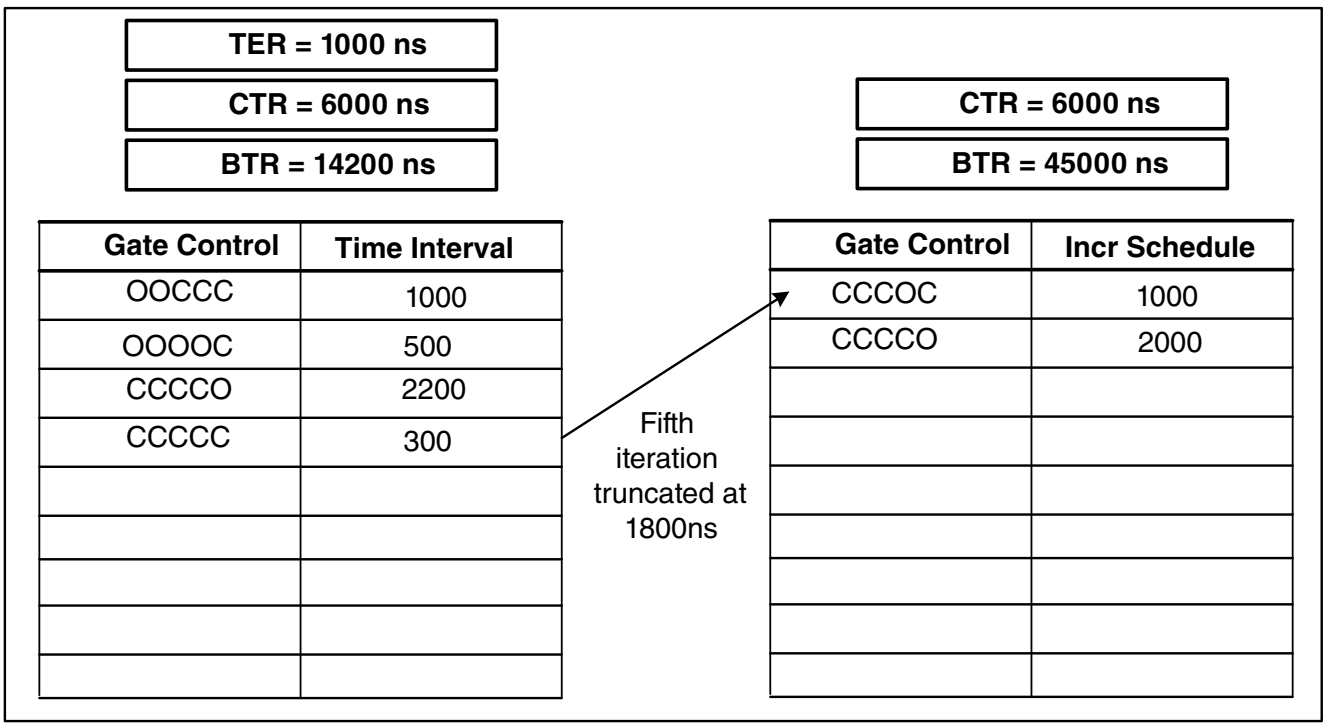


Table 61-22. Switching to New List By Extending the Current List

Current Time	Gate Control Applied	Accumulator Value	BTR (with updates)
38200	OOCCC	1000	38200
39200	OOOOC	1500	38200
39700	CCCCO	3700	38200
41900	CCCCC	4000	38200
42200	OOOOO	0	44200
45000	CCCOC	1000	45000
46000	CCCCO	3000	45000
48000	OOOOO	0	51000

61.3.6.8.1.6 Impact of Transmit Scheduling Algorithms on EST

When EST is used in isolation, the Gate Control List manages the final open/close state of the Queues along with the checks carried out by the Transmission Selection Algorithm in MTL. As the Gate Controls operate on a predefined repetitive schedule, it is recommended to use Strict Priority or Credit Based Shaper (CBS) scheduling algorithms.

Other algorithms such as the Weighted Round Robin (WRR), Deficit Weighted Round Robin (DWRR) and Weighted Fair Queuing (WFQ) implement masking of the queues based on the current winning queue. The algorithm is applied only among the group of queues that open simultaneously. To ensure Queues whose gates are "Open" get priority, these algorithms are modified to treat "gate open" queues and "gate closed" queues as separate groups giving priority to gate open queues.

For example, consider 4 queues (Q3, Q2, Q1, Q0) with weights 4:3:2:1; Q3 and Q2 are in Open state in slot one slot, while Q1 and Q0 are in Open state in another slot. In this case, the scheduler works as follows:

1. In the first slot, the Q3 and Q2 are serviced in the ratio of 4:3 for the duration the slot is open.
2. In the second slot, the queues Q1 and Q0 are serviced in the ratio of 2:1.
3. Fresh arbitration is started every time a slot is opened.

In other words, the traffic does not get distributed in the intended ratio of 4:3:2:1; but as two groups with different ratios and only for the duration of the slot when the gates are open continuously

NOTE

See Programming Guidelines for EST

61.3.6.8.2 Frame Preemption (FPE)

Frame preemption breaks the interfering frame into smaller fragments. Therefore, the guard band needs to be only as large as the largest possible interfering fragment instead of the largest possible interfering frame. During the guard band, only the frames that can complete the transmission of the entire frame before the next gate close event are permitted. This ensures that the high priority traffic can always start at the beginning of the window reserved for it.

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame; the preemptable frame transmission is resumed and completed after the express frame transmission is complete. To support frame preemption, the following two abstractions of the MAC are used:

- A preemptable MAC, called pMAC, which carries the preemptable traffic.
- An express MAC, called eMAC, which carries the express traffic.

In the implementation, only the parts of the MAC that hold the state during preemption are replicated and represented as pMAC and eMAC. The MAC uses the following two ways to put the transmission of the preemtable traffic on hold, in the presence of express traffic:

- The MTL scheduler interrupts the preemptable traffic that is currently being transmitted. When the preemption capability is active, the MAC interrupts the transmission and reception of preemptable frames. A preempted fragment can be followed by zero or more express frames, before the continuation fragments. The preemptable frame can be fragmented any number of times, however, the minimum final and non-final fragment length criterion must be met. However, interleaving of more than 1 preemptable packet is not permitted. This implies that if a preemptable packet is fragmented by an express packet, another preemptable packet cannot be transferred until all the remaining fragments of the first preempted packet are transferred.
- The MTL scheduler prevents starting the transmission of preemptable traffic. When the preemption capability is inactive, the pMAC entity is disabled and only express traffic is transmitted or received. Frame preemption feature can be enabled by setting EFPE field of the MAC_FPE_CTRL_STS register.

61.3.6.8.2.1 GCL Modification to Support FPE

In the EST-only configuration, the GCL entry has up to 24 bits of Time Interval and up to 8 high order bits representing the Gate Open/Close state requirements as shown in the following figure.

Gate Control (up to 5 bits)	Time Interval(ns) 16, 20 , or 24 bits
OOCCC	T0
OOOOC	T1
CCCCO	T2
CCCCC	T3
|	|
|	|
OCOCO	T126
OOOOC	T127

Figure 61-18. Gate Control List When FPE is Disabled

EST only supports SetGate operation, which implies that the gates are set to either open or close at a given time interval. However, when both Frame Preemption (FPE) and EST are enabled, the GCL also supports Set-and-Hold-MAC and Set-and-Release-MAC operations, in addition to the SetGate operation. To enable the support of hold and release

operations, the format of the GCL is slightly changed while configuring and enabling the FPE. The first Queue (Q0) is always programmed to carry preemption traffic and therefore it is always Open. The bit corresponding to Q0 is renamed as Release/Hold MAC control. The TX Queues whose packets are preemptable are indicated by setting the PEC field of the MTL_FPE_CTRL_STS register. The GCL bit of the corresponding Queue are ignored and considered as always "Open". So, even if the software writes a "0" ("C"), it is ignored for such queues.

Gate Control (up to 4 bits)	Release/Hold Indication	Time Interval(ns) 16, 20, or 24 bits
CCCO	0	T0
CCCC	0	T1
OCCC	1	T2
COCC	1	T3
CCOC	1	T4
CCCO	0	T5
CCCC	0	T6

Figure 61-19. Gate Control List When FPE is Enabled

When the Release/Hold bit transitions from a '0' to '1', a Set-and-Hold-MAC operation is performed. This marks the cease of the preemptable traffic. This is achieved by sending a Hold request to MTL "ha" ns in advance (where ha is the time interval mentioned in the Hold Advance (HADV) field of the MTL_FPE_Advance register). When the Release/ Hold bit transitions from a '1' to '0' a Set-and-Release-MAC operation is performed. This marks the resuming of the preemptable traffic. This is achieved by sending a Release request to MTL "ra" ns in advance (where ra is the time interval mentioned in the Release Advance (RADV) field of the MTL_FPE_Advance register). The preemptable traffic is blocked for the time duration the Release/Hold bit is set to a '1' in the Gate Control List.

61.3.6.8.2.2 Impact of Preemption on CBS

In Credit Based Shaper (CBS), the definition of "Transmit" is:

- TRUE for the duration of frame transmission from the queue.
- FALSE when frame transmission from the queue is complete.
- When CBS algorithm is used with frame preemption, the value of "Transmit" is TRUE only while the frame is being transmitted by the MAC. If the frame

transmission is delayed or interrupted (for instance, a preemptable frame transmission is interrupted to allow the transmission of an express frame from a different queue, or the start of an express frame is delayed because a preemptable frame is being transmitted) the value of "Transmit" is FALSE until transmission of the frame commences or is resumed.

Also, the value of "Transmit" is FALSE during the transmission of any overhead that is a consequence of frame preemption, (for example, when additional frame overhead, such as mCRC or Fragment Count, is added to the preemptable frame.

At any given time, if there are no frames in the queue, and the value of Transmit is FALSE, and credit is positive value, the credit value is set to zero if there is no preemptable frame from the queue for which transmission is in progress but has been interrupted.

61.3.6.8.2.3 mPacket Format

When the preemption capability is active, MAC sends mPackets to the PHY. An mPacket can be one of the following:

1. A express packet
2. A preemptable packet
3. An initial fragment of a preemptable packet
4. A continuation fragment of a preemptable packet

Start mPacket Delimiter (SMD)

The value of the SMD indicates whether the mPacket contains an express packet, the start of a preemptable packet (initial fragment or complete packet), or any of continuation fragments of a preemptable packet. Following table shows the valid SMD values.

Table 61-23. Possible SMD Values of mPacket

mPacket Type	Notation	Frame Count	Value
verify packet	SMD-V	-	0x07
respond packet	SMD-R	-	0x19
express packet	SMD-E	-	0xD5
preemptable packet start	SMD-S0	0	0xE6
	SMD-S1	1	0x4C
	SMD-S2	2	0x7F
	SMD-S3	3	0xB3
continuation fragment	SMD-C0	0	0x61
	SMD-C1	1	0x52
	SMD-C2	2	0x9E
	SMD-C3	3	0x2A

frag_count

A frag_count is a modulo-4 counter that increments for each continuation fragment of the preemptable packet. The frag_count protects against mPacket reassembly errors by enabling detection of the loss of up to 3 packet fragments.

The frag_count field is present only in mPackets with SMD-C notation (continuation fragment). The frag_count is zero in the first continuation fragment of each preemptable packet.

Table 61-24. Possible frag_count Values

frag_count	Value
0	0xE6
1	0x4C
2	0x7F
3	0xB3

mData

The contents of the packet from the MAC, starting with the first byte after the SFD to the last byte before the FCS are sent in the mData fields of one or more mPackets for that frame. The minimum size of the mData field is 60 bytes.

CRC The CRC field contains a cyclic redundancy check (CRC) and has an indication of the final mPacket of a frame. In the final mPacket of a frame, the CRC field contains the last 4 octets of the MAC frame (the FCS field).

For other mPackets, the CRC field contains an mCRC value. The mCRC is calculated on the octets of the packet from the first octet of the frame (the octet following the SFD of preemption frames) to the last octet of the packet transmitted in that mPacket by performing an XOR of the calculated 32 bit CRC value of the fragment and a value of 0x0000FFFF.

Summary of Packet Formats

- Express Packet: 7bytes of PREAMBLE, SMD-E, Data, and CRC
- Complete Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, CRC
- Initial Fragment (non-final) of Preemptable Packet: 7 bytes of PREAMBLE, <Current Preemptable packet SMD>, Data, mCRC

- Continuation fragments (non-final) of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, mCRC
- Final fragment of Preemptable packet: 6 bytes of PREAMBLE, <Current Preemptable continuation fragment SMD>, <Current Preemptable continuation fragment FC>, Data, CRC

Table 61-25. Current and Previous SMD Values

Previous Preemptable packet SMD	Current Preemptable packet SMD
SMD-S0	SMD-S1
SMD-S1	SMD-S2
SMD-S2	SMD-S3
SMD-S3	SMD-S0

Table 61-26. Current and Previous SMD Values

Previous Preemptable fragment SMD	Previous Preemptable fragment FC	Current Preemptable continuation fragment SMD	Current Preemptable continuation fragment FC
SMD-S0	NA	SMD-C0	FC0
SMD-S1	NA	SMD-C1	FC0
SMD-S2	NA	SMD-C2	FC0
SMD-S3	NA	SMD-C3	FC0
SMD-C0	FC0	SMD-C0	FC1
SMD-C0	FC1	SMD-C0	FC2
SMD-C0	FC2	SMD-C0	FC3
SMD-C0	FC3	SMD-C0	FC0
SMD-C1	FC0	SMD-C1	FC1
SMD-C1	FC1	SMD-C1	FC2
SMD-C1	FC2	SMD-C1	FC3
SMD-C1	FC3	SMD-C1	FC0
SMD-C2	FC0	SMD-C2	FC1
SMD-C2	FC1	SMD-C2	FC2
SMD-C2	FC2	SMD-C2	FC3
SMD-C2	FC3	SMD-C2	FC0
SMD-C3	FC0	SMD-C3	FC1
SMD-C3	FC1	SMD-C3	FC2
SMD-C3	FC2	SMD-C3	FC3
SMD-C3	FC3	SMD-C3	FC0

61.3.6.8.2.4 Transmit Preemption

When FPE is enabled (setting EFPE=1 in MAC_FPE_CTRL_STS register), the MTL preempts a preemptable frame, when a "hold" request is asserted (EST/Qbv configured and enabled) express frames are available for transmission, that is, frame is present in MTL FIFO and is qualified for arbitration, after ensuring that the minimum mPacket mData field size is met. Therefore, preemption occurs only if at least 60 bytes of the preemptable frame have been transmitted and at least 64 bytes (including the frame CRC) remain to be transmitted.

The earliest starting position of preemption is controlled by the AFSZ field of the MTL_ FPE_CTRL_STS Register. Preemption does not occur until at least 64 * (1+ AFSZ) - 4 bytes of the preemptable frame have been sent.

When preemption occurs, all the preemptable queues are blocked and only the express queues are allowed to arbitrate (if more than one express queue has traffic) and transmit.

Continuation fragment of the preempted frame is the first frame to be transmitted after "release" request is asserted (EST/Qbv configured and enabled) and all the express traffic transmission completes.

NOTE

PTP packets should be transmitted as express packets.

MTL communicates the following frame-type information to the MAC using a 2-bit preemption control signal on the MTI interface qualified by SoF and EoF.

- Express Frame
- Preemption Frame (Full or Fragment)
- Continuation Fragment (non-Final or Final)

Table 61-27. Preemption Control Values on MTI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

MTL should wait for the previous fragment status to be received before resuming the continuation fragments of a preempted frame. Fragment status is described in detail in the Tx Fragment Status section.

61.3.6.8.2.4.1 MAC Tx Preemption

MAC supports preemption by implementing the functionality needed to generate the mPackets as described in “mPacket Format”.

Based on the Preemption Control value received on the MTI interface (qualified with SoF and EoF), MAC determines the frame type (shown in the above table 'Preemption Control Values on MTI Interface for Various Frame Types') and generates mPackets accordingly.

When the preemption capability is active, MAC replaces the SFD of a preemption packet with an SMD-S value. A 2-bit rolling frame count is encoded in the SMD-S value.

The SMD-E value is the same as the SFD value so the SFD of an express packet does not need to be replaced.

61.3.6.8.2.4.1.1 Tx Fragment Status

MAC sends Tx fragment status to indicate successful transmission of fragmented mPackets.

In case of a transmission error (underflow, jabber, and so on) the frame status is sent (and not a fragment status) with an error indication along with all other relevant status fields. In case of receiving an error status for a transmitted fragment, the MTL drops the remaining fragments and does not send any more continuation fragments.

61.3.6.8.2.5 Receive Preemption

61.3.6.8.2.5.1 MAC Receive Preemption

When FPE is enabled, the MAC Receiver pases the incoming packets and differentiates between Express packets and preemptable packets. An SMD containing an SMD-E indicates express packet, and SMD containing an SMD-S indicates the first mPacket of a preemptable packet.

If an mPacket containing an SMD-S is received when MAC has not completed receiving the previous preempted packet, MAC sets a CRC Error status for the previously received partial packet.

When an SMD-S is detected, MAC records the frame count indicated by the SMD and then begins sending data on the MRI interface.

The MAC checks the last four bytes of the mPacket . If the last four bytes of the mPacket do not match CRC, that indicates the end of the packet with or without a CRC error as per the CRC check result. If the last four octets of the mPacket match, that indicates that the packet was preempted.

An SMD containing an SMD-C indicates an mPacket that continues the data for a preempted packet. Upon receiving an SMD value of SMD-C, MAC checks the following:

1. A preempted packet is in progress
2. The frame count indicated by the SMD matches the frame count of the packet in progress
3. The frag_count value indicates the next fragment count.

If any of the above check fails, the mPacket is discarded and MAC sets a CRC Error status for the partially received packet.

If all the checks pass, the next fragment count is incremented modulo 4.

When a packet is preempted, the MAC saves the state of the partially received packet (filter check status, timestamp, length fields etc.) and will be able to process any received Express packets before the continuation fragment is received.

The MAC Receiver sends a "dummy status" for all the mPacket fragments successfully received and sends the Rx status with the final fragment. If an error is detected during any of the fragments the Rx status is sent and the fragment is marked as final fragment. All subsequent continuation fragments received for this packet are dropped in the MAC.

The MAC communicates the following frame type information to the MTL using a 2-bit preemption control signal.

1. Express Frame
2. Preemption Frame (Full or Fragment)
3. Continuation Fragment (non-Final or Final)

Table 61-28. Preemption Control Values on MRI Interface for Various Frame Types

Qualifier	Preemption Control Value	Frame Type
SoF	00	Start Express
SoF	01	Start Preemption
SoF	10	Continuation Fragment
SoF	11	Reserved
EoF	00	End of Frame
EoF	01	End of Fragment

61.3.6.8.2.5.1.1 Data Alignment

When a received frame cannot be fragmented on any byte boundary, MAC retains the unaligned bytes of data in the previous fragment and resends them with the next fragment as shown in the following figure.

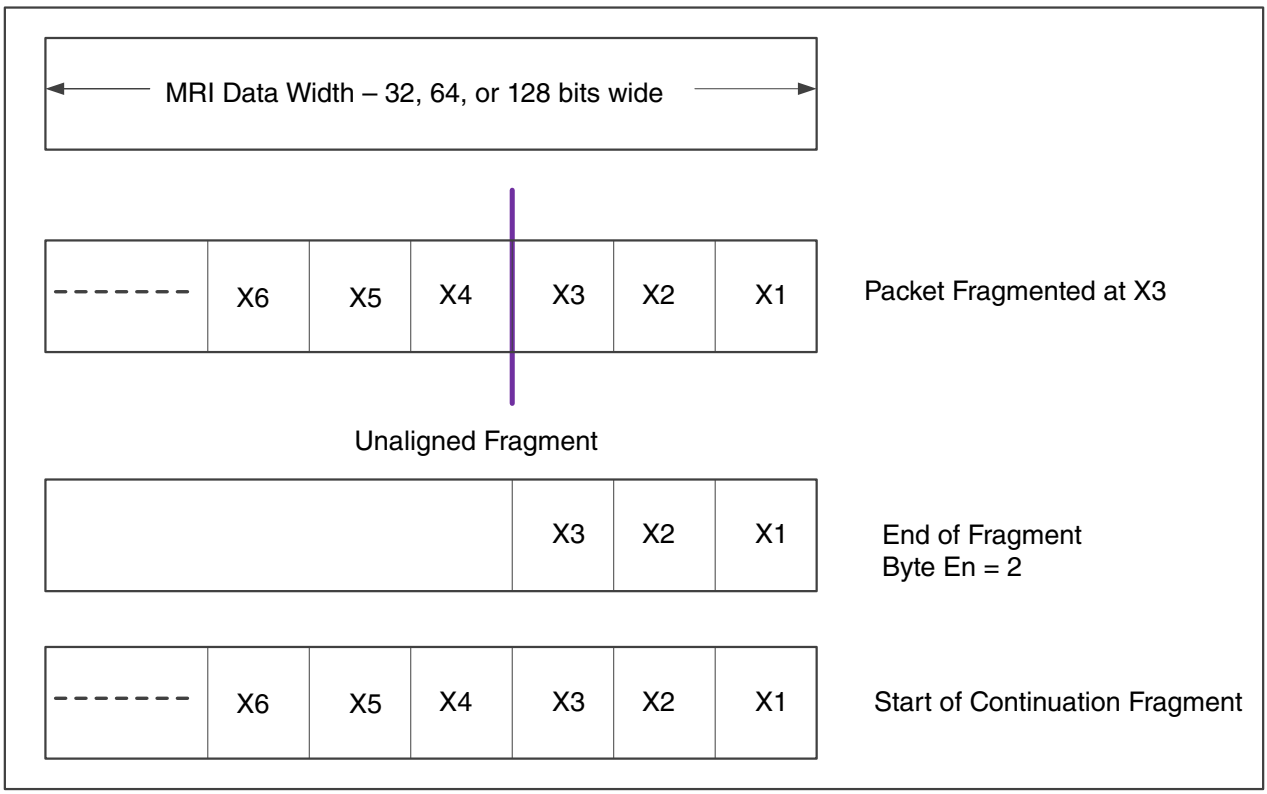


Figure 61-20. Data Alignment Feature of MAC

61.3.6.8.2.5.1.2 MTL Receive Preemption

The MTL Rx must have at least 2 Rx queues to support FPE function as the preemptable packets and the express packets must be routed to separate Rx queues. The destination Rx queue of a received packet is controlled by MAC_RxQ_Ctrl[n] registers. Program these registers such that Preemptable traffic and express traffic are not routed to the same Rx queue. As the queue mapping for tagged packets is based on VLAN user-priority field, this implies that priority of preemptable and express packets are mutually exclusive. In other words, packets of a certain priority (traffic class) are either express or preemptable but cannot be both.

For the preemptable frames, in addition to the PSRQ (Priority Selected in Rx queue) based queue routing, a programmable "Frame Preemption Residue Queue" (FPRQ) is supported to route all other Preemption Packets received (Untagged, SA/ DA or VLAN Filter Fails forwarded due to RA being set or VTFE being reset).

Table 61-29. Preemption Control Values on MRI Interface for Various Frame Types

Preemption Packet Type	Queue Routing
AV Tagged Packets passing the SA/DA and VLAN filters	PSRQ*
DCB/Generic Tagged Packets passing the SA/DA and VLAN filters	PSRQ**
Tagged Packets failing the SA/DA and VLAN filters without setting the Receive All (RA = 0) or setting VTFE = 1	Dropped
All other packets OR when RA = 1	FPRQ

61.3.6.8.2.5.1.3 MTL Receive Arbitration

On the ARI Interface, data is fetched from the MTL Rx FIFO based on the arbitration selected in the MTL_Operation_Mode register. Frame based arbitration can be used only when all the MTL Preemption Queues are operating in Store and Forward mode, else there will be loss of bandwidth on the ARI interface because all the fragments of a preemptable packet is not available. Therefore, any express packets received between the fragments are blocked until all the fragments are received and transferred; this defeats the purpose of express traffic.

When operating in either Threshold (cut through) or Store and forward modes of operation, PBL based arbitration is recommended over Frame based arbitration. In case of PBL based arbitration, the watermark check is always performed and the arbitration/ transfer of data in terms of chunks of "PBL" size of data which is almost similar to the concepts of "fragments". Therefore the express queue packets get blocked for less time as well as the ARI interface transfers the data without loss of efficiency.

61.3.6.8.2.6 Verify and Respond mPackets

When FPE function is present, the MAC can receive and detect the Verify and Respond mPackets, even when FPE is not enabled by software. When MAC detects valid Verify/ Respond mPackets, it notifies the software by setting the RVER and RRSP fields of MAC_FPE_CTRL_STS register respectively. Optionally an interrupt can be generated. As such packets have empty (all-zero) data payload, they are dropped inside the MAC and not forwarded to the MRI.

Software can set the SVER and SRSP fields of MAC_FPE_CTRL_STS register to request MAC to transmit Verify and Respond mPackets respectively. Upon successful transmission of these frames, MAC clears the SVER/SRSP bits and sets the TVER & TRSP fields of MAC_FPE_CTRL_STS register. Optionally an interrupt can be generated when these events occur.

61.3.6.8.2.7 Frame Preemption and MMC Counter and Interrupt Registers

The following MMC counters and associated Interrupt registers are instantiated/present in the MAC when Frame Preemption feature is enabled.

Table 61-30. MMC Counters and Associated Interrupt Registers

Frame Assembly Error Counter	Description	Associated MMC Counter
Frame Assembly Error Counter	A 32-bit counter that provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.	MMC_Rx_Packet_Assembly_Err_Cntr
Frame SMD Error Counter	A 32-bit counter that provides the number of received MAC frames rejected due to arriving with an SMD-C when there was no preceding preempted frame.	MMC_Rx_Packet_SMD_Err_Cntr
Frame Assembly OK Counter	A 32-bit counter that provides the number of MAC frames that were successfully reassembled.	MMC_Rx_Packet_Assembly_Ok_Cntr
MAC Rx Fragment Counter	A 32-bit counter that provides the number of additional mPackets received due to preemption.	MMC_Rx_FPE_Fragment_Cntr
MAC Tx Fragment Counter	A 32-bit counter that provides the number of additional mPackets transmitted due to preemption.	MMC_Tx_FPE_Fragment_Cntr
Hold Request Counter	A 32-bit counter that maintains the count of number of times a hold request is given to MAC.	MMC_Tx_Hold_Req_Cntr

61.3.6.8.2.7.1 Additional Registers Associated With MMC Interrupts

Following are the additional registers associated with MMC interrupts for the MMC error counters:

- MMC_FPE_Tx_Interrupt
- MMC_FPE_Tx_Interrupt_Mask
- MMC_Rx_Interrupt
- MMC_FPE_Rx_Interrupt_Mask

61.3.6.8.3 Time-Based Scheduling

Time-based scheduling feature is suitable for traffic whose periodicity and rate are predictable. To improve the quality-of-service of such traffic,

- The transmit DMA fetches the packet from the host memory for transmission at designated time. This helps the software to setup the Transmit descriptors in advance even before packet is ready/available. It reduces the overhead on the software and

avoids constant monitoring of the time and preparing descriptors just in time when the packet is targeted to be transmitted.
- The MAC transmits the packet only at the designated/pre-determined time even if the packets are fetched in advance. This helps in maintaining a constant transmission rate that can be consumed by the receiver station; therefore avoiding congestion and excessive buffering in the network.

The time-based scheduling feature supports fetching and launching an Ethernet packet at (or after) a pre-determined time. The time-based scheduling is supported only in the following modes/configurations:

- Full duplex mode
- Link speed is 100Mbps or higher

The following figure shows the time-based scheduling flow.

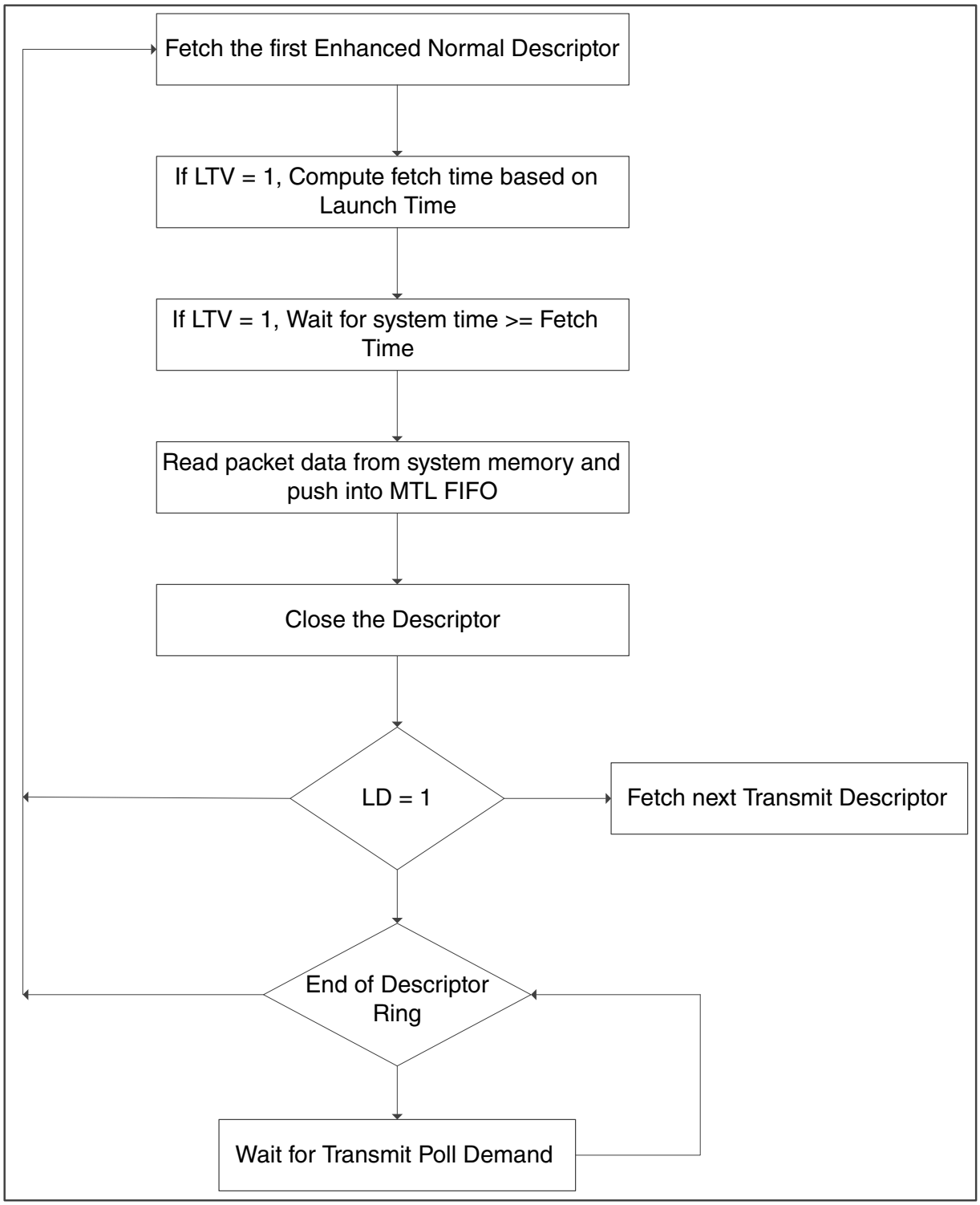


Figure 61-21. Time-Based Scheduling Flow

Following are the definitions specific to the time-based scheduling feature:

61.3.6.8.3.1 Launch Time

Launch time is the time beyond which the MTL can schedule the packet for transmission. The launch time is reset after the transmission of one frame. Following are the two formats of the launch time.

Normal/Absolute Format:

In this format, the launch time is interpreted in the Normal/Absolute format if the ESTM bit of the MTL_TBS_CTRL register is set to 0.

The launch time is a 32-bit value, where most-significant 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns. The launch time is compared against the IEEE 1588 based System/PTP Time (bits[39:8]) and rolls over after 256 seconds.

The maximum value of the lower 32-bits of System Time is 999,999,999 decimal (0x3B9AC9FF) and it wraps to 0 when reaching this value (representing a full second). Therefore, the maximum value of the lower 24 bits of the Launch Time (after multiplying by 256) must be 0x3B9AC9.

As the maximum value of Launch time is 256 seconds,

- Launch Time is greater than current System Time when its value is between System Time[39:8] and System Time[39:8] + 128 sec
- Launch Time is less than current System Time when its value is between "System Time[39:8] + 128sec" and "System Time[39:8] + 256 sec", because this is a modulo 256 computation.

EST/Offset Format

In this format, the Launch time is an offset value relative to the time indicated by the Base Time Register (BTR) of the GCL list provided in the GCL Slot Number (GSN). The value in the BTR is always updated to the start time of the current loop of the GCL.

For each packet, the GSN value and the Launch Time value are specified in the Enhanced Transmit Descriptor in the DMA configurations, or as Control Word in the MTL configurations.

The launch time offset is a 32-bit value; the upper 8-bits represent the time in seconds and the rest 24 bits represent the time in 256 ns, which is added to the BTR value corresponding to GCL Slot Number. The value of the Launch Time Offset must be smaller than the value of the Cycle Time (specified in the CTR register that is implemented when EST is enabled). If the CTR is greater than or equal to 1s, the maximum value of the lower-24 bits of Launch Time offset should be 999,999,999 decimal (0x3B9AC9FF).

In this format, the launch time is a 64-bit value, which is interpreted as, Launch Time = Launch GSN BTR[63:0] + (Launch Time Offset[31:0] << 8), which is compared with the System Time [63:0].

GSN BTR is the Base Time value at which the Launch GSN loop started execution.

GCL Slot Number (GSN) A modulo 16 count of the GCL loop count is implemented and known as GSN. The count is incremented for every new GCL loop; installation of new GCL list does not impact the count. The current GCL slot number can be obtained by reading the CGSN field of the MTL_EST_Status register.

The maximum value of GSN is 15. Therefore GSN values between Current GSN (CGSN field of MTL_EST_Status register) and CGSN+8 represent current or future slots; all other GSN values are interpreted as elapsed slots or past slots. So, for the correct interpretation of time, GSN value must be between CGSN and CGSN+8, for correct interpretation of time.

61.3.6.8.3.2 Launch Expiry Time

Normal/Absolute Mode

In the normal mode, when the LEOV (Launch Expiry Offset Valid bit of MTL_TBS_CTRL register) is set, the Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determines the maximum amount of time a frame is eligible for launch, starting from the time the frame becomes eligible for launch.

The Launch Expiry Offset is a 24-bit value defined in 256ns units, with a maximum possible value of 999,999,999 ns (0x3B9AC9FF).

Launch Expiry Time = (Launch Time[39:8] + LEOS[32:8]) * 256ns

The packet with a specific Launch time is considered eligible for transmission when the Launch time is less than the System Time and (if LEOV is set) the System Time is less than the Launch Expiry time.

When the System time is greater than the Launch Expiry time the frame is categorized as expired and will be dropped from the MTL FIFO.

NOTE

- For correct interpretation and meaningful operation, the Fetch, Launch, and Launch Expiry time should never be set to a value larger than Current System Time + 128 sec; such a value is interpreted as time that has already elapsed.
- In full duplex mode, the frames dropped from the MTL FIFO have Error Summary (Bit 15) and Excessive Deferral (bit 3) of TxStatus set.

EST/Offset Mode

In the EST mode of operation, when the LEOV field of the MTL_TBS_CTRL register is set, the launch expiry GSN offset (LEGOS field of the MTL_TBS_CTRL register) and Launch Expiry Offset (LEOS field of MTL_TBS_CTRL register) determine the maximum amount of time a frame remains eligible for launch, starting from the time the frame becomes eligible for launch.

Launch Expiry Offset = (LEGOS: LEOS)

LEGOS holds the GSN offset (multiples of CTR time) and LEOS holds maximum value of CTR (sub CTR values) in ns.

Launch Expiry offset is a 24-bit value defined in the units of 256 ns, with a maximum possible value of the smaller of 999,999,999ns or CTR-1 ns.

The Launch Expiry GSN is computed as follows:

Launch Expiry GSN = (Launch GSN + LEGOS + CCMA) Where, CCMA is the CTR Carry due to Modulo addition. This value is 1 if ((Launch Time Offset + LEOS) << 8) is equal to or greater than CTR.

When CCMA = 1, Launch Expiry Offset = (Launch Time Offset+ LEOS) - CTR

When CCMA = 0,

Launch Expiry Offset = (Launch Time Offset + LEOS).

Launch Expiry Time = Launch Expiry GSN BTR[63:0] + Launch Expiry Time Offset.

When LEOV is not set, Launch Expiry Time is not checked.

When LEOV is set, and

- the system time is greater than the Launch Expiry time, the frame is dropped from MTL FIFO. The frame is considered as expired.
- the launch time is smaller than the System Time and Launch Expiry Time is greater than system time, the frame is considered eligible for launch.

NOTE

- Max value of LEGOS is 7. This implies that when LEOV is set, the frame has a maximum life time of <8 GCL loop iterations after it becomes eligible for Launch.
- The slot number of the First GCL List executed each time after EST is enabled, is zero.

61.3.6.8.3.3 Fetch Time

The fetch time accounts for all possible delays in the DMA fetch operation and ensures that the frame is present in the MTL FIFO before the launch time.

If FTOV field of DMA_TBS_CTRL register is set, then fetch time can be indicated for each packet. If the FTOV filed is not set, the Fetch Time Offset is not valid and the DMA fetches packets without any time constraints.

61.3.6.8.3.3.1 Normal/Absolute Mode

In the Normal mode fetch time derived/calculated by reducing the time specified in the Fetch Time Offset (FTOS) field of DMA_TBS_CTRL register from the given Launch time.

Normal mode of operation, the Fetch Launch Time is computed as:

Fetch Time[39:8] = (Launch Time[39:8] - FTOS[31:8])

The Fetch time is 32-bits and is compared against System Time[39:8] to determine the Eligible for fetching the frame:

- The Fetch time is defined as "greater" than System Time if the Fetch Time is in the range of "System Time[39:8]" and "System Time[39:8] + 128 sec". The frame is considered as not-eligible for fetch.
- The Fetch time is defined as "smaller" than the System Time if the Fetch time is in the range of "System Time[39:8] + 128 sec" and "System Time[39:8] + 256 sec". The frame is considered as eligible for fetch.

This is a modulo 256 computation.

61.3.6.8.3.3.2 EST/Offset Mode

In the EST/Offset mode, the Fetch GSN Offset (FGOS field of MTL_TBS_CTRL register) provides the Slot Number offset to be deducted from the Launch GSN. In this case the FTOS value should be the smaller of 999,999,999ns or CTR-1 ns.

If (Launch Time Offset >= FTOS):

Fetch Time Offset = ((Launch Time Offset - FTOS) * 256ns)

CBFS(CTR Borrow for Fetch Subtraction) = 0

If (Launch Time Offset < FTOS) Fetch Time Offset = CTR + ((Launch Time Offset - FTOS) * 256ns)

CBFS (CTR Borrow for Fetch Subtraction) = 1

The Fetch GSN is computed as follows:

Fetch GSN = Launch GSN - FGOS - CBFS

Fetch Time = Fetch GSN BTR[63:0] + Fetch Time Offset

The frame is marked eligible for DMA fetch when the fetch time is smaller than the System Time.

61.3.6.8.3.3.3 DMA Operations Sequence

Following is the sequence of operation when FTOV = 1:

1. Fetch the first Enhanced Normal Descriptor (FD is set)
2. If LTV is set and Fetch enabled, compute the Fetch Time based on the Launch Time Wait for the System Time to be greater than Fetch Time
3. Read the Frame (Data) from the host memory and transfer to MTL FIFO
4. Close the Normal Descriptor
5. Fetch the next Normal Descriptor (if the previous Descriptor was not the Last)
6. Repeat steps 4 to 6, until the last descriptor of the frame (LD is set)

After the Last Descriptor of a frame, the next enhanced normal descriptor should be programmed with a new Launch Time and with LTV bit set. Otherwise, the subsequent frames will be processed without any time restrictions.

NOTE

See Programming the Launch Time in Time-Based Scheduling

61.3.6.8.4 Credit-Based Shaper Algorithm

The MTL Queue Scheduler (shown in Figure 61-1) uses the credit-based shaper algorithm to arbitrate the AV traffic in all queues and the legacy Ethernet traffic in Queue 0. You can program the additional queues to use the credit-based shaper algorithm.

The following sections provide information about how you can implement the credit-based shaper algorithm:

- Credit Value
- idleSlopeCredit and sendSlopeCredit Values
- Bandwidth Status

61.3.6.8.4.1 Credit Value

The credit value is part of the credit-based sharper algorithm used by the MTL Queue Scheduler.

The credit value is accumulated every transmit clock cycle, that is, 40 ns for 100 Mbps and 8 ns for 1000 Mbps. The credit to be added or subtracted per cycle can be fractional based on the required idleSlope and sendSlope values, as described in the table below.

Table 61-31. Credit Value Per Transmit Cycle Example

Mode	Values	Description
100 Mbps	• portTransmitRate = 100 Mbps • idleSlope = 70 Mbps (assuming 70% bandwidth reserved for a higher priority traffic class) • sendSlope = 30 Mbps	• credit = 2.8 bits accumulates per cycle (40 ns) for the higher priority traffic class when best-effort packet is being transmitted. • credit = 1.2 bits drains per cycle (40 ns) when higher priority traffic class packet is being transmitted.

The DMA stores the queue traffic in the respective part of the Tx FIFO based on the slot number in the transmit descriptor (if enabled) or the bandwidth availability on the AMBA AHB application bus.

The credit for a queue builds up only when the packet is available but it cannot be transmitted because the MAC is sending a packet from another queue. The module supports another mode in which the credit can build up in advance for a queue in which no packet is available in respective part of the FIFO. This enables sending a burst of high priority traffic in a queue as soon as the data is available. You can enable this mode with Bit 3 of the CBS Control registers of corresponding queues.

When reset, the accumulated credit parameter in the credit-based shaper algorithm is set to zero if there is positive credit, and there is no packet to transmit in a queue. The credit does not accumulate when there is no packet waiting in a queue and other queues are transmitting. When set, the accumulated credit parameter in the credit-based shaper algorithm is not reset to zero if there is positive credit and no packet to transmit in a queue. The credit accumulates even when there is no packet waiting in a queue and other queues are transmitting.

61.3.6.8.4.2 idleSlopeCredit and sendSlopeCredit Values

The idleSlopeCredit and sendSlopeCredit values are programmable.

The software must program the idleSlopeCredit and sendSlopeCredit values. The programmed values should be the credit accumulated or drained per clock cycle scaled by 1024, such as, 2.8 x 1024 = 2867 and 1.2 x 1024 = 1229. In addition, the software must program the hiCredit and loCredit values, scaled by 1024, to adjust for scaling of the idleSlopeCredit and sendSlopeCredit values. This means that if computed hiCredit and

loCredit values are 12,000 bits and 3,036 bits respectively, the values to be programmed in the hiCredit and loCredit registers of the corresponding channel are 12000 x 1024 bits and two's complement of 3036 x 1024, respectively.

61.3.6.8.4.3 Bandwidth Status

The hardware maintains the status of the actual bandwidth consumed by each higher priority queue in the CBS status registers. This enables the software to estimate the average bandwidth consumed by numerically higher traffic classes as compared to the reserved bandwidth.

The CBS status register gives the average number of bits transmitted during the previous programmed slot interval (1, 2, 4, 8, or 16 slots of 125 μs) in a queue. The status register is updated even if the credit-based shaper algorithm is not enabled for a queue. The number of slots over which the average bits transmitted per slot are computed is programmed in Bits[6:4] of the CBS control register of the respective queue. For example, if you have programmed two slots, the average bits are computed over slot numbers 0–1, 2–3, 4–5, and so on.

The value programmed in the idleSlopeCredit register of a queue is proportional to the bandwidth reserved for the queue. The software can allocate any bandwidth that is not used by the higher priority queue to the reserved bandwidth of the lower priority queue.

A lower priority queue, which is using the credit-based shaper algorithm, cannot use the unused reserved bandwidth of any higher priority queue that is using the credit-based shaper algorithm. However, a lower priority queue, which is using the strict-priority algorithm, can use the unused reserved bandwidth of any higher priority queue that uses the credit-based shaper algorithm. For example, Queue 1 and Queue 2 use the credit-based shaper algorithm (with reserved bandwidth of 50% and 25%, respectively) and Queue 0 uses the strict-priority algorithm. If Queue 1 uses only 40% of reserved bandwidth, the remaining 10% is used by Queue 0. Queue 2 cannot exceed the reserved bandwidth of 25%.

61.3.6.9 Queue Modes

Both Tx and Rx queues can handle AV traffic.

Queueing is based on WRR (Weighted Round Robin) or WSP (Weighted Strict Priority) algorithm for generic traffic while it is based on CBS (Credit Base Shaper) and SP (Strict Priority) algorithm for AV traffic.

Whether an Rx Queue is configured for generic or AV-based routing depends on the RXQ#EN field of corresponding queue. Queueing is done based on VLAN tag priority. The VLAN tag priority should match the PSRQ field of the MAC_RXQ_CTRL2 registers. The Rx packets can also be routed to a particular DMA channel based on the DCS field of perfectly-matched MAC Address register.

By default, untagged packets in generic traffic are routed to Receive queue specified in UPQ field of MAC_RXQ_CTRL1 register. Queue 0 is the default value of UPQ field. You can override the default value with any other value, for the UPQ field.

AV control packets (tagged or untagged) are routed based on the AVCPQ field of the MAC_RXQ_CTRL1 register. The PTP over Ethernet packets are routed based on the AVPTPQ field of MAC_RXQ_CTRL1 register.

61.3.6.10 Queue Priorities

Priority of an Rx queue can be programmed in corresponding field of MAC_RxQ_Ctrl2 registers. AV queue (high priority) should be programmed as higher priority than the Best-effort queue (low priority).

61.3.7 TCP/IP Offloading Features

61.3.7.1 Transmit Checksum Offload Engine

The MAC has a Checksum Offload Engine (COE) to support checksum calculation and insertion in the Transmit path, using which, the software can offload the task of checksum insertion to the hardware. In the transmit path MAC calculates the checksum and inserts it in the Tx packet. This feature helps in reducing the load on the software and can improve the overall throughput of the system. This feature is supported for only Q0 queue.

The checksum offload engine module supports two types of checksum calculation and insertion. The checksum engine can be controlled for each packet by setting the CIC bits (TDES3 Bits[17:16]).

Note: The checksum for TCP, UDP, or ICMP is calculated over a complete packet, and then inserted into its corresponding header field. Because of this requirement, when this function is enabled, the Tx FIFO automatically operates in the store-and-forward mode even if IP is configured for Threshold (cut-through) mode.

61.3.7.1.1 IP Header Checksum Engine

In IPv4 datagrams, the integrity of the header fields is indicated by the 16-bit Header Checksum field (the eleventh and twelfth bytes of the IPv4 datagram). The COE detects an IPv4 datagram when the Type field of Ethernet packet has the value 0x0800 and the Version field of IP datagram has the value 0x4. The checksum field of the input packet is ignored during calculation and replaced with the calculated value.

NOTE

IPv6 headers do not have a checksum field. Therefore, the COE does not modify the IPv6 header fields.

The result of this IP header checksum calculation is indicated by the IP Header Error status bit in the Transmit status (TDES3[0]). This status bit is set whenever the values of the Ethernet Type field and the Version field of IP header are not consistent, or when the Ethernet packet does not have enough data, as indicated by the IP header Length field. In other words, this bit is set when an IP header error is asserted under the following circumstances:

- For IPv4 datagrams:
  - The received Ethernet type is 0x0800, but the Version field of IP header is not equal to 0x4.
  - The IPv4 Header Length field indicates a value less than 0x5 (20 bytes).
  - The total packet length is less than the value given in the IPv4 Header Length field.
- For IPv6 datagrams:
  - The Ethernet type is 0x86dd but the IP header Version field is not equal to 0x6.
  - The packet ends before the IPv6 header (40 bytes) or extension header (as given in the corresponding Header Length field in an extension header) is completely received.

61.3.7.1.2 TCP/UDP/ICMP Checksum Engine

The TCP/UDP/ICMP Checksum Engine processes the IPv4 or IPv6 header (including extension headers) and determines whether the encapsulated payload is TCP, UDP, or ICMP. The checksum is calculated for the TCP, UDP, or ICMP payload and inserted into its corresponding field in the header. The Tx COE can work in the following two modes:

- The TCP, UDP, or ICMPv6 pseudo-header is not included in the checksum calculation and is assumed to be present in the Checksum field of the input packet.

This engine includes the Checksum field in the checksum calculation, and then replaces the Checksum field with the final calculated checksum.

- The engine ignores the Checksum field, includes the TCP, UDP, or ICMPv6 pseudo-header data into the checksum calculation, and overwrites the checksum field with the final calculated value.

NOTE

For ICMP-over-IPv4 packets, the Checksum field in the ICMP packet must always be 16'h0000 in both modes, because pseudo-headers are not defined for such packets. If it does not equal 16’h0000, an incorrect checksum may be inserted into the packet.

The result of this operation is indicated by the Payload Checksum Error status bit in the Transmit Status vector (TDES3[12]). This engine sets the Payload Checksum Error status bit when:

- It detects that the packet has been forwarded to the MAC Transmitter engine in the store-and-forward mode without the end of packet (EOP) being written to the FIFO.
- when the packet ends before the number of bytes indicated by the Payload Length field in the IP Header is received.

When the packet is longer than the indicated payload length, the COE ignores them as stuff bytes and no error is reported. When this engine detects the first type of error, it does not modify the TCP, UDP, or ICMP header. For the second error type, it still inserts the calculated checksum into the corresponding header field. The following table describes the functions supported by Transmit Checksum Offload engine based on the packet type. When the MAC does not insert the checksum, it is indicated as "No" in the table.

NOTE

You should not enable checksum insertion for IPv4 or IPv6 packets that are greater than the frame size constraint specified in Description of Transmit Checksum Offload Engine because it may result in incorrect checksum insertion or unexpected behavior.

Table 61-32. Transmit Checksum Offload Engine Functions for Different Packet Types

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/UDPchecksum insertion
Non-IPv4 or IPv6 packet	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in	Yes	Yes

Table continues on the next page...

Table 61-32. Transmit Checksum Offload Engine Functions for Different Packet Types (continued)

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/UDPchecksum insertion
MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.		
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in MAC) but less than or equal to the frame size constraint specified in Transmit Checksum Offload Engine.	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers:		
Hop-by-hop options (in IPv6 main header)	Not Applicable	Yes
Hop-by-hop options (in IPv6 extension header)	Not Applicable	No
Destinations options	Not Applicable	Yes
Routing (with segment left 0)	Not Applicable	No
Routing (with segment left > 0)	Not Applicable	No
TCP, UDP, or ICMP	Not Applicable	Yes
Authentication	Not Applicable	No
Any other next header field in main or extension headers	Not Applicable	No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels:	#160;	#160;
IPv4 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 packet in an IPv4 tunnel	Yes (IPv4 tunnel header)	No
IPv6 Tunnels:	#160;	#160;
IPv4 packet in an IPv6 tunnel	Not applicable	No
IPv6 packet in an IPv6 tunnel	Not applicable	No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled).	Not applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No

Table continues on the next page...

Table 61-32. Transmit Checksum Offload Engine Functions for Different Packet Types (continued)

Packet Type	Hardware IP headerchecksum insertion	Hardware TCP/UDPchecksum insertion
IPv6 frames with security features (such as encapsulated security payload)	Not applicable	No

61.3.7.2 Receive Checksum Offload Engine

IP provides the Checksum Offload Engine that is used to detect any error in an IPv4 or IPv6 packet in the receive path. The MAC verifies the checksum field of the received packet with the internally calculated checksum and provides the status.

The Receive Checksum Offload engine is used to detect errors in IP packets by calculating the header checksum and further matching it with the received header checksum. This engine also identifies a TCP, UDP, or ICMP payload in received IP packets and calculates the checksum of such payloads appropriately.

Here, the IPv4 or IPv6 packets in the received Ethernet packets are detected and processed for data integrity. The MAC receiver identifies IPv4 or IPv6 packets by checking for value 0x0800 or 0x86DD, respectively, in the Type field of the received Ethernet packet. This identification is applicable to single VLAN-tagged packets. It is also applicable to double VLAN-tagged packets when the Enable Double VLAN Processing option is selected and the EDVLP bit of the MAC_VLAN_Tag register is set.

The Rx COE calculates the IPv4 header checksums and checks that they match the received IPv4 header checksums. The result of this operation (pass or fail) is given to the RFC module for insertion into the receive status word. The IP Header Error bit is set for any mismatch between the indicated payload type (Ethernet Type field) and the IP header version, or when the received packet does not have enough bytes, as indicated by the Length field of the IPv4 header (or when fewer than 20 bytes are available in an IPv4 or IPv6 header).

This engine also identifies a TCP, UDP, or ICMP payload in the received IP datagrams (IPv4 or IPv6) and calculates the checksum of such payloads properly, as defined in the TCP, UDP, or ICMP specifications. This engine includes the TCP, UDP, or ICMPv6 pseudo-header bytes for checksum calculation and checks whether the received checksum field matches the calculated value. The result of this operation is given as a Payload Checksum Error bit in the receive status word. This status bit is also set if the length of the TCP, UDP, or ICMP payload does not match the expected payload length given in the IP header.

Following table describes the functions supported by the Rx COE based on the packet type. When the payload of an IP packet is not processed (indicated as "No" in the table), the information (whether the checksum engine is bypassed or not) is given in the receive status.

NOTE

The MAC does not append any payload checksum bytes to the received Ethernet packets.

Table 61-33. Receive Checksum Offload Engine Functions for Different Packet Types

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
Non-IPv4 or IPv6	No	No
IPv4 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Yes	Yes
IPv6 packet is greater than 1,522 bytes (2,000 bytes when IEEE 802.3ad Support for 2K Packets is enabled in the MAC)	Not Applicable	Yes
IPv4 with TCP, UDP, or ICMP	Yes	Yes
IPv4 header's protocol field contains a protocol other than TCP, UDP, or ICMP	Yes	No
IPv4 packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet is an IPv4 fragment	Yes	No
IPv6 packet with the following next header fields in main or extension headers: • Hop-by-hop options (in IPv6 main header) • Hop-by-hop options (in IPv6 extension header) • Destinations options • Routing (with segment left 0) • Routing (with segment left > 0) • TCP, UDP, or ICMP • Any other next header field in main or extension headers	• Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable • Not Applicable	• Yes • No • Yes • Yes • No • Yes • No
IPv4 packet has TCP header with Options fields	Yes	Yes
IPv4 Tunnels: • IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel	• Yes (IPv4 tunnel header) • Yes (IPv4 tunnel header)	• No • No

Table continues on the next page...

Table 61-33. Receive Checksum Offload Engine Functions for Different Packet Types (continued)

Packet type	Hardware IP header checksum checking	Hardware TCP/UDP/ICMP checksum checking
IPv4 Tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	• Not Applicable • Not Applicable	• No • No
IPv4 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Yes	Yes
IPv6 packet has 802.3ac tag (with C-VLAN Tag or S-VLAN Tag when enabled)	Not Applicable	Yes
IPv4 frames with security features (such as encapsulated security payload)	Yes	No
IPv6 frames with security features (such as encapsulated security payload)	Not Applicable	No

Set the IPC bit of the MAC_Configuration register for enabling Receive Checksum offload.

61.3.8 MAC Management Counters

IP supports storing the statistics about the received and transmitted packets in registers that are accessible through the application/software. The MMC module maintains a set of registers for gathering statistics on the received and transmitted packets. The MMC counters are free running.

For MMC register details, see registers "MMC_Control" to "MMC_Rx_FPE_Fragment_Cntr" in ENET_QOS register descriptions.

61.3.9 Flow Control

61.3.9.1 Transmit Flow Control

The Transmit Flow Control involves transmitting Pause packets in full-duplex mode and backpressure in half-duplex mode to control the flow of packets from the remote end.

61.3.9.1.1 Flow control in Full duplex mode

IP supports the IEEE802.3x Pause Packets.

Pause packet structure is shown in the following table.

Table 61-34. Pause Packet Fields

Field	Description
DA	Contains the special multicast address
SA	Contains the MAC address 0
Type	Contains 8808
MAC Control opcode	Contains 0001 for IEEE 802.3x Pause Control packets; 0101 for PFC packets
PT	Contains Pause time specified in the PT field of the MAC_Q#_Tx_Flow_Ctrl register

61.3.9.1.2 Pause Packet Control

When the FCB bit is set, the MAC generates and transmits a single Pause packet. If the FCB bit is set again after the Pause packet transmission is complete, the MAC sends another Pause packet irrespective of whether the pause time is complete or not. To extend the pause or terminate the pause prior to the time specified in the previously-transmitted Pause packet, the application should program the Pause Time register with appropriate value and then again set the FCB bit.

61.3.9.1.3 Flow Control in Half-Duplex Mode

In half-duplex mode, the MAC uses the deferral mechanism for the flow control (backpressure). When the application requests to stop receiving packets, the MAC sends a JAM pattern of 32 bytes when it senses a packet reception, provided the transmit flow control is enabled. This results in a collision and the remote station backs off. If the application requests a packet to be transmitted, it is scheduled and transmitted even when the backpressure is activated. If the backpressure is kept activated for a long time (and more than 16 consecutive collision events occur), the remote stations abort the transmission because of excessive collisions.

Following table describes the flow control in the Tx path for Queue 0 based on the setting of the following bits:

- EHFC bit of MTL_RxQ0_Operation_Mode register
- TFE bit of MAC_Q0_Tx_Flow_Ctrl register
- DM bit of MAC_Configuration register

Flow control is similar for all queues.

Table 61-35. Tx MAC Flow Control

EHFC	TFE	DM	Description
x	0	x	The MAC transmitter does not perform the flow control or backpressure operation.
0	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.
1	1	0	The MAC transmitter performs back-pressure when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx performs back-pressure when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.
0	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1.
1	1	1	The MAC transmitter sends the Pause packet when Bit 0 of MAC_Q0_Tx_Flow_Ctrl register is set or the sideband signal sbd_flowctrl_i is 1. In addition, the MAC Tx sends a Pause packet when Rx Queue level crosses the threshold set by Bits[10:8] of MTL_RxQ0_Operation_Mode register.

61.3.9.1.4 Enabling Transmit Flow Control

To independently enable Transmit Flow Control for each Tx queue, set the TFE bit in the MAC_Q#_Tx_Flow_Ctrl register.

61.3.9.2 Receive Flow Control

In the Receive path, the Flow Control is functional only in the full-duplex mode. If any Pause packet is received in the half-duplex mode, the packet is considered as a normal control packet.

NOTE

Receive pause packets should have a frame size of 64 bytes.

The Receive Flow Control is implemented by the MAC based on the bit value of the respective register, and the destination address and different fields of the received packet.

Following table describes the flow control in the Rx path based on the setting of the following bits:

- RFE bit of MAC_RX_FLOW_CTRL register
- DM bit of MAC_CONFIGURATION register

Table 61-36. Rx MAC Flow Control

RFE	DM	Description
0	x	The MAC receiver does not detect the received Pause packets.
1	0	The MAC receiver does not detect the received Pause packets but recognizes such packets as Control packets.
1	1	The MAC receiver detects or processes the Pause packets and responds to such packets by stopping the MAC transmitter.

61.3.9.2.1 Enabling Receive Flow Control

To enable Pause Flow Control, set the RFE bit in the MAC_Rx_Flow_Ctrl register.

61.3.10 Loopback Mode

The MAC supports Loopback of transmitted packets to its receiver.

The following are some guidelines for using the loopback mode:

- Enable loopback only with the full-duplex mode. In half-duplex mode, the carrier sense signal (crs) or collision (col) signal inputs get sampled which may result into issues such as packet dropping.
- If the loopback mode is enabled without connecting a PHY chip (for example, in FPGA setup), you should externally generate the Tx and Rx clocks and provide these clocks to the MAC.
- Do not loop back big packets. Big packets (>1522 bytes) may get corrupted in the loopback FIFO.

To enable this feature, program the LM bit of the MAC_Configuration register.

You can enable loopback for all PHY interfaces. The data is always looped back through internal asynchronous FIFO on to the internal Receive MII or GMII interface, irrespective of which PHY interface is selected. The loopback data is also passed through the corresponding transmit PHY interface block, on to the Ethernet line.

61.3.11 Descriptors

61.3.11.1 Overview

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. The application creates the descriptors in the system memory. The module supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted.
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted.

Each normal descriptor contains two buffers and two address pointers. These buffers enable the adapter port to be compatible with various types of memory management schemes.

NOTE

There is no limit for the number of descriptors that can be used for a single packet.

61.3.11.2 Descriptor Structure

The module supports the ring structure for DMA descriptor as shown in the following figure.

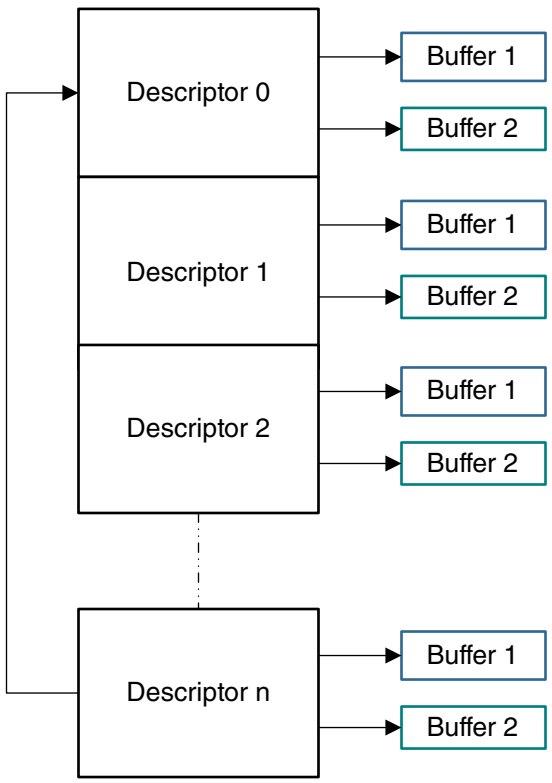


Figure 61-22. Descriptor Ring Structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CH#_Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CH#_TxDesc_Ring_Length)
- Channel n Receive Control 2 register (DMA_CH#_RX_CONTROL2)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process. The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer;

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer;

The DMA automatically wraps around the base address when the end of ring is reached, as shown in the following figure.

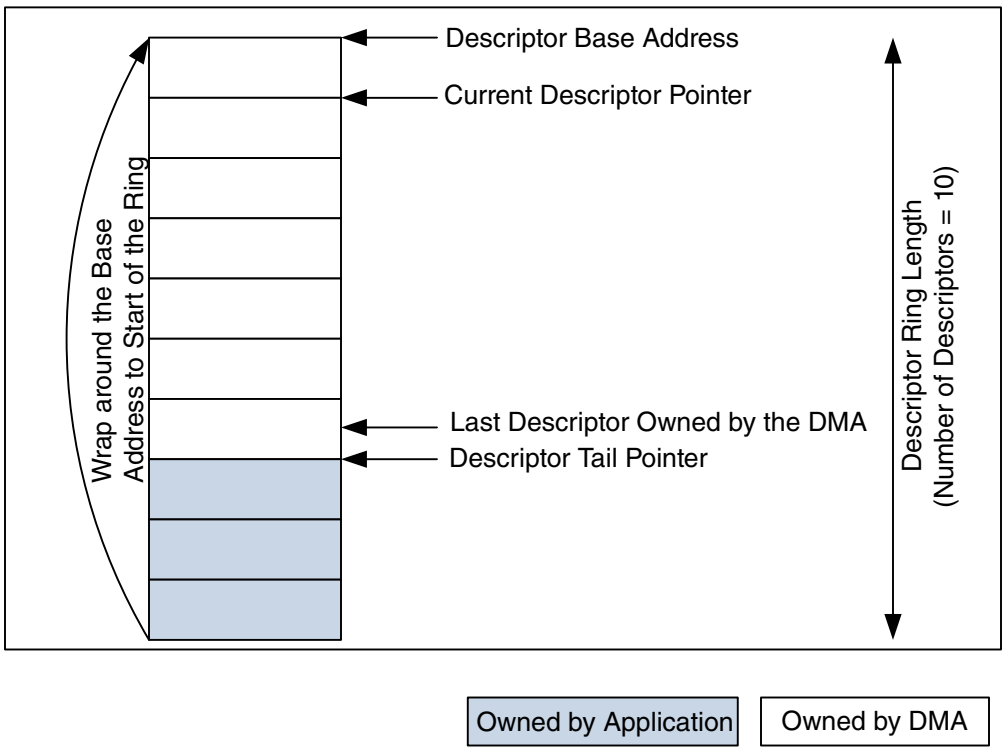


Figure 61-23. DMA Descriptor Ring

For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

61.3.11.3 Transmit Descriptor

The DMA in the module requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format

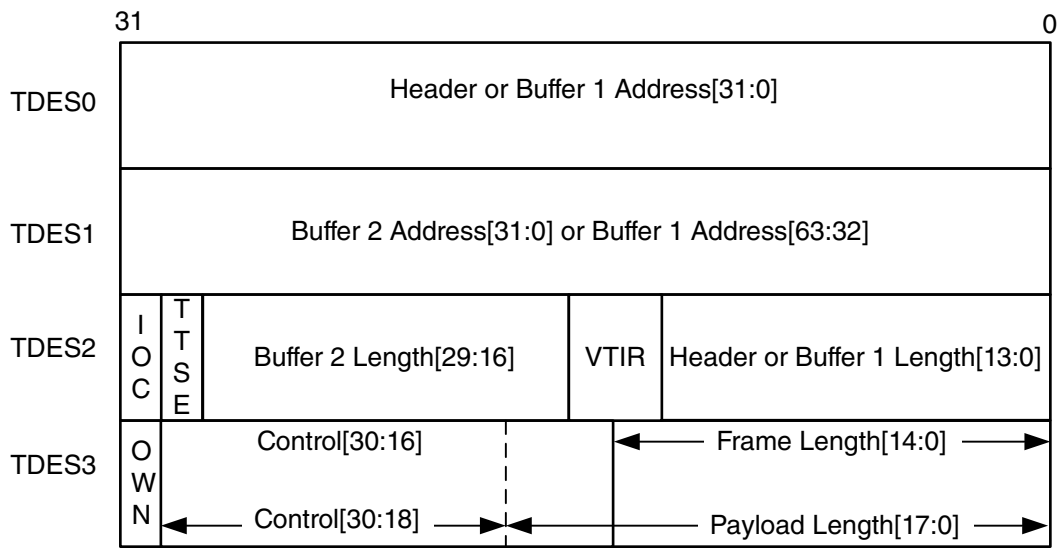
NOTE

TSO Split header is not supported.

61.3.11.3.1 Transmit Normal Descriptor (Read Format)

Following figure shows the Read Format for a Transmit normal descriptor. Table: 'TDES0 Normal Descriptor (Read Format)' through Table: 'TDES3 Normal Descriptor (Read Format)' describe the read format for the Transmit Normal Descriptors: TDES0, TDES1, TDES2, and TDES3.

Figure 61-24. Transmit Descriptor Read Format



61.3.11.3.1.1 TDES0 Normal Descriptor (Read Format)

Table 61-37. TDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: • TSE bit of TDES3 • FD bit of TDES3

61.3.11.3.1.2 TDES1 Normal Descriptor (Read Format)

Table 61-38. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer

Table 61-38. TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
		This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

61.3.11.3.1.3 TDES2 Normal Descriptor (Read Format)

Table 61-39. TDES2 Normal Descriptor (Read Format)

31	30	29-16	15:14	13:0
IOC	TTSE/ TMWD	B2L	VTIR	HL or B1L

Table 61-40. TDES2 Normal Descriptor (Read Format)

Bits	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CH#_Status register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE/TMWD	Transmit Timestamp Enable or External TSO Memory Write Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set. If TSE bit is set and external TSO memory is enabled, setting this bit disables external TSO memory writing for this packet.
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length.
15:14	VTIR	VLAN Tag Insertion or Replacement These bits request the MAC to perform VLAN tagging or untagging before transmitting the packets. The application must set the CRC Pad Control bits appropriately when VLAN Tag Insertion, Replacement, or Deletion is enabled for the packet. The following list describes the values of these bits:

Table continues on the next page...

Table 61-40. TDES2 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		• 2'b00: Do not add a VLAN tag. • 2'b01: Remove the VLAN tag from the packets before transmission. This option should be used only with the VLAN packets. • 2'b10: Insert a VLAN tag with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. • 2'b11: Replace the VLAN tag in packets with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. This option should be used only with the VLAN packets. These bits are valid when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core.
13:0	HL or B1L	Header Length or Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length. If the TCP Segmentation Offload feature is enabled through the TSE bit of TDES3, this field is equal to the header length. When the TSE bit is set in TDES3, the header length includes the length in bytes from Ethernet Source address till the end of the TCP header. The maximum header length supported for TSO feature is 1023 bytes. The maximum header length supported for TSO feature is 1023 bytes. If the TCP Segmentation Offload feature is not enabled, this field is equal to Buffer 1 length.

61.3.11.3.1.4 TDES3 Normal Descriptor (Read Format)

Table 61-41. TDES3 Normal Descriptor (Read Format)

31	30	29	28	27:26	25:23	22:19	18	17:16	15	14:0
OWN	CTXT	FD	LD	CPC	SAIC	SLOTNUM or THL	TSE	CIC/TPL	TPL	FL/TPL

Table 61-42. TDES3 Normal Descriptor (Read Format)

Bits	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of Bits[27:26]: • 2'b00: CRC and Pad Insertion The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. • 2'b01: CRC Insertion (Disable Pad Insertion) The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. • 2'b10: Disable CRC Insertion

Table continues on the next page...

Table 61-42. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. • 2'b11: CRC Replacement The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	SAIC	SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet packet with the value given in the MAC Address 0 or 1 register. The application must set the CRC Pad Control bits appropriately when SA Insertion Control is enabled for the packet. Bit 25 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[24:23]: • 2'b00: Do not include the source address • 2'b01: Include the source address. For reliable transmission, the application must provide frames without source addresses. • 2'b10: Replace the source address. For reliable transmission, the application must provide frames with source addresses. • 2'b11: Reserved

Table continues on the next page...

Table 61-42. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		These bits are valid in the EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core and when the First Segment control bit (TDES3 [29]) is set. This field is valid only for the first descriptor.
22:19	SLOTNUMor THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CH#_Slot_Function_Control_Status. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	TSE	TSO Split header is not supported. The value of this bit is always zero.
17:16	CIC/TPL	Checksum Insertion Control or TCP Payload Length These bits control the checksum calculation and insertion. The following list describes the bit encoding: • 2'b00: Checksum Insertion Disabled. • 2'b01: Only IP header checksum calculation and insertion are enabled.

Table continues on the next page...

Table 61-42. TDES3 Normal Descriptor (Read Format) (continued)

Bits	Name	Description
		• 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. • 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid when the Enable Transmit TCP/IP Checksum Offload option is selected and the TSE bit is reset. When the TSE bit is set, this field contains the upper bits [17:16] of the TCP Payload (or IP Payload for UDP fragmentation). This allows the TCP/UDP packet length field to be spanned across TDES3[17:0] to provide 256 KB packet length support. This field is valid only for the first descriptor.
15	TPL	Reserved or TCP Payload Length When the TSE bit is reset, this bit is reserved. When the TSE bit is set, this is Bit 15 of the TCP payload length [17:0]. This field is valid only when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected while configuring the core.
14:0	FL/TPL	Frame Length or TCP Payload Length This field is equal to the length of the packet to be transmitted in bytes. When the TSE bit is not set, this field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP /IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length When the TSE bit is set, this field is equal to the lower 15 bits of the TCP payload length in case of segmentation and IP payload in case of UDP fragmentation.

Table 61-42. TDES3 Normal Descriptor (Read Format)

Bits	Name	Description
		In case of segmentation, this length does not include Ethernet header or TCP/UDP/IP header length. In case of fragmentation, this length does not include Ethernet header and IP header. When DWRR/WFQ algorithm is NOT enabled, value written into this field is not used when TSE = 0.

61.3.11.3.2 Transmit Normal Descriptor (Write-Back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits.

The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet.

Following figure illustrates the write-back format of the Transmit Descriptor.

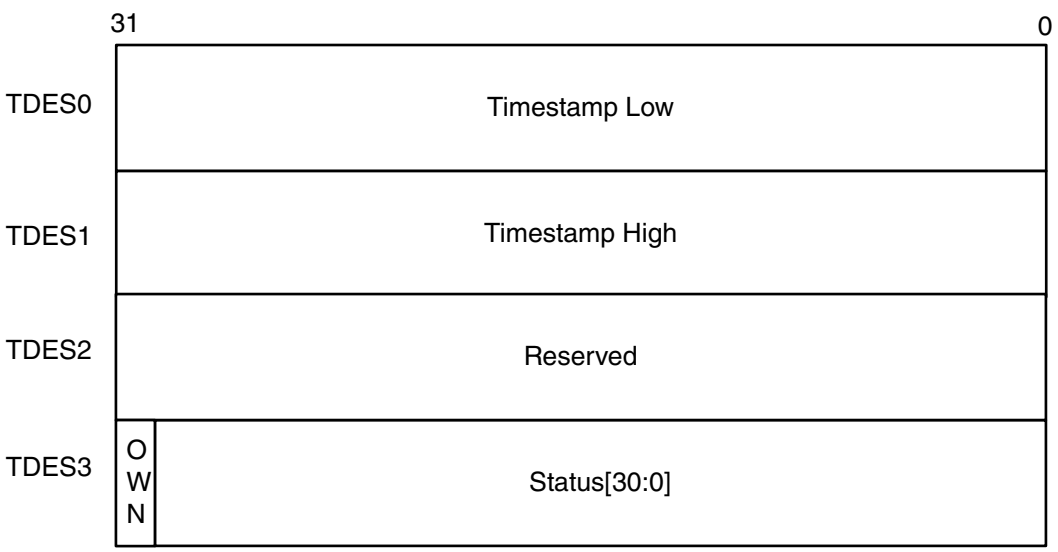


Figure 61-25. Transmit Descriptor Write-Back Format

TDES0 Normal Descriptor (Write-Back Format)

As described in following table, this format is only applicable to the last descriptor of a packet.

Table 61-43. TDES0 Normal Descriptor (Write-Back Format)TDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and the Timestamp status (TTSS) bit is set.

TDES1 Normal Descriptor (Write-Back Format)

As mentioned in the following table, this format is only applicable to the last descriptor of a packet.

Table 61-44. TDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Descriptor bit (LD) in the descriptor is set and Timestamp status (TTSS) bit is set.

TDES2 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 61-45. TDES2 Normal Descriptor (Write-Back Format)

Bit	Description
31:0	Reserved

TDES3 Normal Descriptor (Write-Back Format)

This format is applicable only to the last descriptor of a packet.

Table 61-46. TDES3 Normal Descriptor (Write-Back Format)

31	30	29	28	27:24	23	22:18	17	16	15	14	13	12	11	10	9	8	7:4	3	2	1	0
OWN	CTXT	FD	LD	Rsvd	DE	Rsvd	TTSS	EUE	ES	JT	FF	PCE	LoC	NC	LC	EC	CC	ED	UF	DB	IHE

Table 61-47. TDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the the module DMA owns the descriptor. The DMA clears this bit when it completes the packet transmission. After the write-back is complete, this bit is set to 'b0.
30	CTXT	Context Type This bit should be set to 'b0 for Normal descriptor.
29	FD	First Descriptor This bit indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor This bit is set 'b1 for last descriptor of a packet. The DMA writes the status fields only in the last descriptor of the packet.
27:24	Rsvd	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-back while closing the descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet. • All 1s • CTXT, LD, and FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register

Table continues on the next page...

Table 61-47. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	Rsvd	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	EUE	ECC Uncorrectable Error Status Indicates the ECC uncorrectable error in the TSO memory. Note: Uncorrectable error in Transmit FIFO memory is reported with (Bit 13) FF = 1. This is because, all such packets are flushed by the module.
15	ES	Error Summary This bit indicates the logical OR of the following bits: • TDES3[0]: IP Header Error • TDES3[14]: Jabber Timeout • TDES3[13]: Packet Flush • TDES3[12]: Payload Checksum Error • TDES3[11]: Loss of Carrier • TDES3[10]: No Carrier • TDES3[9]: Late Collision • TDES3[8]: Excessive Collision • TDES3[3]: Excessive Deferral • TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set.
14	JT	Jabber Timeout

Table continues on the next page...

Table 61-47. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	FF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer. When the Full Checksum Offload engine is not enabled, this bit is reserved.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision

Table continues on the next page...

Table 61-47. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that packet transmission was aborted because a collision occurred after the collision window (64 byte times including Preamble in MII mode and 512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: • The DMA encountered an empty Transmit Buffer while transmitting the packet • The application filled the MTL Tx FIFO slower than the MAC transmit rate The transmission process enters the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit

Table continues on the next page...

Table 61-47. TDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

61.3.11.3.3 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature.

The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

NOTE

The VLAN Tag IDs and MSS values, provided by the application in a context descriptor with their corresponding Valid bits set, are stored internally by the DMA. When the outer or inner VLAN tag is provided with the Valid bit set, the DMA always passes the last valid VLAN tag to the MTL. The application cannot invalidate the valid VLAN tag stored by the DMA. The VLAN tag is inserted or replaced based on the control inputs provided for the packet.

The Inner VLAN Tag Control input is used only for the next packet that immediately follows the context descriptor. The application must provide a context descriptor before the normal descriptor of each packet for which the DMA should use the inner VLAN Tag control input.

The following figure shows the format of the Transmit Context descriptor.

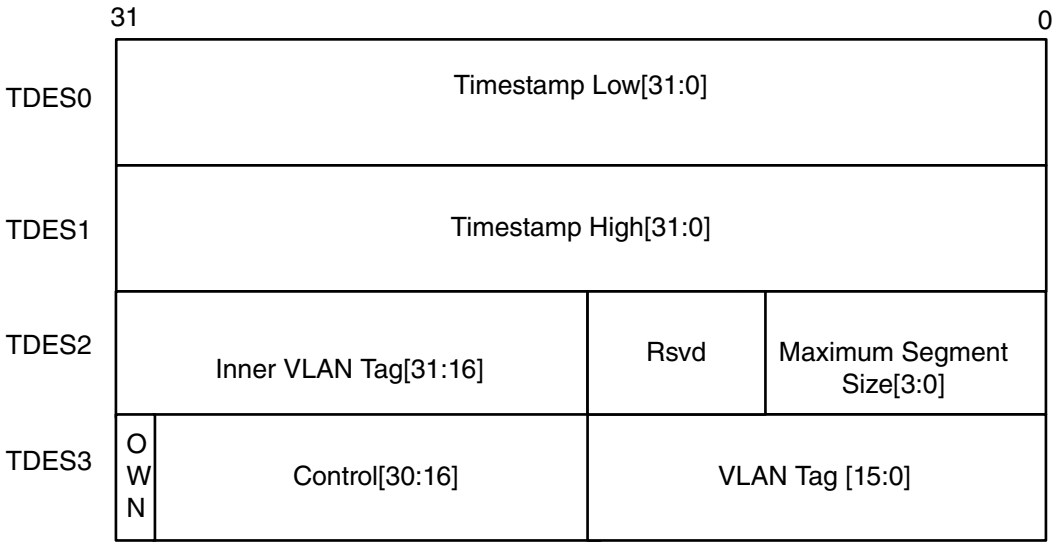


Figure 61-26. Transmit Context Descriptor Format

61.3.11.3.3.1 TDES0 Context Descriptor

Following table describes the format of the TDES0 Context descriptor.

Table 61-48. TDES0 Context Descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

61.3.11.3.3.2 TDES1 Context Descriptor

Following table describes the format of the TDES1 Context descriptor.

Table 61-49. TDES1 Context Descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

61.3.11.3.3.3 TDES2 Context Descriptor

Following table explains the format of the TDES2 Context descriptor.

Table 61-50. TDES2 Context Descriptor

Bit	Name	Description
31:16	IVT	Inner VLAN Tag When the IVLTV bit of TDES3 context descriptor is set and the TCMSSV and OSTC bits of TDES3 context descriptor are reset, TDES2[31:16] contains the inner VLAN Tag to be inserted in the subsequent Transmit packets.
15:14	Rsvd	Reserved
13:0	MSS	TSO Split header is not supported. The value of this bit is always zero.

61.3.11.3.3.4 TDES3 Context Descriptor

Table 61-51. TDES3 Context Descriptor

31	30	29:28	27	26	25:24	23	22:18	17	16	15:0
OWN	CTXT	Rsvd	OSTC	TCMSSV	Rsvd	CDE	Rsvd	IVLTV	VLTV	VT

Follwoing table details the format of the TDES3 Context descriptor.

Table 61-52. TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the

Table continues on the next page...

Table 61-52. TDES3 Context Descriptor (continued)

Bit	Name	Description
		application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1'b1 for Context descriptor.
29:28	Rsvd	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.
25:24	Rsvd	Reserved
23	CDE	Context Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect.The DMA sets this bit during write-back while closing the context descriptor. Descriptor Errors can be: • Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet. • All 1s. • CD, LD, and FD bits set to 1. Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be

Table continues on the next page...

Table 61-52. TDES3 Context Descriptor (continued)

Bit	Name	Description
		considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:20	Rsvd	Reserved
19:18	IVTIR	Inner VLAN Tag Insert or Replace When this bit is set, these bits request the MAC to perform Inner VLAN tagging or un-tagging before transmitting the packets. If the packet is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: • 2'b00: Do not add the inner VLAN tag. • 2'b01: Remove the inner VLAN tag from the packets before transmission. This option should be used only with the VLAN frames. • 2'b10: Insert an inner VLAN tag with the tag value programmed in the MAC_In-ner_VLAN_Incl register or context descriptor. • 2'b11: Replace the inner VLAN tag in packets with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. This option should be used only with the VLAN frames. These bits are valid when the Enable SA and VLAN Insertion on Tx and Enable Double VLAN Processing options are selected.
17	IVLTV	Inner VLAN Tag Valid When this bit is set, it indicates that the IVT field of TDES2 is valid.
16	VLTV	VLAN Tag Valid When this bit is set, it indicates that the VT field of TDES3 is valid.
15:0	VT	VLAN Tag This field contains the VLAN Tag to be inserted or replaced in the packet. This field is used as VLAN Tag only when the VLTI bit of the MAC_VLAN_Incl register is reset.

61.3.11.4 Receive Descriptor

The DMA in the module attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.

The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA will close the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)".

For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

61.3.11.4.1 Receive Normal Descriptor (Read Format)

The read format for a Receive Normal descriptor is made up of a header or Buffer 1 address, reserved field, payload or Buffer 2 or Next Descriptor address, a 30-bit reserved filed, OWN bit, and an interrupt bit.

Following figure shows the Read Format for a Receive Normal Descriptor.

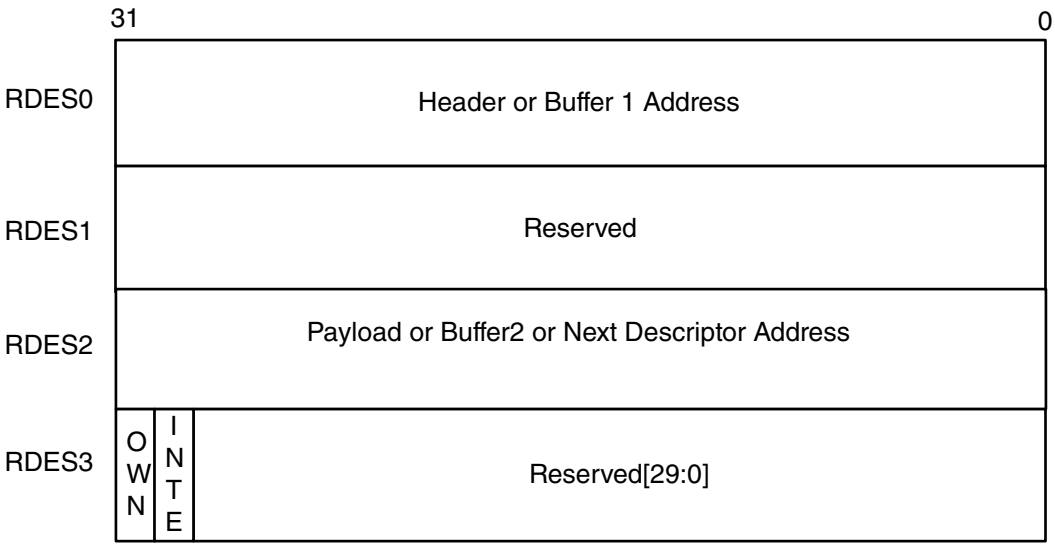


Figure 61-27. Receive Normal Descriptor Read Format

NOTE

In the Receive Descriptor (Read Format), if the Buffer Address field is all 0s, the module does not transfer data to that buffer and skips to the next buffer or next descriptor.

61.3.11.4.1.1 RDES0 Normal Descriptor (Read Format)

Following table explains the read format of the RDES0 Normal Descriptor.

Table 61-53. RDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of Control register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[1:0] (or RDES0[2:0]/[3:0] in case of 64-/128-bit configuration) as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer.

Table 61-53. RDES0 Normal Descriptor (Read Format)

Bit	Name	Description
		If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

61.3.11.4.1.2 RDES1 Normal Descriptor (Read Format)

Following table describes the read format of the RDES1 Normal Descriptor.

Table 61-54. RDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

61.3.11.4.1.3 RDES2 Normal Descriptor (Read Format)

Following table provides the read format of the RDES2 Normal Descriptor.

Table 61-55. RDES2 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 Address Pointer These bits indicate the physical address of Buffer 2. When the SPH bit of the DMA_CH#_Control register is set, the buffer address pointer must be bus width-aligned, that is, RDES2[3:0, 2:0, or 1:0] = 0 corresponding to 128, 64, or 32 bus width. LSBs are ignored internally. When the SPH bit of the DMA_CH#_Control register is reset, there is no limitations on the RDES2 value. However, the RxDMA uses the LS Bits of the pointer address only while transferring the start bytes of a packet. If the BUF2AP is giving the address of a buffer in which the middle or last part of a packet is stored, the DMA ignores BUF2AP[3:0 or 2:0 or 1:0] (corresponding to 128- or 64- or 32-bit data-bus) and writes to the complete location.

61.3.11.4.1.4 RDES3 Normal Descriptor (Read Format)

Following table describes the read format of the RDES3 Normal Descriptor.

31	30	19-26	25	24	23:0
OWN	IOC	Rsvd	BUF2V	BUF1V	Rsvd

Table 61-56. RDES3 Normal Descriptor (Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Rsvd	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Rsvd	Reserved

61.3.11.4.2 Receive Normal Descriptor (Write-Back Format)

The following figure illustrates the write-back format for a Receive Normal descriptor.

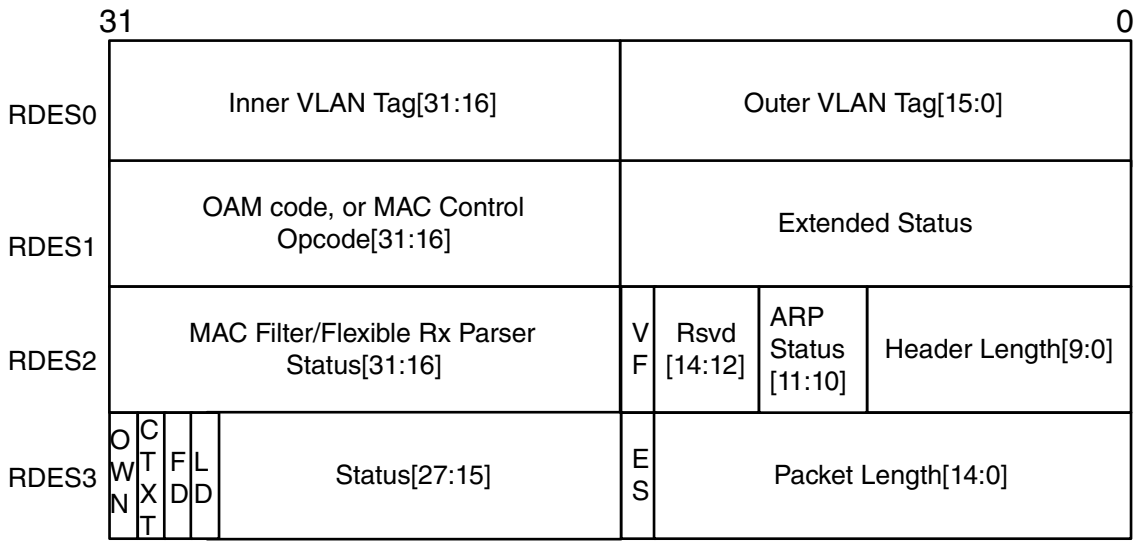


Figure 61-28. Receive Normal Descriptor (Write-Back Format)

NOTE

When Flexible RX Parser is enabled, RDES2[31:16] indicates Parser status, and not the MAC filter status. MAC filter status is not available when Flexible RX parser is enabled.

61.3.11.4.2.1 RDES0 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES0 Normal Descriptor.

Table 61-57. RDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid only when Double VLAN tag processing and VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

61.3.11.4.2.2 RDES1 Normal Descriptor (Write-Back Format)

The Status fields in write-back format are valid only for the last descriptor (RDES3[28] is set). Following table provides the details of the write-back format for RDES1 Normal Descriptor.

31:16	15	14	13	12	11:8	7	6	5	4	3	2:0
OPC	TD	TSA	PV	PFT	PMT	IPCE	IPCB	IPV6	IPV4	IPHE	PT

Table 61-58. RDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If Bits[18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. MAC Control Packet opcode The bits[15:8] of RDES3 contains the subtype and bits [7:0] contains the code.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set. The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.

Table continues on the next page...

Table 61-58. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: • 0000: No PTP message received • 0001: SYNC (all clock types) • 0010: Follow_Up (all clock types) • 0011: Delay_Req (all clock types) • 0100: Delay_Resp (all clock types) • 0101: Pdelay_Req (in peer-to-peer transparent clock) • 0110: Pdelay_Resp (in peer-to-peer transparent clock) • 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) • 1000: Announce • 1001: Management • 1010: Signaling • 1011–1110: Reserved • 1111: PTP packet with Reserved message type These bits are available only when you select the Timestamp feature.
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: • The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment.

Table continues on the next page...

Table 61-58. RDES1 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. • The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the Enable Split Header Feature option is selected and the SPH bit of Control Register of a channel is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present This bit indicates that an IPV4 header is detected. When the SPH bit of RDES3 is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: • The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. • The IP datagram version is not consistent with the Ethernet Type value. • Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
2:0	PT	Payload Type

Table 61-58. RDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
		These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): • 3'b000: Unknown type or IP/AV payload not processed • 3'b001: UDP • 3'b010: TCP • 3'b011: ICMP • 3'b110: AV Tagged Data Packet • 3'b111: AV Tagged Control Packet • 3'b101: AV Untagged Control Packet • 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

61.3.11.4.2.3 RDES2 Normal Descriptor (Write-Back Format)

31:29	28	27	26:19	18	17	16	15	14	13:11	10	9:0
L3L4FM	L4FM	L3FM	MADRM	HF	DAF	SAF	OTS	ITS	Rsvd	ARPNR	HL

Table 61-59. RDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: • 000: Filter 0 • 001: Filter 1 • 010: Filter 2 • 011: Filter 3 • 100: Filter 4 • 101: Filter 5

Table continues on the next page...

Table 61-59. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• 110: Filter 6 • 111: Filter 7 This field is valid only when Bit 28 or Bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter. Note: This status is not available when Flexible RX Parser is enabled.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: • Layer 3 fields are not enabled and all enabled Layer 4 fields match • All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: • All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed • All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset.

Table continues on the next page...

Table 61-59. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register. Note: This status is not available when Flexible RX Parser is enabled.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits[26:19] indicate the hash value. Note: This status is not available when Flexible RX Parser is enabled.
17	DAF/RXPI	Destination Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the DA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet parsing is incomplete (RXPI) due to ECC error. Note: When this bit is set, ES bit of RDES3 is also set.
16	SAF/RXPD	SA Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the SA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet is dropped (RXPD) by the parser. Note: When this bit is set, ES bit of RDES3 is also set.
15	OTS	VLAN Filter Status When set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter. This bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames
14	ITS	Inner VLAN Tag Filter Status (ITS) This bit is valid only for Double VLAN Tagged frames, when Double VLAN Processing is enabled. For more information, see Filter Status.
13:11	Rsvd	Reserved

Table continues on the next page...

Table 61-59. RDES2 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply to earlier ARP request (only one ARP request is processed at a time). This bit is reserved when the Enable IPv4 ARP Offload option is not selected.
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header. This field is valid when the Enable Split Header Feature option is selected.

61.3.11.4.2.4 RDES3 Normal Descriptor (Write-Back Format)

Following table describes the write-back format for the RDES3 Normal Descriptor.

31	30	29	28	27	26	25	24	23	22	21	20	19	18:16	15	14:0
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT	ES	PL

Table 61-60. RDES3 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the module DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true:

Table continues on the next page...

Table 61-60. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} • 2'b00: Intermediate Descriptor • 2'b01: First Descriptor • 2'b10: Reserved • 2'b11: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet. See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid

Table continues on the next page...

Table 61-60. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error

Table continues on the next page...

Table 61-60. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		When this bit is set, it indicates that the gmii_rxer_i signal is asserted while the gmii_rxdv_i signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd!= 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows: • 3'b000: The packet is a length packet • 3'b001: The packet is a type packet. • 3'b011: The packet is a ARP Request packet type • 3'b100: The packet is a type packet with VLAN Tag • 3'b101: The packet is a type packet with Double VLAN Tag • 3'b110: The packet is a MAC Control packet type • 3'b111: The packet is a OAM packet type • 3'b010: Reserved
15	ES	Error Summary When this bit is set, it indicates the logical OR of the following bits: • RDES3[24]: CRC Error • RDES3[19]: Dribble Error • RDES3[20]: Receive Error • RDES3[22]: Watchdog Timeout • RDES3[21]: Overflow Error • RDES3[23]: Giant Packet

Table continues on the next page...

Table 61-60. RDES3 Normal Descriptor (Write-Back Format) (continued)

Bit	Name	Description
		• RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled • RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled This field is valid only when the LD bit of RDES3 is set.
14:0	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current packet.

61.3.11.4.3 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

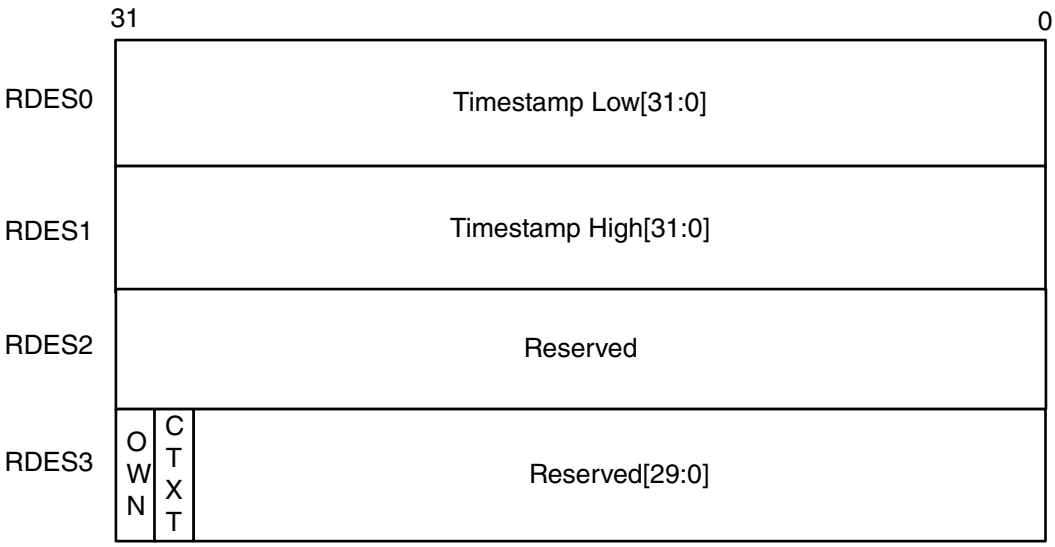


Figure 61-29. Receive Context Descriptor Format

61.3.11.4.3.1 RDES0 Context Descriptor

Table 61-61. RDES0 Context Descriptor

Bit	Name	Description
31:0	RTSL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

61.3.11.4.3.2 RDES1 Context Descriptor

Table 61-62. RDES1 Context Descriptor

Bit	Field	Description
31:0	RTSH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

61.3.11.4.3.3 RDES2 Context Descriptor

Table 61-63. RDES2 Context Descriptor

Bit	Description
31:0	Reserved

61.3.11.4.3.4 RDES3 Context Descriptor

31	30	29	28:0
OWN	CTXT	DE	Rsvd

Table 61-64. RDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: • The DMA completes the packet reception • The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2’b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} • 2’b00: Reserved • 2’b01: Reserved • 2’b10: Context Descriptor • 2’b11: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status

Table continues on the next page...

Table 61-64. RDES3 Context Descriptor (continued)

Bit	Name	Description
		register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using theDE bit along with CTXT bit.
28:0	Rsvd	Reserved

61.3.11.5 Enhanced Descriptor for Time-Based Scheduling

The time-based scheduling feature needs 32 bytes Enhanced Descriptors to be enabled on all the DMA channels that intend to use the feature (by setting the EDSE bit of DMA_CH(#i)_TX_Control register).

The structure of 32-byte Descriptor for the Context and the Normal Descriptor in Read and Write formats are described in the following sections.

61.3.11.5.1 Enhanced Normal Descriptor - Read (32-Bit Mode)

The fields present in the first 16 bytes of the Enhanced Descriptor format of normal descriptor are as follows:

- LTV

  Launch Time Valid indicates the Launch Time (LT) and GSN fields present in the descriptor are valid. The LTV must be set only if the FD bit of the descriptor is not.
- GSN

  This field indicates the GCL slot number associated with the packet.
- Launch Time (LT)

  This field indicates the Launch Time associated with the packet.

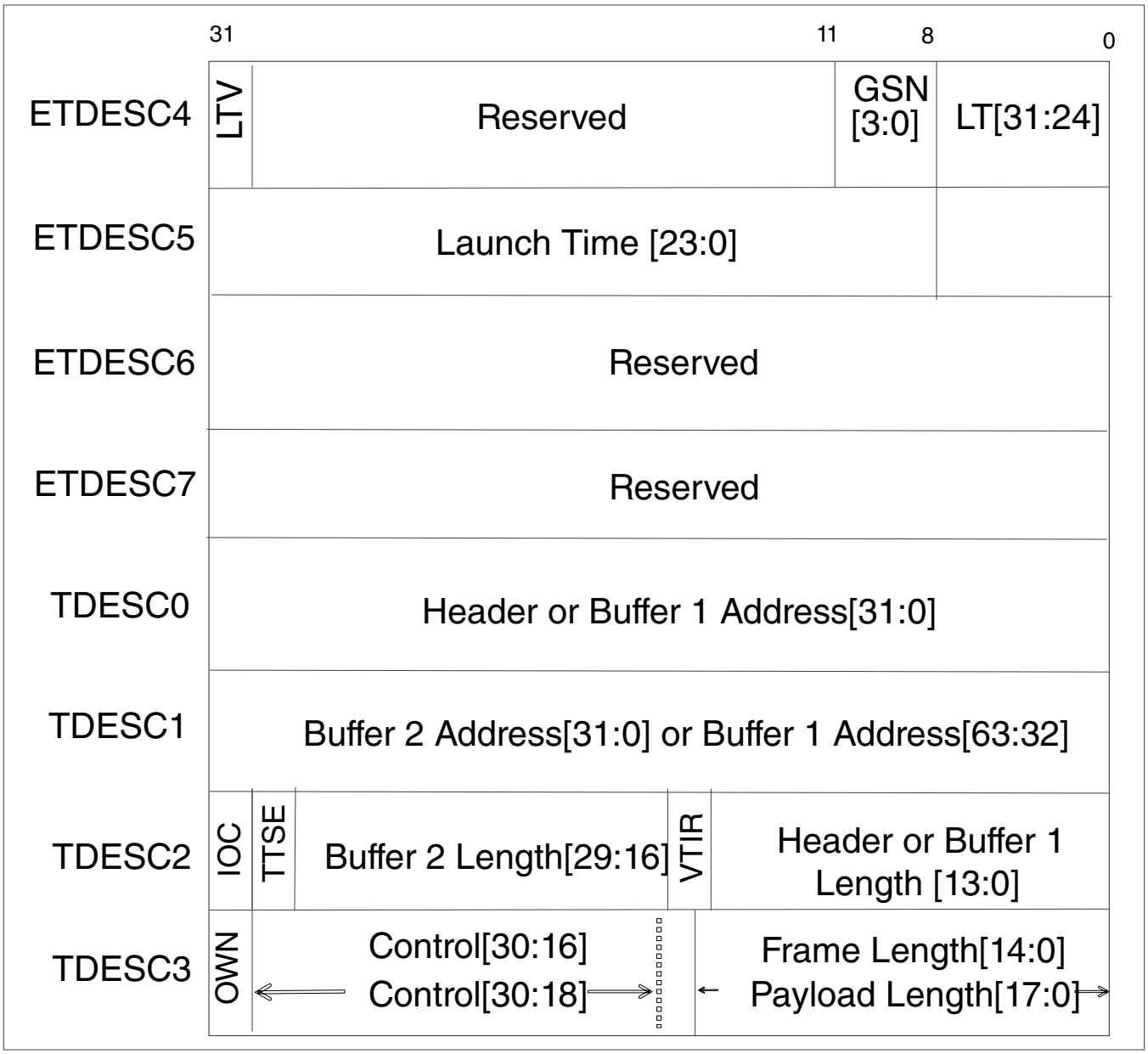


Figure 61-30. Enhanced Normal Descriptor - Read (32-Bit Mode)

61.3.11.5.2 Enhanced Normal Descriptor (Write, 32-Bit Mode)

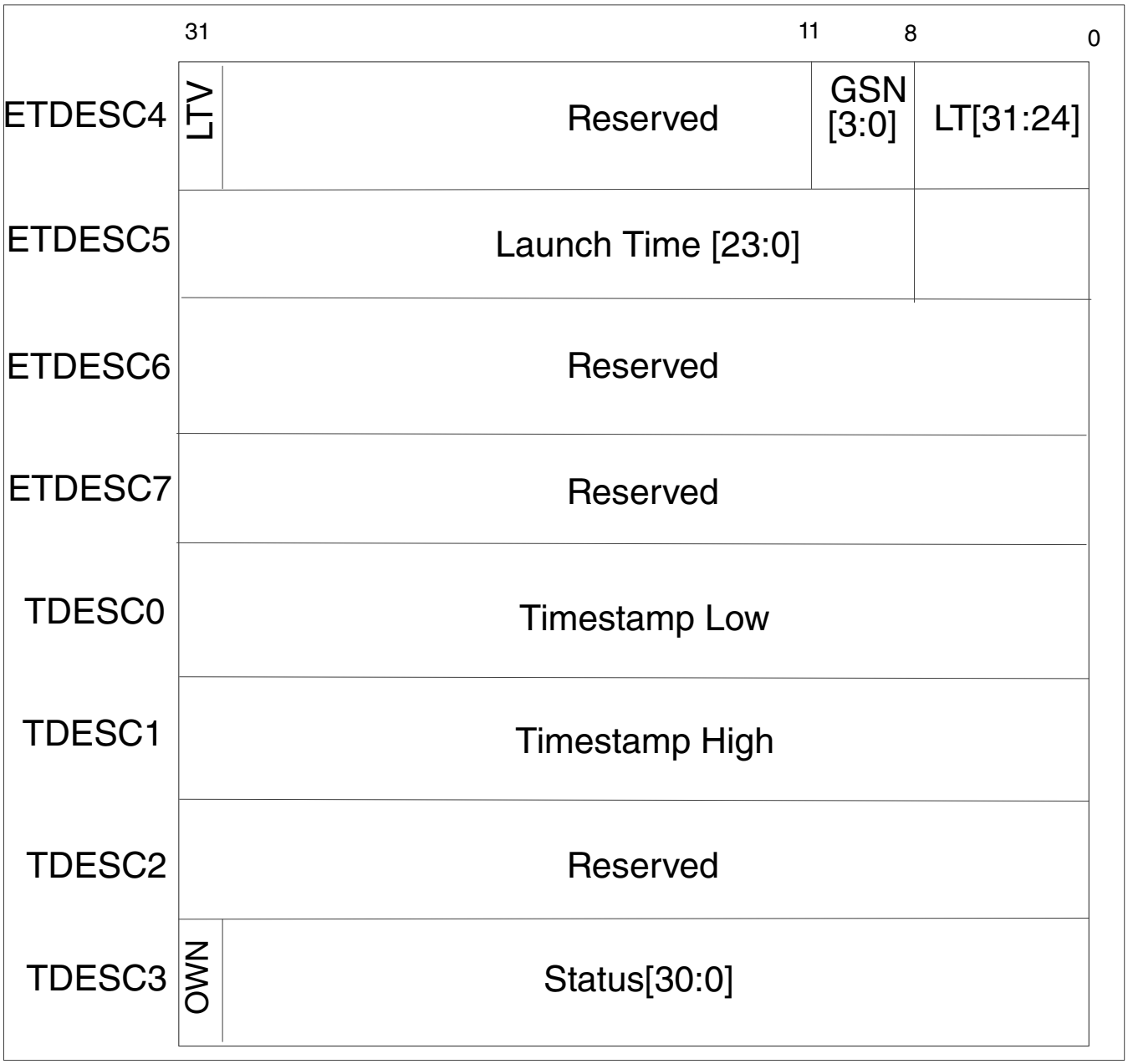


Figure 61-31. Enhanced Normal Descriptor Write (32-Bit Mode)

NOTE

- All the enhanced descriptor examples, assumes a 32-bit data width configurations. However, the same definitions are valid in 64 and 128 bit configurations.

- In both the write back formats, no modifications can be done to the extended 16 bytes. The rest of the 16 bytes (TDESC0 to TDESC3) are written back as per the previous 16 bytes Descriptor format.
- Fetch time when enabled overrides the AV slot function
- When an unaligned new GCL list (with a different CTR) is installed it is recommended not to have traffic during the installation of the new list. Any traffic during the switching of the lists might have unpredictable behavior regarding Fetch, Launch, and Launch expiry as the CTR and BTR values get updated while the frame is being processed

61.3.11.5.3 Enhanced Context Descriptor (Read, 32-bit Mode)

The first 16 bytes of the Enhanced context descriptor (ETDESC4 to ETDESC7) are reserved and must be zeros. The fields present in last bytes of the Descriptor (TDESC0 to TDESC3) are same as the context descriptor (TDESC0 to TDESC3) in 16 byte format.

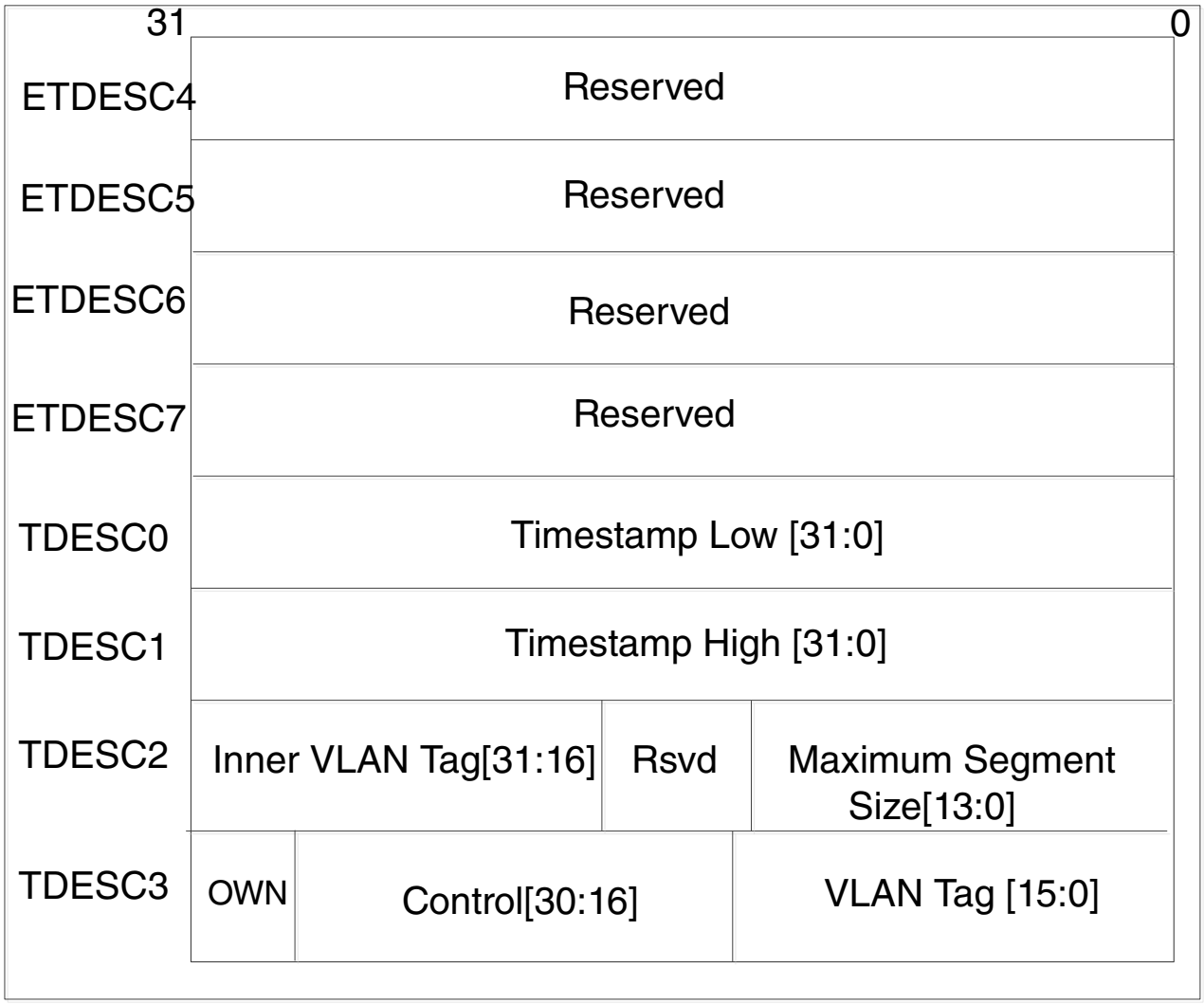


Figure 61-32. Enhanced Context Descriptor - Read

61.3.11.5.4 Enhanced Context Descriptor (Write, 32 Bit Mode)

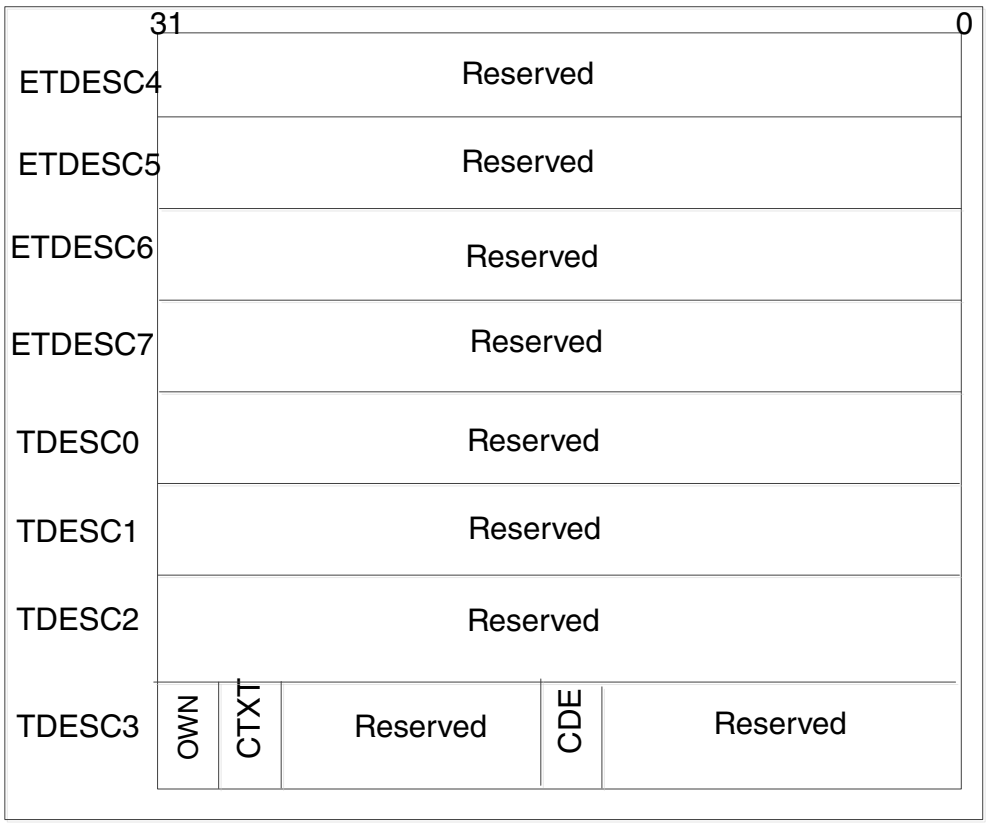


Figure 61-33. Enhanced Context Descriptor - Write

61.3.12 Power management and low power modes

This section describes the power management (PMT) mechanisms supported by the MAC. PMT supports the reception of network (remote) wake-up frames and magic packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and magic packets received by the MAC. The PMT block sits on the receiver path of the MAC and is enabled with remote wake-up frame enable and magic packet enable. These enables are in the PMT control and status register and are programmed by the application.

When the power-down mode is enabled in the PMT, then all received frames are dropped by the Ethernet core and they are not forwarded to the application. The Ethernet comes out of the power-down mode only when either a magic packet or a remote wake-up frame is received and the corresponding detection is enabled.

61.3.12.1 Energy Efficient Ethernet

Energy Efficient Ethernet (EEE) is an operational mode that enables the IEEE 802.3 MAC sublayer along with a family of physical layers to operate in the Low-Power Idle (LPI) mode. The EEE operational mode supports the IEEE 802.3 MAC operation at 100 Mbps. The Ethernet module supports the IEEE 802.3az-2010 for EEE. The LPI mode allows power saving by switching off the parts of the communication device functionality when there is no data to be transmitted and received. The systems on both sides of the link can disable some functionality to save power during periods of low-link utilization. The MAC controls whether the system should enter or exit the LPI mode and communicates this to the PHY. The EEE specifies the capabilities negotiation methods that the link partners can use to determine whether EEE is supported, and then select the set of parameters that are common to both devices.

61.3.12.1.1 Transmit path functions

In the transmit path, the software must set the LPIEN bit of the MAC LPI control status register to indicate to the MAC to stop transmission and initiate the LPI protocol. The MAC completes the transmission in progress, generates its transmission status, and starts transmitting the LPI pattern instead of the IDLE pattern if the link status has been up continuously for a period specified in the LPI LS TIMER(LST) field of MAC LPI timers control register. The PHY link status bit of the LPI control and status register indicates the link status of the PHY.

According to the Energy Efficient Ethernet standard (IEEE 802.3az-2010), the PHY must not stop the transmit clock during the LPI state in the MII (10 or 100) mode.

To make the PHY enter the LPI state, the MAC performs the following tasks:

1. De-asserts transmit enable (TX_EN).
2. Asserts transmit error (TX_ER). Sets TXD[3:0] to 0x1 The MAC maintains the same state of the TX_EN, TX_ER, and TXD signals for the entire duration during which the PHY remains in the LPI state.
3. Updates the status (TLPIEN bit of MAC LPI control status register and generates an interrupt. To bring the PHY out of the LPI state, that is, when the software resets the LPIEN bit, the MAC performs the following tasks:
4. Stops transmitting the LPI pattern and start transmitting the IDLE pattern

5. Starts the LPI TW TIMER
6. Updates the LPI exit status (TLPIEX bit of the MAC LPI control status register and generates an interrupt

The MAC cannot start the transmission until the wake-up time specified for the PHY expires. The auto-negotiated wake-up interval is programmed in the TWT field of the MAC LPI Timers control register.

61.3.12.1.2 Automated entry/exit of LPI mode in TX path

The MAC transmitter can be programmed to enter and exit LPI IDLE mode automatically based on whether it is IDLE for a specific period or has a packet to transfer. These modes are enabled and controlled by the MAC LPI control status register.

When the LPITXA (bit 19) and LPITXEN (bit 16) of the MAC LPI control status register are set, the MAC transmitter enters the LPI IDLE state when the MAC transmit path (including the MTL layers and DMA layers) are idle. The MAC transmitter exits the LPI IDLE state and clears the LPITXEN bit as soon as any of the functions in the TX path (DMA, MTL, or MAC) becomes non-idle due to initiation of a packet transfer. In addition to the above, when bit 20 (LPIATE) is also set, the MAC transmitter enters LPI IDLE state only when the transmit path remains in an idle state (no activity) for the time indicated by the value in MAC LPI Entry Timer. In this mode also, the MAC transmitter exits the LPI IDLE state as soon as any of the functions becomes non-idle. However, the LPITXEN bit is not cleared but remains active so that reentry to LPI IDLE state is possible without any software intervention when the MAC becomes idle again.

When both LPIATE and LPITXA bits are cleared, you can directly control the entry and exit of the LPI IDLE state by programming the LPITXEN bit.

61.3.12.1.3 Receive path functions

In the receive path, when the PHY receives the signals from the link partner to enter into the LPI state, the PHY and MAC perform the following tasks:

1. The PHY asserts RX_ER.
2. The PHY sets RXD[7:0] to 0x01.
3. The PHY de-asserts RX_DV. The PHY maintains the same state of the RX_ER, RXD, and RX_DV signals for the entire duration during which it remains in the LPI state.
4. The MAC updates the RLPIEN bit of the MAC LPI control status register and immediately generates an interrupt.

When the PHY receives signals from the link partner to exit the LPI state, the PHY and MAC perform the following tasks:

1. The PHY de-asserts RX_ER and returns to a normal inter-packet state.
2. The MAC updates the RLPIEX bit of the MAC LPI control status register and generates an interrupt immediately. The sideband signal (synchronous to Rx clock) is also asserted.

61.3.12.1.4 LPI timers

The transmitter maintains the following two timers that are loaded with the respective values from the MAC_LPI_Timers_Control register:

- LPI LS Timer
  - The LPI LS TIMER counts, in milliseconds, the time expired since the link status is up. You can enable the monitoring of the LUD bit of the MAC_PHYIF_Control_Status register by setting the PLSEN bit of the MAC_LPI_Control_Status register. The link status is indicated to the MAC by the LNKSTS bit of the MAC_PHYIF_Control_Status register or the value programmed by the software in the PLS bit of the MAC_LPI_Control_Status register. If the link status is not available in the MAC_PHYIF_Control_Status, the software should get the PHY link status by reading the PHY register and accordingly update the PLS bit. This timer is cleared every time the link goes down. It starts to increment when the link is up again and continues to increment until the value of the timer becomes equal to the terminal count. Once the terminal count is reached, the timer remains at the same value as long as the link is up. The terminal count is the value programmed in bits 25:16 of the MAC_LPI_Timers_Control register. The GMII interface does not assert the LPI pattern unless the terminal count is reached. This ensures a minimum time for which no LPI pattern is asserted after a link is established with the remote station. This period is defined as 1 second in the IEEE 802.3-az-2010. The LPI LS TIMER is 10-bit wide. Therefore, the software can program up to 1,023 ms.
- LPI TW Timer
  - The LPI TW TIMER counts, in microseconds, the time expired since the de-assertion of LPI. The terminal count should be programmed in bits 15:0 of MAC register 53 (LPI timers control register). The terminal count of the timer is the value of resolved transmit TW that is the auto-negotiated time after which the MAC can resume the normal transmit operation. After exiting the LPI mode, the MAC resumes its normal operation after the TW timer reaches the terminal count. The MAC supports the LPI TW TIMER in units of a microsecond. The LPI TW TIMER is 16-bits wide. Therefore, the software can program up to 65,535 µs.

61.3.12.1.5 LPI interrupt

The MAC generates the LPI interrupt when the Tx or Rx side enters or exits the LPI state. The low-power interrupt is asserted when the LPI interrupt status is set. The LPI interrupt can be cleared by reading the MAC LPI control status register.

61.3.12.2 Power management and wake-up

The power management (PMT) block supports the reception of magic packets as wake up packets, as well as the reception of network (remote) wake-up packets as wake up packets.

61.3.12.2.1 Magic packet detection

The magic packet frame is based on a method that uses Advanced Micro Device's magic packet technology to power up the sleeping device on the network. The MAC receives a specific packet of information, called a magic packet, addressed to the node on the network.

Only magic packets that are addressed to the device or a broadcast address are checked to determine whether they meet the wake-up requirements. Magic packets that pass the address filtering (unicast or broadcast)are checked to determine whether they meet the remote wake-on-LAN data format of 6 bytes of all ones followed by a MAC address appearing 16 times.

The application enables magic packet wake-up by writing a 1 to bit 1 of the PMT control and status register. The PMT block constantly monitors each frame addressed to the node for a specific magic packet pattern. Each frame received is checked for a 0xFFFF FFFF FFFF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 0xFFFF FFFF FFFF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame but must be preceded by the synchronization stream (0xFFFF FFFF FFFF). The device also accepts a multicast frame, as long as the 16 duplications of the MAC address are detected.

If the MAC address of a node is 0x0011 2233 4455, then the MAC scans for the data sequence:

- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

- 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
- ...CRC

Magic packet detection is updated in the PMT control and status register for magic packet received. A PMT interrupt to the application triggers a read to the PMT CSR to determine whether a magic packet frame is received. When you enable the power-down mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx FIFO or the application. The MAC comes out of the power-down mode only when a magic packet is received and the corresponding detection is enabled.

61.3.12.2.2 Remote wake-up detection

The PMT block supports the reception of network (remote) wake-up packets as wake-up packets.

When the MAC is in sleep mode and the remote wake-up bit is enabled in PMT control and status register (0x00C0), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers by performing a sequential write to address (0x00C4). The application enables remote wake-up by writing a 1 to bit 2 of the PMT control and status register.

PMT supports 16 programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of filter command, and if filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received.

Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter byte mask determines which bytes of the frame must be examined. The 31st bit of the byte mask must be set to zero.

The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, MII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the PMT control and status register for every remote wake-up frame received. A PMT interrupt to the application triggers a read to the PMT control and status register to determine the reception of a wake-up frame.

61.3.12.3 System considerations during power-down

MAC neither gates nor stops clocks when power-down mode is enabled. Power saving by clock gating must be done outside the core by the application. The receive data path must be clocked with ENET_RX_CLK during power-down mode because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the application path clocks can be gated off during deep-sleep mode.

The PMT interrupt is asserted when a valid wake-up frame is received. This signal is generated in the receive clock domain.

The recommended power-down and wake-up sequences are as follows.

1. Disable the transmit DMA and wait for any previous frame transmissions to complete. These transmissions can be detected when transmit interrupt (see DMA channel status register bit TI) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC configuration register.
3. Wait until the receive DMA empties all the frames from the Rx FIFO. This can be done by reading the appropriate bits of Debug registers in the DMA and MTL CSR space.
4. Enable power-down mode by appropriately configuring the PMT registers.
5. Enable the MAC receiver and enter power-down mode.
6. Gate the application and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter sleep mode.
7. On receiving a valid wake-up frame, the MAC asserts the PMT interrupt signal and exits the power-down mode.
8. On receiving the interrupt, the system must enable the application and transmit clock inputs to the core.
9. Read the PMT status register to clear the interrupt, then enable the other modules in the system and resume normal operation.

61.3.13 Clocks

The table found here describes the clock sources for ENET_QOS. Please see the chip-specific clocking section for clock setting, configuration and gating information.

Table 61-65. ENET_QOS Clocks

Clock name	Description
aclk_i	Bus clock
clk_csr_i	CSR clock, same source as aclk_i
ptp_ref_i	PTP clock

61.3.14 Interrupts

This module supports interrupt coalescing which reduces the number of interrupts generated by IP and reduces the CPU load. Various interrupt event gets captured in status register. If interrupt enable bit of an interrupt is set then corresponding interrupt will be generated based on event status in status registers. INTM (interrupt mode) bit decides that interrupt signal is a level signal or pulse signal.

Interrupt output signals are shown below which are synchronous to control and Status register (CSR) clock.

- Sbd_intr_o – common interrupt
- Sbd_perch_tx_intr_o[max_dma_ch] – Interrupt per Tx channel
- Sbd_perch_rx_intr_o [max_dma_ch] – Interrupt per Rx channel

NOTE

max_dma_ch is equal to number of Tx/Rx queues which is 2.

The sbd_intr_o common interrupt is level signal. When sbd_intr_o interrupt is asserted, then corresponding interrupt event source can be found in DMA_Interrupt status register. DMA_Interrupt status register is read only register which contains the interrupt event source bits corresponding to each DMA channel (Tx and Rx queue pair), MAC transaction layer and MAC blocks. Software must then read the following registers corresponding to the bits that are set.

- MAC_Interrupt_Status register
- MTL_Interrupt_Status register
- DMA_CH(n)_Status register

sbd_intr_o interrupt gets de-asserted only when all the enabled interrupt events are cleared in their respective status registers and correspondingly all the bits in the DMA_Interrupt_Status register are cleared.

The DMA_CH[n]_Status register captures all the interrupt events of that Tx DMA and Rx DMA channel pair. The DMA_CH[n]_Interrupt_Enable register contains the corresponding enable bits for each of the interrupt event. There are two groups of interrupts in the DMA channel namely Normal and Abnormal interrupts. They are indicated by bits[15:14] of DMA_CH[n]_Status register respectively. The normal group is for events that happen during the normal transfer of packets (TI, RI, TBU) while the abnormal interrupt events are for error events

Interrupts are not queued. If the same interrupt event occurs again before the driver responds to the previous one, no additional interrupts are generated.

The common sbd_intr_o output signal is asserted for the transfer complete interrupts only when the corresponding interrupts are enabled in DMA_CH[n]_Interrupt_Enable register.

This module also supports the following per Channel Transfer Complete interrupt signals.

- sbd_perch_tx_intr_o[max_dma_ch] (Transmit Per Channel Interrupts)
- sbd_perch_rx_intr_o[max_dma_ch] (Receive Per Channel Interrupts)

The behavior of the RI/TI/ sbd_perch_tx_intr_o[]/sbd_perch_rx_intr_o[] changes depending on the settings of INTM field bit[17:16] in DMA_Mode. Following table explains the Transfer Complete Interrupt behavior.

Table 61-66. Transfer Complete Interrupt Behavior

Interrupt Mode	Behavior of the sbd_perch_tx_intr_o[] and sbd_perch_rx_intr_o[]	Behavior of TI/RI and sbd_intr_o
INTM=1	These signals reflect the value of corresponding TI/RI bits in DMA_CH[n]_Status register when the corresponding interrupt enable is set. Hence they are level signals and are cleared by the application by writing 1'b1 to the RI/TI status bits. This signal will not be asserted when the corresponding interrupt enable bit is not set.	The TI/RI is set as explained above. However, the sbd_intr_o signal and NIS status bit are not asserted for any RI/TI events.
INTM=2	In this mode, RI/TI interrupts are queued. These signals reflect the value of corresponding TI/RI bits inDMA_CH[n]_Status register when the corresponding interrupt enable is set. They are level signals and cleared by software by writing 1'b1 to the RI/TI status bits. However, it is set again if another TI/RI event(s) is detected before the TI/RI bits are cleared for the previous event.	The RI/TI status bits are set whenever the Transfer Complete event is detected and gets reset whenever software driver clears those bits by writing 1. However, if another Transfer Complete event is detected before it is cleared (serviced) by the software, then the module automatically set these status bits again. However, the sbd_intr_o signal is not generated based on TI/RI

61.3.14.1 PMT interrupts

The PMT interrupt signal is asserted when a valid remote wake-up packet is received. When software resets the PWRDWN bit in the remote wake-Up packet detection register, the MAC comes out of the power-down mode, but this event does not generate the PMT interrupt.

When you enable the power-down mode in the PMT block, the MAC drops all received packets and does not forward any packet to the MTL Rx FIFO or the application. The MAC comes out of the power-down mode only when a remote wake-up packet is received and the corresponding detection is enabled. The PMT block is available in the receive path of MAC.

61.4 External signals

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

Table 61-67. External signals

Port name	Interface	I/O	Description
1588_EVENT_AUX_IN	MII/RMII	I	
1588_EVENT_IN	MII/RMII	I	
1588_EVENT_OUT	MII/RMII	O	
TX_CLK	MII, RGMII	I (for MII) O for (RGMII)	MII: The external PHY provides this transmission clock. This is 25 MHz in 100 Mbps mode, 2.5 MHz in 10 Mbps mode. All transmission signals generated by the MAC are synchronous to this clock. This clock is required for all PHY interfaces. RGMII: The transmit clock signal is always provided by the MAC on the TX_CLK line. Source-synchronous clocking is used: the clock signal that is output (by either the PHY or the MAC) is synchronous with the data signals.
RX_CLK	MII, RGMII	I	MII: The external PHY provides this receive clock. This clock is 25 MHz in 100 Mbps mode, 2.5 MHz in 10 Mbps mode. All MII receive signals received by the MAC are synchronous to this clock. This clock input is required for all PHY interfaces. RGMII: Received clock signal (recovered from incoming received data)

Table continues on the next page...

Table 61-67. External signals (continued)

Port name	Interface	I/O	Description
REF_CLK1	RMII	I/O	This 50Mhz clock used by the RMII interface. When RMII mode is selected, RX_CLK (25Mhz or 2.5Mhz) should be derived from this clock.
TX_EN	MII, RMII, RGMII	O	The MAC drives this signal. This signal has multiple functions depending on the selected PHY interface. • MII: When high, indicates that valid data is being transmitted on the phy_txd_o bus. Synchronous to TX_CLK • RMII: When high, indicates that valid data is being transmitted on the phy_txd_o bus. Synchronous to REF_CLK. • RGMII: Contains TX_EN on the rising edge of TX_CLK, and a logical derivative of TX_EN and TX_ER (TX_EN XOR TX_ER) on the falling edge of TX_CLK.
TX_DATA	MII, RMII, RGMII	O	This is a group of transmit data signals driven by the MAC. It has multiple functions depending on the selected PHY interface. • MII: Bits[3:0] provide the MII transmit data nibble. The data is valid only when the TX_EN signal is high. Synchronous to TX_CLK • RMII: Bits[1:0] provide the RMII transmit data. The data is valid only when the TX_EN signal is high. Synchronous to REF_CLK. • RGMII: Bits[3:0] provide serial output Ethernet data
TX_ER	MII	I	The PHY drives this signal. In MII, indicates an error or carrier extension in the

Table continues on the next page...

Table 61-67. External signals (continued)

Port name	Interface	I/O	Description
			transmitted packet the TX_DATA[3:0] bus. Synchronous to TX_CLK.
CRS	MII	I	This signal is valid only in the MII mode. The PHY drives this signal high when the Transmit or Receive medium is not idle. The PHY drives this signal low when both Transmit and Receive mediums are idle. This signal is not synchronous to any clock.
COL	MII	I	This signal is valid only in the MII mode. The PHY drives this signal high when a collision is detected on the medium. This signal is not synchronous to any clock.
RX_DATA	MII, RMII, RGMII	I	This is a group of data signals received from the PHY. It has multiple functions depending on the selected PHY interface. • MII: Bits [3:0] provide the MII receive data nibble. The data is valid only when the RX_EN signal is high. Synchronous to RX_CLK. • RMII: Bits [1:0] provide the RMII receive data. The data is valid only when the RX_EN signal is high. Synchronous to REF_CLK. • RGMII: Bits [3:0] contains the Ethernet input data transferred from the PHY to the media-access controller.
MDC	MII, RMII, RGMII	O	Management Data Clock. The MAC provides timing reference for the MDIO or MII through this periodic clock. This clock is generated from the application clock through a clock divider controlled by MAC_MDIO_ADDRESS[CR].

Table continues on the next page...

Table 61-67. External signals (continued)

Port name	Interface	I/O	Description
MDIO	MII, RMII, RGMII	I/O	Transfers control information between the external PHY and the MAC. Data is synchronous to MDC. This signal is an input after reset.
RX_ER	MII	I	The PHY drives this signal. In MII, indicates an error or carrier extension in the received packet the RX_DATA[3:0] bus. Synchronous to RX_CLK.
RX_EN	RGMII	I	The PHY drives this signal. It has multiple functions depending on the selected PHY interface. • MII: Indicates that the data on the MII_RXD bus is valid. It remains high continuously from the first recovered byte or nibble of the packet through the final recovered byte or nibble of the packet. Synchronous to RX_CLK. • RMII: Contains the crs and data valid information of the Receive interface. Synchronous to REF_CLK. In RGMII mode, contains receive data valid on the rising edge of RX_CLK, and a logical derivative of receive data valid and RX_ER (receive data valid XOR RX_ER) on the falling edge of RX_CLK.

61.5 Initialization

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

This section provides the instructions for initializing the DMA or MAC registers in the proper sequence.

61.5.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll bit 0 of the DMA_MODE, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SYSBUS_MODE register:
   a. AAL
   b. Fixed burst or undefined burst
   c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.
   d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [7:1])
4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see Descriptors
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= 1) and DMA_CH#_RX_CONTROL2 (for i = 0; i <= 1)). The ring length programmed must be at least 4.

   NOTE

   The descriptor address from the start to the end of the ring must not cross the 4GB boundary.

6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_RXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1))

NOTE

For 40-bit or 48-bit addressing mode, program the higher address List registers (DMA_CH[n]_TXDESC_LIST_HADDRESS, DMA_CH[n]_RXDESC_LIST_HADDRESS).

The tailpointer registers must be advanced to the location immediately after the descriptors that are set, for the DMA to know that additional descriptors are available.

7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:
   - DMA_CH(#i)_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_TX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_RX_Control (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)
8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) and ST (bit 0) of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

61.5.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.

Complete the following steps to initialize the MTL registers:

1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.

2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
   a. Transmit Store And Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
   b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
   c. Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
   a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
   b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
   c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)
   d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

61.5.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.

The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.

1. Provide the MAC address registers: MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. If more than one MAC address is enabled in your configuration, program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_PACKET_FILTER register:
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_TX_FLOW_CTRL register:
   a. Pause time and other Pause frame control bits
   b. Transmit Flow control bits

c. Flow Control Busy
4. Program the MAC_INTERRUPT_ENABLE register, as required, and if applicable, for your configuration.
5. Program the appropriate fields in the MAC_CONFIGURATION register. For ex: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in MAC_CONFIGURATION register to start the MAC transmitter and receiver.

61.6 Application Information

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

61.6.1 Performing Normal Receive and Transmit Operation

During normal operation of the module, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TXDESC_TAIL_POINTER and DMA_CH[n]_RXDESC_TAIL_POINTER).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_CURRENT_APP_TXDESC and DMA_CH[n]_CURRENT_APP_RXDESC).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n]_CURRENT_APP_TXBUFFER and DMA_CH[n]_CURRENT_APP_RXBUFFER).

61.6.2 Stopping and Starting Transmission

Complete the following steps to pause the transmission.

NOTE

The steps are provided for Channel 0.

1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) Register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TXQ0_DEBUG Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Config-uration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TXQ0_DEBUG Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TXQ0_DEBUG Register and RXQSTS is 0 in MTL_RXQ0_DEBUG Register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

NOTE

- Do not change the configuration (such as duplex mode, speed, port, or loop back) when the MAC is actively transmitting or receiving. These parameters are changed by software only when the MAC transmitter and receiver are not active.
- Similarly, do not change the DMA-related configuration when Transmit and Receive DMA are active.

61.6.3 Programming Guidelines for Switching to New Descriptor List in RxDMA

Switching to a new descriptor list is different in the Rx DMA compared to the Tx DMA. Switching to a new descriptor list is permitted when the RxDMA is in SUSPEND state, as clarified by the following points:

- Generally, RxDMA prepares the descriptors in advance.
- If the RxDMA goes to SUSPEND due to descriptors not being available, a major failure occurs (software is not able to free the filled-up descriptors/buffers). If this issue is not rectified immediately, frames will be lost because of an RxFIFO Overflow. Therefore, the software is allowed to create a new descriptor list and program the RxDMA to start using it immediately, without going into STOP state.

61.6.4 Programming Guidelines for Multi-Channel Multi-Queuing

61.6.4.1 Transmit

1. Program the Transmit queue size in the TQS field of MTL_TXQ[n]_OPERATION_MODE register. Based on the value programmed in the TQS field, the size of the queue is determined.

   In the Transmit operation, the number of channels is equal to the number of the queues. Due to this reason, the Channel-to-Queue mapping is fixed.

2. For a queue to be used, the queue must be enabled in TXQEN in the corresponding MTL_TXQ[n]_OPERATION_MODE Register. In DMA configurations, the ST bit of DMA_CH[n]_TX_CONTROL Register and corresponding TXQEN in MTL_TXQ[n]_OPERATION_MODE Register needs to be enabled.
3. The scheduling method needs to be programmed in SCHALG of MTL_OPERATION_MODE register.
4. Program the MTL_TXQ[n]_Quantum_Weight for DCB queue as per the selected algorithm. In case of CBS algorithm in AVB queues, the MTL_TXQ[n]_ETS_CONTROL, MTL_TXQ[n]_SENDSLOPECREDIT, MTL_TXQ[n]_HICREDIT and MTL_TXQ[n]_LOCREDIT registers also need to be programmed as required.
5. If DCB is enabled and PFC function is required, program MAC_TXQ_PRTY_MAP0 Register to assign a fixed priority to the queue. This priority assigned is used for determining if the corresponding queue should stop transmitting packet based on the received PFC packet.

61.6.4.2 Receive

1. Program the Receive queue size in the RQS field of MTL_RXQ[n]_OPERATION_MODE Register. Based on the value programmed in RQS field, the size of the queue is determined.

2. Enable the Receive Queues 0 to 4 in the fields RXQ0EN to RXQ4EN in MAC_RXQ_CTRL0 Register for AV or DCB. In DMA configurations, SR bit of statically or dynamically mapped DMA_CH[n]_RX_CONTROL Register and corresponding RXQ[n]_EN in MAC_RXQ_CTRL0 Register needs to be enabled.
3. The MAC routes the Rx packets to the Rx Queues based on following packet types:
   a. AV PTP Packets: Based on the programming of AVPTPQ in MAC_RXQ_CTRL1 Register.
   b. AV Untagged Control packets: Based on the programming of AVCPQ in MAC_RXQ_CTRL1 Register.
   c. Data Center Bridging (DCB) related Link Layer Discovery Protocol (LLDP) packets. Program DCBCPQ in MAC_RXQ_CTRL1 Register to indicate to MAC which queue should get the DCB packets.
   d. VLAN Tag Priority field in VLAN Tagged packets: Program PSRQ4-0 of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 Register for the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 4.
   e. The AV tagged control and data packets are also routed based on PSRQ field of the MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers.

NOTE

The priorities set in PSRQ4-0 should be unique.

4. If multiple RX DMA channels are enabled, the following programming should be done for proper arbitration and mapping:
   a. Program the RAA field of MTL_OPERATION_MODE register to select the arbitration algorithm to decide which RxQ is read out from the RxFIFO memory.
   b. Program the MTL_RXQ[n]_CONTROL to decide the weights and the packet arbitration for each RxQ.
   c. If static mapping is programmed in MTL_RXQ_DMA_MAP[n] register (RXQ[n]DADMACH is reset to 0), bits RXQx2DMA and others need to be programmed to select the channel for which each queue is mapped.
   d. Set RXQ[n]DADMACH bit in MTL_RXQ_DMA_MAP0 Register to select dynamic mapping of packets in each RxQueue.
   e. In dynamic channel mapping, the routing of a packet to a specific RxDMA channel is decided by the value of DCS field in the lowest MAC Address Register.

61.6.4.2.1 Programming Guidelines for Recovering from DMA Channel Failure

When the DMA channel issues a bus error, follow these steps to recover from the failure.

Recovering from the Receive DMA Channel Failure

1. Set the RPF bit to 1. This flushes all the packets one after the other.

   This step is optional. However, setting this bit prevents HOL (head-of-line) blocking in the Rx queues when packets sent to the RXDMA are stopped due to bus error.

2. Reprogram the specific Registers of the DMA channel.
3. Start the DMA channel.

Recovering from the Transmit DMA Channel Failure

1. Stop the specific DMA channel, even if it is in active state.
2. Flush the corresponding MTL queue.
3. Reprogram the specific Registers of the DMA channel.
4. Start the DMA channel.

NOTE

Due to the known limitations in the design, reprogramming the DMA channel registers might not always be successful in recovering from a Bus Error. If the module is not fully functional after reprogramming the DMA, as a workaround, issue a Soft Reset to recover from the Bus Error.

61.6.5 Programming Guidelines for IEEE 1588 Timestamping

61.6.5.1 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the MAC_TIMESTAMP_CONTROL Register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during the module initialization:

1. Mask the Timestamp Trigger interrupt by clearing the bit 16 of MAC_INTERRUPT_ENABLE Register.
2. Set Bit 0 of MAC_TIMESTAMP_CONTROL Register to enable timestamping.
3. Program MAC_SUB_SECOND_INCREMENT Register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_TIMESTAMP_CONTROL Register.
5. Poll the MAC_TIMESTAMP_CONTROL Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_TIMESTAMP_CONTROL Register to select the Fine Update method (if required).

7. Program MAC_SYSTEM_TIME_SECONDS_UPDATE Register and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE Register with the appropriate time value.
8. Set Bit 2 in MAC_TIMESTAMP_CONTROL Register.
9. The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers. If one-step timestamping is enabled
   a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
   b. Program registers MAC_TIMESTAMP_INGRESS_ASYM_CORR and MAC_TIMESTAMP_EGRESS_ASYM_CORR to update the correction field in PDelay_Req PTP messages.
10. Enable the MAC receiver and transmitter for proper timestamping.

NOTE

If timestamp operation is disabled by clearing Bit 0 of MAC_TIMESTAMP_CONTROL Register, repeat all these steps to restart the timestamp operation.

61.6.5.2 System Time Correction

61.6.5.2.1 Coarse Correction Method

To synchronize or update the system time in one process (coarse correction method):

1. Set the offset (positive or negative) in the Timestamp Update registers(MAC_SYSTEM_TIME_SECONDS_UPDATE and MAC_SYSTEM_TIME_NANOSECONDS_UPDATE).
2. Set Bit 3 (TSUPDT) of MAC_TIMESTAMP_CONTROL REGISTER.

The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

61.6.5.2.2 Fine Correction Method

To synchronize or update the system time to reduce system-time jitter (fine correction method):

1. With the help of the algorithm, calculate the rate by which you want to make the system time increments slower or faster.
2. Update the MAC_TIMESTAMP_ADDEND with the new value and set Bit 5 of the MAC_TIMESTAMP_CONTROL Register.

3. Wait for the time for which you want the new value of the Addend register to be active. You can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.
4. Program the required target time in MAC_PPS[n]_TARGET_TIME_SECONDS Register and MAC_PPS[n]_TARGET_TIME_NANOSECONDS Register.
5. Enable the Timestamp interrupt in bit 12 of MAC_INTERRUPT_ENABLE register.
6. Set bit 4 in Register MAC_Timestamp_Control.
7. When this trigger causes an interrupt, read MAC_INTERRUPT_STATUS Register.
8. Reprogram MAC_TIMESTAMP_ADDEND Register with the old value and set bit 5 again.

61.6.6 Programming Guidelines for AV Feature

After you enable the AV feature in the module controller, there are several programming tasks to be followed:

- Initializing the DMA for QOS-AHB configurations only
- Enabling slot number checking
- Enabling average bits per slot reporting
- Disabling flow control for AV-enabled queues (transmit and receive flow control)

61.6.6.1 Initializing the DMA in Audio Video Feature

The first step in programming the AV feature in a QOS-AHB configuration is to initialize the DMA.

Use this initialization sequence for QOS-AHB/QOS-AXI configurations with AV feature.

1. Provide a software reset to reset all QOS internal registers and logic (bit 0 in DMA_MODE register).
2. Wait for the completion of the reset process. Poll bit 0 of the DMA_MODE register), which is cleared only after the reset operation is completed.
3. Program the fields to initialize the DMA register by setting the values in DMA_MODE register.
4. Create a proper descriptor list for transmit and receive. In addition, ensure that the DMA owns the Transmit and Receive descriptors. When OSF mode is used, at least two TX descriptors are required.

   For more information about descriptors, see Descriptors.

5. Make sure that your software creates three or more different transmit or receive descriptors in the list before reusing any of the descriptors.
6. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TXDESC_RING_LENGTH (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) and DMA_CH#_RX_CONTROL2 (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.
7. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1), DMA_CH(#i)_TXDESC_LIST_ADDRESS (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1)). In addition, you must program the Transmit and Receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_TX_CH-1) and DMA_CH(#i)_RXDESC_TAIL_POINTER (for i = 0; i <= DWC_EQOS_NUM_D-MA_RX_CH-1)).
8. Program the following fields to initialize the mode of operation in the MTL_TXQ0_OPERATION_MODE register:
   a. Transmit Store And Forward (TSF)
   b. Transmit Threshold Control (TTC)
   c. Transmit Queue Enable (TXQEN) to value 2‘b10 to enable Transmit Queue0
   d. Transmit Queue Size (TQS)
9. Enable the interrupts by programming the DMA_CH(#i)_INTERRUPT_ENABLE (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register.
10. Repeat steps 4 through 9 for all additional channels of AV feature.
11. Program the CBS control register, idleSlope, sendSlope, hiCredit, and loCredit registers of the AV Queues.
12. Start the Receive and Transmit DMA by setting bit 0 of the DMA_CH(#i)_TX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1) register and bit 0 of the DMA_CH(#i)_RX_CONTROL (for i = 0; i <= DWC_EQOS_NUM_DMA_RX_CH-1) register.

61.6.6.2 Enabling Slot Number Checking

You can enable slot number checking to specify the intervals at which the DMA channels mapped to AV Queues fetch the frames from the AHB/AXI system bus.

These steps should be completed after step 11 and before step 12 of "Initializing DMA".

You can use the slot number check feature to specify the intervals at which the DMA Channels mapped to AV Queues fetch the frames from the AHB/AXI system bus. This feature is useful for a uniform and periodic transfer of the AV traffic from the host memory. The feature is available only when you enable timestamping and program the MAC_SUB_SECOND_INCREMENT register. To enable the slot number checking:

1. Enable timestamping by following the steps described in "Initialization Guidelines for System Time Generation".
2. Make sure that the SLOTNUM field (bits 22:19) of TDES3 Normal Descriptor (Read Format) contains a valid slot number. You can read the current reference slot number from the DMA_CH(#i)_Slot_Function_Control_Status (for i = 0; i <= DWC_EQOS_NUM_DMA_TX_CH-1).
3. Set Bit 0 (ESC) of the Slot Function Control and Status register of a channel to enable the slot number checking.

61.6.6.3 Enabling Average Bits Per Slot Reporting

You can enable the reporting of the average bits that are transmitted in a slot.

The CBS Status register of the additional AV channels provides information about the average bits that are transmitted in a slot. The software can asynchronously read this register to retrieve information about the average bits transmitted per slot. To enable average bits per slot reporting:

1. Enable timestamping by following the steps described in the Initialization Guidelines for System Time Generation.
2. Program Bits [6:4], SLC, of the MTL_TXQ[n]_ETS_CONTROL register of a channel with the number of slots over which the average transmitted bits per slot need to be computed.
3. Enable Bit 9(ABPSSIE) of the MTL_Q[n]_INTERRUPT_CONTROL_STATUS register of a channel to generate the average bits per slot interrupt.

NOTE

- The frequency of this interrupt depends on the value programmed in step 2. For example, when you program value 0 in the SLC field, the interrupt is generated at every 125 microsecond.
- When not required, you can disable this interrupt to stop the interrupt flooding.

4. Read Bits [16:0], ABS, from the MTL_TXQ[n]_ETS_STATUS register of a channel on each interrupt.

NOTE

The software can read the ABS bits in polling mode even if the ABPSIE bit is not enabled. When high, bit 1 (ABPSIS) of the MTL_TXQ[n]_ETS_STATUS register indicates that a new value is updated in the ABS field.

61.6.6.4 Disabling Flow Control for AV Enabled Queues

61.6.6.4.1 Transmit Flow

Program the EHFC (Enable Hardware Flow Control) bit of corresponding Rx Queue's MTL_RXQ[n]_OPERATION_MODE register to 0.

61.6.6.4.2 Receive Flow Control

Program the PSTQ[n] field corresponding to AV enabled Tx Queue in MAC_TXQ_PRTY_MAP0/1 register to 0.

61.6.6.5 Programming Guidelines for Flexible Pulse-Per-Second Output

After you enable the Flexible Pulse-Per-Second Output feature in the module controller, you can perform the following tasks:

- Generating Single Pulse on PPS
- Generating Next Pulse on PPS
- Generating a Pulse Train on PPS
- Generating an Interrupt without Affecting the PPS

61.6.6.5.1 Generating Single Pulse on PPS

To generate single pulse on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register. This instructs the MAC to use the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) for start time of PPS signal output.
2. Program the start time value in the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS).

3. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= 3) Register.
4. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to 0001. This instructs the MAC to generate single pulse on the PPS signal output at the time programmed in the Target Time registers.

61.6.6.5.2 Generating Next Pulse on PPS

When the PPSCMD is executed (PPSCMD bits = 0), you can cancel the pulse generation by giving the Cancel Start Command (PPSCMD=0011) before the programmed start time elapses. You can also program the behavior of the next pulse in advance.

To program the next pulse:

1. Program the start time for the next pulse in the Target Time registers. This time should be more than the time at which the falling edge occurs for the previous pulse.
2. Program the width of the next PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1).
3. Program Bits [3:0], PPSCMD, of MAC_PPS_CONTROL to generate a single pulse after the time at which the previous pulse is de-asserted. This instructs the MAC to generate single pulse on the PPS signal output, at the time programmed in Target Time registers.

   If you give this command before the previous pulse becomes low, then the new command overwrites the previous command and the QOS may generate only 1 extended pulse.

61.6.6.5.3 Generating a Pulse Train on PPS

To generate a pulse train on PPS:

1. Program 11 or 10 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_Control Register. This instructs the MAC to use the Target Time registers for start time of the PPS signal output.
2. Program the start time value in the Target Time registers.
3. Program the interval value between the train of pulses on the PPS signal output in MAC_PPS(#i)_INTERVAL (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
4. Program the width of the PPS signal output in MAC_PPS(#i)_WIDTH (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register.
5. Program Bits[3:0], PPSCMD, of MAC_PPS_CONTROL Register to 0010. This instructs the MAC to generate train of pulses on the PPS signal output with start time programmed in Target Time registers.

By default, the PPS pulse train is free-running unless stopped by ‘STOP Pulse train at time’ or ‘STOP Pulse Train immediately’ commands.

6. Program the stop value in the Target Time registers. Ensure that Bit 31 (TSTRBUSY) of MAC_PPS(#i)_TARGET_TIME_NANOSECONDS (for i = 0; i <= DWC_EQOS_PPS_OUT_NUM-1) Register is reset before programming the Target Time registers again.
7. Program the PPSCMD field (bit 3:0) of MAC_PPS_CONTROL to 0100. This stops the train of pulses on PPS signal output after the programmed stop time specified in Step 6 elapses.

   You can stop the pulse train at any time by programming 0101 in the PPSCMD field. Similarly, you can cancel the Stop Pulse train command (given in Step 7) by programming 0110 in the PPSCMD field before the time (programmed in Step 6) elapses. You can cancel the pulse train generation by programming 0011 in the PPSCMD field before the programmed start time (in Step 2) elapses.

61.6.6.5.4 Generating an Interrupt without Affecting the PPS

You can use bits[6:5], TRGTMODSEL, of the MAC_PPS_CONTROL Register to program the Target Time registers to do any one of the following:

- Generate only interrupts.
- Generate interrupts and the PPS start and stop time.
- Generate only PPS start and stop time.

To program the Target Time registers to generate only interrupt events:

1. Program 00 (for interrupt) in Bits [6:5], TRGTMODSEL, of MAC_PPS_CONTROL Register. This instructs the MAC to use the Target Time registers for target time interrupt.
2. Program a target time value in the Target Time registers. This instructs the MAC to generate an interrupt when the target time elapses.

   If Bits [6:5], TRGTMODSEL, are changed (for example, to control the PPS), then the interrupt generation is overwritten with the new mode and new programmed Target Time register value.

61.6.6.6 Programming Guidelines for VLAN Filtering on Receive

To program VLAN filtering on receive:

1. Program MAC_VLAN_TAG register for the following bit to select the filtering method:
   a. ETV: Enable 12-Bit VLAN Tag Comparison or 16-bit VLAN Tag comparison.
   b. VTHM: VLAN Tag Hash Table Match Enable.
   c. ERIVLT: Enable inner VLAN Tag or outer VLAN Tag (to enable the inner or outer VLAN Tag filtering, Double VLAN Processing should enabled by setting EDVLP)
   d. ERSVLM: Enable Receive S-VLAN Match or C-VLAN match (for S-VLAN processing to be enabled, set ESVL)
   e. DOVLTC: Ignores VLAN Type for Tag Match
   f. VTIM: to enable VLAN Tag Inverse Match instead of the normal VLAN Tag matching
2. Program VL of MAC_VLAN_Tag Register for the 12-bit or 16-bit VLAN tag.
3. If Hash filtering of VLAN tag is enabled, program MAC_VLAN_HASH_TABLE Register. When ETV bit is reset, upper 4 bits of the calculated CRC-32 of VLAN Tag are inverted and used to index the content of MAC_VLAN_Hash_Table register. When ETV bit is set, upper 4 bits of calculated CRC-32 of VLAN Tag are used to index the content of MAC_VLAN_HASH_TABLE register. For example, when ETV bit is set, a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table. When ETV bit is reset a hash value of 4'b1000 selects Bit 7 of the VLAN Hash table.

61.6.6.7 Programming Guidelines for Extended VLAN Filtering and Routing on Receive

For the indirect access of the per VLAN Tag registers, follow these steps.

- Write
  a. Write the required data into the MAC VLAN Tag Data register.
  b. Program the VLAN Tag Control Register's OFS field with the required Filter Register's offset and command type to the CT field. For a write command, set this bit to 0.
  c. Write 1 to the OB field and wait till the OB bit is reset to do the next write. This will guarantee that the appropriate VLAN Tag Filter Register has been programmed.
- Read
  a. Program the VLAN Tag Control Register's OFS Field with the required register's offset and command type to the CT field. For a read command, set this bit to 1.

b. Write 1 to the OB field and wait till the OB bit is reset. The appropriate VLAN Tag Filter register's value is available in the MAC VLAN Tag Data Register.

61.6.6.8 Programming Sequence for Queue/Channel Based VLAN Inclusion Register

To program the queue/channel-based VLAN inclusion register:

NOTE

Indirect VLAN include registers are not accessed while setting the CBTI bit.

1. Set the CBTI bit of MAC_VLAN_INCL register to 1, to enable queue/channel based VLAN Tag insertion on all transmitted packets. This bit must be set before any indirect access to the queue/channel specific MAC_VLAN_INCL(#i) register.
2. Program the VLAN Tag and VLAN Type to be inserted in packets from a particular queue/channel in VLT and CSVL fields of MAC_VLAN_INCL register; corresponding offset address in ADDR field (0 for queue/channel 0, 1 for queue/channel 1, and so on) must be set. Set the RDWR bit to 0 to indicate write access. The write to byte 0 (byte 3 in Big Endian mode) of MAC_VLAN_INCL register initiates access to indirect access MAC_VLAN_INCL(#i) register.
3. The BUSY bit of MAC_VLAN_INCL register is set by the module to indicate the progress of access to indirect access MAC_VLAN_INCL(#i) register. On completion of the access, the BUSY bit is cleared. The application must not attempt subsequent access to MAC_VLAN_INCL(#i) register when the BUSY bit is 1.
4. Repeat step 2 and step 3 to program VLAN Tag and VLAN Type to be inserted in packets from the remaining queues/channels. The application must ensure that the required VLAN Tag and VLAN Type for all the queues/channels are programmed; otherwise unintended VLAN Tag and VLAN Type might be inserted.

61.6.7 Programming Guidelines for EST

Program the gate control values and time intervals in the Software Owned Gate Control List (SWOL) along with the other EST related registers that are described in ENET_QOS register descriptions to appropriate values. The following sub-sections provide step by step details for programming the GCL and the other EST related registers.

61.6.7.1 Programming the GCL and GCL Linked Registers

To program the Gate Control List (GCL) and the four other registers implemented per GCL:

1. The GCL and the four other GCL-linked registers should be accessed via indirect addressing using the MTL_EST_GCL_CONTROL and MTL_EST_GCL_DATA registers. The SWOL bit of the MTL_EST_STATUS register indicates if GCL0 or GCL1 is owned by software.
2. To program the GCL, write the 32-bit write data to MTL_EST_GCL_DATA register. Then program TL_EST_GCL_CONTROL register to write the write address and other control information.
3. In the MTL_EST_GCL_DATA register, Write Data consists of up to 8 bits (5 configurable, 3 reserved) of Gate Controls and up to 24 bits (configurable) of Time Interval. Gate Close is indicated by programming a '0' and Gate Open is indicated by programming a '1'. For a 4-TC and 20-bit Time Interval Configuration, the data width is 24-bits and the remaining 8-bits are reserved/read-only. The data should be written in the following format.

   {8'h0, TC3, TC2, TC1, TC0, 20-bit Time Interval} where TCx = 0 or 1.

4. MTL_EST_GCL_CONTROL register: Program the SRWO bit to 1 (to start a Write Op) and program the address and R/W fields appropriately.
5. Poll and check for the SRWO bit to be cleared by hardware to indicate the completion of the previous operation before initiating a new R/W operation via the same indirect addressing mode.
6. Repeat steps 3, 4, 5 until the programming of the GCL is completed.
7. Using the same indirect addressing method described above, program the BTR, CTR, TER and LLR registers. Set the GCRR bit in the MTL_EST_GCL_CONTROL register appropriately. The GCRR bit interprets the address field as belonging to these registers (instead of the GCL).

After programming of the GCL and the related registers, program the MTL_EST_CONTROL register to allow hardware to own and process the GCL.When the List Length (as indicated in LLR) is 1, the associated Time Interval should be smaller than the value of the Cycle Time Register. Otherwise, an error is reported (as detailed in the Error handling section) as a single set of gate controls add no value in the TSN context.

NOTE

The time unit in all the GCL related registers is seconds and nanoseconds. In cases where internally generated PTP System Time is used, the nanoseconds field must be programmed to use

the Digital Rollover mode. (TSCTRLSSR field of MAC_TIMESTAMP_CONTROL must be set to value '1')

61.6.7.2 Programming the EST Registers

After completing the steps mentioned in "Programming the GCL and GCL Linked Registers" , program the MTL_EST_CONTROL register.

1. Set the Current Time Offset Value and Time Internal Left Shift fields of the MTL_EST_CONTROL register appropriately. Also, set the Enable EST (EEST) and Switch to SWOL (SWOL) bits.

NOTE

The CTOV recommendations for GMII for SPRAM configurations are:

- 96 * Tx clock period, for 32 and 64 bit data width configurations.
- 128 * Tx clock period for 128 bit data width configurations.

The CTOV recommendations for GMII for non-SPRAM configurations are:

- 30 * Tx clock period, when SA/VLAN insertion is enabled.
- 22 * Tx clock period, when SA/VLAN insertion is not enabled.

2. This will enable the hardware to own and process the new GCL and make a switch to the new GCL at the BTR value. The hardware provides an interrupt (if enabled) when the switch to the new list happens.
3. Appropriate action has to be taken by software to address any other interrupts (explained in the Inter-rupts section) received during the hardware execution of the GCL.

Set the SSWL (Switch to Software Owned List) bit in the MTL_EST_CONTROL register to handoff to hardware. Software is not allowed to write to the GCL and GCL linked registers when the SSWL bit is set, because the new GCL might be in use by the hardware.

The SSWL is reset/cleared by the hardware when it successfully switches to the new list. The hardware also flips the SWOL bit to indicate the new GCL that the software owns.

To install a new GCL, program the GCL it owns (indicated by SWOL bit) as described in "Programming the GCL and GCL Linked Registers" and then program the MTL_EST_CONTROL register as described above. Ensure that the new BTR is set to an appropriate value to avoid BTRError that might need software intervention in some cases.

To avoid transmission overruns, the packet length (frame size) information should be available at all times. Therefore, in the DMA configurations, program the packet length in the first descriptor of every Tx frame. Similarly, in the MTL configuration, provide the packet length in the control word.

NOTE

To avoid transmission overruns, the Packet Length provided in the Transmit Descriptor should account for the SA and VLAN insertion, if applicable, that is, Packet Length should represent the actual packet length on the Ethernet line.

61.6.8 Programming the Launch Time in Time-Based Scheduling

The Launch Time is programmed in the Enhanced Normal Transmit Descriptors in DMA configurations and is driven as a control word in MTL configurations as follows:

The OSTC and Launch Time features are mutually exclusive and should not be used together; in case of a conflict (if OSTC = LTV = 1 in MTL configuration), priority is given to OSTC and the Launch Time is ignored.

In DMA configuration, if a context descriptor is received with a valid OSTC values immediately before receiving a first normal descriptor with LTV bit set, then the LTV is ignored.

61.6.9 Programming Guidelines for Media Clock Generation and Recovery

61.6.9.1 Programming Guidelines for Media Clock Generation

1. To set the PPS instance in Media Clock generation mode, program the appropriate Presentation Time Control (supported generation modes "1001-1011") to

PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_CONTROL register.
2. Based on the selected PPS instance, application must drive the appropriate trigger signal to the corresponding mcg_pst_trig_i[#i].
3. Based on the programmed mode, DUT captures the timestamp and programs it in the MAC_PPS(#i)_TARGET_TIME_SECONDS register. Then, mcgr_dma_req_o[#i] is set.
4. For every request generated by the DUT, the module reads the corresponding MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the read request by asserting the corresponding mcgr_dma_ack_i[#i].

61.6.9.2 Programming Guidelines for Media Clock Recovery

1. Enable the Current Presentation Time (CPT) Counter by setting the PTGE field of MAC_TIMESTAMP_CONTROL register. In addition to programming the initialization values for the System Time, update the MAC_PRESN_TIME_UPDT register with the equivalent Presentation Time init value. Only then the TSINIT field of MAC_TIMESTAMP_CONTROL register is set.

2. The Increment values used for the System time is also used for the Current Presentation time. This is because, the increment value is in Sub-seconds and Sub-nanoseconds.

3. When an update is needed, along with programming the update values for the System Time, the MAC_PRESN_TIME_UPDT register should be updated with the equivalent Presentation Time update value (32-bit ns). Finally, TSUPDT field of MAC_TIMESTAMP_CONTROL register must be set for the update to happen.

4. Program the MCGREN#i of MAC_PPS_CONTROL register to enable the PPS instance to operate in MCGR mode.

5. To set the PPS instance in Media Clock recovery mode, program the appropriate Presentation Time Control (supported recovery modes "0001 - 0011") to PPSCTRL_PPSCMD (for 0th instance)/PPSCMD#i (for 1,2,3 instances) of MAC_PPS_Control register.

6. After PPS instance is set in recovery mode, corresponding mcgr_dma_req_o[#i] is set by the DUT. For every request generated by the DUT, program the target presentation time into the MAC_PPS(#i)_TARGET_TIME_SECONDS register and acknowledge the request by asserting the corresponding mcgr_d-ma_ack_i[#i]. Acknowledgment can be given before/after programming the Target presentation time.

7. Observe the recovered media clock on the corresponding ptp_pps_o[#i].

61.6.10 Programming Guidelines for ECC Protection for Memories

1. Enable the ECC feature for the respective memory by setting the appropriate bit in MTL_ECC_CONTROL register.
2. A correctable interrupt (sbd_sfty_ce_intr_o) or an uncorrectable interrupt (sbd_sfty_ue_intr_o) are generated to indicate to the application if any correctable/ uncorrectable/address errors are detected. Appropriate status is indicated in the DMA/MTL_ECC_INTERRUPT_STATUS register.
3. Debug mode is provided for each memory to specify errors.

NOTE

Enable the ECC feature before the traffic is online (or after the reset), otherwise false interrupts might be triggered.

61.6.10.1 Programming Guidelines for ECC Hardware Error Injection (Debug Mode)

1. Enable the ECC error injection feature for MTL TX/RX and DMA TSO, by setting the appropriate bit in the MTL_DBG_CTL register.
2. Enable the ECC error injection feature for the EST memory by setting the appropriate bit in the EST_GCL_CONTROL register.
3. Enable the ECC error injection for RX Parser memory by setting the appropriate bit in the MTL_RXP_INDIRECT_ACC_CONTROL_STATUS register. To access Rx Parser memory in debug mode, disable the Rx Parser feature by setting the FRPE bit of MTL_OPERATION_MODE register to 0.
4. Where multiple CSR writes are required, for writing single data word into the memory, the application should ensure that all the CSR writes corresponding to one memory write should have the same value for the error injection control word.

61.6.11 Programming Guidelines for On-Chip Datapath Parity Protection

Follow the below steps for on-chip datapath parity protection:

1. Set the EDPP bit of MTL_DPP_CONTROL register to enable the generation of parity and detection of parity Mismatch on datapath.
2. An uncorrectable interrupt (sbd_sfty_ue_intr_o) is generated to indicate to the application whether any parity Mismatches are detected and the appropriate status will be indicated in DMA/MTL_SAFETY_INTERRUPT_STATUS and MAC_DPP_FSM_INTERRUPT_STATUS register.
3. Debug mode is supported for each parity generator, to insert error.

NOTE

To avoid false safety interrupts, enable the datapath parity protection before the start of Rx or Tx traffic.

61.6.12 Programming Guidelines for FSM Parity and Timeout

To program the FSM parity and timeout:

1. Set the PRTYEN and TMOUTEN fields of MAC_FSM_CONTROL register to enable the FSM parity and timeout feature respectively.
2. For parity error detection, force the FSMs to enter into a state with even number of 1s and wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
3. For Error Injection mode in FSM parity, select the appropriate clock domain for which parity error injection should be set using the [23:16] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt to be set with FSMPES of MAC_DPP_FSM_INTERRUPT_STATUS.
4. For FSM Timeouts, program the large and normal mode values indicated by LTMRMD and NTMRMD of MAC_FSM_ACT_TIMER register.
5. Select large or normal mode tic generation per clock domain by using the [31:24] bits of MAC_FSM_CONTROL Register.
6. Set the TMR field of the MAC_ACT_TIMER field with the appropriate number of CSR clock cycles used to generate 1-microsecond tic.

7. Force the FSM in a particular clock domain to be in an active state for Timeout duration (for example, T-2T). Check for the safety interrupt and the status bit for the corresponding clock domain getting set in [15:8] bits of MAC_DPP_FSM_INTERRUPT_STATUS register.
8. For Error Injection mode in FSM timeout, select the appropriate clock domain for which timeout error injection must be set using the [15:8] bits of MAC_FSM_CONTROL register. Wait for the safety uncorrectable interrupt along with the corresponding error status set in MAC_DPP_FSM_INTERRUPT_STATUS.
9. For application/CSR interface timeout, only normal mode tic generation is used. Based on the timeouts of the configured interface, appropriate fields (MSTTES, SLVTES) of MAC_DPP_FSM_INTERRUPT_STATUS are set along with the safety interrupt.

61.7 Memory Map and register definition

NOTE

This section and all its sub-sections are Synopsys Proprietary. Used with permission.

This section includes the ENET module memory map and detailed descriptions of all registers.

61.7.1 ENET_QOS register descriptions

61.7.1.1 ENET_QOS memory map

ENET_QOS base address: 4043_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	MAC Configuration Register (MAC_CONFIGURATION)	32	RW	0000_0000h
4h	MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)	32	RW	0000_0000h
8h	MAC Packet Filter (MAC_PACKET_FILTER)	32	RW	0000_0000h
Ch	Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
10h	MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)	32	RW	0000_0000h
14h	MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)	32	RW	0000_0000h
50h	MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)	32	RW	0000_0000h
54h	MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)	32	RW	0000_0000h
58h	MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)	32	RW	0000_0000h
60h	VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)	32	RW	0000_0000h
64h	MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)	32	RW	0000_0000h
70h	MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)	32	RW	0000_0000h
74h	MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)	32	RW	0000_0000h
78h	MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)	32	RW	0000_0000h
7Ch	MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)	32	RW	0000_0000h
80h	MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)	32	RW	0000_0000h
90h	MAC Rx Flow Control (MAC_RX_FLOW_CTRL)	32	RW	0000_0000h
94h	Receive Queue Control 4 (MAC_RXQ_CTRL4)	32	RW	0000_0000h
98h	Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)	32	RW	0000_0000h
9Ch	Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)	32	RW	0000_0000h
A0h	Receive Queue Control 0 (MAC_RXQ_CTRL0)	32	RW	0000_0000h
A4h	Receive Queue Control 1 (MAC_RXQ_CTRL1)	32	RW	0000_0000h
A8h	Receive Queue Control 2 (MAC_RXQ_CTRL2)	32	RW	0000_0000h
ACh	Receive Queue Control 3 (MAC_RXQ_CTRL3)	32	RW	0000_0000h
B0h	Interrupt Status (MAC_INTERRUPT_STATUS)	32	R	0000_0000h
B4h	Interrupt Enable (MAC_INTERRUPT_ENABLE)	32	RW	0000_0000h
B8h	Receive Transmit Status (MAC_RX_TX_STATUS)	32	R	0000_0000h
C0h	PMT Control and Status (MAC_PMT_CONTROL_STATUS)	32	RW	0000_0000h
C4h	Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)	32	RW	0000_0000h
D0h	LPI Control and Status (MAC_LPI_CONTROL_STATUS)	32	RW	0000_0000h
D4h	LPI Timers Control (MAC_LPI_TIMERS_CONTROL)	32	RW	03E8_0000h
D8h	Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)	32	RW	0000_0000h
DCh	One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)	32	RW	0000_0063h
F8h	PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)	32	RW	0000_0000h
110h	MAC Version (MAC_VERSION)	32	R	0000_1052h
114h	MAC Debug (MAC_DEBUG)	32	R	0000_0000h
11Ch	Optional Features or Functions 0 (MAC_HW_FEATURE0)	32	R	0EFD_71F7h
120h	Optional Features or Functions 1 (MAC_HW_FEATURE1)	32	R	4199_39A6h
124h	Optional Features or Functions 2 (MAC_HW_FEATURE2)	32	R	4410_4104h
128h	Optional Features or Functions 3 (MAC_HW_FEATURE3)	32	R	0C39_5632h
200h	MDIO Address (MAC_MDIO_ADDRESS)	32	RW	0000_0000h
204h	MAC MDIO Data (MAC_MDIO_DATA)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
230h	CSR Software Control (MAC_CSR_SW_CTRL)	32	RW	0000_0000h
234h	Frame Preemption Control (MAC_FPE_CTRL_STS)	32	RW	0000_0000h
240h	32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)	32	R	0000_0000h
244h	MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)	32	RW	0000_0000h
300h	MAC Address0 High (MAC_ADDRESS0_HIGH)	32	RW	8000_FFFFh
304h	MAC Address0 Low (MAC_ADDRESS0_LOW)	32	RW	FFFF_FFFFh
308h	MAC Address1 High (MAC_ADDRESS1_HIGH)	32	RW	0000_FFFFh
30Ch	MAC Address1 Low (MAC_ADDRESS1_LOW)	32	RW	FFFF_FFFFh
310h	MAC Address2 High (MAC_ADDRESS2_HIGH)	32	RW	0000_FFFFh
314h	MAC Address2 Low (MAC_ADDRESS2_LOW)	32	RW	FFFF_FFFFh
318h	MAC Address3 High (MAC_ADDRESS3_HIGH)	32	RW	0000_FFFFh
31Ch	MAC Address3 Low (MAC_ADDRESS3_LOW)	32	RW	FFFF_FFFFh
320h	MAC Address4 High (MAC_ADDRESS4_HIGH)	32	RW	0000_FFFFh
324h	MAC Address4 Low (MAC_ADDRESS4_LOW)	32	RW	FFFF_FFFFh
328h	MAC Address5 High (MAC_ADDRESS5_HIGH)	32	RW	0000_FFFFh
32Ch	MAC Address5 Low (MAC_ADDRESS5_LOW)	32	RW	FFFF_FFFFh
330h	MAC Address6 High (MAC_ADDRESS6_HIGH)	32	RW	0000_FFFFh
334h	MAC Address6 Low (MAC_ADDRESS6_LOW)	32	RW	FFFF_FFFFh
338h	MAC Address7 High (MAC_ADDRESS7_HIGH)	32	RW	0000_FFFFh
33Ch	MAC Address7 Low (MAC_ADDRESS7_LOW)	32	RW	FFFF_FFFFh
340h	MAC Address8 High (MAC_ADDRESS8_HIGH)	32	RW	0000_FFFFh
344h	MAC Address8 Low (MAC_ADDRESS8_LOW)	32	RW	FFFF_FFFFh
348h	MAC Address9 High (MAC_ADDRESS9_HIGH)	32	RW	0000_FFFFh
34Ch	MAC Address9 Low (MAC_ADDRESS9_LOW)	32	RW	FFFF_FFFFh
350h	MAC Address10 High (MAC_ADDRESS10_HIGH)	32	RW	0000_FFFFh
354h	MAC Address10 Low (MAC_ADDRESS10_LOW)	32	RW	FFFF_FFFFh
358h	MAC Address11 High (MAC_ADDRESS11_HIGH)	32	RW	0000_FFFFh
35Ch	MAC Address11 Low (MAC_ADDRESS11_LOW)	32	RW	FFFF_FFFFh
360h	MAC Address12 High (MAC_ADDRESS12_HIGH)	32	RW	0000_FFFFh
364h	MAC Address12 Low (MAC_ADDRESS12_LOW)	32	RW	FFFF_FFFFh
368h	MAC Address13 High (MAC_ADDRESS13_HIGH)	32	RW	0000_FFFFh
36Ch	MAC Address13 Low (MAC_ADDRESS13_LOW)	32	RW	FFFF_FFFFh
370h	MAC Address14 High (MAC_ADDRESS14_HIGH)	32	RW	0000_FFFFh
374h	MAC Address14 Low (MAC_ADDRESS14_LOW)	32	RW	FFFF_FFFFh
378h	MAC Address15 High (MAC_ADDRESS15_HIGH)	32	RW	0000_FFFFh
37Ch	MAC Address15 Low (MAC_ADDRESS15_LOW)	32	RW	FFFF_FFFFh
380h	MAC Address16 High (MAC_ADDRESS16_HIGH)	32	RW	0000_FFFFh
384h	MAC Address16 Low (MAC_ADDRESS16_LOW)	32	RW	FFFF_FFFFh
388h	MAC Address17 High (MAC_ADDRESS17_HIGH)	32	RW	0000_FFFFh
38Ch	MAC Address17 Low (MAC_ADDRESS17_LOW)	32	RW	FFFF_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
390h	MAC Address18 High (MAC_ADDRESS18_HIGH)	32	RW	0000_FFFFh
394h	MAC Address18 Low (MAC_ADDRESS18_LOW)	32	RW	FFFF_FFFFh
398h	MAC Address19 High (MAC_ADDRESS19_HIGH)	32	RW	0000_FFFFh
39Ch	MAC Address19 Low (MAC_ADDRESS19_LOW)	32	RW	FFFF_FFFFh
3A0h	MAC Address20 High (MAC_ADDRESS20_HIGH)	32	RW	0000_FFFFh
3A4h	MAC Address20 Low (MAC_ADDRESS20_LOW)	32	RW	FFFF_FFFFh
3A8h	MAC Address21 High (MAC_ADDRESS21_HIGH)	32	RW	0000_FFFFh
3ACh	MAC Address21 Low (MAC_ADDRESS21_LOW)	32	RW	FFFF_FFFFh
3B0h	MAC Address22 High (MAC_ADDRESS22_HIGH)	32	RW	0000_FFFFh
3B4h	MAC Address22 Low (MAC_ADDRESS22_LOW)	32	RW	FFFF_FFFFh
3B8h	MAC Address23 High (MAC_ADDRESS23_HIGH)	32	RW	0000_FFFFh
3BCh	MAC Address23 Low (MAC_ADDRESS23_LOW)	32	RW	FFFF_FFFFh
3C0h	MAC Address24 High (MAC_ADDRESS24_HIGH)	32	RW	0000_FFFFh
3C4h	MAC Address24 Low (MAC_ADDRESS24_LOW)	32	RW	FFFF_FFFFh
3C8h	MAC Address25 High (MAC_ADDRESS25_HIGH)	32	RW	0000_FFFFh
3CCh	MAC Address25 Low (MAC_ADDRESS25_LOW)	32	RW	FFFF_FFFFh
3D0h	MAC Address26 High (MAC_ADDRESS26_HIGH)	32	RW	0000_FFFFh
3D4h	MAC Address26 Low (MAC_ADDRESS26_LOW)	32	RW	FFFF_FFFFh
3D8h	MAC Address27 High (MAC_ADDRESS27_HIGH)	32	RW	0000_FFFFh
3DCh	MAC Address27 Low (MAC_ADDRESS27_LOW)	32	RW	FFFF_FFFFh
3E0h	MAC Address28 High (MAC_ADDRESS28_HIGH)	32	RW	0000_FFFFh
3E4h	MAC Address28 Low (MAC_ADDRESS28_LOW)	32	RW	FFFF_FFFFh
3E8h	MAC Address29 High (MAC_ADDRESS29_HIGH)	32	RW	0000_FFFFh
3ECh	MAC Address29 Low (MAC_ADDRESS29_LOW)	32	RW	FFFF_FFFFh
3F0h	MAC Address30 High (MAC_ADDRESS30_HIGH)	32	RW	0000_FFFFh
3F4h	MAC Address30 Low (MAC_ADDRESS30_LOW)	32	RW	FFFF_FFFFh
3F8h	MAC Address31 High (MAC_ADDRESS31_HIGH)	32	RW	0000_FFFFh
3FCh	MAC Address31 Low (MAC_ADDRESS31_LOW)	32	RW	FFFF_FFFFh
400h	MAC Address32 High (MAC_ADDRESS32_HIGH)	32	RW	0000_FFFFh
404h	MAC Address32 Low (MAC_ADDRESS32_LOW)	32	RW	FFFF_FFFFh
408h	MAC Address33 High (MAC_ADDRESS33_HIGH)	32	RW	0000_FFFFh
40Ch	MAC Address33 Low (MAC_ADDRESS33_LOW)	32	RW	FFFF_FFFFh
410h	MAC Address34 High (MAC_ADDRESS34_HIGH)	32	RW	0000_FFFFh
414h	MAC Address34 Low (MAC_ADDRESS34_LOW)	32	RW	FFFF_FFFFh
418h	MAC Address35 High (MAC_ADDRESS35_HIGH)	32	RW	0000_FFFFh
41Ch	MAC Address35 Low (MAC_ADDRESS35_LOW)	32	RW	FFFF_FFFFh
420h	MAC Address36 High (MAC_ADDRESS36_HIGH)	32	RW	0000_FFFFh
424h	MAC Address36 Low (MAC_ADDRESS36_LOW)	32	RW	FFFF_FFFFh
428h	MAC Address37 High (MAC_ADDRESS37_HIGH)	32	RW	0000_FFFFh
42Ch	MAC Address37 Low (MAC_ADDRESS37_LOW)	32	RW	FFFF_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
430h	MAC Address38 High (MAC_ADDRESS38_HIGH)	32	RW	0000_FFFFh
434h	MAC Address38 Low (MAC_ADDRESS38_LOW)	32	RW	FFFF_FFFFh
438h	MAC Address39 High (MAC_ADDRESS39_HIGH)	32	RW	0000_FFFFh
43Ch	MAC Address39 Low (MAC_ADDRESS39_LOW)	32	RW	FFFF_FFFFh
440h	MAC Address40 High (MAC_ADDRESS40_HIGH)	32	RW	0000_FFFFh
444h	MAC Address40 Low (MAC_ADDRESS40_LOW)	32	RW	FFFF_FFFFh
448h	MAC Address41 High (MAC_ADDRESS41_HIGH)	32	RW	0000_FFFFh
44Ch	MAC Address41 Low (MAC_ADDRESS41_LOW)	32	RW	FFFF_FFFFh
450h	MAC Address42 High (MAC_ADDRESS42_HIGH)	32	RW	0000_FFFFh
454h	MAC Address42 Low (MAC_ADDRESS42_LOW)	32	RW	FFFF_FFFFh
458h	MAC Address43 High (MAC_ADDRESS43_HIGH)	32	RW	0000_FFFFh
45Ch	MAC Address43 Low (MAC_ADDRESS43_LOW)	32	RW	FFFF_FFFFh
460h	MAC Address44 High (MAC_ADDRESS44_HIGH)	32	RW	0000_FFFFh
464h	MAC Address44 Low (MAC_ADDRESS44_LOW)	32	RW	FFFF_FFFFh
468h	MAC Address45 High (MAC_ADDRESS45_HIGH)	32	RW	0000_FFFFh
46Ch	MAC Address45 Low (MAC_ADDRESS45_LOW)	32	RW	FFFF_FFFFh
470h	MAC Address46 High (MAC_ADDRESS46_HIGH)	32	RW	0000_FFFFh
474h	MAC Address46 Low (MAC_ADDRESS46_LOW)	32	RW	FFFF_FFFFh
478h	MAC Address47 High (MAC_ADDRESS47_HIGH)	32	RW	0000_FFFFh
47Ch	MAC Address47 Low (MAC_ADDRESS47_LOW)	32	RW	FFFF_FFFFh
480h	MAC Address48 High (MAC_ADDRESS48_HIGH)	32	RW	0000_FFFFh
484h	MAC Address48 Low (MAC_ADDRESS48_LOW)	32	RW	FFFF_FFFFh
488h	MAC Address49 High (MAC_ADDRESS49_HIGH)	32	RW	0000_FFFFh
48Ch	MAC Address49 Low (MAC_ADDRESS49_LOW)	32	RW	FFFF_FFFFh
490h	MAC Address50 High (MAC_ADDRESS50_HIGH)	32	RW	0000_FFFFh
494h	MAC Address50 Low (MAC_ADDRESS50_LOW)	32	RW	FFFF_FFFFh
498h	MAC Address51 High (MAC_ADDRESS51_HIGH)	32	RW	0000_FFFFh
49Ch	MAC Address51 Low (MAC_ADDRESS51_LOW)	32	RW	FFFF_FFFFh
4A0h	MAC Address52 High (MAC_ADDRESS52_HIGH)	32	RW	0000_FFFFh
4A4h	MAC Address52 Low (MAC_ADDRESS52_LOW)	32	RW	FFFF_FFFFh
4A8h	MAC Address53 High (MAC_ADDRESS53_HIGH)	32	RW	0000_FFFFh
4ACh	MAC Address53 Low (MAC_ADDRESS53_LOW)	32	RW	FFFF_FFFFh
4B0h	MAC Address54 High (MAC_ADDRESS54_HIGH)	32	RW	0000_FFFFh
4B4h	MAC Address54 Low (MAC_ADDRESS54_LOW)	32	RW	FFFF_FFFFh
4B8h	MAC Address55 High (MAC_ADDRESS55_HIGH)	32	RW	0000_FFFFh
4BCh	MAC Address55 Low (MAC_ADDRESS55_LOW)	32	RW	FFFF_FFFFh
4C0h	MAC Address56 High (MAC_ADDRESS56_HIGH)	32	RW	0000_FFFFh
4C4h	MAC Address56 Low (MAC_ADDRESS56_LOW)	32	RW	FFFF_FFFFh
4C8h	MAC Address57 High (MAC_ADDRESS57_HIGH)	32	RW	0000_FFFFh
4CCh	MAC Address57 Low (MAC_ADDRESS57_LOW)	32	RW	FFFF_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4D0h	MAC Address58 High (MAC_ADDRESS58_HIGH)	32	RW	0000_FFFFh
4D4h	MAC Address58 Low (MAC_ADDRESS58_LOW)	32	RW	FFFF_FFFFh
4D8h	MAC Address59 High (MAC_ADDRESS59_HIGH)	32	RW	0000_FFFFh
4DCh	MAC Address59 Low (MAC_ADDRESS59_LOW)	32	RW	FFFF_FFFFh
4E0h	MAC Address60 High (MAC_ADDRESS60_HIGH)	32	RW	0000_FFFFh
4E4h	MAC Address60 Low (MAC_ADDRESS60_LOW)	32	RW	FFFF_FFFFh
4E8h	MAC Address61 High (MAC_ADDRESS61_HIGH)	32	RW	0000_FFFFh
4ECh	MAC Address61 Low (MAC_ADDRESS61_LOW)	32	RW	FFFF_FFFFh
4F0h	MAC Address62 High (MAC_ADDRESS62_HIGH)	32	RW	0000_FFFFh
4F4h	MAC Address62 Low (MAC_ADDRESS62_LOW)	32	RW	FFFF_FFFFh
4F8h	MAC Address63 High (MAC_ADDRESS63_HIGH)	32	RW	0000_FFFFh
4FCh	MAC Address63 Low (MAC_ADDRESS63_LOW)	32	RW	FFFF_FFFFh
700h	MMC Control (MAC_MMC_CONTROL)	32	RW	0000_0000h
704h	MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)	32	R	0000_0000h
708h	MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)	32	R	0000_0000h
70Ch	MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)	32	RW	0000_0000h
710h	MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)	32	RW	0000_0000h
714h	Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)	32	R	0000_0000h
718h	Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)	32	R	0000_0000h
71Ch	Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)	32	R	0000_0000h
720h	Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)	32	R	0000_0000h
724h	Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
728h	Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
72Ch	Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
730h	Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
734h	Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
738h	Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
73Ch	Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)	32	R	0000_0000h
740h	Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)	32	R	0000_0000h
744h	Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
748h	Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)	32	R	0000_0000h
74Ch	Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)	32	R	0000_0000h
750h	Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)	32	R	0000_0000h
754h	Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)	32	R	0000_0000h
758h	Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)	32	R	0000_0000h
75Ch	Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)	32	R	0000_0000h
760h	Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)	32	R	0000_0000h
764h	Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)	32	R	0000_0000h
768h	Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)	32	R	0000_0000h
76Ch	Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)	32	R	0000_0000h
770h	Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)	32	R	0000_0000h
774h	Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)	32	R	0000_0000h
778h	Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)	32	R	0000_0000h
780h	Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)	32	R	0000_0000h
784h	Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)	32	R	0000_0000h
788h	Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)	32	R	0000_0000h
78Ch	Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)	32	R	0000_0000h
790h	Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)	32	R	0000_0000h
794h	CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)	32	R	0000_0000h
798h	Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)	32	R	0000_0000h
79Ch	Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)	32	R	0000_0000h
7A0h	Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)	32	R	0000_0000h
7A4h	Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)	32	R	0000_0000h
7A8h	Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)	32	R	0000_0000h
7ACh	Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7B0h	Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
7B4h	Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
7B8h	Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
7BCh	Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
7C0h	Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)	32	R	0000_0000h
7C4h	Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)	32	R	0000_0000h
7C8h	Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)	32	R	0000_0000h
7CCh	Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)	32	R	0000_0000h
7D0h	Pause Packets Received (MAC_RX_PAUSE_PACKETS)	32	R	0000_0000h
7D4h	Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)	32	R	0000_0000h
7D8h	Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)	32	R	0000_0000h
7DCh	Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)	32	R	0000_0000h
7E0h	Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)	32	R	0000_0000h
7E4h	Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)	32	R	0000_0000h
7ECh	Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)	32	R	0000_0000h
7F0h	Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)	32	R	0000_0000h
7F4h	Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)	32	R	0000_0000h
7F8h	Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)	32	R	0000_0000h
800h	MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)	32	RW	0000_0000h
808h	MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)	32	R	0000_0000h
810h	Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)	32	R	0000_0000h
814h	IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)	32	R	0000_0000h
818h	IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)	32	R	0000_0000h
81Ch	IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)	32	R	0000_0000h
820h	IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)	32	R	0000_0000h
824h	Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
828h	IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)	32	R	0000_0000h
82Ch	IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)	32	R	0000_0000h
830h	IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)	32	R	0000_0000h
834h	IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)	32	R	0000_0000h
838h	IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)	32	R	0000_0000h
83Ch	IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)	32	R	0000_0000h
840h	IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)	32	R	0000_0000h
844h	IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)	32	R	0000_0000h
850h	Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)	32	R	0000_0000h
854h	Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)	32	R	0000_0000h
858h	Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)	32	R	0000_0000h
85Ch	Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)	32	R	0000_0000h
860h	Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)	32	R	0000_0000h
864h	Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)	32	R	0000_0000h
868h	Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)	32	R	0000_0000h
86Ch	Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)	32	R	0000_0000h
870h	Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)	32	R	0000_0000h
874h	Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)	32	R	0000_0000h
878h	Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)	32	R	0000_0000h
87Ch	Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)	32	R	0000_0000h
880h	Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)	32	R	0000_0000h
884h	Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)	32	R	0000_0000h
8A0h	MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8A4h	MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)	32	RW	0000_0000h
8A8h	MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)	32	R	0000_0000h
8ACh	MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)	32	R	0000_0000h
8C0h	MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)	32	R	0000_0000h
8C4h	MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)	32	RW	0000_0000h
8C8h	MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)	32	R	0000_0000h
8CCh	MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)	32	R	0000_0000h
8D0h	MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)	32	R	0000_0000h
8D4h	MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)	32	R	0000_0000h
900h	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)	32	RW	0000_0000h
904h	Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)	32	RW	0000_0000h
910h	Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)	32	RW	0000_0000h
914h	Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)	32	RW	0000_0000h
918h	Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)	32	RW	0000_0000h
91Ch	Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)	32	RW	0000_0000h
930h	Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)	32	RW	0000_0000h
934h	Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)	32	RW	0000_0000h
940h	Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)	32	RW	0000_0000h
944h	Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)	32	RW	0000_0000h
948h	Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)	32	RW	0000_0000h
94Ch	Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)	32	RW	0000_0000h
960h	Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)	32	RW	0000_0000h
964h	Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)	32	RW	0000_0000h
970h	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)	32	RW	0000_0000h
974h	Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)	32	RW	0000_0000h
978h	Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)	32	RW	0000_0000h
97Ch	Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)	32	RW	0000_0000h
990h	Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)	32	RW	0000_0000h
994h	Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)	32	RW	0000_0000h
9A0h	Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)	32	RW	0000_0000h
9A4h	Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)	32	RW	0000_0000h
9A8h	Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)	32	RW	0000_0000h
9ACh	Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)	32	RW	0000_0000h
9C0h	Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
9C4h	Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)	32	RW	0000_0000h
9D0h	Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)	32	RW	0000_0000h
9D4h	Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)	32	RW	0000_0000h
9D8h	Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)	32	RW	0000_0000h
9DCh	Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)	32	RW	0000_0000h
9F0h	Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)	32	RW	0000_0000h
9F4h	Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)	32	RW	0000_0000h
A00h	Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)	32	RW	0000_0000h
A04h	Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)	32	RW	0000_0000h
A08h	Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)	32	RW	0000_0000h
A0Ch	Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)	32	RW	0000_0000h
A20h	Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)	32	RW	0000_0000h
A24h	Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)	32	RW	0000_0000h
A30h	Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)	32	RW	0000_0000h
A34h	Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)	32	RW	0000_0000h
A38h	Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)	32	RW	0000_0000h
A3Ch	Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)	32	RW	0000_0000h
A50h	Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)	32	RW	0000_0000h
A54h	Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)	32	RW	0000_0000h
A60h	Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)	32	RW	0000_0000h
A64h	Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)	32	RW	0000_0000h
A68h	Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)	32	RW	0000_0000h
A6Ch	Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)	32	RW	0000_0000h
A70h	MAC_INDIR_ACCESS_CTRL (MAC_INDIR_ACCESS_CTRL)	32	RW	0000_0000h
A74h	MAC_INDIR_ACCESS_DATA (MAC_INDIR_ACCESS_DATA)	32	RW	0000_0000h
B00h	Timestamp Control (MAC_TIMESTAMP_CONTROL)	32	RW	0000_2000h
B04h	Subsecond Increment (MAC_SUB_SECOND_INCREMENT)	32	RW	0000_0000h
B08h	System Time Seconds (MAC_SYSTEM_TIME_SECONDS)	32	R	0000_0000h
B0Ch	System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)	32	R	0000_0000h
B10h	System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)	32	RW	0000_0000h
B14h	System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)	32	RW	0000_0000h
B18h	Timestamp Addend (MAC_TIMESTAMP_ADDEND)	32	RW	0000_0000h
B1Ch	System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)	32	RW	0000_0000h
B20h	Timestamp Status (MAC_TIMESTAMP_STATUS)	32	R	0000_0000h
B30h	Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B34h	Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)	32	R	0000_0000h
B40h	Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)	32	RW	0000_0000h
B48h	Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)	32	R	0000_0000h
B4Ch	Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)	32	R	0000_0000h
B50h	Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)	32	RW	0000_0000h
B54h	Timestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)	32	RW	0000_0000h
B58h	Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)	32	RW	0000_0000h
B5Ch	Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)	32	RW	0000_0000h
B60h	Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000h
B64h	Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)	32	RW	0000_0000h
B68h	Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)	32	R	0000_0000h
B6Ch	Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)	32	R	0000_0000h
B70h	PPS Control (MAC_PPS_CONTROL)	32	RW	0000_0000h
B80h	PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)	32	RW	0000_0000h
B84h	PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000h
B88h	PPS0 Interval (MAC_PPS0_INTERVAL)	32	RW	0000_0000h
B8Ch	PPS0 Width (MAC_PPS0_WIDTH)	32	RW	0000_0000h
B90h	PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)	32	RW	0000_0000h
B94h	PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000h
B98h	PPS1 Interval (MAC_PPS1_INTERVAL)	32	RW	0000_0000h
B9Ch	PPS1 Width (MAC_PPS1_WIDTH)	32	RW	0000_0000h
BA0h	PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)	32	RW	0000_0000h
BA4h	PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000h
BA8h	PPS2 Interval (MAC_PPS2_INTERVAL)	32	RW	0000_0000h
BACh	PPS2 Width (MAC_PPS2_WIDTH)	32	RW	0000_0000h
BB0h	PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
BB4h	PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)	32	RW	0000_0000h
BB8h	PPS3 Interval (MAC_PPS3_INTERVAL)	32	RW	0000_0000h
BBCh	PPS3 Width (MAC_PPS3_WIDTH)	32	RW	0000_0000h
BC0h	PTP Offload Engine Control (MAC_PTO_CONTROL)	32	RW	0000_0000h
BC4h	Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)	32	RW	0000_0000h
BC8h	Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)	32	RW	0000_0000h
BCCh	Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)	32	RW	0000_0000h
BD0h	Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)	32	RW	0000_0000h
C00h	MTL Operation Mode (MTL_OPERATION_MODE)	32	RW	0000_0000h
C08h	FIFO Debug Access Control and Status (MTL_DBG_CTL)	32	RW	0000_0000h
C0Ch	FIFO Debug Status (MTL_DBG_STS)	32	RW	0000_0018h
C10h	FIFO Debug Data (MTL_FIFO_DEBUG_DATA)	32	RW	0000_0000h
C20h	MTL Interrupt Status (MTL_INTERRUPT_STATUS)	32	R	0000_0000h
C30h	Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)	32	RW	0000_0000h
C34h	Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)	32	RW	0000_0000h
C40h	Time Based Scheduling Control (MTL_TBS_CTRL)	32	RW	0000_0000h
C50h	Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)	32	RW	0000_0000h
C54h	MTL_EST_EXT_CONTROL (MTL_EST_EXT_CONTROL)	32	RW	0000_0000h
C58h	Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)	32	RW	0000_0000h
C60h	EST Scheduling Error (MTL_EST_SCH_ERROR)	32	RW	0000_0000h
C64h	EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)	32	RW	0000_0000h
C68h	EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)	32	R	0000_0000h
C70h	EST Interrupt Enable (MTL_EST_INTR_ENABLE)	32	RW	0000_0000h
C80h	EST GCL Control (MTL_EST_GCL_CONTROL)	32	RW	0000_0000h
C84h	EST GCL Data (MTL_EST_GCL_DATA)	32	RW	0000_0000h
C90h	Frame Preemption Control and Status (MTL_FPE_CTRL_STS)	32	RW	0000_0000h
C94h	Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)	32	RW	0000_0000h
CA0h	RXP Control Status (MTL_RXP_CONTROL_STATUS)	32	RW	80FF_00FFh
CA4h	RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
CA8h	RXP Drop Count (MTL_RXP_DROP_CNT)	32	R	0000_0000h
CACh	RXP Error Count (MTL_RXP_ERROR_CNT)	32	R	0000_0000h
CB0h	RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)	32	RW	0000_0000h
CB4h	RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)	32	RW	0000_0000h
CB8h	MTL_RXP_BYPASS_CNT (MTL_RXP_BYPASS_CNT)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
D00h	Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)	32	RW	0000_0000h
D04h	Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)	32	R	0000_0000h
D08h	Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)	32	R	0000_0000h
D14h	Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)	32	R	0000_0000h
D18h	Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)	32	RW	0000_0000h
D2Ch	Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
D30h	Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)	32	RW	0000_0000h
D34h	Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000h
D38h	Queue 0 Receive Debug (MTL_RXQ0_DEBUG)	32	R	0000_0000h
D3Ch	Queue 0 Receive Control (MTL_RXQ0_CONTROL)	32	RW	0000_0000h
D40h	Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)	32	RW	0000_0000h
D44h	Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)	32	R	0000_0000h
D48h	Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)	32	R	0000_0000h
D50h	Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)	32	RW	0000_0000h
D54h	Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)	32	R	0000_0000h
D58h	Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)	32	RW	0000_0000h
D5Ch	Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)	32	RW	0000_0000h
D60h	Queue 1 hiCredit (MTL_TXQ1_HICREDIT)	32	RW	0000_0000h
D64h	Queue 1 loCredit (MTL_TXQ1_LOCREDIT)	32	RW	0000_0000h
D6Ch	Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
D70h	Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)	32	RW	0000_0000h
D74h	Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000h
D78h	Queue 1 Receive Debug (MTL_RXQ1_DEBUG)	32	R	0000_0000h
D7Ch	Queue 1 Receive Control (MTL_RXQ1_CONTROL)	32	RW	0000_0000h
D80h	Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)	32	RW	0000_0000h
D84h	Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)	32	R	0000_0000h
D88h	Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)	32	R	0000_0000h
D90h	Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)	32	RW	0000_0000h
D94h	Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)	32	R	0000_0000h
D98h	Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)	32	RW	0000_0000h
D9Ch	Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)	32	RW	0000_0000h
DA0h	Queue 2 hiCredit (MTL_TXQ2_HICREDIT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
DA4h	Queue 2 loCredit (MTL_TXQ2_LOCREDIT)	32	RW	0000_0000h
DACh	Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
DB0h	Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)	32	RW	0000_0000h
DB4h	Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000h
DB8h	Queue 2 Receive Debug (MTL_RXQ2_DEBUG)	32	R	0000_0000h
DBCh	Queue 2 Receive Control (MTL_RXQ2_CONTROL)	32	RW	0000_0000h
DC0h	Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)	32	RW	0000_0000h
DC4h	Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)	32	R	0000_0000h
DC8h	Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)	32	R	0000_0000h
DD0h	Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)	32	RW	0000_0000h
DD4h	Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)	32	R	0000_0000h
DD8h	Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)	32	RW	0000_0000h
DDCh	Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)	32	RW	0000_0000h
DE0h	Queue 3 hiCredit (MTL_TXQ3_HICREDIT)	32	RW	0000_0000h
DE4h	Queue 3 loCredit (MTL_TXQ3_LOCREDIT)	32	RW	0000_0000h
DECh	Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
DF0h	Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)	32	RW	0000_0000h
DF4h	Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000h
DF8h	Queue 3 Receive Debug (MTL_RXQ3_DEBUG)	32	R	0000_0000h
DFCh	Queue 3 Receive Control (MTL_RXQ3_CONTROL)	32	RW	0000_0000h
E00h	Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)	32	RW	0000_0000h
E04h	Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)	32	R	0000_0000h
E08h	Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)	32	R	0000_0000h
E10h	Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)	32	RW	0000_0000h
E14h	Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)	32	R	0000_0000h
E18h	Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)	32	RW	0000_0000h
E1Ch	Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)	32	RW	0000_0000h
E20h	Queue 4 hiCredit (MTL_TXQ4_HICREDIT)	32	RW	0000_0000h
E24h	Queue 4 loCredit (MTL_TXQ4_LOCREDIT)	32	RW	0000_0000h
E2Ch	Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)	32	RW	0000_0000h
E30h	Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
E34h	Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)	32	R	0000_0000h
E38h	Queue 4 Receive Debug (MTL_RXQ4_DEBUG)	32	R	0000_0000h
E3Ch	Queue 4 Receive Control (MTL_RXQ4_CONTROL)	32	RW	0000_0000h
1000h	DMA Bus Mode (DMA_MODE)	32	RW	0000_0000h
1004h	DMA System Bus Mode (DMA_SYSBUS_MODE)	32	RW	0101_0000h
1008h	DMA Interrupt Status (DMA_INTERRUPT_STATUS)	32	R	0000_0000h
100Ch	DMA Debug Status 0 (DMA_DEBUG_STATUS0)	32	R	0000_0000h
1010h	DMA Debug Status 1 (DMA_DEBUG_STATUS1)	32	R	0000_0000h
1040h	AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)	32	RW	0000_0000h
1050h	DMA_TBS_CTRL0 (DMA_TBS_CTRL0)	32	RW	0000_0000h
1054h	DMA_TBS_CTRL1 (DMA_TBS_CTRL1)	32	RW	0000_0000h
1058h	DMA_TBS_CTRL2 (DMA_TBS_CTRL2)	32	RW	0000_0000h
105Ch	DMA_TBS_CTRL3 (DMA_TBS_CTRL3)	32	RW	0000_0000h
1100h	DMA Channel 0 Control (DMA_CH0_CONTROL)	32	RW	0000_0000h
1104h	DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)	32	RW	0000_0000h
1108h	DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)	32	RW	0000_0000h
1114h	Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)	32	RW	0000_0000h
111Ch	Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)	32	RW	0000_0000h
1120h	Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)	32	RW	0000_0000h
1128h	Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)	32	RW	0000_0000h
112Ch	Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)	32	RW	0000_0000h
1130h	Channel 0 Receive Control 2 register (DMA_CH0_RX_CONTROL2)	32	RW	0000_0000h
1134h	Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)	32	RW	0000_0000h
1138h	Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000h
113Ch	Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0h
1144h	Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)	32	R	0000_0000h
114Ch	Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)	32	R	0000_0000h
1154h	Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)	32	R	0000_0000h
115Ch	Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)	32	R	0000_0000h
1160h	DMA Channel 0 Status (DMA_CH0_STATUS)	32	RW	0000_0000h
1164h	Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1168h	Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)	32	R	0000_0000h
1180h	DMA Channel 1 Control (DMA_CH1_CONTROL)	32	RW	0000_0000h
1184h	DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)	32	RW	0000_0000h
1188h	DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)	32	RW	0000_0000h
1194h	Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)	32	RW	0000_0000h
119Ch	Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)	32	RW	0000_0000h
11A0h	Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)	32	RW	0000_0000h
11A8h	Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)	32	RW	0000_0000h
11ACh	Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)	32	RW	0000_0000h
11B0h	Channel 1 Receive Control 2 register (DMA_CH1_RX_CONTROL2)	32	RW	0000_0000h
11B4h	Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)	32	RW	0000_0000h
11B8h	Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000h
11BCh	Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0h
11C4h	Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)	32	R	0000_0000h
11CCh	Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)	32	R	0000_0000h
11D4h	Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)	32	R	0000_0000h
11DCh	Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)	32	R	0000_0000h
11E0h	DMA Channel 1 Status (DMA_CH1_STATUS)	32	RW	0000_0000h
11E4h	Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)	32	R	0000_0000h
11E8h	Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)	32	R	0000_0000h
1200h	DMA Channel 2 Control (DMA_CH2_CONTROL)	32	RW	0000_0000h
1204h	DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)	32	RW	0000_0000h
1208h	DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)	32	RW	0000_0000h
1214h	Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)	32	RW	0000_0000h
121Ch	Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)	32	RW	0000_0000h
1220h	Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)	32	RW	0000_0000h
1228h	Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
122Ch	Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)	32	RW	0000_0000h
1230h	Channel 2 Receive Control 2 register (DMA_CH2_RX_CONTROL2)	32	RW	0000_0000h
1234h	Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)	32	RW	0000_0000h
1238h	Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000h
123Ch	Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0h
1244h	Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)	32	R	0000_0000h
124Ch	Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)	32	R	0000_0000h
1254h	Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)	32	R	0000_0000h
125Ch	Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)	32	R	0000_0000h
1260h	DMA Channel 2 Status (DMA_CH2_STATUS)	32	RW	0000_0000h
1264h	Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)	32	R	0000_0000h
1268h	Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)	32	R	0000_0000h
1280h	DMA Channel 3 Control (DMA_CH3_CONTROL)	32	RW	0000_0000h
1284h	DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)	32	RW	0000_0000h
1288h	DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)	32	RW	0000_0000h
1294h	Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)	32	RW	0000_0000h
129Ch	Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)	32	RW	0000_0000h
12A0h	Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)	32	RW	0000_0000h
12A8h	Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)	32	RW	0000_0000h
12ACh	Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)	32	RW	0000_0000h
12B0h	Channel 3 Receive Control 2 register (DMA_CH3_RX_CONTROL2)	32	RW	0000_0000h
12B4h	Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)	32	RW	0000_0000h
12B8h	Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000h
12BCh	Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0h
12C4h	Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)	32	R	0000_0000h
12CCh	Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)	32	R	0000_0000h
12D4h	Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)	32	R	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
12DCh	Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)	32	R	0000_0000h
12E0h	DMA Channel 3 Status (DMA_CH3_STATUS)	32	RW	0000_0000h
12E4h	Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)	32	R	0000_0000h
12E8h	Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)	32	R	0000_0000h
1300h	Channel 4 Control (DMA_CH4_CONTROL)	32	RW	0000_0000h
1304h	DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)	32	RW	0000_0000h
1308h	DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)	32	RW	0000_0000h
1314h	Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)	32	RW	0000_0000h
131Ch	Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)	32	RW	0000_0000h
1320h	Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)	32	RW	0000_0000h
1328h	Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)	32	RW	0000_0000h
132Ch	Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)	32	RW	0000_0000h
1330h	DMA Channel 4 Receive Control 2 register (DMA_CH4_RX_CONTROL2)	32	RW	0000_0000h
1334h	Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)	32	RW	0000_0000h
1338h	Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)	32	RW	0000_0000h
133Ch	Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)	32	RW	0000_07C0h
1344h	Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)	32	R	0000_0000h
134Ch	Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)	32	R	0000_0000h
1354h	Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)	32	R	0000_0000h
135Ch	Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)	32	R	0000_0000h
1360h	DMA Channel 4 Status (DMA_CH4_STATUS)	32	RW	0000_0000h
1364h	Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)	32	R	0000_0000h
1368h	Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)	32	R	0000_0000h

61.7.1.2 MAC Configuration Register (MAC_CONFIGURATION)

61.7.1.2.1 Offset

Register	Offset
MAC_CONFIGURATION	0h

61.7.1.2.2 Function

The MAC Configuration Register establishes the operating mode of the MAC.

61.7.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	Reserved	SARC			IPC	IPG			GPSLCE	S2KP	CST	ACS	WD	BE	JD	J E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PS	FES	DM	LM	ECRSFD	DO	DCRS	DR	Reserved	BL		DC	PRELEN		TE	RE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.2.4 Fields

Field	Function
31 —	Reserved
30-28 SARC	Source Address Insertion or Replacement Control This field controls the source address insertion or replacement for all transmitted packets. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits[29:28]: 2'b0x: - The mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation. 2'b10: - If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected while configuring the core, the MAC inserts the content of the MAC Address 1 registers in the SA field of all transmitted packets.

Table continues on the next page...

Field	Function
	2'b11: - If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers in the SA field of all transmitted packets. - If Bit 30 is set to 1 and the MAC Address Register 1 is enabled, the MAC replaces the content of the MAC Address 1 registers in the SA field of all transmitted packets. Note: - Changes to this field take effect only on the start of a packet. If you write to this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 000b - mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation 010b - Contents of MAC Addr-0 inserted in SA field 011b - Contents of MAC Addr-0 replaces SA field 110b - Contents of MAC Addr-1 inserted in SA field 111b - Contents of MAC Addr-1 replaces SA field
27 IPC	Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled. The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. 0b - IP header/payload checksum checking is disabled 1b - IP header/payload checksum checking is enabled
26-24 IPG	Inter-Packet Gap These bits control the minimum IPG between packets during transmission. Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode. In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG. The above function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_EXT_CONFIGURATION register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_EXT_CONFIGURATION register. 000b - 96 bit times IPG 001b - 88 bit times IPG 010b - 80 bit times IPG 011b - 72 bit times IPG 100b - 64 bit times IPG 101b - 56 bit times IPG 110b - 48 bit times IPG 111b - 40 bit times IPG
23 GPSLCE	Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_EXT_CONFIGURATION register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet).

Table continues on the next page...

Field	Function
	The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. 0b - Giant Packet Size Limit Control is disabled 1b - Giant Packet Size Limit Control is enabled
22 S2KP	IEEE 802. IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets. When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For more information about how the setting of this bit and the JE bit impact the Giant packet status, see the Table, Gaint Packet Status based on S2KP and JE Bits. Note: When the JE bit is set, setting this bit has no effect on the giant packet status. 0b - Support upto 2K packet is disabled 1b - Support upto 2K packet is Enabled
21 CST	CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits. 0b - CRC stripping for Type packets is disabled 1b - CRC stripping for Type packets is enabled
20 ACS	Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit . 0b - Automatic Pad or CRC Stripping is disabled 1b - Automatic Pad or CRC Stripping is enabled
19 WD	Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes. When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. 0b - Watchdog is enabled 1b - Watchdog is disabled

Table continues on the next page...

Field	Function
18 BE	Packet Burst Enable When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode. 0b - Packet Burst is disabled 1b - Packet Burst is enabled
17 JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes. When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet. 0b - Jabber is enabled 1b - Jabber is disabled
16 JE	Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. 0b - Jumbo packet is disabled 1b - Jumbo packet is enabled
15 PS	Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit. 0b - For 1000 or 2500 Mbps operations 1b - For 10 or 100 Mbps operations
14 FES	Speed This bit selects the speed mode. The mac_speed_o[0] signal reflects the value of this bit. 0b - 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1b - 100 Mbps when PS bit is 1 and 2.5 Gbps when PS bit is 0
13 DM	Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations. 0b - Half-duplex mode 1b - Full-duplex mode
12 LM	Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back. 0b - Loopback is disabled 1b - Loopback is enabled
11 ECRSFD	Enable Carrier Sense Before Transmission in Full-Duplex Mode When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low. When this bit is reset, the MAC transmitter ignores the status of the CRS signal. 0b - ECRSFD is disabled 1b - ECRSFD is enabled
10	Disable Receive Own

Table continues on the next page...

Field	Function
DO	When this bit is set, the MAC disables the reception of packets when the gmii_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY. This bit is not applicable in the full-duplex mode. 0b - Enable Receive Own 1b - Disable Receive Own
9 DCRS	Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. 0b - Enable Carrier Sense During Transmission 1b - Disable Carrier Sense During Transmission
8 DR	Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status. When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. 0b - Enable Retry 1b - Disable Retry
7 —	Reserved
6-5 BL	Back-Off Limit The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range 0 <= r < 2^k This bit is applicable only in the half-duplex mode. 00b - k = min(n,10) 01b - k = min(n,8) 10b - k = min(n,4) 11b - k = min(n,1)
4 DC	Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted.

Table continues on the next page...

Field	Function
	When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. 0b - Deferral check function is disabled 1b - Deferral check function is enabled
3-2 PRELEN	Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. 00b - 7 bytes of preamble 01b - 5 bytes of preamble 10b - 3 bytes of preamble 11b - Reserved
1 TE	Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII or MII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. 0b - Transmitter is disabled 1b - Transmitter is enabled
0 RE	Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII or MII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII or MII interface. 0b - Receiver is disabled 1b - Receiver is enabled

61.7.1.3 MAC Extended Configuration Register (MAC_EXT_CONFIGURATION)

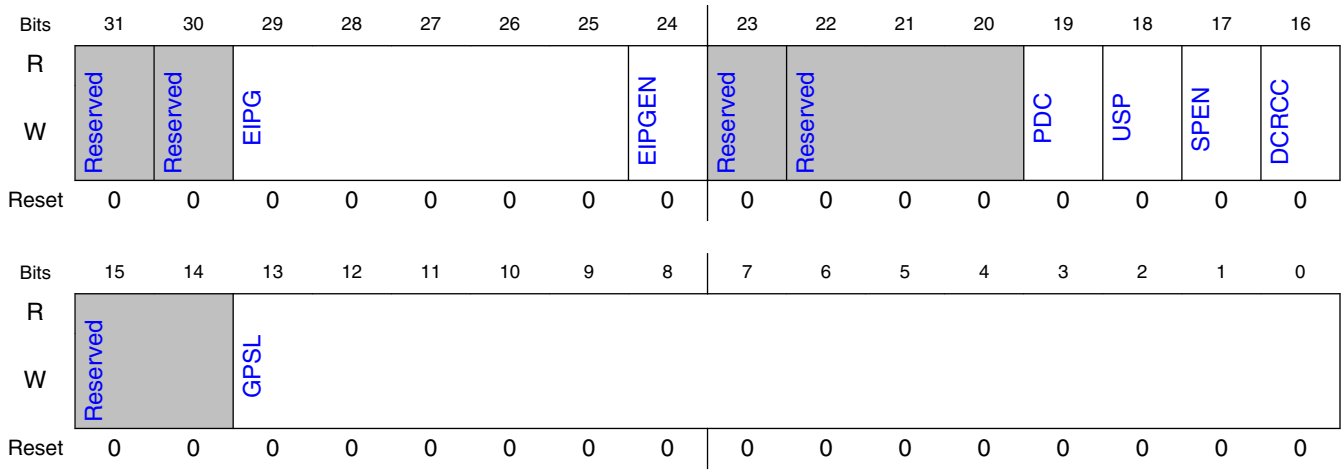
61.7.1.3.1 Offset

Register	Offset
MAC_EXT_CONFIGURATION	4h

61.7.1.3.2 Function

The MAC Extended Configuration Register establishes the operating mode of the MAC.

61.7.1.3.3 Diagram



61.7.1.3.4 Fields

Field	Function
31 —	Reserved
30 —	Reserved
29-25 EIPG	Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in CONFIGURATION register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: {EIPG, IPG} 8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ---------- 8'hFF - 2144 bit times
24 EIPGEN	Extended Inter-Packet Gap Enable When this bit is set, the MAC interprets EIPG field and IPG field in CONFIGURATION register together as minimum IPG greater than 96 bit times in steps of 8 bit times. When this bit is reset, the MAC ignores EIPG field and interprets IPG field in CONFIGURATION register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times. Note: The extended Inter-Packet Gap feature must be enabled when operating in Full-Duplex mode only. There may be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode. 0b - Extended Inter-Packet Gap is disabled 1b - Extended Inter-Packet Gap is enabled
23	Reserved

Table continues on the next page...

Field	Function
—	
22-20 —	Reserved
19 PDC	Packet Duplication Control When this bit is set, the received packet with Multicast/Broadcast Destination address is routed to multiple Receive DMA Channels. The Receive DMA Channels is identified by the DCS field of MAC_ADDRESS(#i)_HIGH register corresponding to the MAC Address register that matches the Multicast/Broadcast Destination address in the received packet. The DCS field is interpreted to be a one-hot value, each bit corresponding to the Receive DMA Channel. When this bit is reset, the received packet is routed to single Receive DMA Channel. The Receive DMA Channel is identified by the DCS field of ADDRESS(#i)_High register corresponding to the MAC Address register that matches the Destination address in the received packet. The DCS field is interpreted as a binary value. 0b - Packet Duplication Control is disabled 1b - Packet Duplication Control is enabled
18 USP	Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. 0b - Unicast Slow Protocol Packet Detection is disabled 1b - Unicast Slow Protocol Packet Detection is enabled
17 SPEN	Slow Protocol Detection Enable When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Rx status. The MAC discards the Slow Protocol packets with invalid sub-types. When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. 0b - Slow Protocol Detection is disabled 1b - Slow Protocol Detection is enabled
16 DCRCC	Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. 0b - CRC Checking is enabled 1b - CRC Checking is disabled
15-14 —	Reserved
13-0 GPSL	Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_CONFIGURATION register.

61.7.1.4 MAC Packet Filter (MAC_PACKET_FILTER)

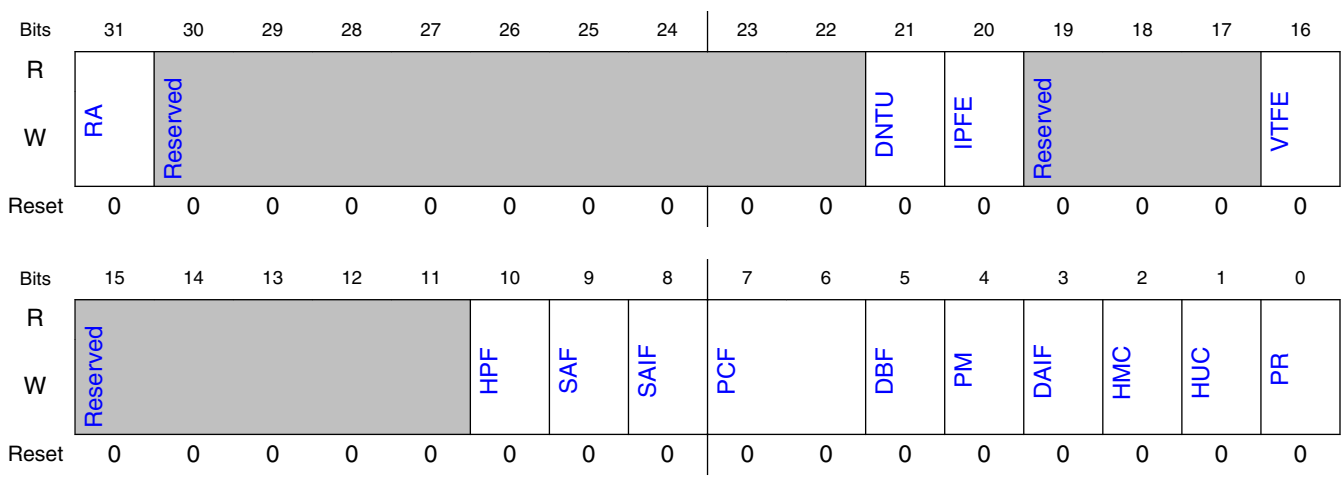
61.7.1.4.1 Offset

Register	Offset
MAC_PACKET_FILTER	8h

61.7.1.4.2 Function

The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

61.7.1.4.3 Diagram



61.7.1.4.4 Fields

Field	Function
31 RA	Receive All When this bit is set, the MAC Receiver module passes all received packets to the application, irrespective of whether they pass the address filter or not. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bit in the Rx Status Word. When this bit is reset, the Receiver module passes only those packets to the application that pass the SA or DA address filter.

Table continues on the next page...

Field	Function
	0b - Receive All is disabled 1b - Receive All is enabled
30-22 —	Reserved
21 DNTU	Drop Non-TCP/UDP over IP Packets When this bit is set, the MAC drops the non-TCP or UDP over IP packets. The MAC forward only those packets that are processed by the Layer 4 filter. When this bit is reset, the MAC forwards all non-TCP or UDP over IP packets. 0b - Forward Non-TCP/UDP over IP Packets 1b - Drop Non-TCP/UDP over IP Packets
20 IPFE	Layer 3 and Layer 4 Filter Enable When this bit is set, the MAC drops packets that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect. When this bit is reset, the MAC forwards all packets irrespective of the match status of the Layer 3 and Layer 4 fields. 0b - Layer 3 and Layer 4 Filters are disabled 1b - Layer 3 and Layer 4 Filters are enabled
19-17 —	Reserved
16 VTFE	VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. 0b - VLAN Tag Filter is disabled 1b - VLAN Tag Filter is enabled
15-11 —	Reserved
10 HPF	Hash or Perfect Filter When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. 0b - Hash or Perfect Filter is disabled 1b - Hash or Perfect Filter is enabled
9 SAF	Source Address Filter Enable When this bit is set, the MAC compares the SA field of the received packets with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the packet. When this bit is reset, the MAC forwards the received packet to the application with updated SAF bit of the Rx Status depending on the SA address comparison. Note: According to the IEEE specification, Bit 47 of the SA is reserved. However, in DWC_ether_qos, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA. 0b - SA Filtering is disabled 1b - SA Filtering is enabled
8 SAIF	SA Inverse Filtering

Table continues on the next page...

Field	Function
	When this bit is set, the Address Check block operates in the inverse filtering mode for SA address comparison. If the SA of a packet matches the values programmed in the SA registers, it is marked as failing the SA Address filter. When this bit is reset, if the SA of a packet does not match the values programmed in the SA registers, it is marked as failing the SA Address filter. 0b - SA Inverse Filtering is disabled 1b - SA Inverse Filtering is enabled
7-6 PCF	Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). 00b - MAC filters all control packets from reaching the application 01b - MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 10b - MAC forwards all control packets to the application even if they fail the Address filter 11b - MAC forwards the control packets that pass the Address filter
5 DBF	Disable Broadcast Packets When this bit is set, the AFM module blocks all incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. 0b - Enable Broadcast Packets 1b - Disable Broadcast Packets
4 PM	Pass All Multicast When this bit is set, it indicates that all received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. 0b - Pass All Multicast is disabled 1b - Pass All Multicast is enabled
3 DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed. 0b - DA Inverse Filtering is disabled 1b - DA Inverse Filtering is enabled
2 HMC	Hash Multicast When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table. When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers. 0b - Hash Multicast is disabled 1b - Hash Multicast is enabled
1 HUC	Hash Unicast When this bit is set, the MAC performs the destination address filtering of unicast packets according to the hash table. When this bit is reset, the MAC performs a perfect destination address filtering for unicast packets, that is, it compares the DA field with the values programmed in DA registers. 0b - Hash Unicast is disabled 1b - Hash Unicast is enabled
0	Promiscuous Mode

Field	Function
PR	When this bit is set, the Address Filtering module passes all incoming packets irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Rx Status Word are always cleared when PR is set. 0b - Promiscuous Mode is disabled 1b - Promiscuous Mode is enabled

61.7.1.5 Watchdog Timeout (MAC_WATCHDOG_TIMEOUT)

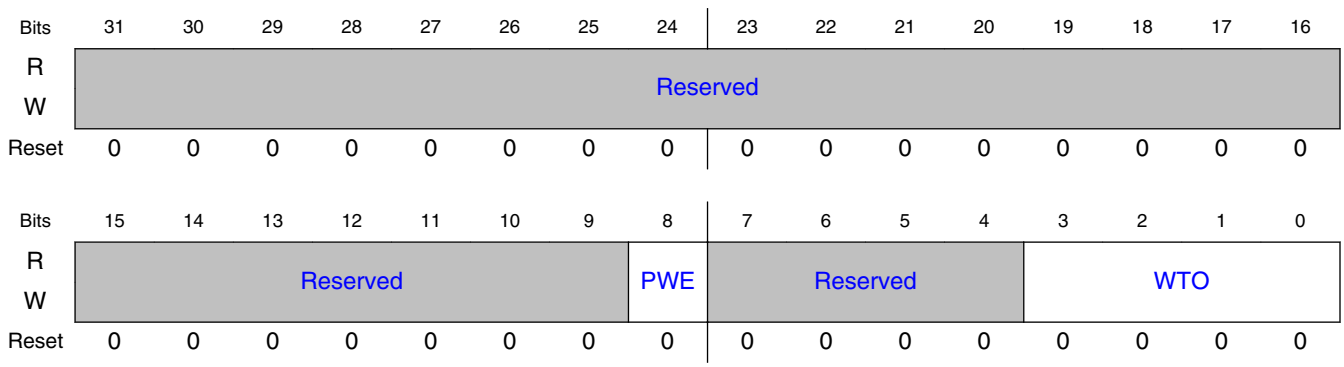
61.7.1.5.1 Offset

Register	Offset
MAC_WATCHDOG_TIMEOUT	Ch

61.7.1.5.2 Function

The Watchdog Timeout register controls the watchdog timeout for received packets.

61.7.1.5.3 Diagram



61.7.1.5.4 Fields

Field	Function
31-9 —	Reserved
8 PWE	Programmable Watchdog Enable

Table continues on the next page...

Field	Function
	When this bit is set and the WD bit of the MAC_CONFIGURATION register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_CONFIGURATION register. 0b - Programmable Watchdog is disabled 1b - Programmable Watchdog is enabled
7-4 —	Reserved
3-0 WTO	Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_CONFIGURATION register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet. Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped. 0000b - 2 KB 0001b - 3 KB 0010b - 4 KB 0011b - 5 KB 0100b - 6 KB 0101b - 7 KB 0110b - 8 KB 0111b - 9 KB 1000b - 10 KB 1001b - 11 KB 1010b - 12 KB 1011b - 13 KB 1100b - 14 KB 1101b - 15 KB 1110b - 16383 Bytes 1111b - Reserved

61.7.1.6 MAC Hash Table Register 0 (MAC_HASH_TABLE_REG0)

61.7.1.6.1 Offset

Register	Offset
MAC_HASH_TABLE_REG0	10h

61.7.1.6.2 Function

The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash table is 128 or 256 bits.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

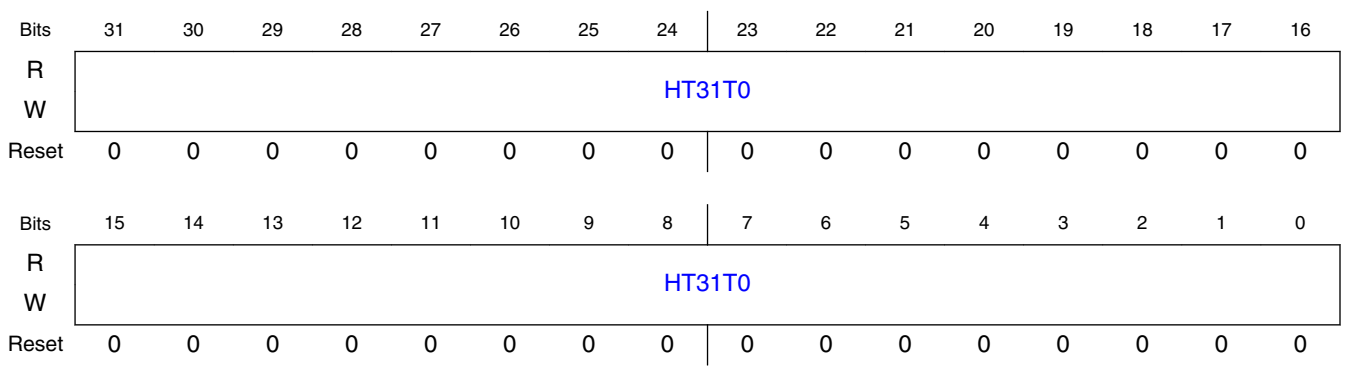
- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.6.3 Diagram



61.7.1.6.4 Fields

Field	Function
31-0 HT31T0	MAC Hash Table First 32 Bits This field contains the first 32 Bits [31:0] of the Hash table.

61.7.1.7 MAC Hash Table Register 1 (MAC_HASH_TABLE_REG1)

61.7.1.7.1 Offset

Register	Offset
MAC_HASH_TABLE_REG1	14h

61.7.1.7.2 Function

The Hash Table Register 1 contains the second 32 bits of the hash table.

The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

- Perform bitwise reversal for the value obtained in Step 1.

- Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HT63T32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.7.4 Fields

Field	Function
31-0 HT63T32	MAC Hash Table Second 32 Bits This field contains the second 32 Bits [63:32] of the Hash table.

61.7.1.8 MAC VLAN Tag Control (MAC_VLAN_TAG_CTRL)

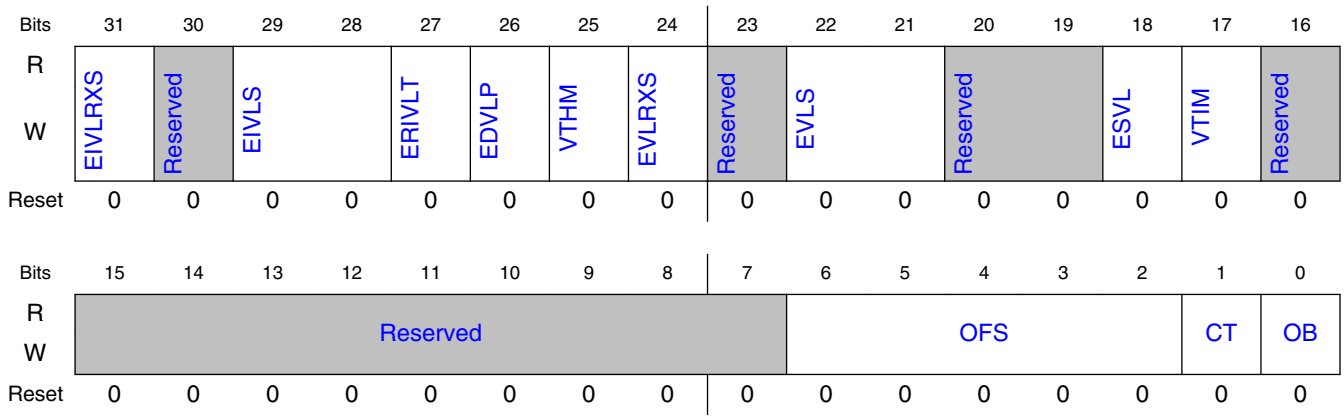
61.7.1.8.1 Offset

Register	Offset
MAC_VLAN_TAG_CTRL	50h

61.7.1.8.2 Function

This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing. It contains the address offset, command type and Busy Bit for CSR access of the Per VLAN Tag registers.

61.7.1.8.3 Diagram



61.7.1.8.4 Fields

Field	Function
31 EIVLRXS	Enable Inner VLAN Tag in Rx Status When this bit is set, the MAC provides the inner VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the inner VLAN Tag in Rx status. 0b - Inner VLAN Tag in Rx status is disabled 1b - Inner VLAN Tag in Rx status is enabled
30 —	Reserved
29-28 EIVLS	Enable Inner VLAN Tag Stripping on Receive This field indicates the stripping operation on inner VLAN Tag in received packet. 00b - Do not strip 01b - Strip if VLAN filter passes 10b - Strip if VLAN filter fails 11b - Always strip
27 ERIVLT	ERIVLT 0b - Inner VLAN tag is disabled 1b - Inner VLAN tag is enabled
26 EDVLP	Enable Double VLAN Processing When this bit is set, the MAC enables processing of up to two VLAN Tags on Tx and Rx (if present). When this bit is reset, the MAC enables processing of up to one VLAN Tag on Tx and Rx (if present). 0b - Double VLAN Processing is disabled 1b - Double VLAN Processing is enabled
25 VTHM	VLAN Tag Hash Table Match Enable When this bit is set, the most significant four bits of CRC of VLAN Tag (ones-complement of most significant four bits of CRC of VLAN Tag when ETV bit is reset) are used to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the packet matched the VLAN hash table. When the ETV bit is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison. When the ETV bit is reset, the ones-complement of the CRC of the 16-bit VLAN tag is used for comparison.

Table continues on the next page...

Field	Function
	When this bit is reset, the VLAN Hash Match operation is not performed. 0b - VLAN Tag Hash Table Match is disabled 1b - VLAN Tag Hash Table Match is enabled
24 EVLRXS	Enable VLAN Tag in Rx status When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status. 0b - VLAN Tag in Rx status is disabled 1b - VLAN Tag in Rx status is enabled
23 —	Reserved
22-21 EVLS	Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. 00b - Do not strip 01b - Strip if VLAN filter passes 10b - Strip if VLAN filter fails 11b - Always strip
20-19 —	Reserved
18 ESVL	Enable S-VLAN When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets. 0b - S-VLAN is disabled 1b - S-VLAN is enabled
17 VTIM	VLAN Tag Inverse Match Enable When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. 0b - VLAN Tag Inverse Match is disabled 1b - VLAN Tag Inverse Match is enabled
16-7 —	Reserved
6-2 OFS	Offset This field holds the address offset of the MAC VLAN Tag Filter Register which the application is trying to access. The width of the field depends on the number of MAC VLAN Tag Registers enabled.
1 CT	Command Type This bit indicates if the current register access is a read or a write. When set, it indicate a read operation. When reset, it indicates a write operation. 0b - Write operation 1b - Read operation
0 OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to per VLAN Tag Filter register. This bit is reset when the read or write command to per VLAN Tag Filter indirect access register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into the Per VLAN Tag register.

Field	Function
	During a read operation, the data should be read from the VLAN_TAG_DATA register only after this bit is reset. 0b - Operation Busy is disabled 1b - Operation Busy is enabled

61.7.1.9 MAC VLAN Tag Data (MAC_VLAN_TAG_DATA)

61.7.1.9.1 Offset

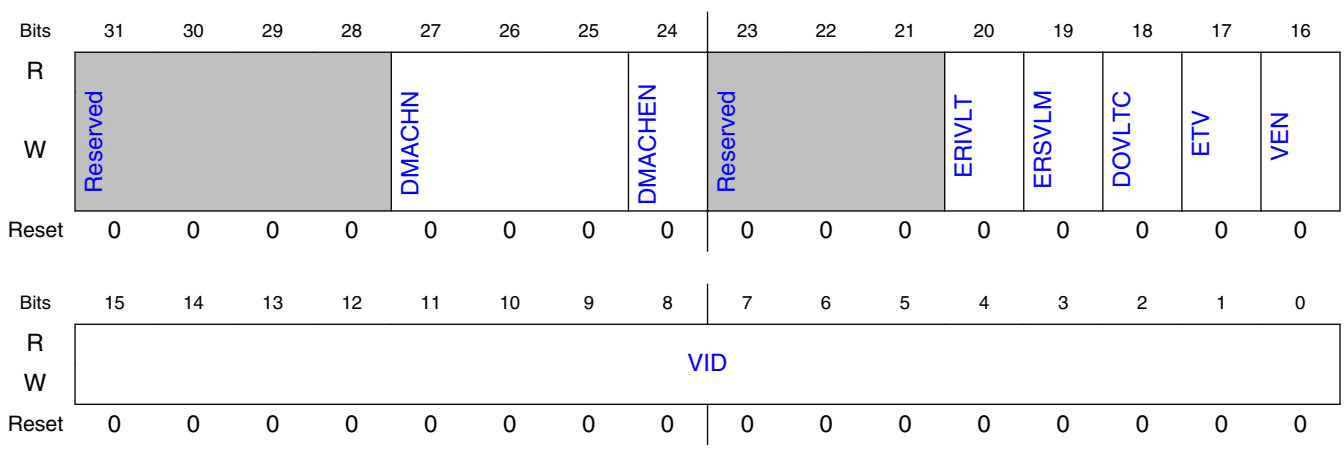
Register	Offset
MAC_VLAN_TAG_DATA	54h

61.7.1.9.2 Function

This register holds the read/write data for Indirect Access of the Per VLAN Tag registers. During the read access, this field contains valid read data only after the OB bit is reset.

During the write access, this field should be valid prior to setting the OB bit in the MAC_VLAN_Tag_Ctrl Register.

61.7.1.9.3 Diagram



61.7.1.9.4 Fields

Field	Function
31-28 —	Reserved
27-25 DMACHN	DMA Channel Number The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field. If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.
24 DMACHEN	DMA Channel Number Enable This bit is the Enable for the DMA Channel Number value programmed in the field DMACH. When this bit is reset, the Routing does not occur based on VLAN Filter result. The frame is routed based on DA Based DMA Channel Routing. 0b - DMA Channel Number is disabled 1b - DMA Channel Number is enabled
23-21 —	Reserved
20 ERIVLT	Enable Inner VLAN Tag Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit and the EDVLP field are set, the MAC receiver enables operation on the inner VLAN Tag (if present). When this bit is reset, the MAC receiver enables operation on the outer VLAN Tag (if present). 0b - Inner VLAN tag comparison is disabled 1b - Inner VLAN tag comparison is enabled
19 ERSVLM	Enable S-VLAN Match for received Frames This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. 0b - Receive S-VLAN Match is disabled 1b - Receive S-VLAN Match is enabled
18 DOVLTC	Disable VLAN Type Comparison This bit is valid only when VLAN Tag Enable of the Filter is set. When this bit is set, the MAC does not check whether the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit is of type S-VLAN or C-VLAN. When this bit is reset, the MAC filters or matches the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit only when VLAN Tag type is similar to the one specified by the Enable S-VLAN Match for received Frames bit. 0b - VLAN type comparison is enabled 1b - VLAN type comparison is disabled
17 ETV	12bits or 16bits VLAN comparison This bit is valid only when VEN of the Filter is set.

Table continues on the next page...

Field	Function
	When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet. 0b - 16 bit VLAN comparison 1b - 12 bit VLAN comparison
16 VEN	VLAN Tag Enable This bit is used to enable or disable the VLAN Tag. When this bit is set, the MAC compares the VLAN Tag of received packet with the VLAN Tag ID. When this bit is reset, no comparison is performed irrespective of the programming of the other fields. 0b - VLAN Tag is disabled 1b - VLAN Tag is enabled
15-0 VID	VLAN Tag ID This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.

61.7.1.10 MAC VLAN Hash Table (MAC_VLAN_HASH_TABLE)

61.7.1.10.1 Offset

Register	Offset
MAC_VLAN_HASH_TABLE	58h

61.7.1.10.2 Function

When VTHM bit of the MAC_VLAN_TAG register is set, the 16-bit VLAN Hash Table register is used for group address filtering based on the VLAN tag. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_TAG Register) in the incoming packet is passed through the CRC logic. When ETV bit of MAC_VLAN_TAG register is set, the upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. When ETV bit of MAC_VLAN_TAG register is reset, the ones-complement of upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. For example, when ETV bit is set a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table, whereas when ETV bit is reset a hash value of 4b'1000 selects Bit 7 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

- Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3).

- Perform bitwise reversal for the value obtained in step 1.

- Take the upper four bits from the value obtained in step 2.

If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[15:8] (in little-endian mode) or Bits[7:0] (in big-endian mode) of this register are written.

- If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VLHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.10.4 Fields

Field	Function
31-16 —	Reserved
15-0 VLHT	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.

61.7.1.11 VLAN Tag Inclusion or Replacement (MAC_VLAN_INCL)

61.7.1.11.1 Offset

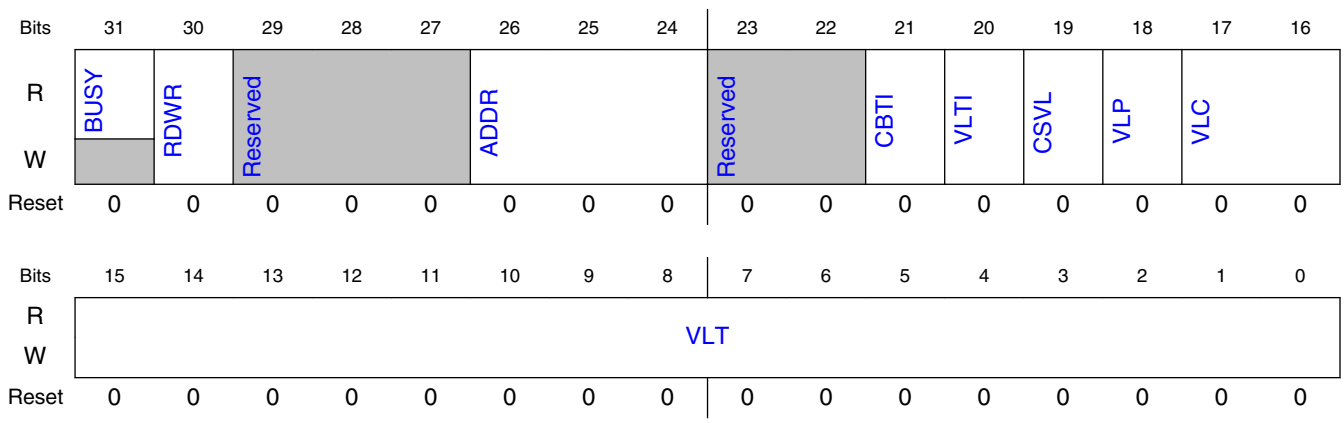
Register	Offset
MAC_VLAN_INCL	60h

61.7.1.11.2 Function

The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets.

The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets. It also contains the VLAN tag insertion controls.

61.7.1.11.3 Diagram



61.7.1.11.4 Fields

Field	Function
31 BUSY	Busy This bit indicates the status of the read/write operation of indirect access to the queue/channel specific VLAN inclusion register. For write operation write to a register is complete when this bit is reset. For read operation the read data is valid when the bit is reset. The application must make sure that this bit is reset before attempting subsequent access to this register. 0b - Busy status not detected 1b - Busy status detected
30 RDWR	Read write control This bit controls the read or write operation for indirectly accessing the queue/channel specific VLAN Inclusion register. When set indicates write operation and when reset indicates read operation. This does not have any effect when CBTI is reset. 0b - Read operation of indirect access 1b - Write operation of indirect access
29-27 —	Reserved

Table continues on the next page...

Field	Function
26-24 ADDR	Address This field selects one of the queue/channel specific VLAN Inclusion register for read/write access. This does not have any effect when CBTI is reset.
23-22 —	Reserved
21 CBTI	Channel based tag insertion When this bit is set, outer VLAN tag is inserted for every packets transmitted by the MAC. The tag value is taken from the queue/channel specific VLAN tag register. The VLTI, VLP, VLC, and VLT fields of this register are ignored when this bit is set. When this bit is set, a write operation to byte 3 of this register initiates the read/write access to the indirect register. When reset, outer VLAN operation is based on the setting of VLTI, VLP, VLC and VLT fields of this register. 0b - Channel based tag insertion is disabled 1b - Channel based tag insertion is enabled
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0b - VLAN Tag Input is disabled 1b - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0b - C-VLAN type (0x8100) is inserted or replaced 1b - S-VLAN type (0x88A8) is inserted or replaced
18 VLP	VLAN Priority Control When this bit is set, the control bits[17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and bits[17:16] are ignored. 0b - VLAN Priority Control is disabled 1b - VLAN Priority Control is enabled
17-16 VLC	VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 13 and 14. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted packets (Bytes 13 and 14 are 0x8100 or 0x88a8).

Table continues on the next page...

Field	Function
	Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00b - No VLAN tag deletion, insertion, or replacement 01b - VLAN tag deletion 10b - VLAN tag insertion 11b - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

61.7.1.12 MAC Inner VLAN Tag Inclusion or Replacement (MAC_INNER_VLAN_INCL)

61.7.1.12.1 Offset

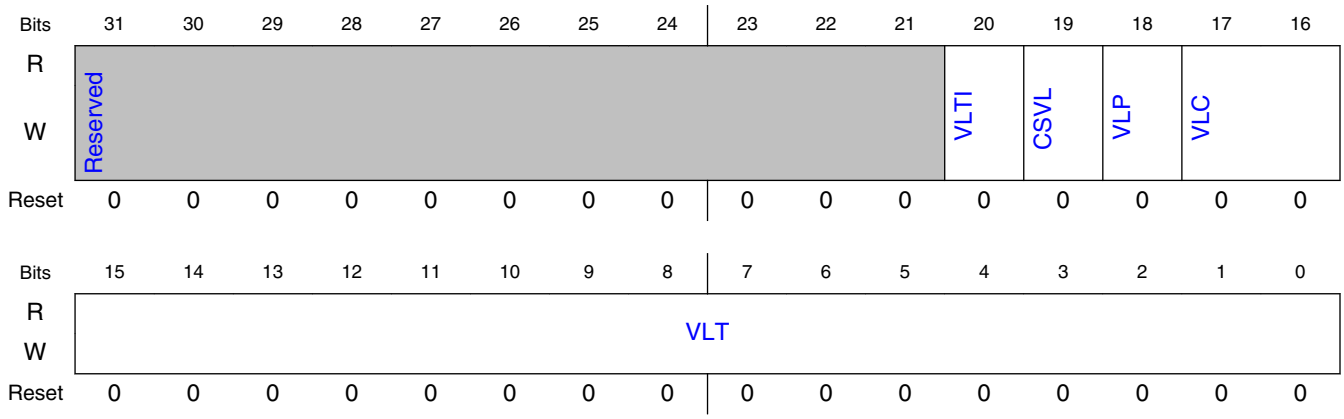
Register	Offset
MAC_INNER_VLAN_INCL	64h

61.7.1.12.2 Function

The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet.

The Inner VLAN Tag Inclusion or Replacement register contains the inner VLAN tag to be inserted or replaced in the Transmit packet. It also contains the inner VLAN tag insertion controls.

61.7.1.12.3 Diagram



61.7.1.12.4 Fields

Field	Function
31-21 —	Reserved
20 VLTI	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: - The Tx descriptor 0b - VLAN Tag Input is disabled 1b - VLAN Tag Input is enabled
19 CSVL	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. 0b - C-VLAN type (0x8100) is inserted 1b - S-VLAN type (0x88A8) is inserted
18 VLP	VLAN Priority Control When this bit is set, the VLC field is used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and the VLC field is ignored. 0b - VLAN Priority Control is disabled 1b - VLAN Priority Control is enabled
17-16 VLC	VLAN Tag Control in Transmit Packets - 2'b00: No VLAN tag deletion, insertion, or replacement - 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 17 and 18) and VLAN tag (bytes 19 and 20) of all transmitted packets with VLAN tags. - 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 19 and 20 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 17 and 18. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. - 2'b11: VLAN tag replacement

Table continues on the next page...

Field	Function
	The MAC replaces VLT in bytes 19 and 20 of all VLAN-type transmitted packets (Bytes 17 and 18 are 0x8100 or 0x88a8). Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value. 00b - No VLAN tag deletion, insertion, or replacement 01b - VLAN tag deletion 10b - VLAN tag insertion 11b - VLAN tag replacement
15-0 VLT	VLAN Tag for Transmit Packets This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase. Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag. The following list describes the bits of this field: - Bits[15:13]: User Priority - Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) - Bits[11:0]: VLAN Identifier (VID) field of VLAN tag

61.7.1.13 MAC Q0 Tx Flow Control (MAC_Q0_TX_FLOW_CTRL)

61.7.1.13.1 Offset

Register	Offset
MAC_Q0_TX_FLOW_CTRL	70h

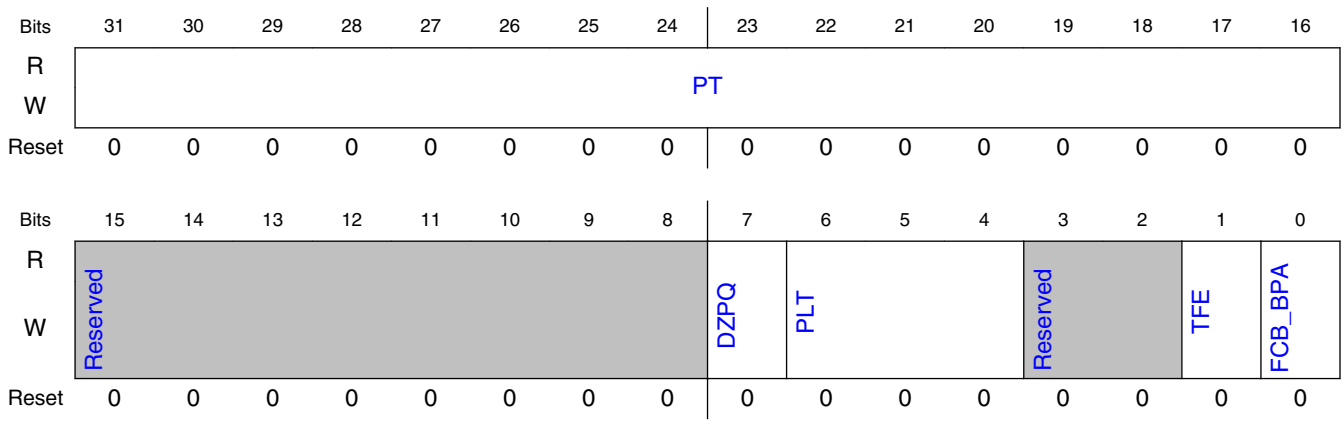
61.7.1.13.2 Function

The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC.

The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC. A Write to a register with the Busy bit set to 1 triggers the Flow Control block to generate a Pause packet. The fields of the control packet are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control packet. The Busy bit remains set until the control packet is transferred onto the cable. The application must make sure that the Busy bit is cleared before writing to the register.

When the PFCE bit in the MAC_Q0_RX_FLOW_CTRL register is enabled, this register controls the generation of Priority Flow Control (PFC) frames with priorities mapped according to PSRQ0 in the MAC_Q0_RXQ_CTRL2 register.

61.7.1.13.3 Diagram



61.7.1.13.4 Fields

Field	Function
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0b - Zero-Quanta Pause packet generation is enabled 1b - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.

Table continues on the next page...

Field	Function
	This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000b - Pause Time minus 4 Slot Times (PT -4 slot times) 001b - Pause Time minus 28 Slot Times (PT -28 slot times) 010b - Pause Time minus 36 Slot Times (PT -36 slot times) 011b - Pause Time minus 144 Slot Times (PT -144 slot times) 100b - Pause Time minus 256 Slot Times (PT -256 slot times) 101b - Pause Time minus 512 Slot Times (PT -512 slot times) 110b - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable Full-Duplex Mode: In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. Half-Duplex Mode: In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled. 0b - Transmit Flow Control is disabled 1b - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. Full-Duplex Mode: In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Half-Duplex Mode: When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flow Control Busy or Backpressure Activate is disabled 1b - Flow Control Busy or Backpressure Activate is enabled

61.7.1.14 MAC Q1 Tx Flow Control (MAC_Q1_TX_FLOW_CTRL)

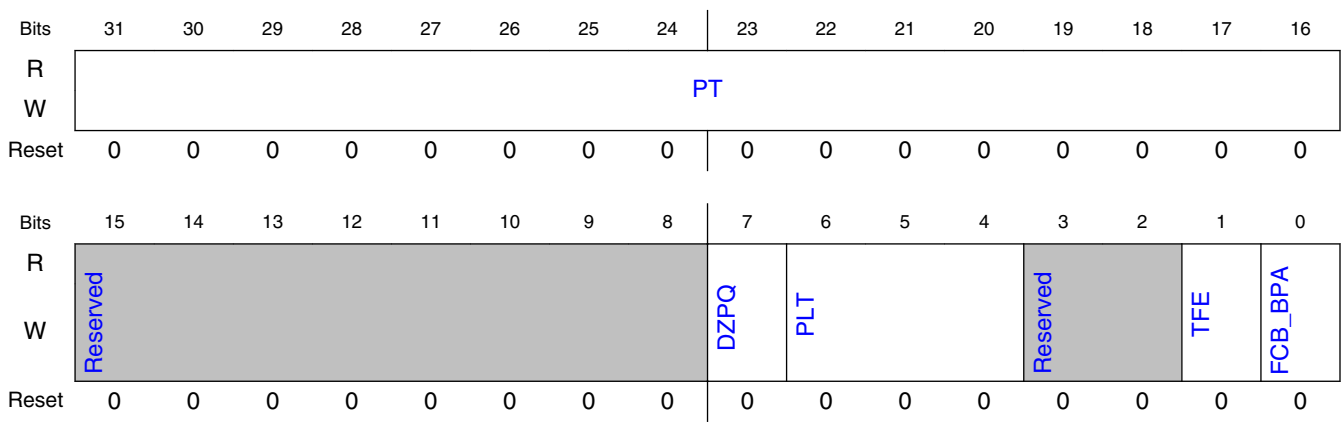
61.7.1.14.1 Offset

Register	Offset
MAC_Q1_TX_FLOW_CTRL	74h

61.7.1.14.2 Function

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q1_RXQ_CTRL2/MAC_Q1_RXQ_CTRL3 registers.

61.7.1.14.3 Diagram



61.7.1.14.4 Fields

Field	Function
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0b - Zero-Quanta Pause packet generation is enabled 1b - Zero-Quanta Pause packet generation is disabled
6-4	Pause Low Threshold

Table continues on the next page...

Field	Function
PLT	This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000b - Pause Time minus 4 Slot Times (PT -4 slot times) 001b - Pause Time minus 28 Slot Times (PT -28 slot times) 010b - Pause Time minus 36 Slot Times (PT -36 slot times) 011b - Pause Time minus 144 Slot Times (PT -144 slot times) 100b - Pause Time minus 256 Slot Times (PT -256 slot times) 101b - Pause Time minus 512 Slot Times (PT -512 slot times) 110b - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0b - Transmit Flow Control is disabled 1b - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flow Control Busy or Backpressure Activate is disabled 1b - Flow Control Busy or Backpressure Activate is enabled

61.7.1.15 MAC Q2 Tx Flow Control (MAC_Q2_TX_FLOW_CTRL)

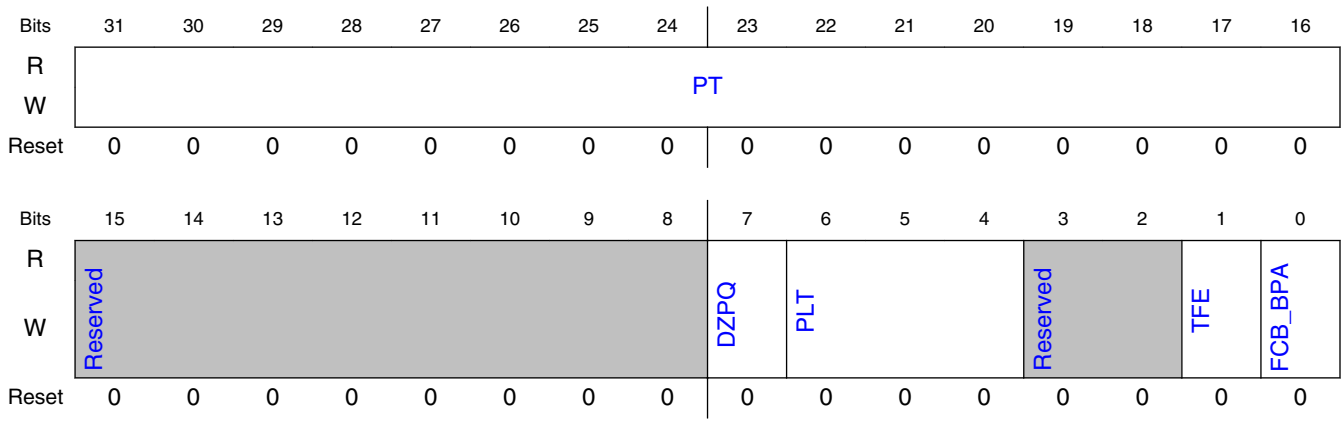
61.7.1.15.1 Offset

Register	Offset
MAC_Q2_TX_FLOW_CTRL	78h

61.7.1.15.2 Function

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q2_RXQ_CTRL2/MAC_Q2_RXQ_CTRL3 registers.

61.7.1.15.3 Diagram



61.7.1.15.4 Fields

Field	Function
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0b - Zero-Quanta Pause packet generation is enabled 1b - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.

Table continues on the next page...

Field	Function
	This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000b - Pause Time minus 4 Slot Times (PT -4 slot times) 001b - Pause Time minus 28 Slot Times (PT -28 slot times) 010b - Pause Time minus 36 Slot Times (PT -36 slot times) 011b - Pause Time minus 144 Slot Times (PT -144 slot times) 100b - Pause Time minus 256 Slot Times (PT -256 slot times) 101b - Pause Time minus 512 Slot Times (PT -512 slot times) 110b - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0b - Transmit Flow Control is disabled 1b - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flow Control Busy or Backpressure Activate is disabled 1b - Flow Control Busy or Backpressure Activate is enabled

61.7.1.16 MAC Q3 Tx Flow Control (MAC_Q3_TX_FLOW_CTRL)

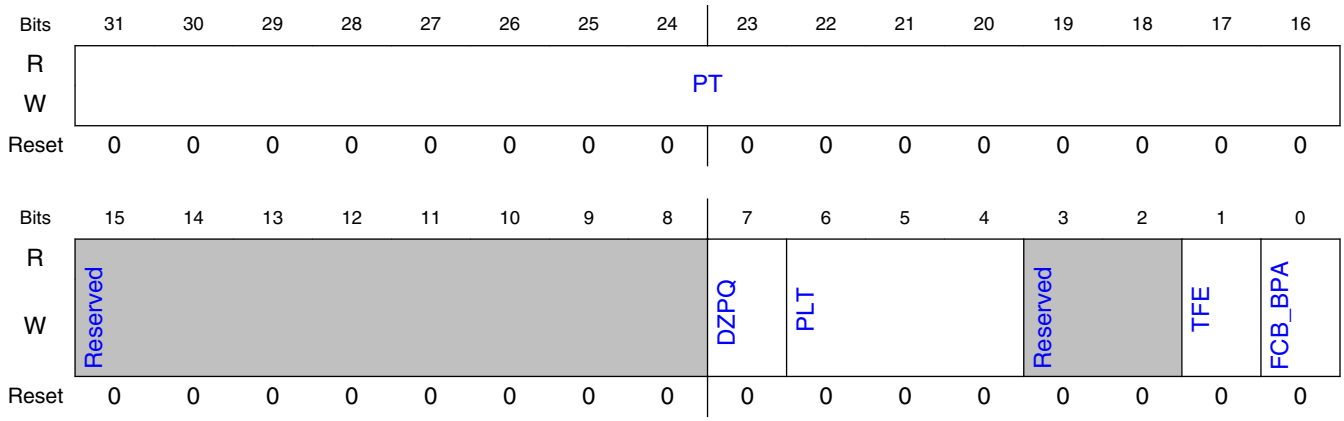
61.7.1.16.1 Offset

Register	Offset
MAC_Q3_TX_FLOW_CTRL	7Ch

61.7.1.16.2 Function

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q3_RXQ_CTRL2/MAC_Q3_RXQ_CTRL3 registers.

61.7.1.16.3 Diagram



61.7.1.16.4 Fields

Field	Function
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0b - Zero-Quanta Pause packet generation is enabled 1b - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000b - Pause Time minus 4 Slot Times (PT -4 slot times) 001b - Pause Time minus 28 Slot Times (PT -28 slot times) 010b - Pause Time minus 36 Slot Times (PT -36 slot times) 011b - Pause Time minus 144 Slot Times (PT -144 slot times) 100b - Pause Time minus 256 Slot Times (PT -256 slot times)

Table continues on the next page...

Field	Function
	101b - Pause Time minus 512 Slot Times (PT -512 slot times) 110b - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0b - Transmit Flow Control is disabled 1b - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flow Control Busy or Backpressure Activate is disabled 1b - Flow Control Busy or Backpressure Activate is enabled

61.7.1.17 MAC Q4 Tx Flow Control (MAC_Q4_TX_FLOW_CTRL)

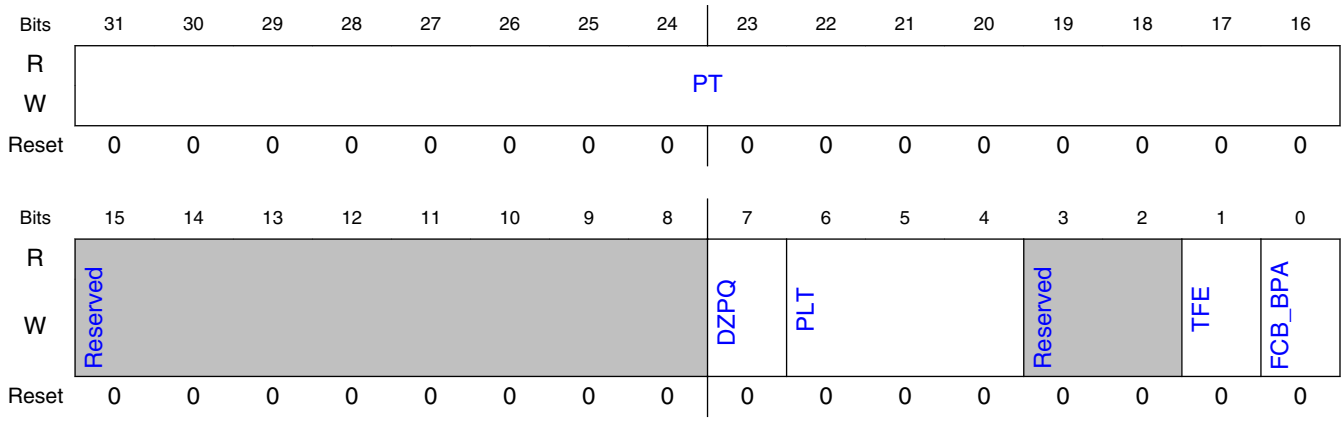
61.7.1.17.1 Offset

Register	Offset
MAC_Q4_TX_FLOW_CTRL	80h

61.7.1.17.2 Function

This register controls the generation of PFC Control packets of priorities mapped as per the PSRQi field in the MAC_Q4_RXQ_CTRL2/MAC_Q4_RXQ_CTRL3 registers.

61.7.1.17.3 Diagram



61.7.1.17.4 Fields

Field	Function
31-16 PT	Pause Time This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15-8 —	Reserved
7 DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. 0b - Zero-Quanta Pause packet generation is enabled 1b - Zero-Quanta Pause packet generation is disabled
6-4 PLT	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted. The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface. This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times. 000b - Pause Time minus 4 Slot Times (PT -4 slot times) 001b - Pause Time minus 28 Slot Times (PT -28 slot times) 010b - Pause Time minus 36 Slot Times (PT -36 slot times) 011b - Pause Time minus 144 Slot Times (PT -144 slot times) 100b - Pause Time minus 256 Slot Times (PT -256 slot times)

Table continues on the next page...

Field	Function
	101b - Pause Time minus 512 Slot Times (PT -512 slot times) 110b - Reserved
3-2 —	Reserved
1 TFE	Transmit Flow Control Enable When this bit is set in full-duplex mode, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. 0b - Transmit Flow Control is disabled 1b - Transmit Flow Control is enabled
0 FCB_BPA	Flow Control Busy This bit initiates a PFC packet if the TFE bit is set. To initiate a PFC packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When PFC packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flow Control Busy or Backpressure Activate is disabled 1b - Flow Control Busy or Backpressure Activate is enabled

61.7.1.18 MAC Rx Flow Control (MAC_RX_FLOW_CTRL)

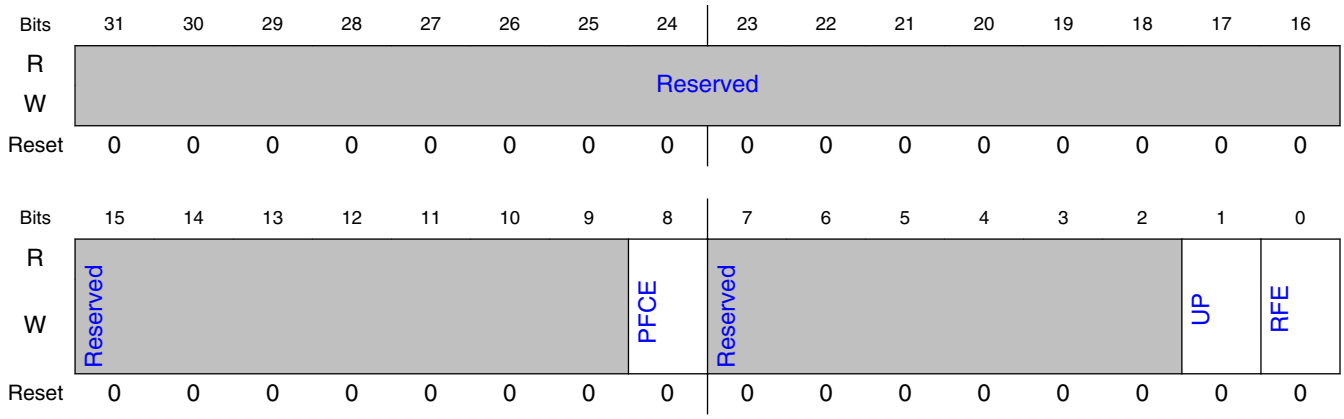
61.7.1.18.1 Offset

Register	Offset
MAC_RX_FLOW_CTRL	90h

61.7.1.18.2 Function

The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet.

61.7.1.18.3 Diagram



61.7.1.18.4 Fields

Field	Function
31-9 —	Reserved
8 PFCE	Priority Based Flow Control Enable When this bit is set, it enables generation and reception of priority-based flow control (PFC) packets. When this bit is reset, it enables generation and reception of 802.3x Pause control packets. 0b - Priority Based Flow Control is disabled 1b - Priority Based Flow Control is enabled
7-2 —	Reserved
1 UP	Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_ADDRESS0_HIGH and MAC_ADDRESS0_LOW. When this bit is reset, the MAC only detects Pause packets with unique multicast address. Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011. 0b - Unicast Pause Packet Detect disabled 1b - Unicast Pause Packet Detect enabled
0 RFE	Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time. 0b - Receive Flow Control is disabled 1b - Receive Flow Control is enabled

61.7.1.19 Receive Queue Control 4 (MAC_RXQ_CTRL4)

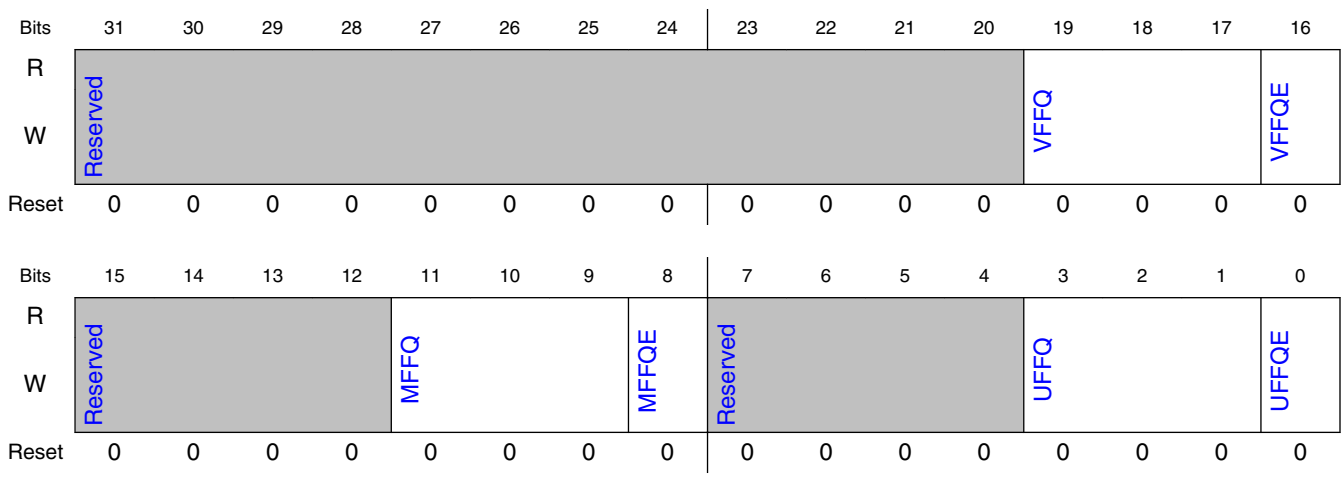
61.7.1.19.1 Offset

Register	Offset
MAC_RXQ_CTRL4	94h

61.7.1.19.2 Function

The Receive Queue Control 4 register controls the routing of unicast and multicast packets that fail the Destination or Source address filter to the Rx queues.

61.7.1.19.3 Diagram



61.7.1.19.4 Fields

Field	Function
31-20 —	Reserved
19-17 VFFQ	VLAN Tag Filter Fail Packets Queue This field holds the Rx queue number to which the tagged packets failing the Destination or Source Address filter (and UFFQE/MFFQE not enabled) or failing the VLAN tag filter must be routed to. This field is valid only when the VFFQE bit is set.
16 VFFQE	VLAN Tag Filter Fail Packets Queuing Enable

Table continues on the next page...

Field	Function
	When this bit is set, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter, are routed to the Rx Queue Number programmed in the VFFQ. When this bit is reset, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter are routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0b - VLAN tag Filter Fail Packets Queuing is disabled 1b - VLAN tag Filter Fail Packets Queuing is enabled
15-12 —	Reserved
11-9 MFFQ	Multicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Multicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the MFFQE bit is set.
8 MFFQE	Multicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Multicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the MFFQ. When this bit is reset, the Multicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0b - Multicast Address Filter Fail Packets Queuing is disabled 1b - Multicast Address Filter Fail Packets Queuing is enabled
7-4 —	Reserved
3-1 UFFQ	Unicast Address Filter Fail Packets Queue. Unicast Address Filter Fail Packets Queue. This field holds the Rx queue number to which the Unicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the UFFQE bit is set.
0 UFFQE	Unicast Address Filter Fail Packets Queuing Enable. When this bit is set, the Unicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the UFFQ. When this bit is reset, the Unicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set. 0b - Unicast Address Filter Fail Packets Queuing is disabled 1b - Unicast Address Filter Fail Packets Queuing is enabled

61.7.1.20 Transmit Queue Priority Mapping 0 (MAC_TXQ_PRTY_MAP0)

61.7.1.20.1 Offset

Register	Offset
MAC_TXQ_PRTY_MAP0	98h

61.7.1.20.2 Function

The Transmit Queue Priority Mapping 0 register contains the priority values assigned to Tx Queue 0 through Tx Queue 3.

61.7.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSTQ3								PSTQ2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSTQ1								PSTQ0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.20.4 Fields

Field	Function
31-24 PSTQ3	Priorities Selected in Transmit Queue 3 This bit is similar to the PSTQ0 bit.
23-16 PSTQ2	Priorities Selected in Transmit Queue 2 This bit is similar to the PSTQ0 bit.
15-8 PSTQ1	Priorities Selected in Transmit Queue 1 This bit is similar to the PSTQ0 bit.
7-0 PSTQ0	Priorities Selected in Transmit Queue 0 This field holds the priorities assigned to Tx Queue 0 by the software. This field determines if Tx Queue 0 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

61.7.1.21 Transmit Queue Priority Mapping 1 (MAC_TXQ_PRTY_MAP1)

61.7.1.21.1 Offset

Register	Offset
MAC_TXQ_PRTY_MAP1	9Ch

61.7.1.21.2 Function

The Transmit Queue Priority Mapping 1 register contains the priority values assigned to Tx Queue 4.

61.7.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PSTQ4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.21.4 Fields

Field	Function
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSTQ4	Priorities Selected in Transmit Queue 4 This field holds the priorities assigned to Tx Queue 4 by the software. This field determines if Tx Queue 4 should be blocked from transmitting specified pause time when a PFC packet is received with priorities matching the priorities programmed in this field. If the content of this field is not mutually exclusive to corresponding fields of other Transmit queues, that is, same priority is mapped to multiple Tx queues, the MAC blocks all queues with matching priority for specified time.

61.7.1.22 Receive Queue Control 0 (MAC_RXQ_CTRL0)

61.7.1.22.1 Offset

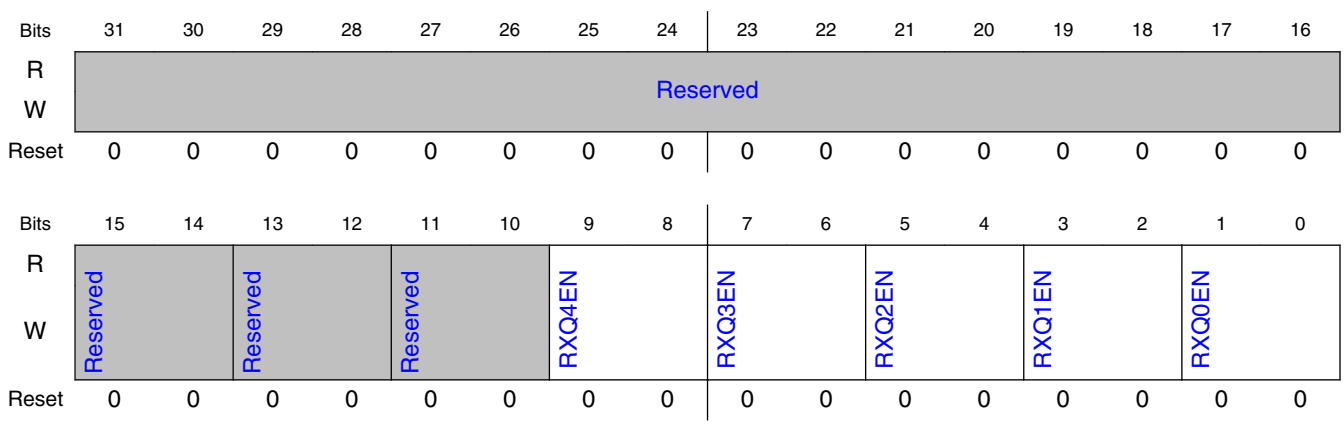
Register	Offset
MAC_RXQ_CTRL0	A0h

61.7.1.22.2 Function

The Receive Queue Control 0 register controls the queue management in the MAC Receiver.

Note: In multiple Rx queues configuration, all the queues are disabled by default. Enable the Rx queue by programming the corresponding field in this register.

61.7.1.22.3 Diagram



61.7.1.22.4 Fields

Field	Function
31-16 —	Reserved
15-14 —	Reserved
13-12	Reserved

Table continues on the next page...

Field	Function
—	
11-10 —	Reserved
9-8 RXQ4EN	Receive Queue 4 Enable This field is similar to the RXQ0EN field. 00b - Queue not enabled 01b - Queue enabled for AV 10b - Queue enabled for DCB/Generic 11b - Reserved
7-6 RXQ3EN	Receive Queue 3 Enable This field is similar to the RXQ0EN field. 00b - Queue not enabled 01b - Queue enabled for AV 10b - Queue enabled for DCB/Generic 11b - Reserved
5-4 RXQ2EN	Receive Queue 2 Enable This field is similar to the RXQ0EN field. 00b - Queue not enabled 01b - Queue enabled for AV 10b - Queue enabled for DCB/Generic 11b - Reserved
3-2 RXQ1EN	Receive Queue 1 Enable This field is similar to the RXQ0EN field. 00b - Queue not enabled 01b - Queue enabled for AV 10b - Queue enabled for DCB/Generic 11b - Reserved
1-0 RXQ0EN	Receive Queue 0 Enable This field indicates whether Rx Queue 0 is enabled for AV or DCB. 00b - Queue not enabled 01b - Queue enabled for AV 10b - Queue enabled for DCB/Generic 11b - Reserved

61.7.1.23 Receive Queue Control 1 (MAC_RXQ_CTRL1)

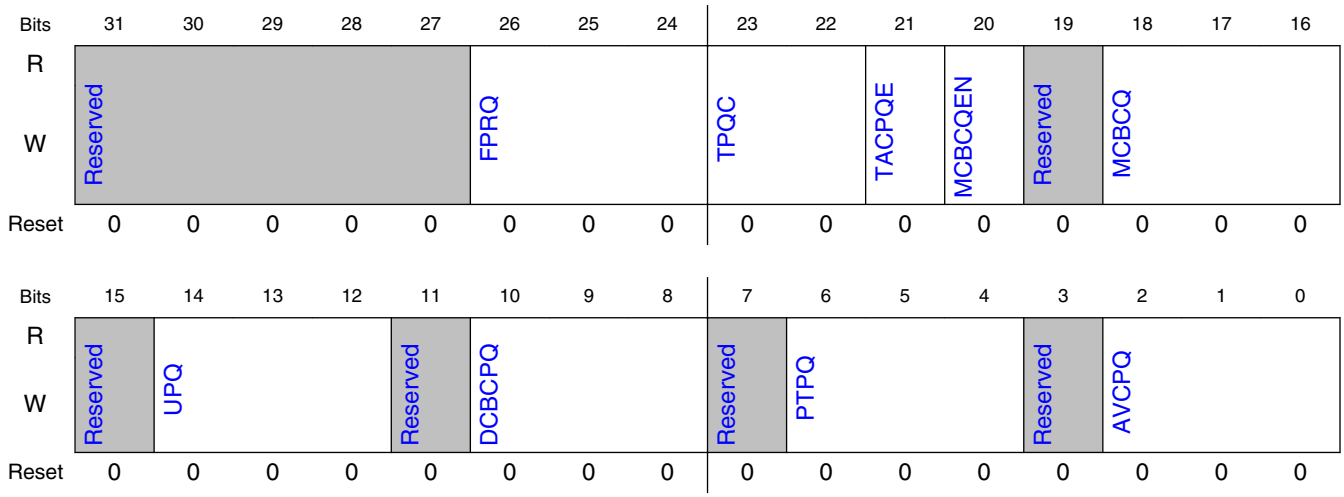
61.7.1.23.1 Offset

Register	Offset
MAC_RXQ_CTRL1	A4h

61.7.1.23.2 Function

The Receive Queue Control 1 register controls the routing of multicast, broadcast, AV, DCB, and untagged packets to the Rx queues.

61.7.1.23.3 Diagram



61.7.1.23.4 Fields

Field	Function
31-27 —	Reserved
26-24 FPRQ	Frame Preemption Residue Queue This field holds the Rx queue number to which the residual preemption frames must be forwarded. Preemption frames that are tagged and pass the SA/DA/VLAN filtering are routed based on PSRQ and all other frames are treated as residual frames and is routed to the queue number mentioned in this field. The Queue-0 is used as a default queue for express frames, so this field cannot be programmed to a value 0.
23-22 TPQC	Tagged PTP over Ethernet Packets Queuing Control. This field controls the routing of the VLAN Tagged PTPoE packets. If DWC_EQOS_AV_ENABLE is selected in the configuration, the following programmable options are allowed. - 2'b00: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for only non-AV enabled Rx Queues). - 2'b01: VLAN Tagged PTPoE packets are routed to Rx Queue specified by PTPQ field (That Rx Queue can be enabled for AV or non-AV traffic). - 2'b10: VLAN Tagged PTPoE packets are routed to only AV enabled Rx Queues based on PSRQ. - 2'b11: Reserved If DWC_EQOS_AV_ENABLE is not selected in the configuration, the following programmable options are allowed. - 1'b0: VLAN Tagged PTPoE packets are routed as generic VLAN Tagged packet (based on PSRQ for DCB/Generic enabled Rx Queues). - 1'b1: VLAN Tagged PTPoE packets are routed to Rx Queues specified by PTPQ field.
21 TACPQE	Tagged AV Control Packets Queuing Enable.

Table continues on the next page...

Field	Function
	When set, the MAC routes the received Tagged AV Control packets to the Rx queue specified by AVCPQ field. When reset, the MAC routes the received Tagged AV Control packets based on the tag priority matching the PSRQ fields in MAC_RXQ_CTRL2 and MAC_RXQ_CTRL3 registers. 0b - Tagged AV Control Packets Queuing is disabled 1b - Tagged AV Control Packets Queuing is enabled
20 MCBCQEN	Multicast and Broadcast Queue Enable This bit specifies that Multicast or Broadcast packets routing to the Rx Queue is enabled and the Multicast or Broadcast packets must be routed to Rx Queue specified in MCBCQ field. 0b - Multicast and Broadcast Queue is disabled 1b - Multicast and Broadcast Queue is enabled
19 —	Reserved
18-16 MCBCQ	Multicast and Broadcast Queue This field specifies the Rx Queue onto which Multicast or Broadcast Packets are routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Multicast or Broadcast Packets. 000b - Receive Queue 0 001b - Receive Queue 1 010b - Receive Queue 2 011b - Receive Queue 3 100b - Receive Queue 4 101b - Reserved 110b - Reserved 111b - Reserved
15 —	Reserved
14-12 UPQ	Untagged Packet Queue This field indicates the Rx Queue to which Untagged Packets are to be routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Untagged Packets. 000b - Receive Queue 0 001b - Receive Queue 1 010b - Receive Queue 2 011b - Receive Queue 3 100b - Receive Queue 4 101b - Reserved 110b - Reserved 111b - Reserved
11 —	Reserved
10-8 DCBCPQ	DCB Control Packets Queue This field specifies the Rx queue on which the received DCB control packets are routed. The DCB data packets are routed based on the PSRQ field of the Transmit Flow Control Register of corresponding queue. 000b - Receive Queue 0 001b - Receive Queue 1 010b - Receive Queue 2 011b - Receive Queue 3 100b - Receive Queue 4

Table continues on the next page...

Field	Function
	101b - Reserved 110b - Reserved 111b - Reserved
7 —	Reserved
6-4 PTPQ	PTP Packets Queue This field specifies the Rx queue on which the PTP packets sent over the Ethernet payload (not over IPv4 or IPv6) are routed. When the AV8021ASMEN bit of MAC_TIMESTAMP_CONTROL register is set, only untagged PTP over Ethernet packets are routed on an Rx Queue. If the bit is not set, then based on programming of TPQC field, both tagged and untagged PTPoE packets can be routed to this Rx Queue. 000b - Receive Queue 0 001b - Receive Queue 1 010b - Receive Queue 2 011b - Receive Queue 3 100b - Receive Queue 4 101b - Reserved 110b - Reserved 111b - Reserved
3 —	Reserved
2-0 AVCPQ	AV Untagged Control Packets Queue This field specifies the Receive queue on which the received AV tagged and untagged control packets are routed. The AV tagged (when TACPQE bit is set) and untagged control packets are routed to Receive queue specified by this field. 000b - Receive Queue 0 001b - Receive Queue 1 010b - Receive Queue 2 011b - Receive Queue 3 100b - Receive Queue 4 101b - Reserved 110b - Reserved 111b - Reserved

61.7.1.24 Receive Queue Control 2 (MAC_RXQ_CTRL2)

61.7.1.24.1 Offset

Register	Offset
MAC_RXQ_CTRL2	A8h

61.7.1.24.2 Function

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the RxQueues 0 to 3.

61.7.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PSRQ3								PSRQ2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSRQ1								PSRQ0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.24.4 Fields

Field	Function
31-24 PSRQ3	Priorities Selected in the Receive Queue 3 This field decides the priorities assigned to Rx Queue 3. All packets with priorities that match the values set in this field are routed to Rx Queue 3. For example, if PSRQ3[6, 3] are set, packets with USP field equal to 3 or 6 are routed to Rx Queue 3. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 3 crosses the flow control threshold settings.
23-16 PSRQ2	Priorities Selected in the Receive Queue 2 This field decides the priorities assigned to Rx Queue 2. All packets with priorities that match the values set in this field are routed to Rx Queue 2. For example, if PSRQ2[1, 0] are set, packets with USP field equal to 1 or 0 are routed to Rx Queue 2. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 2 crosses the flow control threshold settings.
15-8 PSRQ1	Priorities Selected in the Receive Queue 1 This field decides the priorities assigned to Rx Queue 1. All packets with priorities that match the values set in this field are routed to Rx Queue 1. For example, if PSRQ1[4] is set, packets with USP field equal to 4 are routed to Rx Queue 1. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.

Table continues on the next page...

Field	Function
	this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 1 crosses the flow control threshold settings.
7-0 PSRQ0	Priorities Selected in the Receive Queue 0 This field decides the priorities assigned to Rx Queue 0. All packets with priorities that match the values set in this field are routed to Rx Queue 0. For example, if PSRQ0[5] is set, packets with USP field equal to 5 are routed to Rx Queue 0. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. this field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 0 crosses the flow control threshold settings.

61.7.1.25 Receive Queue Control 3 (MAC_RXQ_CTRL3)

61.7.1.25.1 Offset

Register	Offset
MAC_RXQ_CTRL3	ACh

61.7.1.25.2 Function

This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to Rx Queue 4.

61.7.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PSRQ4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.25.4 Fields

Field	Function
31-24 —	Reserved
23-16 —	Reserved
15-8 —	Reserved
7-0 PSRQ4	Priorities Selected in the Receive Queue 4 This field decides the priorities assigned to Rx Queue 4. All packets with priorities that match the values set in this field are routed to Rx Queue 4. For example, if PSRQ4[7:4] is set, packets with USP field equal to 7, 6, 5, or 4 are routed to Rx Queue 4. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues. This field also determines the priorities to be included in the PFC packet sent to remote station when Rx Queue 4 crosses the flow control threshold settings.

61.7.1.26 Interrupt Status (MAC_INTERRUPT_STATUS)

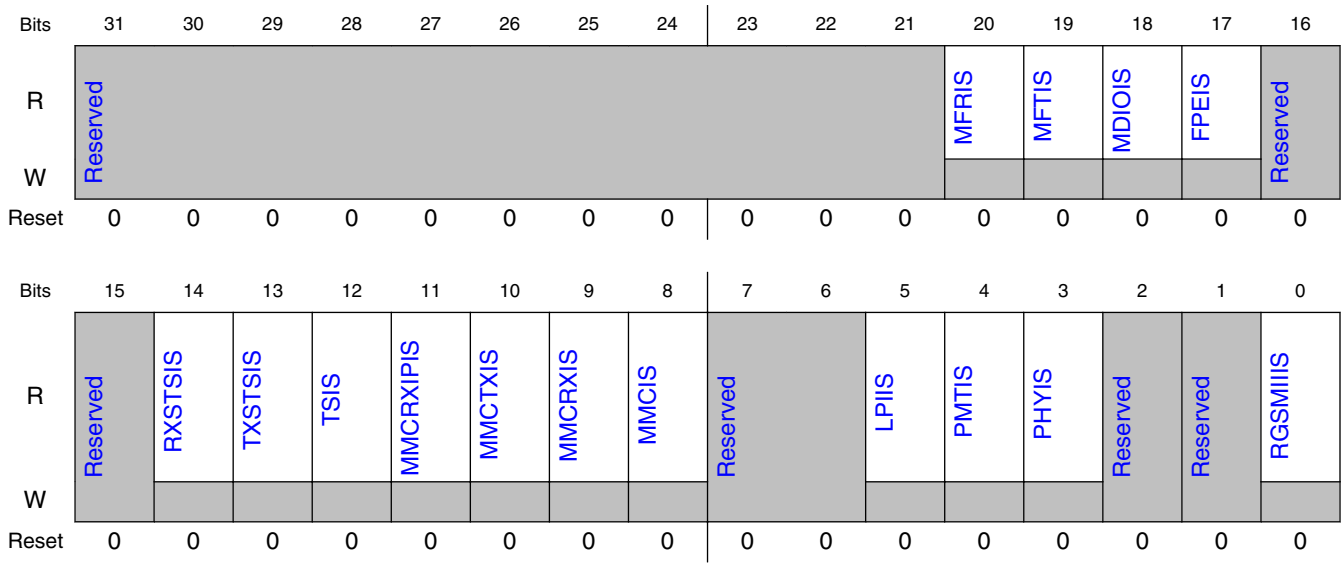
61.7.1.26.1 Offset

Register	Offset
MAC_INTERRUPT_STATUS	B0h

61.7.1.26.2 Function

The Interrupt Status register contains the status of interrupts.

61.7.1.26.3 Diagram



61.7.1.26.4 Fields

Field	Function
31-21 —	Reserved
20 MFRIS	MMC FPE Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0b - MMC FPE Receive Interrupt status not active 1b - MMC FPE Receive Interrupt status active
19 MFTIS	MMC FPE Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC FPE Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option along with FPE support. 0b - MMC FPE Transmit Interrupt status not active 1b - MMC FPE Transmit Interrupt status active
18 MDIOIS	MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit of CSR_SW_CTRL register is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - MDIO Interrupt status not active 1b - MDIO Interrupt status active
17 FPEIS	Frame Preemption Interrupt Status

Table continues on the next page...

Field	Function
	This bit indicates an interrupt event during the operation of Frame Preemption (Bits[19:16] of MAC_FPE_CTRL_STS register is set). To reset this bit, the application must clear the event in MAC_FPE_CTRL_STS that has caused the Interrupt. 0b - Frame Preemption Interrupt status not active 1b - Frame Preemption Interrupt status active
16 —	Reserved
15 —	Reserved
14 RXSTSIS	Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_RX_TX_STATUS register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0b - Receive Interrupt status not active 1b - Receive Interrupt status active
13 TXSTSIS	Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_RX_TX_STATUS register: - Excessive Collision (EXCOL) - Late Collision (LCOL) - Excessive Deferral (EXDEF) - Loss of Carrier (LCARR) - No Carrier (NCARR) - Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_RX_TX_STATUS register. 0b - Transmit Interrupt status not active 1b - Transmit Interrupt status active
12 TSIS	Timestamp Interrupt Status If the Timestamp feature is enabled, this bit is set when any of the following conditions is true: - The system time value is equal to or exceeds the value specified in the Target Time High and Low registers. - There is an overflow in the Seconds register. - The Target Time Error occurred, that is, programmed target time already elapsed. If the Auxiliary Snapshot feature is enabled, this bit is set when the auxiliary snapshot trigger is asserted. In configurations other than EQOS_CORE, when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TXTIMESTAMP_STATUS_SECONDS registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets.

Table continues on the next page...

Field	Function
	This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set) in the MAC_TIMESTAMP_STATUS register. 0b - Timestamp Interrupt status not active 1b - Timestamp Interrupt status active
11 MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) and Enable Receive TCP/IP Checksum Check options. 0b - MMC Receive Checksum Offload Interrupt status not active 1b - MMC Receive Checksum Offload Interrupt status active
10 MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0b - MMC Transmit Interrupt status not active 1b - MMC Transmit Interrupt status active
9 MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0b - MMC Receive Interrupt status not active 1b - MMC Receive Interrupt status active
8 MMCIS	MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. 0b - MMC Interrupt status not active 1b - MMC Interrupt status active
7-6 —	Reserved
5 LPIIS	LPI Interrupt Status When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit in the MAC Transmitter or Receiver. This bit is cleared when the corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is read (or corresponding interrupt source bit of MAC_LPI_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). 0b - LPI Interrupt status not active 1b - LPI Interrupt status active
4 PMTIS	PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_CONTROL_STATUS register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_CONTROL_STATUS register (or corresponding interrupt source bit of MAC_PMT_CONTROL_STATUS register is written to 1 when RCWE bit of CSR_SW_CTRL register is set).

Table continues on the next page...

Field	Function
	This bit is valid only when you select the Enable Power Management option. 0b - PMT Interrupt status not active 1b - PMT Interrupt status active
3 PHYIS	PHY Interrupt This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of CSR_SW_CTRL register is set). 0b - PHY Interrupt not detected 1b - PHY Interrupt detected
2 —	Reserved
1 —	Reserved
0 RGSMIIIS	RGMII or SMII Interrupt Status This bit is set because of any change in value of the Link Status of RGMII or SMII interface (LNKSTS bit in MAC_PHYIF_CONTROL_STATUS register). This bit is cleared when the PHYIF_CONTROL_STATUS register is read (or LNKSTS bit of MAC_PHYIF_CONTROL_STATUS register is written to 1 when RCWE bit of MAC_CSR_SW_CTRL register is set). This bit is valid only when you select the optional RGMII or SMII PHY interface. 0b - RGMII or SMII Interrupt Status is not active 1b - RGMII or SMII Interrupt Status is active

61.7.1.27 Interrupt Enable (MAC_INTERRUPT_ENABLE)

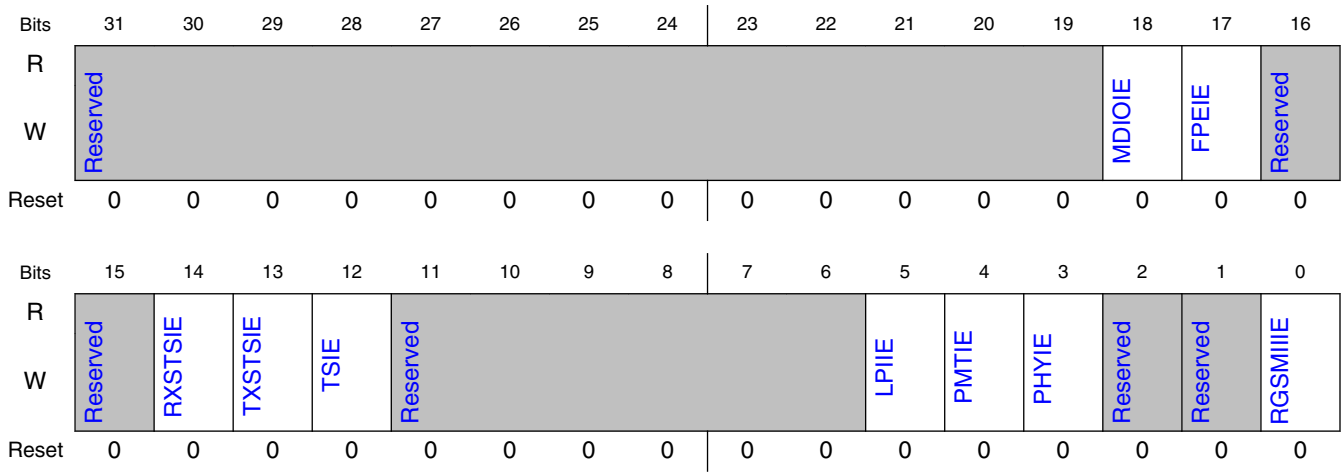
61.7.1.27.1 Offset

Register	Offset
MAC_INTERRUPT_ENABLE	B4h

61.7.1.27.2 Function

The Interrupt Enable register contains the masks for generating the interrupts.

61.7.1.27.3 Diagram



61.7.1.27.4 Fields

Field	Function
31-19 —	Reserved
18 MDIOIE	MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_INTERRUPT_STATUS register. 0b - MDIO Interrupt is disabled 1b - MDIO Interrupt is enabled
17 FPEIE	Frame Preemption Interrupt Enable When this bit is set, it enables the assertion of the interrupt when FPEIS field is set in the MAC_INTERRUPT_STATUS. 0b - Frame Preemption Interrupt is disabled 1b - Frame Preemption Interrupt is enabled
16 —	Reserved
15 —	Reserved
14 RXSTSIE	Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[RXSTSIS]. 0b - Receive Status Interrupt is disabled 1b - Receive Status Interrupt is enabled
13 TXSTSIE	Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TXSTSIS]. 0b - Timestamp Status Interrupt is disabled 1b - Timestamp Status Interrupt is enabled
12 TSIE	Timestamp Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[TSIS]. 0b - Timestamp Interrupt is disabled 1b - Timestamp Interrupt is enabled
11-6 —	Reserved

Table continues on the next page...

Field	Function
5 LPIIE	LPI Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[LPIIS]. 0b - LPI Interrupt is disabled 1b - LPI Interrupt is enabled
4 PMTIE	PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PMTIS]. 0b - PMT Interrupt is disabled 1b - PMT Interrupt is enabled
3 PHYIE	PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of MAC_INTERRUPT_STATUS[PHYIS]. 0b - PHY Interrupt is disabled 1b - PHY Interrupt is enabled
2 —	Reserved
1 —	Reserved
0 RGSMIIIE	RGMII or SMII Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RGSMIIIS bit in MAC_INTERRUPT_STATUS register. 0b - RGMII or SMII Interrupt is disabled 1b - RGMII or SMII Interrupt is enabled

61.7.1.28 Receive Transmit Status (MAC_RX_TX_STATUS)

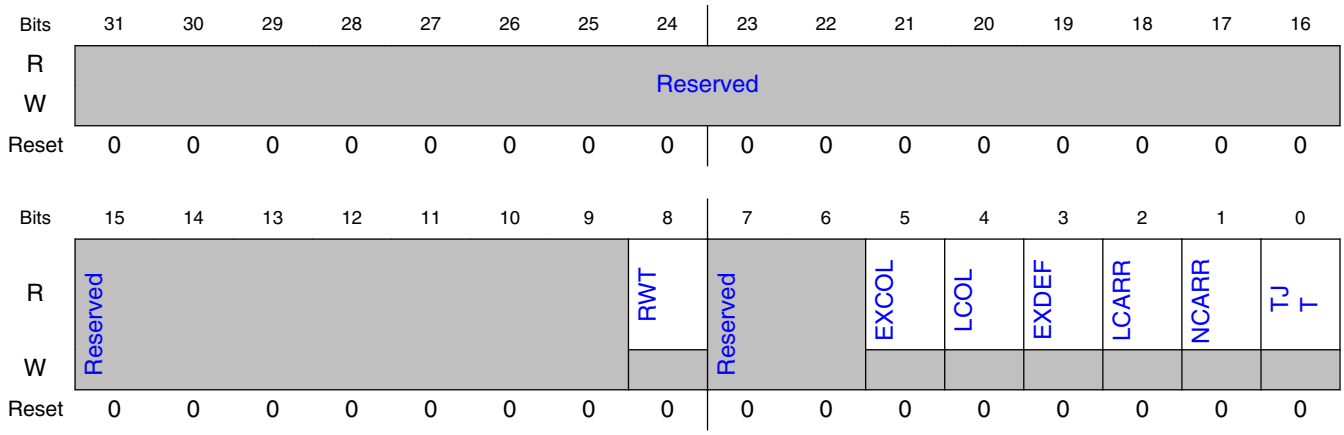
61.7.1.28.1 Offset

Register	Offset
MAC_RX_TX_STATUS	B8h

61.7.1.28.2 Function

The Receive Transmit Status register contains the Receive and Transmit Error status.

61.7.1.28.3 Diagram



61.7.1.28.4 Fields

Field	Function
31-9 —	Reserved
8 RWT	Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_CONFIGURATION register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No receive watchdog timeout 1b - Receive watchdog timed out
7-6 —	Reserved
5 EXCOL	Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. Excessive Collisions When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the CONFIGURATION register, this bit is set after the first collision and the packet transmission is aborted. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No collision 1b - Excessive collision is sensed

Table continues on the next page...

Field	Function
4 LCOL	Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode). Late Collision When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No collision 1b - Late collision is sensed
3 EXDEF	Excessive Deferral When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Excessive Deferral When the DTXSTS bit is set in the MAC_OPERATION_MODE register and the DC bit is set in the MAC_CONFIGURATION register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No Excessive deferral 1b - Excessive deferral
2 LCARR	Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. Loss of Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Carrier is present 1b - Loss of carrier
1 NCARR	No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. No Carrier When the DTXSTS bit is set in the MAC_OPERATION_MODE register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Carrier is present 1b - No carrier

Table continues on the next page...

Field	Function
0 TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_CONFIGURATION register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_CONFIGURATION register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No Transmit Jabber Timeout 1b - Transmit Jabber Timeout occurred

61.7.1.29 PMT Control and Status (MAC_PMT_CONTROL_STATUS)

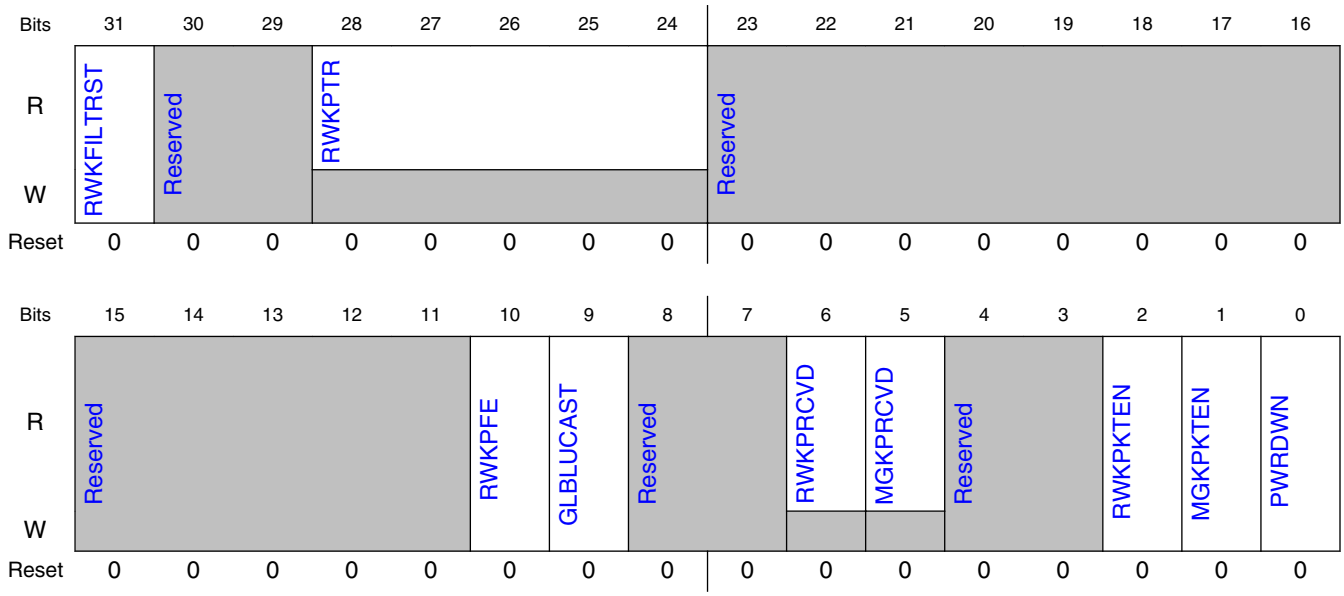
61.7.1.29.1 Offset

Register	Offset
MAC_PMT_CONTROL_STATUS	C0h

61.7.1.29.2 Function

The PMT Control and Status Register.

61.7.1.29.3 Diagram



61.7.1.29.4 Fields

Field	Function
31 RWKFILTRST	Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. It is automatically cleared after 1 clock cycle. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Remote Wake-Up Packet Filter Register Pointer is not Reset 1b - Remote Wake-Up Packet Filter Register Pointer is Reset
30-29 —	Reserved
28-24 RWKPTR	Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the clk_rx_i domain when a Write occurs to that register.
23-11 —	Reserved
10 RWKPFE	Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. Remote Wake-up Packet Forwarding Enable

Table continues on the next page...

Field	Function
	When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high. Note: If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Remote Wake-up Packet Forwarding is disabled 1b - Remote Wake-up Packet Forwarding is enabled
9 GLBLUCAST	Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. 0b - Global unicast is disabled 1b - Global unicast is enabled
8-7 —	Reserved
6 RWKPRCVD	Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Remote wake-up packet is received 1b - Remote wake-up packet is received
5 MGKPRCVD	Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - No Magic packet is received 1b - Magic packet is received
4-3 —	Reserved
2 RWKPKTEN	Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. 0b - Remote wake-up packet is disabled 1b - Remote wake-up packet is enabled
1	Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet.

Table continues on the next page...

Field	Function
MGKPKTEN	0b - Magic Packet is disabled 1b - Magic Packet is enabled
0 PWRDWN	Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Power down is disabled 1b - Power down is enabled

61.7.1.30 Remote Wakeup Filter (MAC_RWK_PACKET_FILTER)

61.7.1.30.1 Offset

Register	Offset
MAC_RWK_PACKET_FILTER	C4h

61.7.1.30.2 Function

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register.

The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_PACKET_FILTER register. When the Remote Wakeup Filters are to be programmed, the entire set of wkuppktfilter_reg registers must be written. The wkuppktfilter_reg register is programmed by sequentially writing the eight, sixteen or thirty-two register values in MAC_RWK_PACKET_FILTER register for wkuppktfilter_REG0, wkuppktfilter_REG1, ..., wkuppktfilter_REG31 respectively. The

wkuppktfilter_reg register is read in a similar way. The MAC updates the wkuppktfilter_reg register current pointer value in RWKPTR field of MAC_PMT_CONTROL_STATUS register.

The Remote Wakeup Filters are arranged in blocks of 4 filters each and each such block have eight 32-bit wide registers, viz. wkuppktfilter_REG0-7, wkuppktfilter_reg8-15, wkuppktfilter_REG16-23 and wkuppktfilter_REG24-31. The fields of Remote Wakeup Filter are described as follows:

Filter i Byte Mask: The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3, .., 15) to determine whether or not a packet is a wake-up packet.

- The MSB (31st bit) must be zero.

- Bit j[30:0] is the byte mask.

- If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored.

Filter i Command: The 4-bit filter i command controls the filter i operation.

- Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet.

- Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC-16 value. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2".

- Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. The details are provided below:

-- The And_Previous bit setting is applicable within a set of 4 filters.

-- Setting of And_Previous bit of filter that is not enabled has no effect, that is setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.

-- If And_Previous bit is set for filter to form AND chained filter, the AND chain breaks at the point any filter is not enabled. For example: If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set) but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result is considered. If Filter 2 And_Previous bit is set (bit 1 in

Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 1 is not enabled (bit 0 in Filter 1 command is reset), then only Filter 2 result ANDed with Filter 3 result is considered. If Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 3 And_Previous bit is set (bit 1 in Filter 3 command is set), but Filter 2 is not enabled (bit 0 in Filter 2 command is reset), then since setting of Filter 2 And_Previous bit has no effect only Filter 1 result ORed with Filter 3 result is considered.

-- If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type.

- Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled.

Filter i Offset: The filter i offset register defines the offset (within the packet) from which the filter i examines the packets.

- This 8-bit pattern-offset is the offset for the filter i first byte to be examined.

- The minimum allowed offset is 12, which refers to the 13th byte of the packet.

- The offset value 0 refers to the first byte of the packet.

Filter i CRC-16: The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register.

- The 16-bit CRC calculation uses the following polynomial: $G(x) = x^{16} + x^{15} + x^2 + 1$

- Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:

-- 32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC-16 calculation.

-- 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

The structure of the RWK Filter registers is shown in "Figure 10-11 Remote Wake-Up Packet Filter Register".

61.7.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WKUPFRMFTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.30.4 Fields

Field	Function
31-0 WKUPFRMFTR	RWK Packet Filter This field contains the various controls of RWK Packet filter.

61.7.1.31 LPI Control and Status (MAC_LPI_CONTROL_STATUS)

61.7.1.31.1 Offset

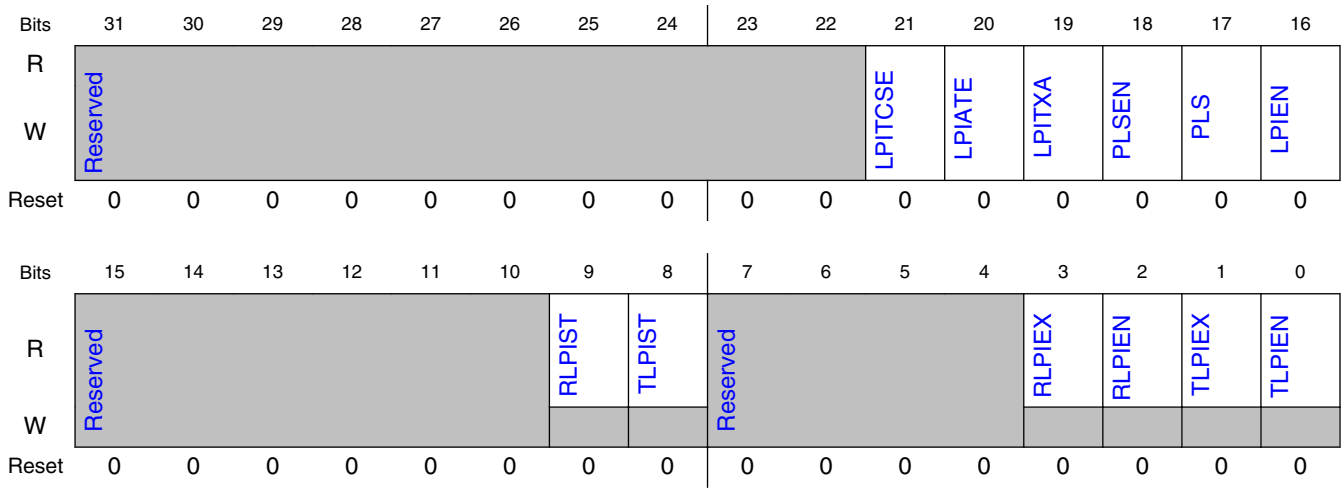
Register	Offset
MAC_LPI_CONTROL_STATUS	D0h

61.7.1.31.2 Function

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status.

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

61.7.1.31.3 Diagram



61.7.1.31.4 Fields

Field	Function
31-22 —	Reserved
21 LPITCSE	LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. LPI Tx Clock Stop Enable When this bit is set, the MAC asserts sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode to indicate that the Tx clock to MAC can be stopped. When this bit is reset, the MAC does not assert sbd_tx_clk_gating_ctrl_o signal high after it enters Tx LPI mode. If RGMII Interface is selected, the Tx clock is required for transmitting the LPI pattern. The Tx Clock cannot be gated and so the LPITCSE bit cannot be programmed. 0b - LPI Tx Clock Stop is disabled 1b - LPI Tx Clock Stop is enabled
20 LPIATE	LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. LPI Timer Enable This bit controls the automatic entry of the MAC Transmitter into and exit out of the LPI state. When LPIATE, LPITXA and LPIEN bits are set, the MAC Transmitter enters LPI state only when the complete MAC TX data path is IDLE for a period indicated by the MAC_LPI_ENTRY_TIMER register. After entering LPI state, if the data path becomes non-IDLE (due to a new packet being accepted for transmission), the Transmitter exits LPI state but does not clear LPIEN bit. This enables the re-entry into LPI state when it is IDLE again. When LPIATE is 0, the LPI Auto timer is disabled and MAC Transmitter enters LPI state based on the settings of LPITXA and LPIEN bit descriptions.

Table continues on the next page...

Field	Function
	0b - LPI Timer is disabled 1b - LPI Timer is enabled
19 LPITXA	LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. LPI Tx Automate This bit controls the behavior of the MAC when it is entering or coming out of the LPI mode on the Transmit side. This bit is not functional in the EQOS-CORE configurations in which the Tx clock gating is done during the LPI mode. If the LPITXA and LPIEN bits are set to 1, the MAC enters the LPI mode only after all outstanding packets (in the core) and pending packets (in the application interface) have been transmitted. The MAC comes out of the LPI mode when the application sends any packet for transmission or the application issues a Tx FIFO Flush command. In addition, the MAC automatically clears the LPIEN bit when it exits the LPI state. If Tx FIFO Flush is set in the FTQ bit of MAC_TXQ0_OPERATION_MODE register, when the MAC is in the LPI mode, it exits the LPI mode. When this bit is 0, the LPIEN bit directly controls behavior of the MAC when it is entering or coming out of the LPI mode. 0b - LPI Tx Automate is disabled 1b - LPI Tx Automate is enabled
18 PLSEN	PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. PHY Link Status Enable This bit enables the link status received on the RGMII, SGMII, or SMII Receive paths to be used for activating the LPI LS TIMER. When this bit is set, the MAC uses the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and the PLS bit for the LPI LS Timer trigger. When this bit is reset, the MAC ignores the link-status bits of the MAC_PHYIF_CONTROL_STATUS register and takes only the PLS bit. 0b - PHY Link Status is disabled 1b - PHY Link Status is enabled
17 PLS	PHY Link Status This bit indicates the link status of the PHY. PHY Link Status This bit indicates the link status of the PHY. The MAC Transmitter asserts the LPI pattern only when the link status is up (OKAY) at least for the time indicated by the LPI LS TIMER. When this bit is set, the link is considered to be okay (UP) and when this bit is reset, the link is considered to be down. 0b - link is down 1b - link is okay (UP)
16 LPIEN	LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. LPI Enable When this bit is set, it instructs the MAC Transmitter to enter the LPI state. When this bit is reset, it instructs the MAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the MAC exits the LPI state because of the arrival of a new packet for transmission. 0b - LPI state is disabled 1b - LPI state is enabled
15-10 —	Reserved

Table continues on the next page...

Field	Function
9 RLPIST	Receive LPI State When this bit is set, it indicates that the MAC is receiving the LPI pattern on the GMII or MII interface. 0b - Receive LPI state not detected 1b - Receive LPI state detected
8 TLPIST	Transmit LPI State When this bit is set, it indicates that the MAC is transmitting the LPI pattern on the GMII or MII interface. 0b - Transmit LPI state not detected 1b - Transmit LPI state detected
7-4 —	Reserved
3 RLPIEX	Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. Receive LPI Exit When this bit is set, it indicates that the MAC Receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0b - Receive LPI exit not detected 1b - Receive LPI exit detected
2 RLPIEN	Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. Receive LPI Entry When this bit is set, it indicates that the MAC Receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Note: This bit may not be set if the MAC stops receiving the LPI pattern for a very short duration, such as, less than three clock cycles of CSR clock. 0b - Receive LPI entry not detected 1b - Receive LPI entry detected
1 TLPIEX	Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. Transmit LPI Exit When this bit is set, it indicates that the MAC transmitter exited the LPI state after the application cleared the LPIEN bit and the LPI TW Timer has expired. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0b - Transmit LPI exit not detected 1b - Transmit LPI exit detected
0 TLPIEN	Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. Transmit LPI Entry When this bit is set, it indicates that the MAC Transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). 0b - Transmit LPI entry not detected 1b - Transmit LPI entry detected

61.7.1.32 LPI Timers Control (MAC_LPI_TIMERS_CONTROL)

61.7.1.32.1 Offset

Register	Offset
MAC_LPI_TIMERS_CONTROL	D4h

61.7.1.32.2 Function

The LPI Timers Control register controls the timeout values in the LPI states.

The LPI Timers Control register controls the timeout values in the LPI states. It specifies the time for which the MAC transmits the LPI pattern and also the time for which the MAC waits before resuming the normal transmission.

61.7.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						LST									
Reset	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.32.4 Fields

Field	Function
31-26 —	Reserved
25-16 LST	LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. LPI LS Timer

Table continues on the next page...

Field	Function
	This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (OKAY) before the LPI pattern can be transmitted to the PHY. The MAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec) as defined in the IEEE standard.
15-0 TWT	LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. LPI TW Timer This field specifies the minimum time (in microseconds) for which the MAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.

61.7.1.33 Tx LPI Entry Timer Control (MAC_LPI_ENTRY_TIMER)

61.7.1.33.1 Offset

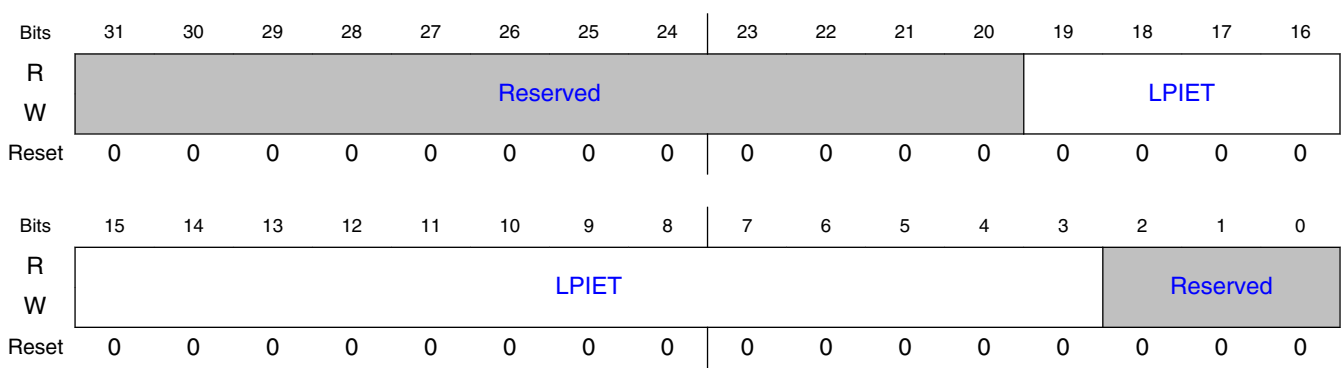
Register	Offset
MAC_LPI_ENTRY_TIMER	D8h

61.7.1.33.2 Function

This register controls the Tx LPI entry timer.

This register controls the Tx LPI entry timer. This counter is enabled only when bit[20] (LPITE) bit of MAC_LPI_CONTROL_STATUS is set to 1.

61.7.1.33.3 Diagram



61.7.1.33.4 Fields

Field	Function
31-20 —	Reserved
19-3 LPIET	LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. LPI Entry Timer This field specifies the time in microseconds the MAC waits to enter LPI mode, after it has transmitted all the frames. This field is valid and used only when LPITE and LPITXA are set to 1. Bits [2:0] are read-only so that the granularity of this timer is in steps of 8 micro-seconds.
2-0 —	Reserved

61.7.1.34 One-microsecond Reference Timer (MAC_ONEUS_TIC_COUNTER)

61.7.1.34.1 Offset

Register	Offset
MAC_ONEUS_TIC_COUNTER	DCh

61.7.1.34.2 Function

This register controls the generation of the Reference time (1 microsecond tic) for all the LPI timers. This timer has to be programmed by the software initially.

61.7.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				TIC_1US_CNTR											
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1

61.7.1.34.4 Fields

Field	Function
31-12 —	Reserved
11-0 TIC_1US_CNTR	1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. 1US TIC Counter The application must program this counter so that the number of clock cycles of CSR clock is 1us. (Subtract 1 from the value before programming). For example if the CSR clock is 100MHz then this field needs to be programmed to value 100 - 1 = 99 (which is 0x63). This is required to generate the 1US events that are used to update some of the EEE related counters.

61.7.1.35 PHY Interface Control and Status (MAC_PHYIF_CONTROL_STATUS)

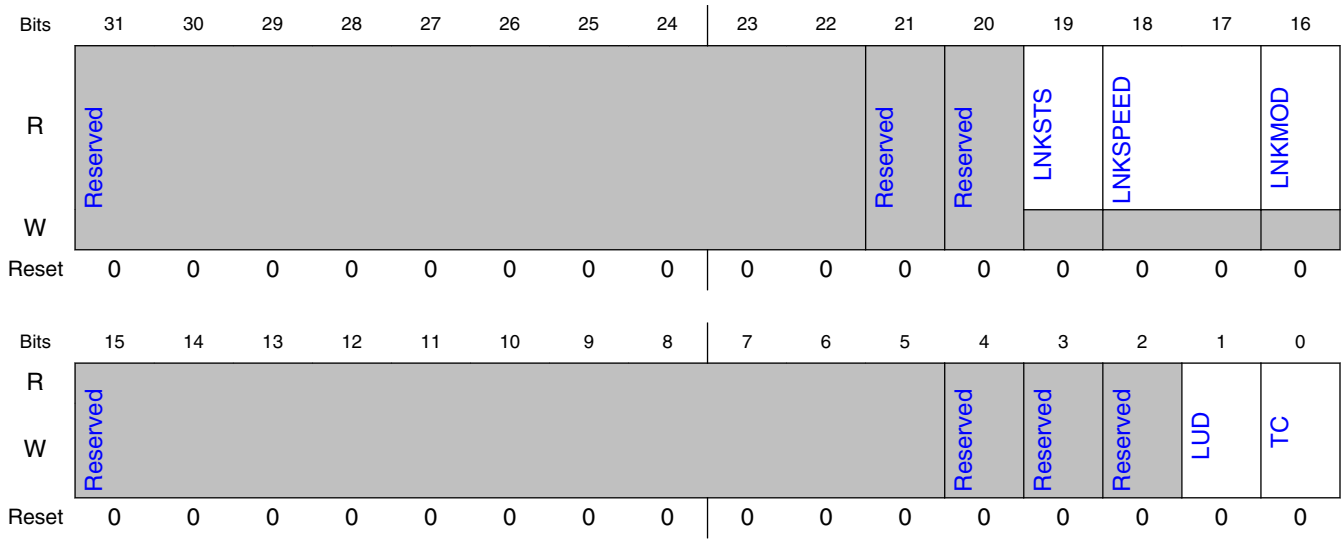
61.7.1.35.1 Offset

Register	Offset
MAC_PHYIF_CONTROL_STATUS	F8h

61.7.1.35.2 Function

The PHY Interface Control and Status register indicates the status signals received by the SGMII, RGMII, or SMII interface (selected at reset) from the PHY.

61.7.1.35.3 Diagram



61.7.1.35.4 Fields

Field	Function
31-22 —	Reserved
21 —	Reserved
20 —	Reserved
19 LNKSTS	Link Status This bit indicates whether the link is up (1'b1) or down (1'b0). 0b - Link down 1b - Link up
18-17 LNKSPEED	Link Speed This bit indicates the current speed of the link. 00b - 2.5 MHz 01b - 25 MHz 10b - 125 MHz 11b - Reserved
16 LNKMOD	Link Mode This bit indicates the current mode of operation of the link. 0b - Half-duplex mode 1b - Full-duplex mode
15-5 —	Reserved
4 —	Reserved
3 —	Reserved
2	Reserved

Table continues on the next page...

Field	Function
—	
1 LUD	Link Up or Down This bit indicates whether the link is up or down during transmission of configuration in the RGMII, SGMII, or SMII interface. 0b - Link down 1b - Link up
0 TC	Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. The details of this feature are provided in the following sections: - "Reduced Gigabit Media Independent Interface" - "Serial Media Independent Interface" - "Serial Gigabit Media Independent Interface" 0b - Disable Transmit Configuration in RGMII, SGMII, or SMII 1b - Enable Transmit Configuration in RGMII, SGMII, or SMII

61.7.1.36 MAC Version (MAC_VERSION)

61.7.1.36.1 Offset

Register	Offset
MAC_VERSION	110h

61.7.1.36.2 Function

The version register identifies the version of the DWC_ether_qos.

The version register identifies the version of the DWC_ether_qos. This register contains two bytes: one that we uses to identify the core release number, and the other that you set while configuring the core.

61.7.1.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	USERVER								SNPSVER							
W																
Reset	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0

61.7.1.36.4 Fields

Field	Function
31-16 —	Reserved
15-8 USERVER	User-defined Version (8'h10)
7-0 SNPSVER	Synopsys-defined Version

61.7.1.37 MAC Debug (MAC_DEBUG)

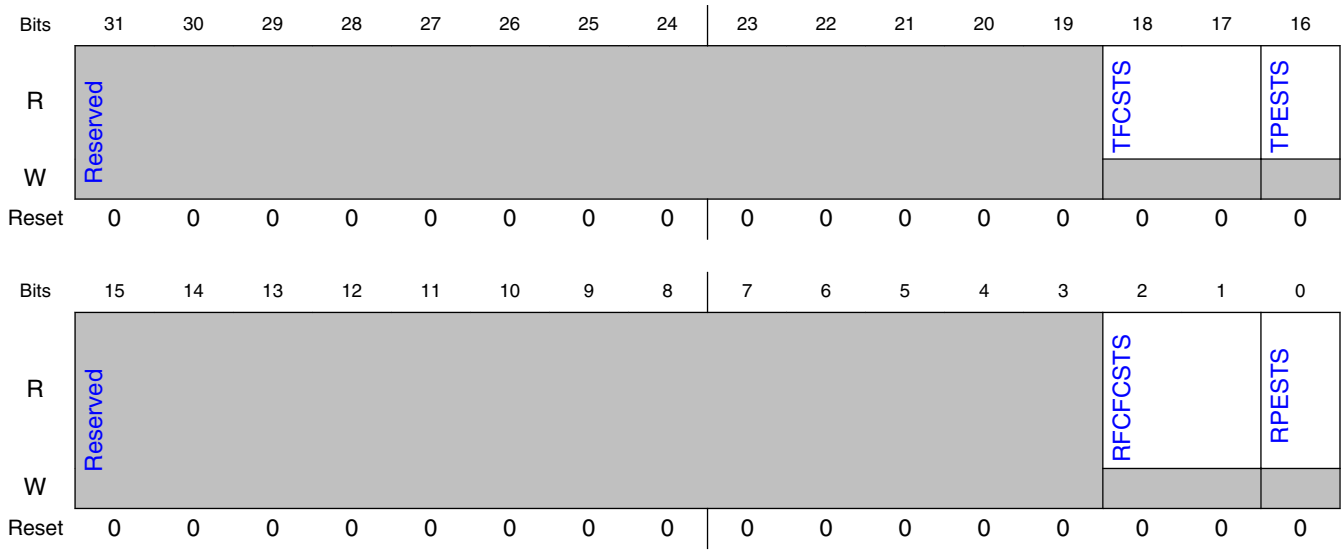
61.7.1.37.1 Offset

Register	Offset
MAC_DEBUG	114h

61.7.1.37.2 Function

The Debug register provides the debug status of various MAC blocks.

61.7.1.37.3 Diagram



61.7.1.37.4 Fields

Field	Function
31-19 —	Reserved
18-17 TFCSTS	MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. 00b - Idle state 01b - Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 10b - Generating and transmitting a Pause control packet (in full-duplex mode) 11b - Transferring input packet for transmission
16 TPESTS	MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. 0b - MAC GMII or MII Transmit Protocol Engine Status not detected 1b - MAC GMII or MII Transmit Protocol Engine Status detected
15-3 —	Reserved
2-1 RFCFCSTS	MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.
0 RPESTS	MAC GMII or MII Receive Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII receive protocol engine is actively receiving data, and it is not in the Idle state. 0b - MAC GMII or MII Receive Protocol Engine Status not detected 1b - MAC GMII or MII Receive Protocol Engine Status detected

61.7.1.38 Optional Features or Functions 0 (MAC_HW_FEATURE0)

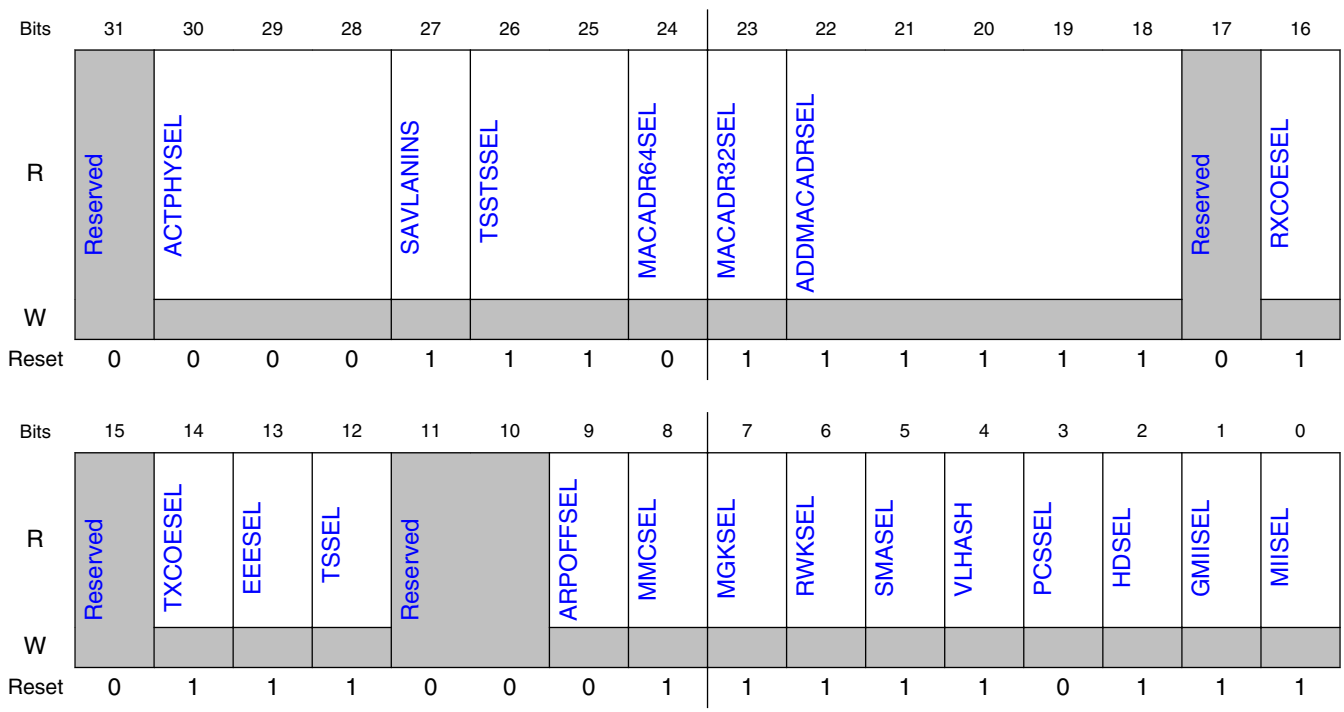
61.7.1.38.1 Offset

Register	Offset
MAC_HW_FEATURE0	11Ch

61.7.1.38.2 Function

This register indicates the presence of first set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

61.7.1.38.3 Diagram



61.7.1.38.4 Fields

Field	Function
31 —	Reserved

Table continues on the next page...

Field	Function
30-28 ACTPHYSEL	Active PHY Selected When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion. 000b - GMII or MII 001b - RGMII 010b - SGMII 011b - TBI 100b - RMII 101b - RTBI 110b - SMII 111b - RevMII
27 SAVLANINS	Source Address or VLAN Insertion Enable This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected 0b - Source Address or VLAN Insertion Enable option is not selected 1b - Source Address or VLAN Insertion Enable option is selected
26-25 TSSTSSEL	Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected 00b - Reserved 01b - Internal 10b - External 11b - Both
24 MACADR64SEL	MAC Addresses 64-127 Selected This bit is set to 1 when the Enable Additional 64 MAC Address Registers (64-127) option is selected 0b - MAC Addresses 64-127 Select option is not selected 1b - MAC Addresses 64-127 Select option is selected
23 MACADR32SEL	MAC Addresses 32-63 Selected This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected 0b - MAC Addresses 32-63 Select option is not selected 1b - MAC Addresses 32-63 Select option is selected
22-18 ADDMACADRSEL	MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option
17 —	Reserved
16 RXCOESEL	Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected 0b - Receive Checksum Offload Enable option is not selected 1b - Receive Checksum Offload Enable option is selected
15 —	Reserved
14 TXCOESEL	Transmit Checksum Offload Enabled This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected 0b - Transmit Checksum Offload Enable option is not selected 1b - Transmit Checksum Offload Enable option is selected
13 EEESEL	Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected 0b - Energy Efficient Ethernet Enable option is not selected 1b - Energy Efficient Ethernet Enable option is selected
12 TSSEL	IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected 0b - IEEE 1588-2008 Timestamp Enable option is not selected 1b - IEEE 1588-2008 Timestamp Enable option is selected

Table continues on the next page...

Field	Function
11-10 —	Reserved
9 ARPOFFSEL	ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected 0b - ARP Offload Enable option is not selected 1b - ARP Offload Enable option is selected
8 MMCSEL	RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected 0b - RMON Module Enable option is not selected 1b - RMON Module Enable option is selected
7 MGKSEL	PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected 0b - PMT Magic Packet Enable option is not selected 1b - PMT Magic Packet Enable option is selected
6 RWKSEL	PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected 0b - PMT Remote Wake-up Packet Enable option is not selected 1b - PMT Remote Wake-up Packet Enable option is selected
5 SMASEL	SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected 0b - SMA (MDIO) Interface not selected 1b - SMA (MDIO) Interface selected
4 VLHASH	VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected 0b - VLAN Hash Filter not selected 1b - VLAN Hash Filter selected
3 PCSSEL	PCS Registers (TBI, SGMII, or RTBI PHY interface) This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected 0b - No PCS Registers (TBI, SGMII, or RTBI PHY interface) 1b - PCS Registers (TBI, SGMII, or RTBI PHY interface)
2 HDSEL	Half-duplex Support This bit is set to 1 when the half-duplex mode is selected 0b - No Half-duplex support 1b - Half-duplex support
1 GMIISEL	1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation 0b - No 1000 Mbps support 1b - 1000 Mbps support
0 MIISEL	10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation 0b - No 10 or 100 Mbps support 1b - 10 or 100 Mbps support

61.7.1.39 Optional Features or Functions 1 (MAC_HW_FEATURE1)

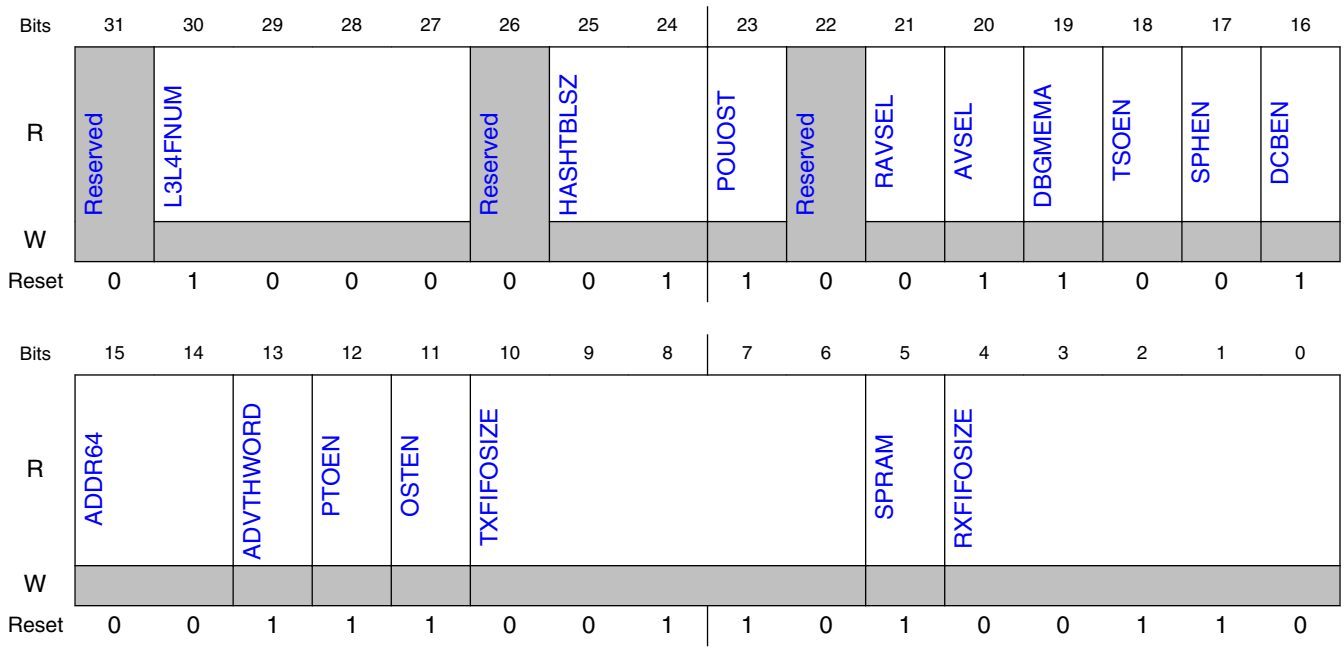
61.7.1.39.1 Offset

Register	Offset
MAC_HW_FEATURE1	120h

61.7.1.39.2 Function

This register indicates the presence of second set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

61.7.1.39.3 Diagram



61.7.1.39.4 Fields

Field	Function
31 —	Reserved
30-27 L3L4FNUM	Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: 0000b - No L3 or L4 Filter 0001b - 1 L3 or L4 Filter 0010b - 2 L3 or L4 Filters 0011b - 3 L3 or L4 Filters 0100b - 4 L3 or L4 Filters 0101b - 5 L3 or L4 Filters 0110b - 6 L3 or L4 Filters 0111b - 7 L3 or L4 Filters 1000b - 8 L3 or L4 Filters
26 —	Reserved

Table continues on the next page...

Field	Function
25-24 HASHTBLSZ	Hash Table Size This field indicates the size of the hash table: 00b - No hash table 01b - 64 10b - 128 11b - 256
23 POUOST	One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. 0b - One Step for PTP over UDP/IP Feature is not selected 1b - One Step for PTP over UDP/IP Feature is selected
22 —	Reserved
21 RAVSEL	Rx Side Only AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. 0b - Rx Side Only AV Feature is not selected 1b - Rx Side Only AV Feature is selected
20 AVSEL	AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. 0b - AV Feature is not selected 1b - AV Feature is selected
19 DBGMEMA	DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected 0b - DMA Debug Registers option is not selected 1b - DMA Debug Registers option is selected
18 TSOEN	TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected 0b - TCP Segmentation Offload Feature is not selected 1b - TCP Segmentation Offload Feature is selected
17 SPHEN	Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected 0b - Split Header Feature is not selected 1b - Split Header Feature is selected
16 DCBEN	DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is selected 0b - DCB Feature is not selected 1b - DCB Feature is selected
15-14 ADDR64	Address Width. Address Width. This field indicates the configured address width: 00b - 32 01b - 40 10b - 48 11b - Reserved
13 ADVTHWORD	IEEE 1588 High Word Register Enable This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected 0b - IEEE 1588 High Word Register option is not selected 1b - IEEE 1588 High Word Register option is selected
12 PTOEN	PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. 0b - PTP Offload feature is not selected 1b - PTP Offload feature is selected
11 OSTEN	One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. 0b - One-Step Timestamping feature is not selected 1b - One-Step Timestamping feature is selected

Table continues on the next page...

Field	Function
10-6 TXFIFOSIZE	MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(TXFIFO_SIZE) -7: 0_0000b - 128 bytes 0_0001b - 256 bytes 0_0010b - 512 bytes 0_0011b - 1024 bytes 0_0100b - 2048 bytes 0_0101b - 4096 bytes 0_0110b - 8192 bytes 0_0111b - 16384 bytes 0_1000b - 32 KB 0_1001b - 64 KB 0_1010b - 128 KB 0_1011b - Reserved
5 SPRAM	Single Port RAM Enable This bit is set to 1 when the Use single port RAM Feature is selected. 0b - Single Port RAM feature is not selected 1b - Single Port RAM feature is selected
4-0 RXFIFOSIZE	MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(RXFIFO_SIZE) -7: 0_0000b - 128 bytes 0_0001b - 256 bytes 0_0010b - 512 bytes 0_0011b - 1024 bytes 0_0100b - 2048 bytes 0_0101b - 4096 bytes 0_0110b - 8192 bytes 0_0111b - 16384 bytes 0_1000b - 32 KB 0_1001b - 64 KB 0_1010b - 128 KB 0_1011b - 256 KB 0_1100b - Reserved

61.7.1.40 Optional Features or Functions 2 (MAC_HW_FEATURE2)

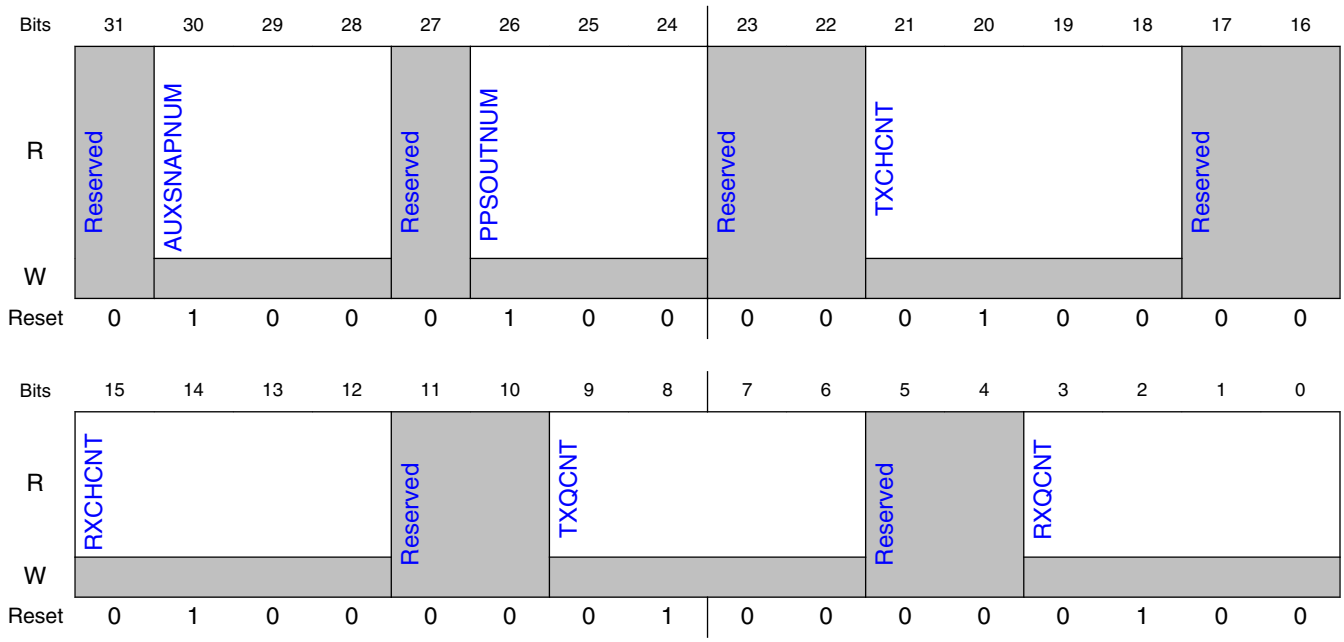
61.7.1.40.1 Offset

Register	Offset
MAC_HW_FEATURE2	124h

61.7.1.40.2 Function

This register indicates the presence of third set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

61.7.1.40.3 Diagram



61.7.1.40.4 Fields

Field	Function
31 —	Reserved
30-28 AUXSNAPNUM	Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: 000b - No auxiliary input 001b - 1 auxiliary input 010b - 2 auxiliary input 011b - 3 auxiliary input 100b - 4 auxiliary input 101b - Reserved
27 —	Reserved
26-24 PPSOUTNUM	Number of PPS Outputs This field indicates the number of PPS outputs: 000b - No PPS output 001b - 1 PPS output 010b - 2 PPS output 011b - 3 PPS output 100b - 4 PPS output 101b - Reserved
23-22 —	Reserved
21-18 TXCHCNT	Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: 0000b - 1 MTL Tx Channel

Table continues on the next page...

Field	Function
	0001b - 2 MTL Tx Channels 0010b - 3 MTL Tx Channels 0011b - 4 MTL Tx Channels 0100b - 5 MTL Tx Channels 0101b - Reserved 0110b - Reserved 0111b - Reserved
17-16 —	Reserved
15-12 RXCHCNT	Number of DMA Receive Channels This field indicates the number of DMA Receive channels: 0000b - 1 MTL Rx Channel 0001b - 2 MTL Rx Channels 0010b - 3 MTL Rx Channels 0011b - 4 MTL Rx Channels 0100b - 5 MTL Rx Channels 0101b - Reserved 0110b - Reserved 0111b - Reserved
11-10 —	Reserved
9-6 TXQCNT	Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: 0000b - 1 MTL Tx Queue 0001b - 2 MTL Tx Queues 0010b - 3 MTL Tx Queues 0011b - 4 MTL Tx Queues 0100b - 5 MTL Tx Queues 0101b - Reserved 0110b - Reserved 0111b - Reserved
5-4 —	Reserved
3-0 RXQCNT	Number of MTL Receive Queues This field indicates the number of MTL Receive queues: 0000b - 1 MTL Rx Queue 0001b - 2 MTL Rx Queues 0010b - 3 MTL Rx Queues 0011b - 4 MTL Rx Queues 0100b - 5 MTL Rx Queues 0101b - Reserved 0110b - Reserved 0111b - Reserved

61.7.1.41 Optional Features or Functions 3 (MAC_HW_FEATURE3)

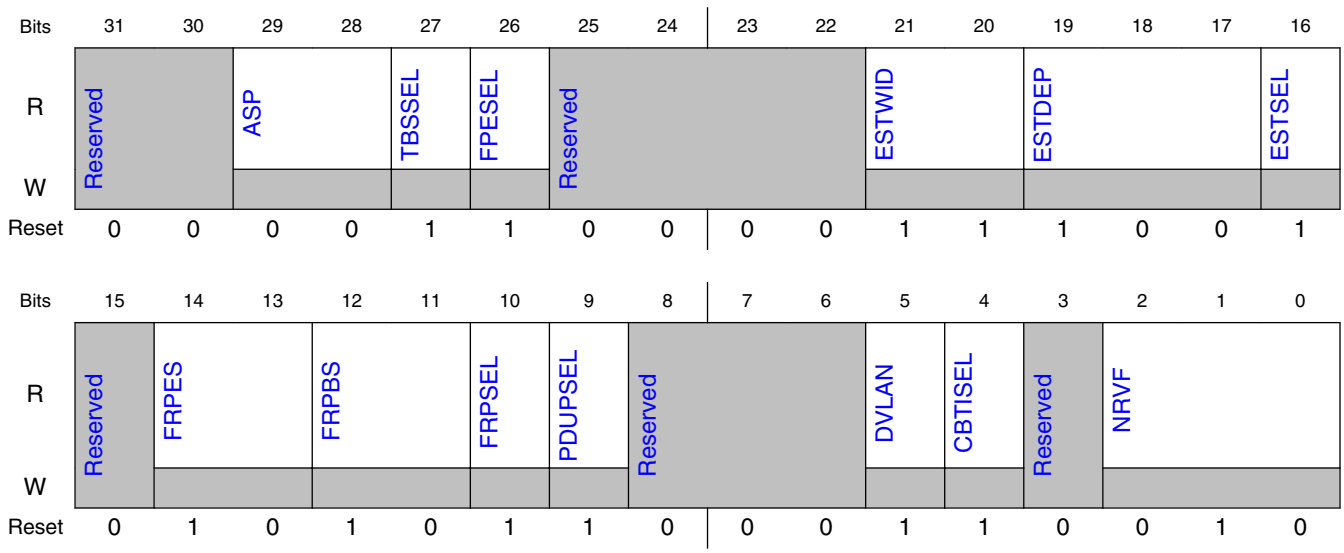
61.7.1.41.1 Offset

Register	Offset
MAC_HW_FEATURE3	128h

61.7.1.41.2 Function

This register indicates the presence of fourth set the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

61.7.1.41.3 Diagram



61.7.1.41.4 Fields

Field	Function
31-30 —	Reserved
29-28 ASP	Automotive Safety Package Following are the encoding for the different Safety features 00b - No Safety features selected 01b - Only "ECC protection for external memory" feature is selected 10b - All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 11b - All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature
27 TBSSEL	Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. 0b - Time Based Scheduling Enable feature is not selected 1b - Time Based Scheduling Enable feature is selected
26 FPESEL	Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. 0b - Frame Preemption Enable feature is not selected 1b - Frame Preemption Enable feature is selected
25-22	Reserved

Table continues on the next page...

Field	Function
—	
21-20 ESTWID	Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field 00b - Width not configured 01b - 16 10b - 20 11b - 24
19-17 ESTDEP	Depth of the Gate Control List This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5 000b - No Depth configured 001b - 64 010b - 128 011b - 256 100b - 512 101b - 1024 110b - Reserved
16 ESTSEL	Enhancements to Scheduling Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. 0b - Enable Enhancements to Scheduling Traffic feature is not selected 1b - Enable Enhancements to Scheduling Traffic feature is selected
15 —	Reserved
14-13 FRPES	Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. 00b - 64 Entries 01b - 128 Entries 10b - 256 Entries 11b - Reserved
12-11 FRPBS	Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. 00b - 64 Bytes 01b - 128 Bytes 10b - 256 Bytes 11b - Reserved
10 FRPSEL	Flexible Receive Parser Selected This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. 0b - Flexible Receive Parser feature is not selected 1b - Flexible Receive Parser feature is selected
9 PDUPSEL	Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. 0b - Broadcast/Multicast Packet Duplication feature is not selected 1b - Broadcast/Multicast Packet Duplication feature is selected
8-6 —	Reserved
5 DVLAN	Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. 0b - Double VLAN option is not selected 1b - Double VLAN option is selected
4 CBTISEL	Queue/Channel based VLAN tag insertion on Tx Enable This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. 0b - Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 1b - Enable Queue/Channel based VLAN tag insertion on Tx feature is selected

Table continues on the next page...

Field	Function
3 —	Reserved
2-0 NRVF	Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: 000b - No Extended Rx VLAN Filters 001b - 4 Extended Rx VLAN Filters 010b - 8 Extended Rx VLAN Filters 011b - 16 Extended Rx VLAN Filters 100b - 24 Extended Rx VLAN Filters 101b - 32 Extended Rx VLAN Filters 110b - Reserved

61.7.1.42 MDIO Address (MAC_MDIO_ADDRESS)

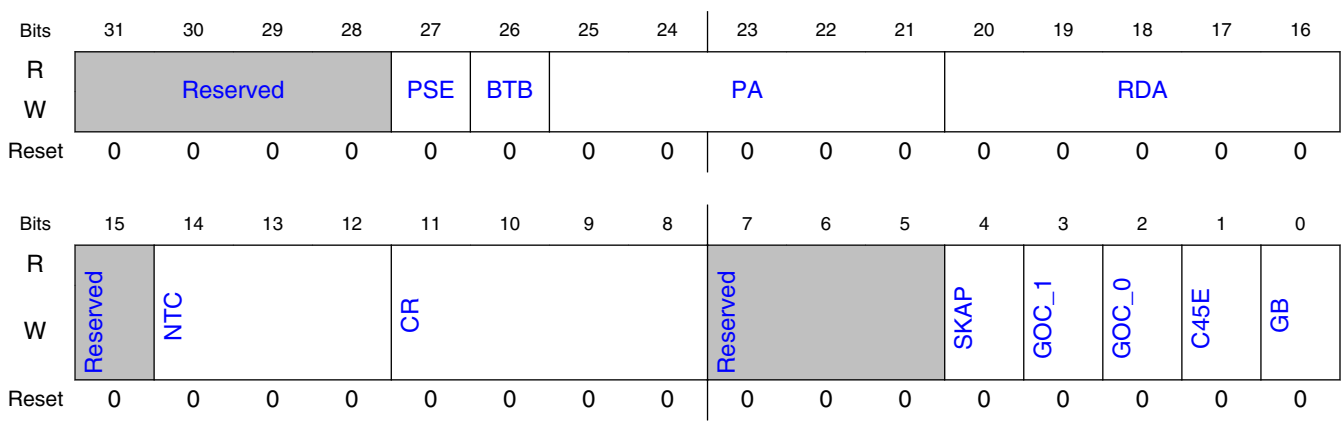
61.7.1.42.1 Offset

Register	Offset
MAC_MDIO_ADDRESS	200h

61.7.1.42.2 Function

The MDIO Address register controls the management cycles to external PHY through a management interface.

61.7.1.42.3 Diagram



61.7.1.42.4 Fields

Field	Function
31-28 —	Reserved
27 PSE	Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. 0b - Preamble Suppression disabled 1b - Preamble Suppression enabled
26 BTB	Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. 0b - Back to Back transactions disabled 1b - Back to Back transactions enabled
25-21 PA	Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20-16 RDA	Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15 —	Reserved
14-12 NTC	Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. Number of Trailing Clocks

Table continues on the next page...

Field	Function
	This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.
11-8 CR	CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: - 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1. CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: - 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 - 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 - 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 - 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 - 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 - 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 - 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 - 0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz freqency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is above the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks: - 1000: CSR clock/4 - 1001: CSR clock/6 - 1010: CSR clock/8 - 1011: CSR clock/10 - 1100: CSR clock/12 - 1101: CSR clock/14 - 1110: CSR clock/16 - 1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.
7-5 —	Reserved
4 SKAP	Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. Skip Address Packet

Table continues on the next page...

Field	Function
	When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. 0b - Skip Address Packet is disabled 1b - Skip Address Packet is enabled
3 GOC_1	GMII Operation Command 1 This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_O is encoded as follows: - 00: Reserved - 01: Write - 10: Post Read Increment Address for Clause 45 PHY - 11: Read When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid. 0b - GMII Operation Command 1 is disabled 1b - GMII Operation Command 1 is enabled
2 GOC_0	GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. 0b - GMII Operation Command 0 is disabled 1b - GMII Operation Command 0 is enabled
1 C45E	Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO. 0b - Clause 45 PHY is disabled 1b - Clause 45 PHY is enabled
0 GB	GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_ADDRESS and MAC_MDIO_DATA registers as long as this bit is set. For write transfers, the application must first write 16-bit data in the GDI field (and also RA field when C45E is set) in MAC_MDIO_DATA register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_DATA register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_DATA register. Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - GMII Busy is disabled 1b - GMII Busy is enabled

61.7.1.43 MAC MDIO Data (MAC_MDIO_DATA)

61.7.1.43.1 Offset

Register	Offset
MAC_MDIO_DATA	204h

61.7.1.43.2 Function

The MAC_MDIO_DATA register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_ADDRESS. This register also stores the Read data from the PHY register located at the address specified by MAC_MDIO_ADDRESS register.

61.7.1.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.43.4 Fields

Field	Function
31-16 RA	Register Address This field is valid only when C45E is set. Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.
15-0 GD	GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

61.7.1.44 CSR Software Control (MAC_CSR_SW_CTRL)

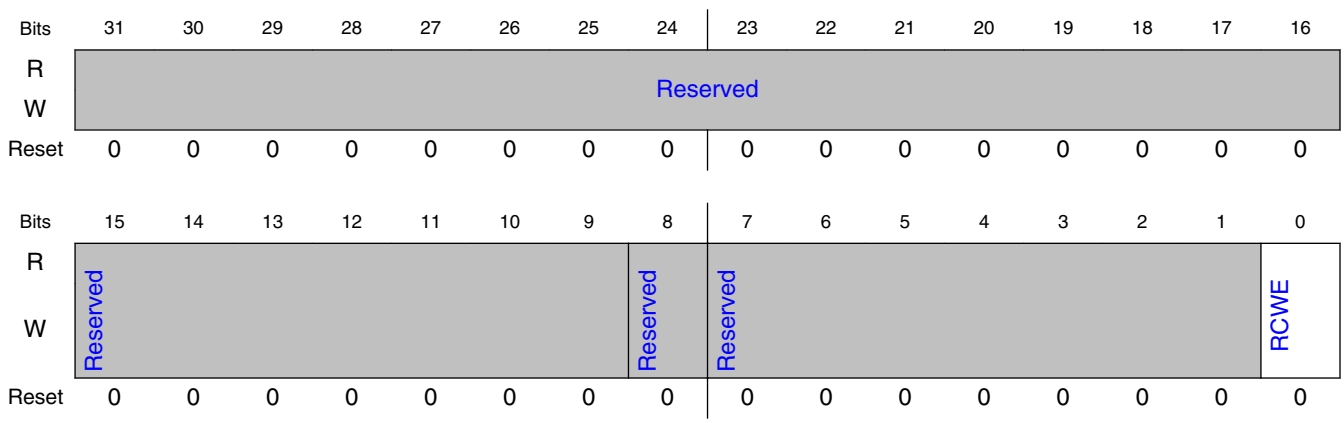
61.7.1.44.1 Offset

Register	Offset
MAC_CSR_SW_CTRL	230h

61.7.1.44.2 Function

This register contains SW programmable controls for changing the CSR access response and status bits clearing.

61.7.1.44.3 Diagram



61.7.1.44.4 Fields

Field	Function
31-9 —	Reserved
8 —	Reserved
7-1 —	Reserved
0 RCWE	Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it.

Field	Function
	When this bit is reset, the access mode of these register fields remain as Clear on Read. 0b - Register Clear on Write 1 is disabled 1b - Register Clear on Write 1 is enabled

61.7.1.45 Frame Preemption Control (MAC_FPE_CTRL_STS)

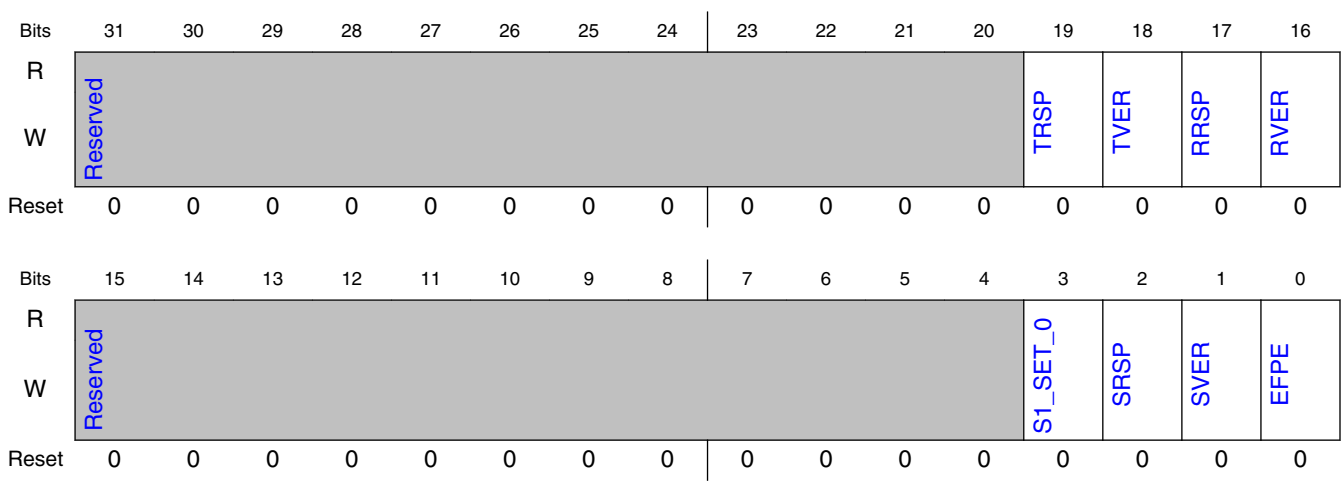
61.7.1.45.1 Offset

Register	Offset
MAC_FPE_CTRL_STS	234h

61.7.1.45.2 Function

This register controls the operation of Frame Preemption.

61.7.1.45.3 Diagram



61.7.1.45.4 Fields

Field	Function
31-20 —	Reserved
19	Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field).

Table continues on the next page...

Field	Function
TRSP	Transmitted Respond Frame Set when a Respond mPacket is transmitted (triggered by setting SRSP field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Not transmitted Respond Frame 1b - transmitted Respond Frame
18 TVER	Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). Transmitted Verify Frame Set when a Verify mPacket is transmitted (triggered by setting SVER field). An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Not transmitted Verify Frame 1b - transmitted Verify Frame
17 RRSP	Received Respond Frame Set when a Respond mPacket is received. Received Respond Frame Set when a Respond mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Not received Respond Frame 1b - Received Respond Frame
16 RVER	Received Verify Frame Set when a Verify mPacket is received. Received Verify Frame Set when a Verify mPacket is received. An interrupt can be generated for this event if FPEIE bit of MAC_INTERRUPT_ENABLE is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Not received Verify Frame 1b - Received Verify Frame
15-4 —	Reserved
3 S1_SET_0	Reserved, Must be set to "0". Reserved, Must be set to "0". This field is reserved for Synopsys internal use, and must always be set to "0" unless instructed by Synopsys. Setting to "1" might cause unexpected behavior in the IP.
2 SRSP	Send Respond mPacket When set indicates hardware to send a Respond mPacket. Send Respond mPacket When set indicates hardware to send a Respond mPacket. Reset by hardware after sending the Respond mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

Table continues on the next page...

Field	Function
	0b - Send Respond mPacket is disabled 1b - Send Respond mPacket is enabled
1 SVER	Send Verify mPacket When set indicates hardware to send a verify mPacket. Send Verify mPacket When set indicates hardware to send a verify mPacket. Reset by hardware after sending the Verify mPacket. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Send Verify mPacket is disabled 1b - Send Verify mPacket is enabled
0 EFPE	Enable Tx Frame Preemption When set Frame Preemption Tx functionality is enabled. 0b - Tx Frame Preemption is disabled 1b - Tx Frame Preemption is enabled

61.7.1.46 32-bit Binary Rollover Equivalent Time (MAC_PRESN_TIME_NS)

61.7.1.46.1 Offset

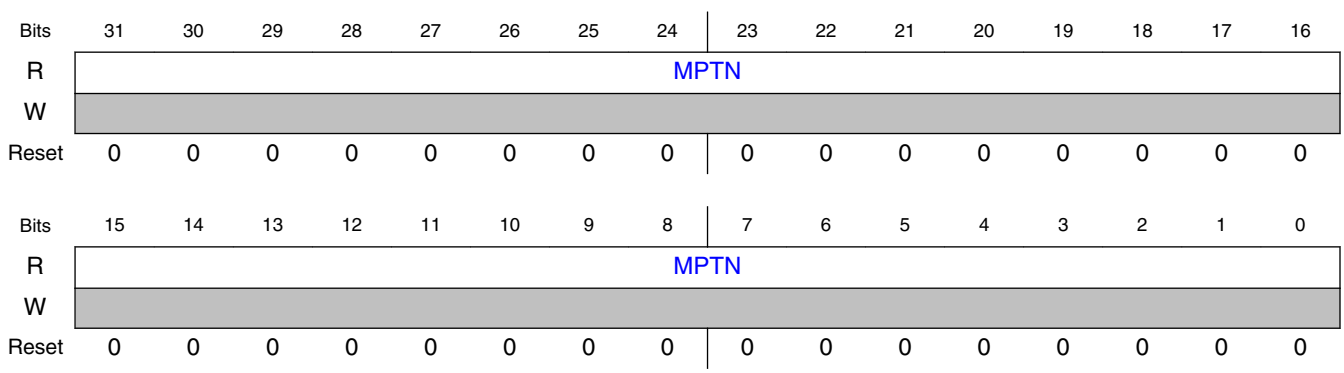
Register	Offset
MAC_PRESN_TIME_NS	240h

61.7.1.46.2 Function

This register contains the 32-bit binary rollover equivalent time of the PTP System Time in ns

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

61.7.1.46.3 Diagram



61.7.1.46.4 Fields

Field	Function
31-0 MPTN	MAC 1722 Presentation Time in ns These bits indicate the value of the 32-bit binary rollover equivalent time of the PTP System Time in ns

61.7.1.47 MAC 1722 Presentation Time (MAC_PRESN_TIME_UPDT)

61.7.1.47.1 Offset

Register	Offset
MAC_PRESN_TIME_UPDT	244h

61.7.1.47.2 Function

This field holds the 32-bit value of MAC 1722 Presentation Time in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in TIMESTAMP_CONTROL register).

Exists when DWC_EQOS_FLEXI_PPS_OUT_EN is configured

61.7.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MPTU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MPTU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.47.4 Fields

Field	Function
31-0 MPTU	MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. MAC 1722 Presentation Time Update This field holds the init value or the update value for the presentation time. When used for update, this field holds the 32-bit value in ns, that should be added to the Current Presentation Time Counter value. Init happens when TSINIT is set, and update happens when the TSUPDT bit is set (TSINIT and TSINIT defined in MAC_TIMESTAMP_CONTROL register). When ADDSUB field of MAC_SYSTEM_TIME_NANOSECONDS_UPDATE is set, this value is directly used for subtraction

61.7.1.48 MAC Address0 High (MAC_ADDRESS0_HIGH)

61.7.1.48.1 Offset

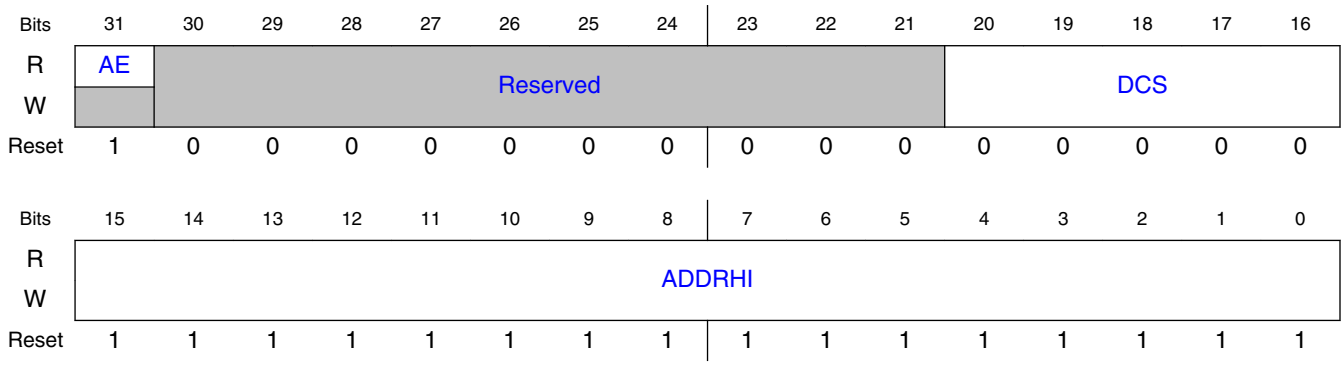
Register	Offset
MAC_ADDRESS0_HIGH	300h

61.7.1.48.2 Function

The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, then the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.48.3 Diagram



61.7.1.48.4 Fields

Field	Function
31 AE	Address Enable This bit is always set to 1. 0b - INVALID : This bit must be always set to 1 1b - This bit is always set to 1
30-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address0 content is routed.
15-0 ADDRHI	MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

61.7.1.49 MAC Address0 Low (MAC_ADDRESS0_LOW)

61.7.1.49.1 Offset

Register	Offset
MAC_ADDRESS0_LOW	304h

61.7.1.49.2 Function

The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station.

61.7.1.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.49.4 Fields

Field	Function
31-0 ADDRLO	MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

61.7.1.50 MAC Address1 High (MAC_ADDRESS1_HIGH)

61.7.1.50.1 Offset

Register	Offset
MAC_ADDRESS1_HIGH	308h

61.7.1.50.2 Function

The MAC ADDRESS1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.50.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS1[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address

Table continues on the next page...

Field	Function
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS1 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS1_High[15:8] - Bit 28: ADDRESS1_High[7:0] - Bit 27: ADDRESS1_Low[31:24] - .. - Bit 24: ADDRESS1_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS1 [47:32] This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.

61.7.1.51 MAC Address1 Low (MAC_ADDRESS1_LOW)

61.7.1.51.1 Offset

Register	Offset
MAC_ADDRESS1_LOW	30Ch

61.7.1.51.2 Function

The MAC ADDRESS1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

61.7.1.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.51.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. MAC ADDRESS1 [31:0] This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.52 MAC Address2 High (MAC_ADDRESS2_HIGH)

61.7.1.52.1 Offset

Register	Offset
MAC_ADDRESS2_HIGH	310h

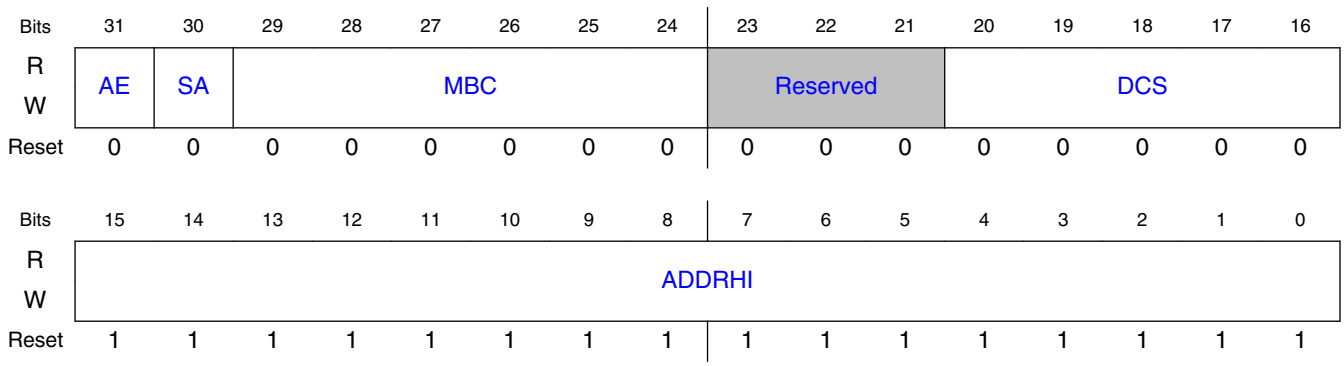
61.7.1.52.2 Function

The MAC ADDRESS2 High register holds the upper 16 bits of the third 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS2 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.52.3 Diagram



61.7.1.52.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the third MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the third MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS2[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS2[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS2[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS2 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS2_High[15:8] - Bit 28: ADDRESS2_High[7:0] - Bit 27: ADDRESS2_Low[31:24] - ..

Table continues on the next page...

Field	Function
	- Bit 24: ADDRESS2_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address2 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address2 content is routed.
15-0 ADDRHI	MAC ADDRESS2 [47:32] This field contains the upper 16 bits[47:32] of the third 6-byte MAC address.

61.7.1.53 MAC Address2 Low (MAC_ADDRESS2_LOW)

61.7.1.53.1 Offset

Register	Offset
MAC_ADDRESS2_LOW	314h

61.7.1.53.2 Function

The MAC ADDRESS2 Low register holds the lower 32 bits of the third 6-byte MAC address of the station.

61.7.1.53.3 Diagram

61.7.1.53.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS2 [31:0] This field contains the lower 32 bits of third 6-byte MAC address. MAC ADDRESS2 [31:0] This field contains the lower 32 bits of third 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.54 MAC Address3 High (MAC_ADDRESS3_HIGH)

61.7.1.54.1 Offset

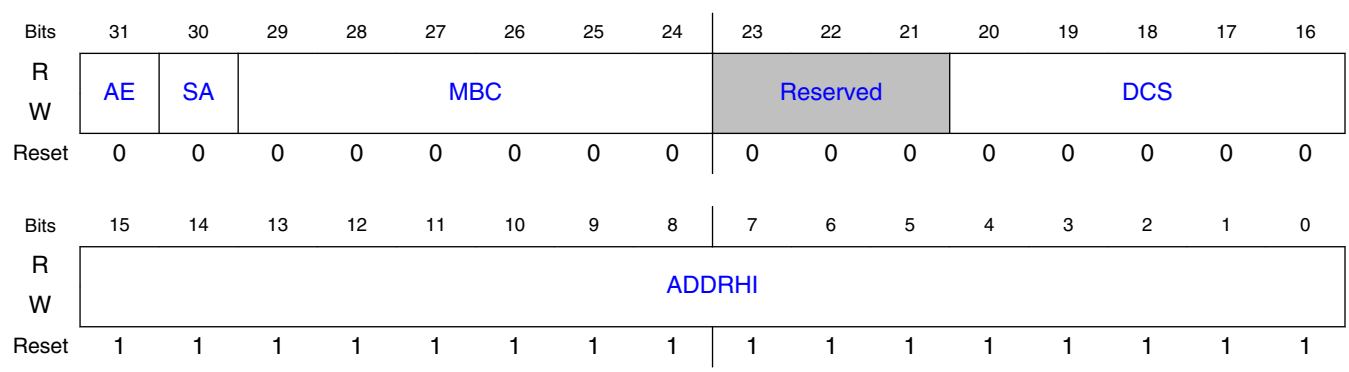
Register	Offset
MAC_ADDRESS3_HIGH	318h

61.7.1.54.2 Function

The MAC ADDRESS3 High register holds the upper 16 bits of the fourth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS3 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.54.3 Diagram



61.7.1.54.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the fourth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS3[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS3[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS3[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS3 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS3_High[15:8] - Bit 28: ADDRESS3_High[7:0] - Bit 27: ADDRESS3_Low[31:24] - .. - Bit 24: ADDRESS3_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address3 content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address3 content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address3 content is routed.
15-0 ADDRHI	MAC ADDRESS3 [47:32] This field contains the upper 16 bits[47:32] of the fourth 6-byte MAC address.

61.7.1.55 MAC Address3 Low (MAC_ADDRESS3_LOW)

61.7.1.55.1 Offset

Register	Offset
MAC_ADDRESS3_LOW	31Ch

61.7.1.55.2 Function

The MAC ADDRESS3 Low register holds the lower 32 bits of the fourth 6-byte MAC address of the station.

61.7.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.55.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS3 [31:0] This field contains the lower 32 bits of fourth 6-byte MAC address. MAC ADDRESS3 [31:0] This field contains the lower 32 bits of fourth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.56 MAC Address4 High (MAC_ADDRESS4_HIGH)

61.7.1.56.1 Offset

Register	Offset
MAC_ADDRESS4_HIGH	320h

61.7.1.56.2 Function

The MAC ADDRESS4 High register holds the upper 16 bits of the fifth 6-byte MAC address of the station.

The MAC ADDRESS4 High register holds the upper 16 bits of the fifth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS4 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.56.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the fifth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the fifth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored

Table continues on the next page...

Field	Function
	1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS4[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS4[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS4[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS4 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS4_High[15:8] - Bit 28: ADDRESS4_High[7:0] - Bit 27: ADDRESS4_Low[31:24] - .. - Bit 24: ADDRESS4_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS4 [47:32] This field contains the upper 16 bits[47:32] of the fifth 6-byte MAC address.

61.7.1.57 MAC Address4 Low (MAC_ADDRESS4_LOW)

61.7.1.57.1 Offset

Register	Offset
MAC_ADDRESS4_LOW	324h

61.7.1.57.2 Function

The MAC ADDRESS4 Low register holds the lower 32 bits of the fifth 6-byte MAC address of the station.

61.7.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.57.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS4 [31:0] This field contains the lower 32 bits of fifth 6-byte MAC address. MAC ADDRESS4 [31:0] This field contains the lower 32 bits of fifth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.58 MAC Address5 High (MAC_ADDRESS5_HIGH)

61.7.1.58.1 Offset

Register	Offset
MAC_ADDRESS5_HIGH	328h

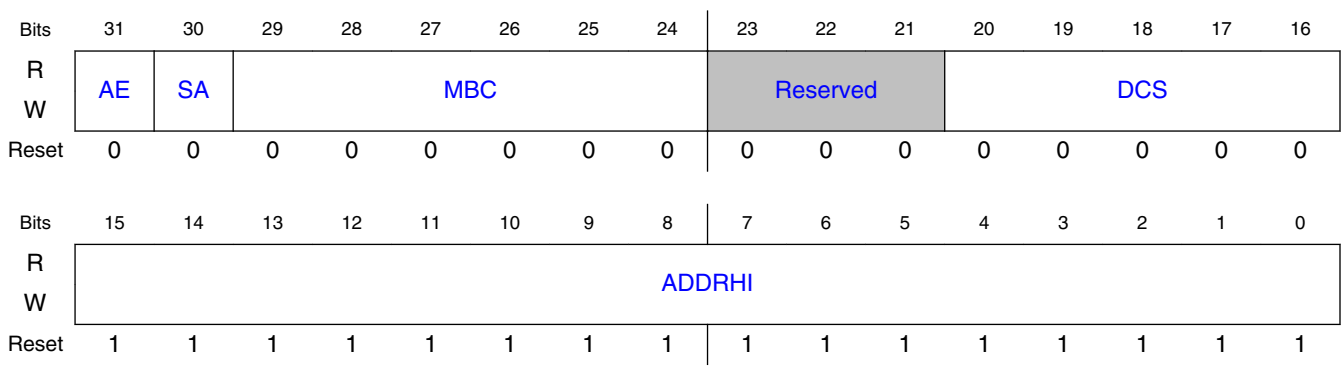
61.7.1.58.2 Function

The MAC ADDRESS5 High register holds the upper 16 bits of the sixth 6-byte MAC address of the station.

The MAC ADDRESS5 High register holds the upper 16 bits of the sixth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS5 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.58.3 Diagram



61.7.1.58.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the sixth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the sixth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS5[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS5[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS5[47:0] is used to compare with the DA fields of the received packet.

Table continues on the next page...

Field	Function
	0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS5 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS5_High[15:8] - Bit 28: ADDRESS5_High[7:0] - Bit 27: ADDRESS5_Low[31:24] - .. - Bit 24: ADDRESS5_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS5 [47:32] This field contains the upper 16 bits[47:32] of the sixth 6-byte MAC address.

61.7.1.59 MAC Address5 Low (MAC_ADDRESS5_LOW)

61.7.1.59.1 Offset

Register	Offset
MAC_ADDRESS5_LOW	32Ch

61.7.1.59.2 Function

The MAC ADDRESS5 Low register holds the lower 32 bits of the sixth 6-byte MAC address of the station.

61.7.1.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.59.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS5 [31:0] This field contains the lower 32 bits of sixth 6-byte MAC address. MAC ADDRESS5 [31:0] This field contains the lower 32 bits of sixth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.60 MAC Address6 High (MAC_ADDRESS6_HIGH)

61.7.1.60.1 Offset

Register	Offset
MAC_ADDRESS6_HIGH	330h

61.7.1.60.2 Function

The MAC ADDRESS6 High register holds the upper 16 bits of the seventh 6-byte MAC address of the station.

The MAC ADDRESS6 High register holds the upper 16 bits of the seventh 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS6 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.60.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the seventh MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the seventh MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS6[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS6[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS6[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS6 registers. Each bit controls the masking of the bytes as follows:

Table continues on the next page...

Field	Function
	- Bit 29: ADDRESS6_High[15:8] - Bit 28: ADDRESS6_High[7:0] - Bit 27: ADDRESS6_Low[31:24] - .. - Bit 24: ADDRESS6_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS6 [47:32] This field contains the upper 16 bits[47:32] of the seventh 6-byte MAC address.

61.7.1.61 MAC Address6 Low (MAC_ADDRESS6_LOW)

61.7.1.61.1 Offset

Register	Offset
MAC_ADDRESS6_LOW	334h

61.7.1.61.2 Function

The MAC ADDRESS6 Low register holds the lower 32 bits of the seventh 6-byte MAC address of the station.

61.7.1.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.61.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS6 [31:0] This field contains the lower 32 bits of seventh 6-byte MAC address. MAC ADDRESS6 [31:0] This field contains the lower 32 bits of seventh 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.62 MAC Address7 High (MAC_ADDRESS7_HIGH)

61.7.1.62.1 Offset

Register	Offset
MAC_ADDRESS7_HIGH	338h

61.7.1.62.2 Function

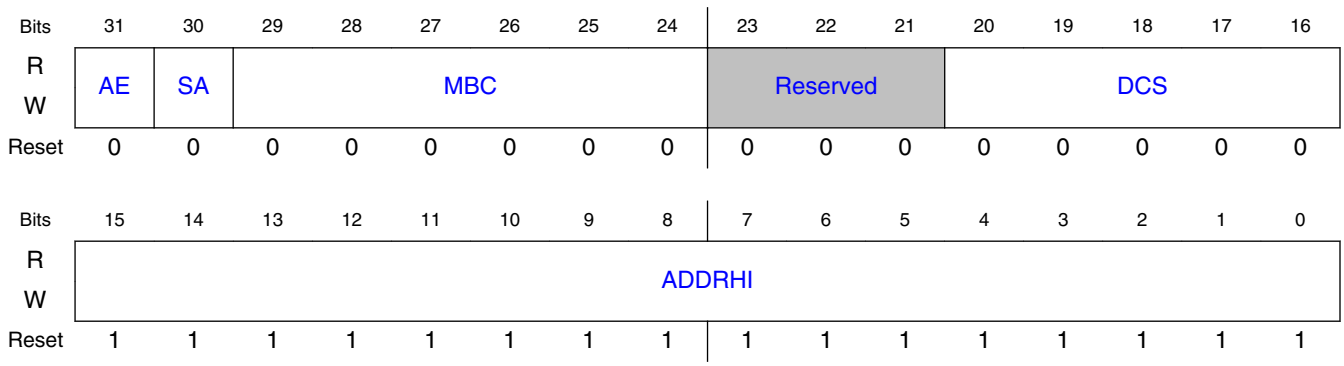
The MAC ADDRESS7 High register holds the upper 16 bits of the eighth 6-byte MAC address of the station.

The MAC ADDRESS7 High register holds the upper 16 bits of the eighth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS7 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.62.3 Diagram



61.7.1.62.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the eighth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the eighth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS7[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS7[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS7[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS7 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS7_High[15:8] - Bit 28: ADDRESS7_High[7:0] - Bit 27: ADDRESS7_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS7_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS7 [47:32] This field contains the upper 16 bits[47:32] of the eighth 6-byte MAC address.

61.7.1.63 MAC Address7 Low (MAC_ADDRESS7_LOW)

61.7.1.63.1 Offset

Register	Offset
MAC_ADDRESS7_LOW	33Ch

61.7.1.63.2 Function

The MAC ADDRESS7 Low register holds the lower 32 bits of the eighth 6-byte MAC address of the station.

61.7.1.63.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.63.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS7 [31:0] This field contains the lower 32 bits of eighth 6-byte MAC address. MAC ADDRESS7 [31:0] This field contains the lower 32 bits of eighth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.64 MAC Address8 High (MAC_ADDRESS8_HIGH)

61.7.1.64.1 Offset

Register	Offset
MAC_ADDRESS8_HIGH	340h

61.7.1.64.2 Function

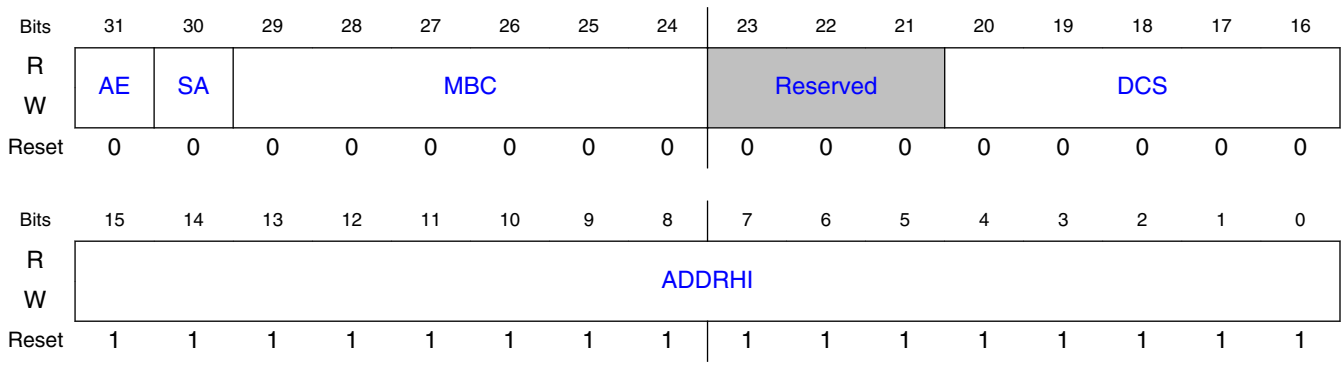
The MAC ADDRESS8 High register holds the upper 16 bits of the ninth 6-byte MAC address of the station.

The MAC ADDRESS8 High register holds the upper 16 bits of the ninth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS8 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.64.3 Diagram



61.7.1.64.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the ninth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the ninth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS8[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS8[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS8[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS8 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS9_High[15:8] - Bit 28: ADDRESS9_High[7:0] - Bit 27: ADDRESS9_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS9_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS8 [47:32] This field contains the upper 16 bits[47:32] of the ninth 6-byte MAC address.

61.7.1.65 MAC Address8 Low (MAC_ADDRESS8_LOW)

61.7.1.65.1 Offset

Register	Offset
MAC_ADDRESS8_LOW	344h

61.7.1.65.2 Function

The MAC ADDRESS8 Low register holds the lower 32 bits of the ninth 6-byte MAC address of the station.

61.7.1.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.65.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS8 [31:0] This field contains the lower 32 bits of ninth 6-byte MAC address. MAC ADDRESS8 [31:0] This field contains the lower 32 bits of ninth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.66 MAC Address9 High (MAC_ADDRESS9_HIGH)

61.7.1.66.1 Offset

Register	Offset
MAC_ADDRESS9_HIGH	348h

61.7.1.66.2 Function

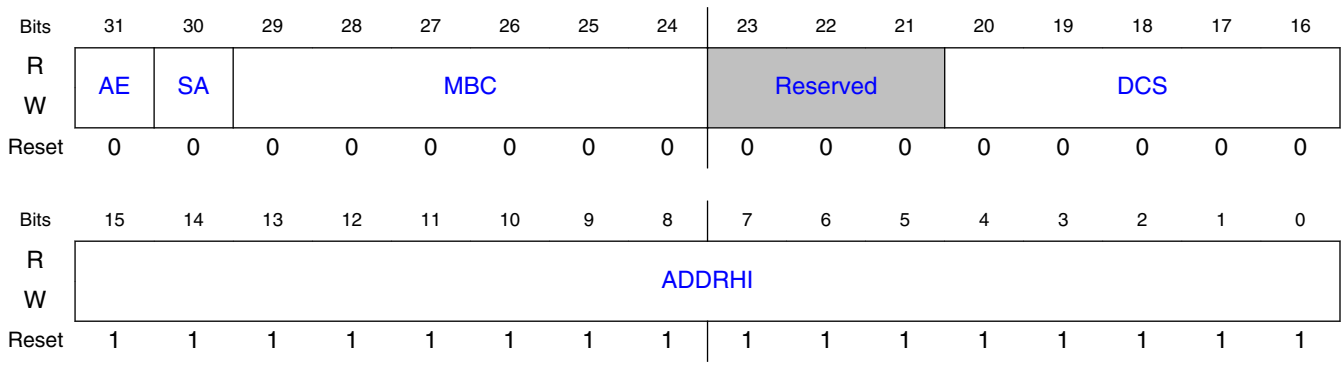
The MAC ADDRESS9 High register holds the upper 16 bits of the tenth 6-byte MAC address of the station.

The MAC ADDRESS9 High register holds the upper 16 bits of the tenth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS9 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.66.3 Diagram



61.7.1.66.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the tenth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the tenth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS9[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS9[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS9[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS9 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS9_High[15:8] - Bit 28: ADDRESS9_High[7:0] - Bit 27: ADDRESS9_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS9_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS9 [47:32] This field contains the upper 16 bits[47:32] of the tenth 6-byte MAC address.

61.7.1.67 MAC Address9 Low (MAC_ADDRESS9_LOW)

61.7.1.67.1 Offset

Register	Offset
MAC_ADDRESS9_LOW	34Ch

61.7.1.67.2 Function

The MAC ADDRESS9 Low register holds the lower 32 bits of the tenth 6-byte MAC address of the station.

61.7.1.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.67.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS9 [31:0] This field contains the lower 32 bits of tenth 6-byte MAC address. MAC ADDRESS9 [31:0] This field contains the lower 32 bits of tenth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.68 MAC Address10 High (MAC_ADDRESS10_HIGH)

61.7.1.68.1 Offset

Register	Offset
MAC_ADDRESS10_HIGH	350h

61.7.1.68.2 Function

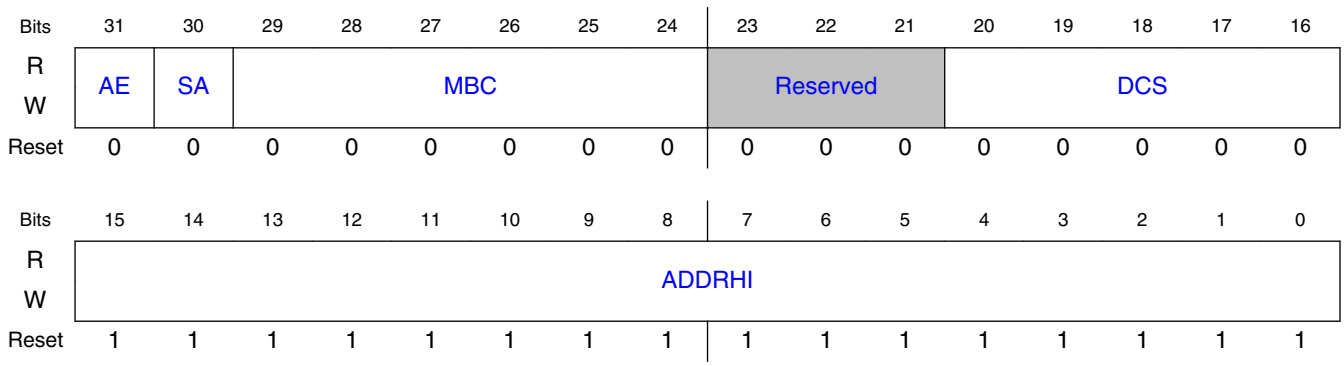
The MAC ADDRESS10 High register holds the upper 16 bits of the eleventh 6-byte MAC address of the station.

The MAC ADDRESS10 High register holds the upper 16 bits of the eleventh 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS10 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.68.3 Diagram



61.7.1.68.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the eleventh MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the eleventh MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS10[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS10[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS10[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS10 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS10_High[15:8] - Bit 28: ADDRESS10_High[7:0] - Bit 27: ADDRESS10_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS10_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS10 [47:32] This field contains the upper 16 bits[47:32] of the eleventh 6-byte MAC address.

61.7.1.69 MAC Address10 Low (MAC_ADDRESS10_LOW)

61.7.1.69.1 Offset

Register	Offset
MAC_ADDRESS10_LOW	354h

61.7.1.69.2 Function

The MAC ADDRESS10 Low register holds the lower 32 bits of the eleventh 6-byte MAC address of the station.

61.7.1.69.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.69.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS10 [31:0] This field contains the lower 32 bits of eleventh 6-byte MAC address. MAC ADDRESS10 [31:0] This field contains the lower 32 bits of eleventh 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.70 MAC Address11 High (MAC_ADDRESS11_HIGH)

61.7.1.70.1 Offset

Register	Offset
MAC_ADDRESS11_HIGH	358h

61.7.1.70.2 Function

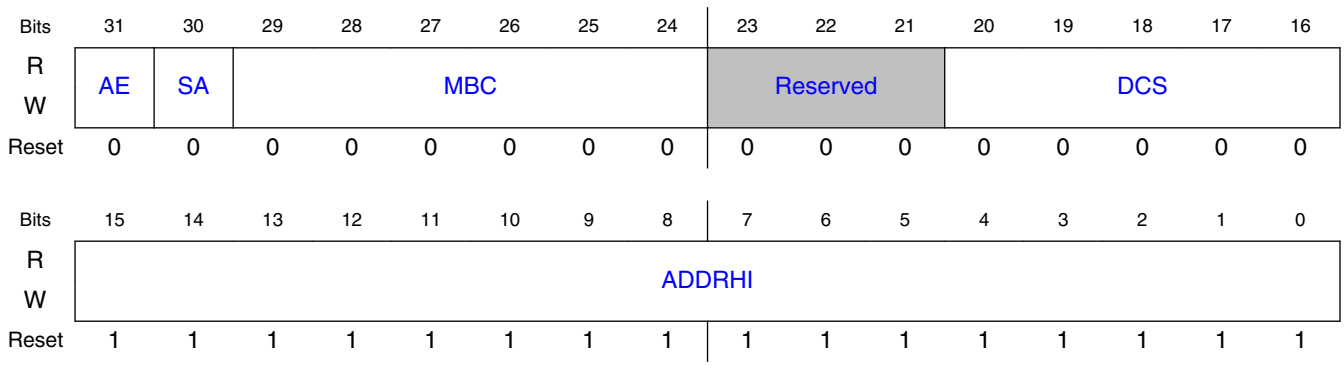
The MAC ADDRESS11 High register holds the upper 16 bits of the twelfth 6-byte MAC address of the station.

The MAC ADDRESS11 High register holds the upper 16 bits of the twelfth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS11 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.70.3 Diagram



61.7.1.70.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the twelfth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the twelfth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS11[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS11[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS11[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS11 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS11_High[15:8] - Bit 28: ADDRESS11_High[7:0] - Bit 27: ADDRESS11_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS11_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS11 [47:32] This field contains the upper 16 bits[47:32] of the twelfth 6-byte MAC address.

61.7.1.71 MAC Address11 Low (MAC_ADDRESS11_LOW)

61.7.1.71.1 Offset

Register	Offset
MAC_ADDRESS11_LOW	35Ch

61.7.1.71.2 Function

The MAC ADDRESS11 Low register holds the lower 32 bits of the twelfth 6-byte MAC address of the station.

61.7.1.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.71.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS11 [31:0] This field contains the lower 32 bits of twelfth 6-byte MAC address. MAC ADDRESS11 [31:0] This field contains the lower 32 bits of twelfth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.72 MAC Address12 High (MAC_ADDRESS12_HIGH)

61.7.1.72.1 Offset

Register	Offset
MAC_ADDRESS12_HIGH	360h

61.7.1.72.2 Function

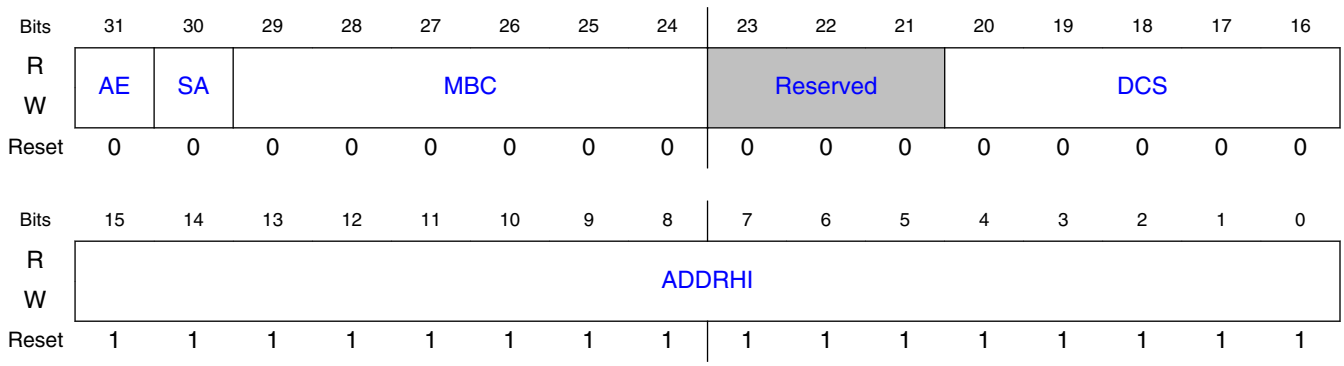
The MAC ADDRESS12 High register holds the upper 16 bits of the thirteenth 6-byte MAC address of the station.

The MAC ADDRESS12 High register holds the upper 16 bits of the thirteenth 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS12 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.72.3 Diagram



61.7.1.72.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the thirteenth MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the thirteenth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS12[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS12[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS12[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS12 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS12_High[15:8] - Bit 28: ADDRESS12_High[7:0] - Bit 27: ADDRESS12_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS12_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS12 [47:32] This field contains the upper 16 bits[47:32] of the thirteenth 6-byte MAC address.

61.7.1.73 MAC Address12 Low (MAC_ADDRESS12_LOW)

61.7.1.73.1 Offset

Register	Offset
MAC_ADDRESS12_LOW	364h

61.7.1.73.2 Function

The MAC ADDRESS12 Low register holds the lower 32 bits of the thirteenth 6-byte MAC address of the station.

61.7.1.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.73.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS12 [31:0] This field contains the lower 32 bits of thirteenth 6-byte MAC address. MAC ADDRESS12 [31:0] This field contains the lower 32 bits of thirteenth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.74 MAC Address13 High (MAC_ADDRESS13_HIGH)

61.7.1.74.1 Offset

Register	Offset
MAC_ADDRESS13_HIGH	368h

61.7.1.74.2 Function

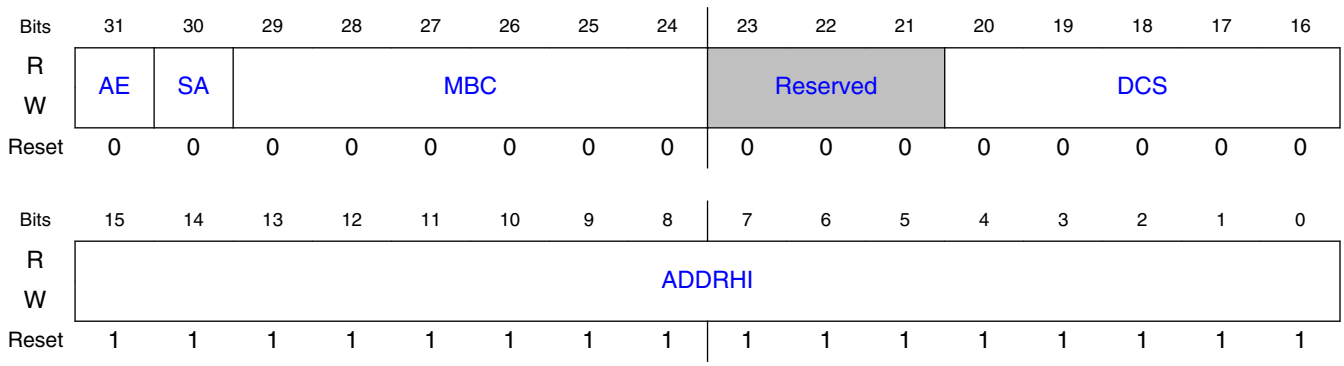
The MAC ADDRESS13 High register holds the upper 16 bits of the 14th 6-byte MAC address of the station.

The MAC ADDRESS13 High register holds the upper 16 bits of the 14th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS13 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.74.3 Diagram



61.7.1.74.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 14th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 14th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS13[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS13[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS13[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS13 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS13_High[15:8] - Bit 28: ADDRESS13_High[7:0] - Bit 27: ADDRESS13_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS13_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS13 [47:32] This field contains the upper 16 bits[47:32] of the 14th 6-byte MAC address.

61.7.1.75 MAC Address13 Low (MAC_ADDRESS13_LOW)

61.7.1.75.1 Offset

Register	Offset
MAC_ADDRESS13_LOW	36Ch

61.7.1.75.2 Function

The MAC ADDRESS13 Low register holds the lower 32 bits of the 14th 6-byte MAC address of the station.

61.7.1.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.75.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS13 [31:0] This field contains the lower 32 bits of 14th 6-byte MAC address. MAC ADDRESS13 [31:0] This field contains the lower 32 bits of 14th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.76 MAC Address14 High (MAC_ADDRESS14_HIGH)

61.7.1.76.1 Offset

Register	Offset
MAC_ADDRESS14_HIGH	370h

61.7.1.76.2 Function

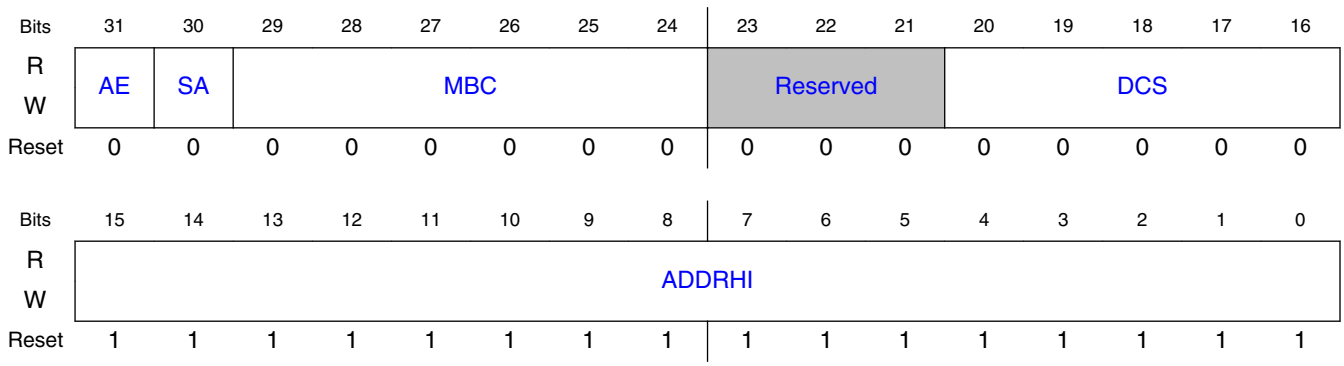
The MAC ADDRESS14 High register holds the upper 16 bits of the 15th 6-byte MAC address of the station.

The MAC ADDRESS14 High register holds the upper 16 bits of the 15th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS14 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.76.3 Diagram



61.7.1.76.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 15th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 15th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS14[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS14[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS14[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS14 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS14_High[15:8] - Bit 28: ADDRESS14_High[7:0] - Bit 27: ADDRESS14_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS14_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS14 [47:32] This field contains the upper 16 bits[47:32] of the 15th 6-byte MAC address.

61.7.1.77 MAC Address14 Low (MAC_ADDRESS14_LOW)

61.7.1.77.1 Offset

Register	Offset
MAC_ADDRESS14_LOW	374h

61.7.1.77.2 Function

The MAC ADDRESS14 Low register holds the lower 32 bits of the 15th 6-byte MAC address of the station.

61.7.1.77.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.77.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS14 [31:0] This field contains the lower 32 bits of 15th 6-byte MAC address. MAC ADDRESS14 [31:0] This field contains the lower 32 bits of 15th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.78 MAC Address15 High (MAC_ADDRESS15_HIGH)

61.7.1.78.1 Offset

Register	Offset
MAC_ADDRESS15_HIGH	378h

61.7.1.78.2 Function

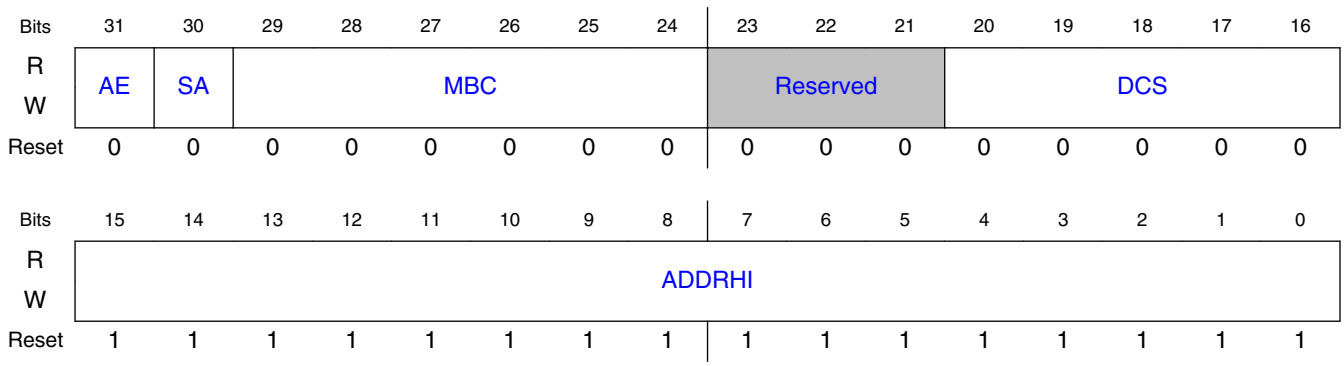
The MAC ADDRESS15 High register holds the upper 16 bits of the 16th 6-byte MAC address of the station.

The MAC ADDRESS15 High register holds the upper 16 bits of the 16th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS15 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.78.3 Diagram



61.7.1.78.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 16th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 16th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS15[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS15[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS15[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS15 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS15_High[15:8] - Bit 28: ADDRESS15_High[7:0] - Bit 27: ADDRESS15_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS15_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS15 [47:32] This field contains the upper 16 bits[47:32] of the 16th 6-byte MAC address.

61.7.1.79 MAC Address15 Low (MAC_ADDRESS15_LOW)

61.7.1.79.1 Offset

Register	Offset
MAC_ADDRESS15_LOW	37Ch

61.7.1.79.2 Function

The MAC ADDRESS15 Low register holds the lower 32 bits of the 16th 6-byte MAC address of the station.

61.7.1.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.79.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS15 [31:0] This field contains the lower 32 bits of 16th 6-byte MAC address. MAC ADDRESS15 [31:0] This field contains the lower 32 bits of 16th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.80 MAC Address16 High (MAC_ADDRESS16_HIGH)

61.7.1.80.1 Offset

Register	Offset
MAC_ADDRESS16_HIGH	380h

61.7.1.80.2 Function

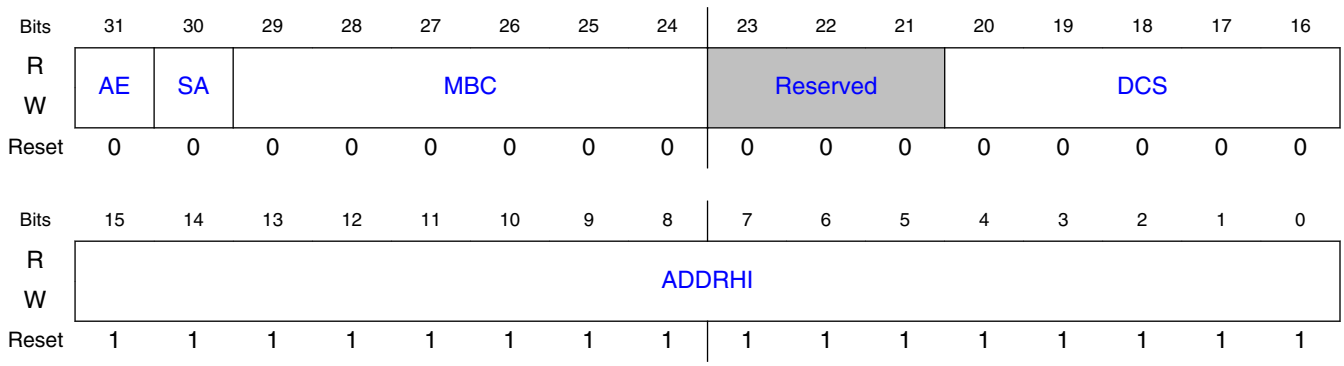
The MAC ADDRESS16 High register holds the upper 16 bits of the 17th 6-byte MAC address of the station.

The MAC ADDRESS16 High register holds the upper 16 bits of the 17th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS16 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.80.3 Diagram



61.7.1.80.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 17th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 17th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS16[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS16[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS16[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS16 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS16_High[15:8] - Bit 28: ADDRESS16_High[7:0] - Bit 27: ADDRESS16_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS16_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS16 [47:32] This field contains the upper 16 bits[47:32] of the 17th 6-byte MAC address.

61.7.1.81 MAC Address16 Low (MAC_ADDRESS16_LOW)

61.7.1.81.1 Offset

Register	Offset
MAC_ADDRESS16_LOW	384h

61.7.1.81.2 Function

The MAC ADDRESS16 Low register holds the lower 32 bits of the 17th 6-byte MAC address of the station.

61.7.1.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.81.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS16 [31:0] This field contains the lower 32 bits of 17th 6-byte MAC address. MAC ADDRESS16 [31:0] This field contains the lower 32 bits of 17th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.82 MAC Address17 High (MAC_ADDRESS17_HIGH)

61.7.1.82.1 Offset

Register	Offset
MAC_ADDRESS17_HIGH	388h

61.7.1.82.2 Function

The MAC ADDRESS17 High register holds the upper 16 bits of the 18th 6-byte MAC address of the station.

The MAC ADDRESS17 High register holds the upper 16 bits of the 18th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS17 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.82.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 18th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 18th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS17[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS17[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS17[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS17 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS17_High[15:8] - Bit 28: ADDRESS17_High[7:0] - Bit 27: ADDRESS17_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS17_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS17 [47:32] This field contains the upper 16 bits[47:32] of the 18th 6-byte MAC address.

61.7.1.83 MAC Address17 Low (MAC_ADDRESS17_LOW)

61.7.1.83.1 Offset

Register	Offset
MAC_ADDRESS17_LOW	38Ch

61.7.1.83.2 Function

The MAC ADDRESS17 Low register holds the lower 32 bits of the 18th 6-byte MAC address of the station.

61.7.1.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.83.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS17 [31:0] This field contains the lower 32 bits of 18th 6-byte MAC address. MAC ADDRESS17 [31:0] This field contains the lower 32 bits of 18th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.84 MAC Address18 High (MAC_ADDRESS18_HIGH)

61.7.1.84.1 Offset

Register	Offset
MAC_ADDRESS18_HIGH	390h

61.7.1.84.2 Function

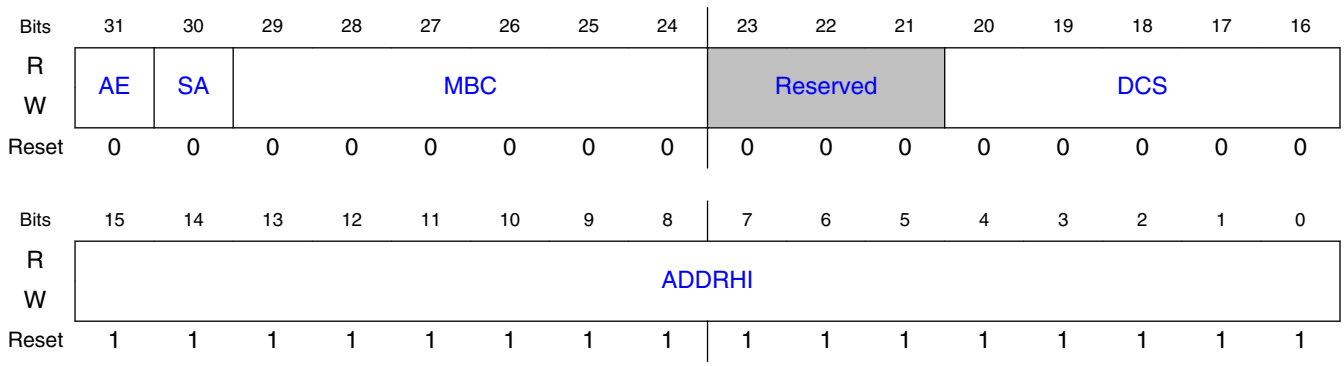
The MAC ADDRESS18 High register holds the upper 16 bits of the 19th 6-byte MAC address of the station.

The MAC ADDRESS18 High register holds the upper 16 bits of the 19th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS18 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.84.3 Diagram



61.7.1.84.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 19th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 19th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS18[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS18[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS18[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS18 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS18_High[15:8] - Bit 28: ADDRESS18_High[7:0] - Bit 27: ADDRESS18_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS18_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS18 [47:32] This field contains the upper 16 bits[47:32] of the 19th 6-byte MAC address.

61.7.1.85 MAC Address18 Low (MAC_ADDRESS18_LOW)

61.7.1.85.1 Offset

Register	Offset
MAC_ADDRESS18_LOW	394h

61.7.1.85.2 Function

The MAC ADDRESS18 Low register holds the lower 32 bits of the 19th 6-byte MAC address of the station.

61.7.1.85.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.85.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS18 [31:0] This field contains the lower 32 bits of 19th 6-byte MAC address. MAC ADDRESS18 [31:0] This field contains the lower 32 bits of 19th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.86 MAC Address19 High (MAC_ADDRESS19_HIGH)

61.7.1.86.1 Offset

Register	Offset
MAC_ADDRESS19_HIGH	398h

61.7.1.86.2 Function

The MAC ADDRESS19 High register holds the upper 16 bits of the 20th 6-byte MAC address of the station.

The MAC ADDRESS19 High register holds the upper 16 bits of the 20th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS19 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.86.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.86.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 20th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 20th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS19[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS19[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS19[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS19 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS19_High[15:8] - Bit 28: ADDRESS19_High[7:0] - Bit 27: ADDRESS19_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS19_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS19 [47:32] This field contains the upper 16 bits[47:32] of the 20th 6-byte MAC address.

61.7.1.87 MAC Address19 Low (MAC_ADDRESS19_LOW)

61.7.1.87.1 Offset

Register	Offset
MAC_ADDRESS19_LOW	39Ch

61.7.1.87.2 Function

The MAC ADDRESS19 Low register holds the lower 32 bits of the 20th 6-byte MAC address of the station.

61.7.1.87.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.87.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS19 [31:0] This field contains the lower 32 bits of 20th 6-byte MAC address. MAC ADDRESS19 [31:0] This field contains the lower 32 bits of 20th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.88 MAC Address20 High (MAC_ADDRESS20_HIGH)

61.7.1.88.1 Offset

Register	Offset
MAC_ADDRESS20_HIGH	3A0h

61.7.1.88.2 Function

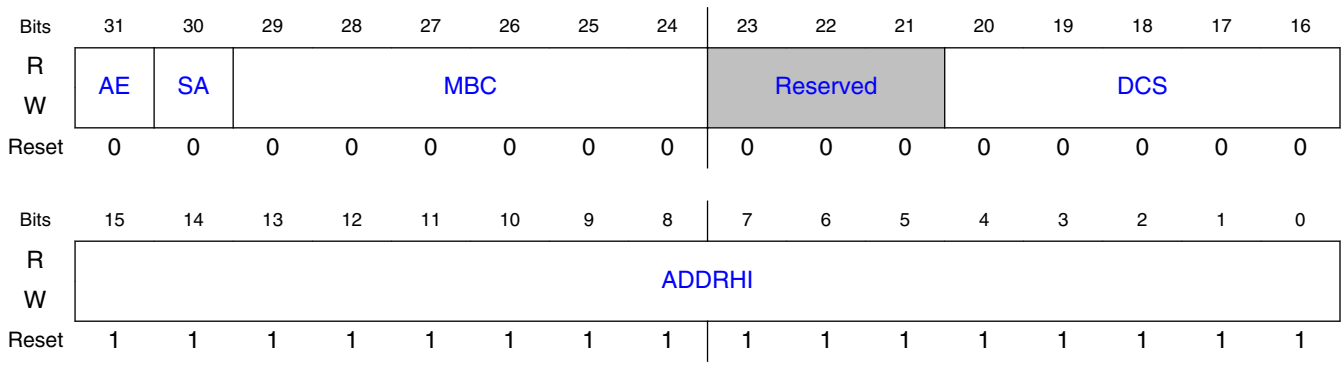
The MAC ADDRESS20 High register holds the upper 16 bits of the 21st 6-byte MAC address of the station.

The MAC ADDRESS20 High register holds the upper 16 bits of the 21st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS20 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.88.3 Diagram



61.7.1.88.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 21st MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 21st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS20[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS20[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS20[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS20 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS20_High[15:8] - Bit 28: ADDRESS20_High[7:0] - Bit 27: ADDRESS20_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS20_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS20 [47:32] This field contains the upper 16 bits[47:32] of the 21st 6-byte MAC address.

61.7.1.89 MAC Address20 Low (MAC_ADDRESS20_LOW)

61.7.1.89.1 Offset

Register	Offset
MAC_ADDRESS20_LOW	3A4h

61.7.1.89.2 Function

The MAC ADDRESS20 Low register holds the lower 32 bits of the 21st 6-byte MAC address of the station.

61.7.1.89.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.89.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS20 [31:0] This field contains the lower 32 bits of 21st 6-byte MAC address. MAC ADDRESS20 [31:0] This field contains the lower 32 bits of 21st 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.90 MAC Address21 High (MAC_ADDRESS21_HIGH)

61.7.1.90.1 Offset

Register	Offset
MAC_ADDRESS21_HIGH	3A8h

61.7.1.90.2 Function

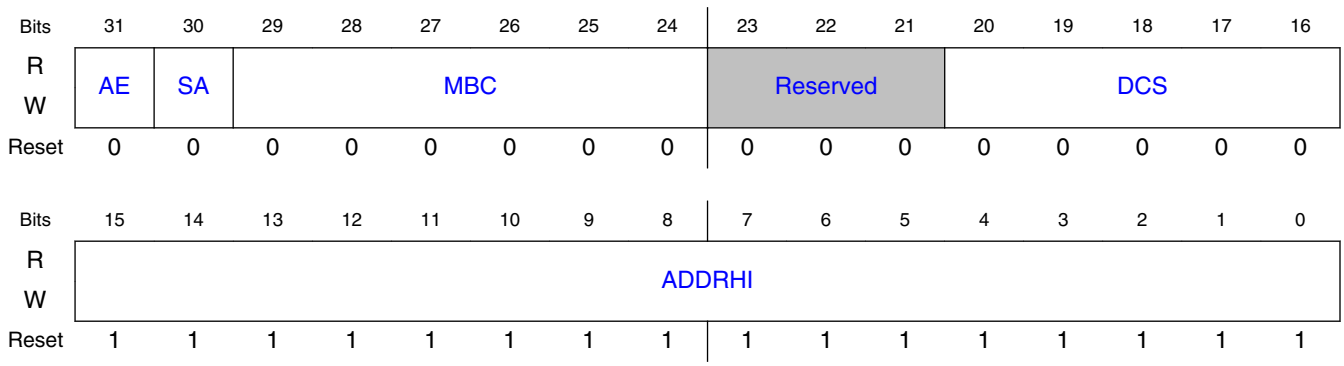
The MAC ADDRESS21 High register holds the upper 16 bits of the 22nd 6-byte MAC address of the station.

The MAC ADDRESS21 High register holds the upper 16 bits of the 22nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS21 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.90.3 Diagram



61.7.1.90.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 22nd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 22nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS21[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS21[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS21[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS21 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS21_High[15:8] - Bit 28: ADDRESS21_High[7:0] - Bit 27: ADDRESS21_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS21_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS21 [47:32] This field contains the upper 16 bits[47:32] of the 22nd 6-byte MAC address.

61.7.1.91 MAC Address21 Low (MAC_ADDRESS21_LOW)

61.7.1.91.1 Offset

Register	Offset
MAC_ADDRESS21_LOW	3ACh

61.7.1.91.2 Function

The MAC ADDRESS21 Low register holds the lower 32 bits of the 22nd 6-byte MAC address of the station.

61.7.1.91.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.91.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS21 [31:0] This field contains the lower 32 bits of 22nd 6-byte MAC address. MAC ADDRESS21 [31:0] This field contains the lower 32 bits of 22nd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.92 MAC Address22 High (MAC_ADDRESS22_HIGH)

61.7.1.92.1 Offset

Register	Offset
MAC_ADDRESS22_HIGH	3B0h

61.7.1.92.2 Function

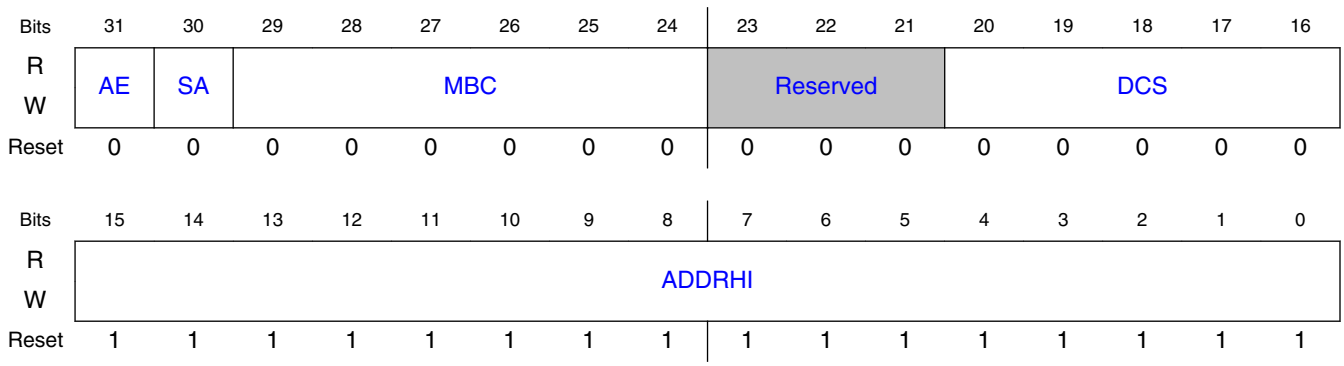
The MAC ADDRESS22 High register holds the upper 16 bits of the 23rd 6-byte MAC address of the station.

The MAC ADDRESS22 High register holds the upper 16 bits of the 23rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS22 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.92.3 Diagram



61.7.1.92.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 23rd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 23rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS22[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS22[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS22[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS22 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS22_High[15:8] - Bit 28: ADDRESS22_High[7:0] - Bit 27: ADDRESS22_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS22_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS22 [47:32] This field contains the upper 16 bits[47:32] of the 23rd 6-byte MAC address.

61.7.1.93 MAC Address22 Low (MAC_ADDRESS22_LOW)

61.7.1.93.1 Offset

Register	Offset
MAC_ADDRESS22_LOW	3B4h

61.7.1.93.2 Function

The MAC ADDRESS22 Low register holds the lower 32 bits of the 23rd 6-byte MAC address of the station.

61.7.1.93.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.93.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS22 [31:0] This field contains the lower 32 bits of 23rd 6-byte MAC address. MAC ADDRESS22 [31:0] This field contains the lower 32 bits of 23rd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.94 MAC Address23 High (MAC_ADDRESS23_HIGH)

61.7.1.94.1 Offset

Register	Offset
MAC_ADDRESS23_HIGH	3B8h

61.7.1.94.2 Function

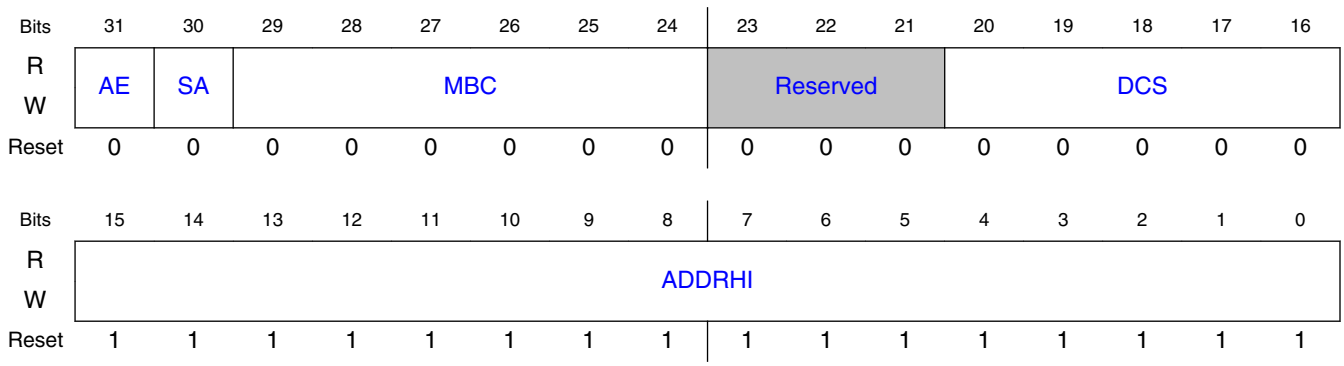
The MAC ADDRESS23 High register holds the upper 16 bits of the 24th 6-byte MAC address of the station.

The MAC ADDRESS23 High register holds the upper 16 bits of the 24th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS23 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.94.3 Diagram



61.7.1.94.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 24th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 24th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS23[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS23[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS23[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS23 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS23_High[15:8] - Bit 28: ADDRESS23_High[7:0] - Bit 27: ADDRESS23_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS23_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS23 [47:32] This field contains the upper 16 bits[47:32] of the 24th 6-byte MAC address.

61.7.1.95 MAC Address23 Low (MAC_ADDRESS23_LOW)

61.7.1.95.1 Offset

Register	Offset
MAC_ADDRESS23_LOW	3BCh

61.7.1.95.2 Function

The MAC ADDRESS23 Low register holds the lower 32 bits of the 24th 6-byte MAC address of the station.

61.7.1.95.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.95.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS23 [31:0] This field contains the lower 32 bits of 24th 6-byte MAC address. MAC ADDRESS23 [31:0] This field contains the lower 32 bits of 24th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.96 MAC Address24 High (MAC_ADDRESS24_HIGH)

61.7.1.96.1 Offset

Register	Offset
MAC_ADDRESS24_HIGH	3C0h

61.7.1.96.2 Function

The MAC ADDRESS24 High register holds the upper 16 bits of the 25th 6-byte MAC address of the station.

The MAC ADDRESS24 High register holds the upper 16 bits of the 25th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS24 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.96.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.96.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 25th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 25th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS24[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS24[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS24[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS24 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS24_High[15:8] - Bit 28: ADDRESS24_High[7:0] - Bit 27: ADDRESS24_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS24_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS24 [47:32] This field contains the upper 16 bits[47:32] of the 25th 6-byte MAC address.

61.7.1.97 MAC Address24 Low (MAC_ADDRESS24_LOW)

61.7.1.97.1 Offset

Register	Offset
MAC_ADDRESS24_LOW	3C4h

61.7.1.97.2 Function

The MAC ADDRESS24 Low register holds the lower 32 bits of the 25th 6-byte MAC address of the station.

61.7.1.97.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.97.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS24 [31:0] This field contains the lower 32 bits of 25th 6-byte MAC address. MAC ADDRESS24 [31:0] This field contains the lower 32 bits of 25th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.98 MAC Address25 High (MAC_ADDRESS25_HIGH)

61.7.1.98.1 Offset

Register	Offset
MAC_ADDRESS25_HIGH	3C8h

61.7.1.98.2 Function

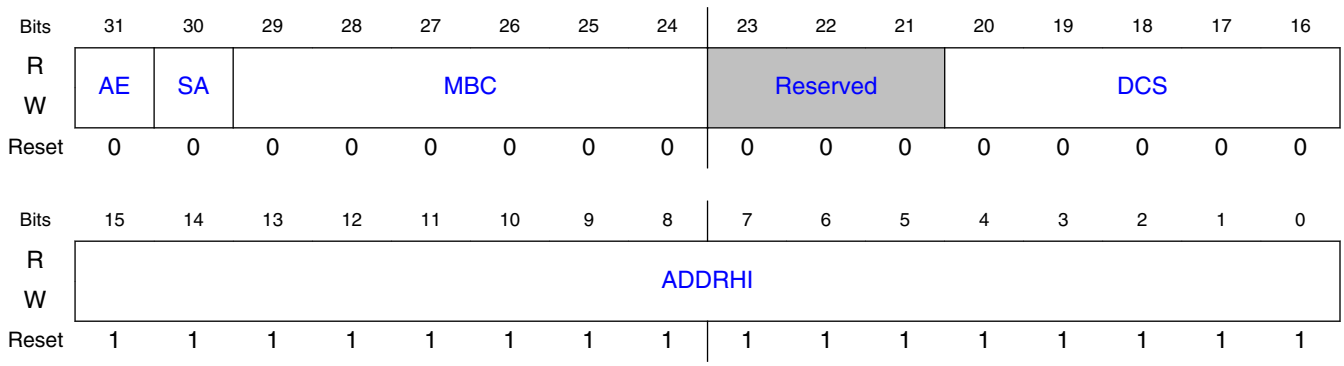
The MAC ADDRESS25 High register holds the upper 16 bits of the 26th 6-byte MAC address of the station.

The MAC ADDRESS25 High register holds the upper 16 bits of the 26th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS25 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.98.3 Diagram



61.7.1.98.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 26th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 26th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS25[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS25[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS25[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS25 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS25_High[15:8] - Bit 28: ADDRESS25_High[7:0] - Bit 27: ADDRESS25_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS25_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS25 [47:32] This field contains the upper 16 bits[47:32] of the 26th 6-byte MAC address.

61.7.1.99 MAC Address25 Low (MAC_ADDRESS25_LOW)

61.7.1.99.1 Offset

Register	Offset
MAC_ADDRESS25_LOW	3CCh

61.7.1.99.2 Function

The MAC ADDRESS25 Low register holds the lower 32 bits of the 26th 6-byte MAC address of the station.

61.7.1.99.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.99.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS25 [31:0] This field contains the lower 32 bits of 26th 6-byte MAC address. MAC ADDRESS25 [31:0] This field contains the lower 32 bits of 26th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.100 MAC Address26 High (MAC_ADDRESS26_HIGH)

61.7.1.100.1 Offset

Register	Offset
MAC_ADDRESS26_HIGH	3D0h

61.7.1.100.2 Function

The MAC ADDRESS26 High register holds the upper 16 bits of the 27th 6-byte MAC address of the station.

The MAC ADDRESS26 High register holds the upper 16 bits of the 27th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS26 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.100.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.100.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 27th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 27th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS26[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS26[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS26[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS26 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS26_High[15:8] - Bit 28: ADDRESS26_High[7:0] - Bit 27: ADDRESS26_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS26_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS26 [47:32] This field contains the upper 16 bits[47:32] of the 27th 6-byte MAC address.

61.7.1.101 MAC Address26 Low (MAC_ADDRESS26_LOW)

61.7.1.101.1 Offset

Register	Offset
MAC_ADDRESS26_LOW	3D4h

61.7.1.101.2 Function

The MAC ADDRESS26 Low register holds the lower 32 bits of the 27th 6-byte MAC address of the station.

61.7.1.101.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.101.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS26 [31:0] This field contains the lower 32 bits of 27th 6-byte MAC address. MAC ADDRESS26 [31:0] This field contains the lower 32 bits of 27th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.102 MAC Address27 High (MAC_ADDRESS27_HIGH)

61.7.1.102.1 Offset

Register	Offset
MAC_ADDRESS27_HIGH	3D8h

61.7.1.102.2 Function

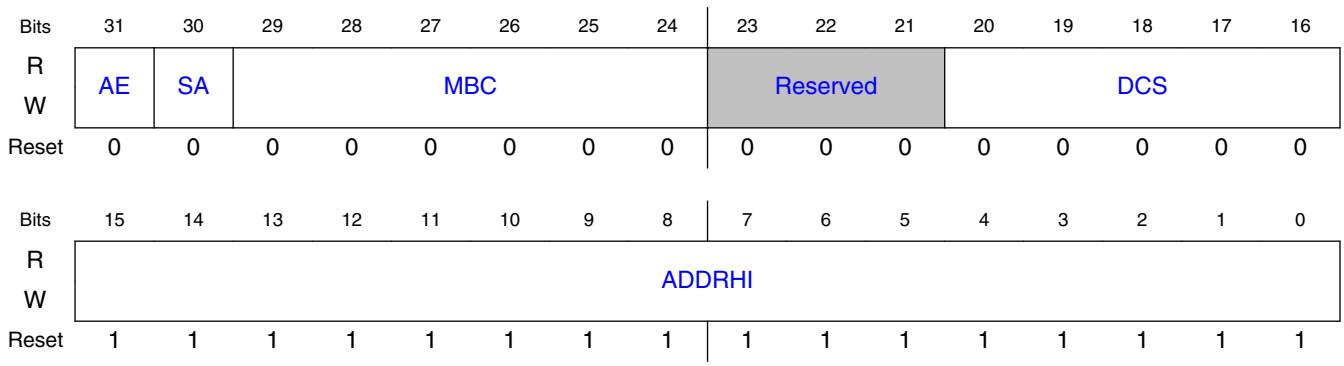
The MAC ADDRESS27 High register holds the upper 16 bits of the 28th 6-byte MAC address of the station.

The MAC ADDRESS27 High register holds the upper 16 bits of the 28th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS27 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.102.3 Diagram



61.7.1.102.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 28th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 28th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS27[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS27[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS27[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS27 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS27_High[15:8] - Bit 28: ADDRESS27_High[7:0] - Bit 27: ADDRESS27_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS27_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS27 [47:32] This field contains the upper 16 bits[47:32] of the 28th 6-byte MAC address.

61.7.1.103 MAC Address27 Low (MAC_ADDRESS27_LOW)

61.7.1.103.1 Offset

Register	Offset
MAC_ADDRESS27_LOW	3DCh

61.7.1.103.2 Function

The MAC ADDRESS27 Low register holds the lower 32 bits of the 28th 6-byte MAC address of the station.

61.7.1.103.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.103.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS27 [31:0] This field contains the lower 32 bits of 28th 6-byte MAC address. MAC ADDRESS27 [31:0] This field contains the lower 32 bits of 28th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.104 MAC Address28 High (MAC_ADDRESS28_HIGH)

61.7.1.104.1 Offset

Register	Offset
MAC_ADDRESS28_HIGH	3E0h

61.7.1.104.2 Function

The MAC ADDRESS28 High register holds the upper 16 bits of the 29th 6-byte MAC address of the station.

The MAC ADDRESS28 High register holds the upper 16 bits of the 29th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS28 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.104.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.104.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 29th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 29th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS28[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS28[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS28[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS28 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS28_High[15:8] - Bit 28: ADDRESS28_High[7:0] - Bit 27: ADDRESS28_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS28_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS28 [47:32] This field contains the upper 16 bits[47:32] of the 29th 6-byte MAC address.

61.7.1.105 MAC Address28 Low (MAC_ADDRESS28_LOW)

61.7.1.105.1 Offset

Register	Offset
MAC_ADDRESS28_LOW	3E4h

61.7.1.105.2 Function

The MAC ADDRESS28 Low register holds the lower 32 bits of the 29th 6-byte MAC address of the station.

61.7.1.105.3 Diagram

61.7.1.105.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS28 [31:0] This field contains the lower 32 bits of 29th 6-byte MAC address. MAC ADDRESS28 [31:0] This field contains the lower 32 bits of 29th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.106 MAC Address29 High (MAC_ADDRESS29_HIGH)

61.7.1.106.1 Offset

Register	Offset
MAC_ADDRESS29_HIGH	3E8h

61.7.1.106.2 Function

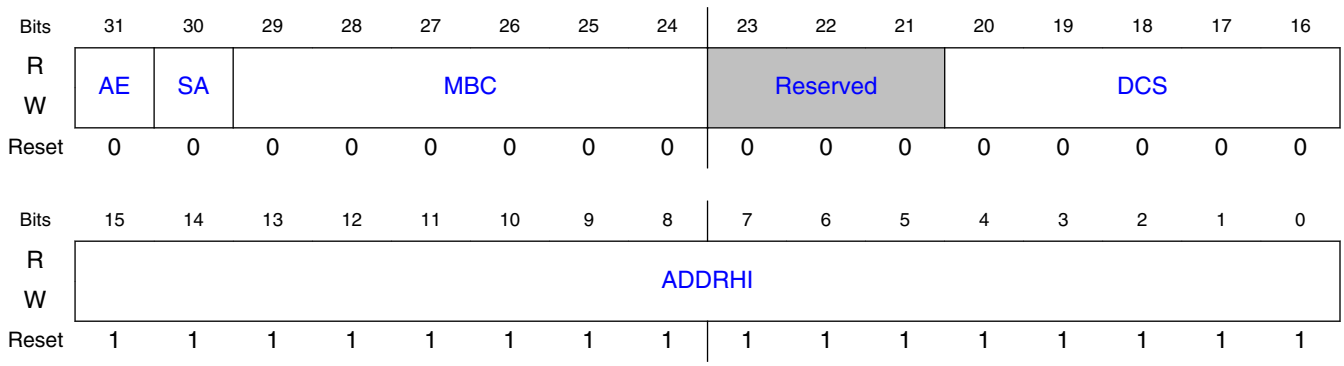
The MAC ADDRESS29 High register holds the upper 16 bits of the 30th 6-byte MAC address of the station.

The MAC ADDRESS29 High register holds the upper 16 bits of the 30th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS29 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.106.3 Diagram



61.7.1.106.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 30th MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 30th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS29[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS29[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS29[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS29 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS29_High[15:8] - Bit 28: ADDRESS29_High[7:0] - Bit 27: ADDRESS29_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS29_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS29 [47:32] This field contains the upper 16 bits[47:32] of the 30th 6-byte MAC address.

61.7.1.107 MAC Address29 Low (MAC_ADDRESS29_LOW)

61.7.1.107.1 Offset

Register	Offset
MAC_ADDRESS29_LOW	3ECh

61.7.1.107.2 Function

The MAC ADDRESS29 Low register holds the lower 32 bits of the 30th 6-byte MAC address of the station.

61.7.1.107.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.107.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS29 [31:0] This field contains the lower 32 bits of 30th 6-byte MAC address. MAC ADDRESS29 [31:0] This field contains the lower 32 bits of 30th 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.108 MAC Address30 High (MAC_ADDRESS30_HIGH)

61.7.1.108.1 Offset

Register	Offset
MAC_ADDRESS30_HIGH	3F0h

61.7.1.108.2 Function

The MAC ADDRESS30 High register holds the upper 16 bits of the 31st 6-byte MAC address of the station.

The MAC ADDRESS30 High register holds the upper 16 bits of the 31st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS30 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.108.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	SA	MBC						Reserved			DCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.108.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 31st MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 31st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS30[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS30[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS30[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS30 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS30_High[15:8] - Bit 28: ADDRESS30_High[7:0] - Bit 27: ADDRESS30_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS30_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS30 [47:32] This field contains the upper 16 bits[47:32] of the 31st 6-byte MAC address.

61.7.1.109 MAC Address30 Low (MAC_ADDRESS30_LOW)

61.7.1.109.1 Offset

Register	Offset
MAC_ADDRESS30_LOW	3F4h

61.7.1.109.2 Function

The MAC ADDRESS30 Low register holds the lower 32 bits of the 31st 6-byte MAC address of the station.

61.7.1.109.3 Diagram

61.7.1.109.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS30 [31:0] This field contains the lower 32 bits of 31st 6-byte MAC address. MAC ADDRESS30 [31:0] This field contains the lower 32 bits of 31st 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.110 MAC Address31 High (MAC_ADDRESS31_HIGH)

61.7.1.110.1 Offset

Register	Offset
MAC_ADDRESS31_HIGH	3F8h

61.7.1.110.2 Function

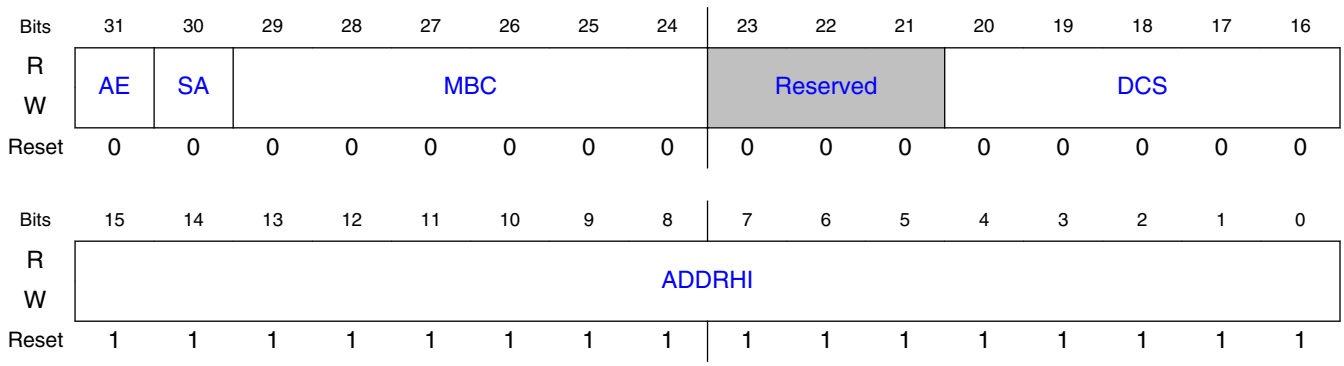
The MAC ADDRESS31 High register holds the upper 16 bits of the 32nd 6-byte MAC address of the station.

The MAC ADDRESS31 High register holds the upper 16 bits of the 32nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC ADDRESS31 Low Register are

written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

61.7.1.110.3 Diagram



61.7.1.110.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the address filter module uses the 32nd MAC address for perfect filtering. Address Enable When this bit is set, the address filter module uses the 32nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30 SA	Source Address When this bit is set, the MAC ADDRESS31[47:0] is used to compare with the SA fields of the received packet. Source Address When this bit is set, the MAC ADDRESS31[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC ADDRESS31[47:0] is used to compare with the DA fields of the received packet. 0b - Compare with Destination Address 1b - Compare with Source Address
29-24 MBC	Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. Mask Byte Control These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC ADDRESS31 registers. Each bit controls the masking of the bytes as follows: - Bit 29: ADDRESS31_High[15:8] - Bit 28: ADDRESS31_High[7:0] - Bit 27: ADDRESS31_Low[31:24]

Table continues on the next page...

Field	Function
	- .. - Bit 24: ADDRESS31_Low[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.
23-21 —	Reserved
20-16 DCS	DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. DMA Channel Select If the PDC bit of MAC_EXT_CONFIGURATION register is not set: This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address(#i) content is routed. If the PDC bit of MAC_EXT_CONFIGURATION register is set: This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.
15-0 ADDRHI	MAC ADDRESS31 [47:32] This field contains the upper 16 bits[47:32] of the 32nd 6-byte MAC address.

61.7.1.111 MAC Address31 Low (MAC_ADDRESS31_LOW)

61.7.1.111.1 Offset

Register	Offset
MAC_ADDRESS31_LOW	3FCh

61.7.1.111.2 Function

The MAC ADDRESS31 Low register holds the lower 32 bits of the 32nd 6-byte MAC address of the station.

61.7.1.111.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.111.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS31 [31:0] This field contains the lower 32 bits of 32nd 6-byte MAC address. MAC ADDRESS31 [31:0] This field contains the lower 32 bits of 32nd 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.

61.7.1.112 MAC Address32 High (MAC_ADDRESS32_HIGH)

61.7.1.112.1 Offset

Register	Offset
MAC_ADDRESS32_HIGH	400h

61.7.1.112.2 Function

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

The MAC ADDRESS32 High register holds the upper 16 bits of the 33rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.112.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.112.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 33rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS32 content is routed.
15-0 ADDRHI	MAC ADDRESS32 [47:32] This field contains the upper 16 bits (47:32) of the 33rd 6-byte MAC address.

61.7.1.113 MAC Address32 Low (MAC_ADDRESS32_LOW)

61.7.1.113.1 Offset

Register	Offset
MAC_ADDRESS32_LOW	404h

61.7.1.113.2 Function

The MAC ADDRESS32 Low register holds the lower 16 bits of the 33rd 6-byte MAC address of the station.

61.7.1.113.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.113.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. MAC ADDRESS32 [31:0] This field contains the lower 32 bits of the 33rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.114 MAC Address33 High (MAC_ADDRESS33_HIGH)

61.7.1.114.1 Offset

Register	Offset
MAC_ADDRESS33_HIGH	408h

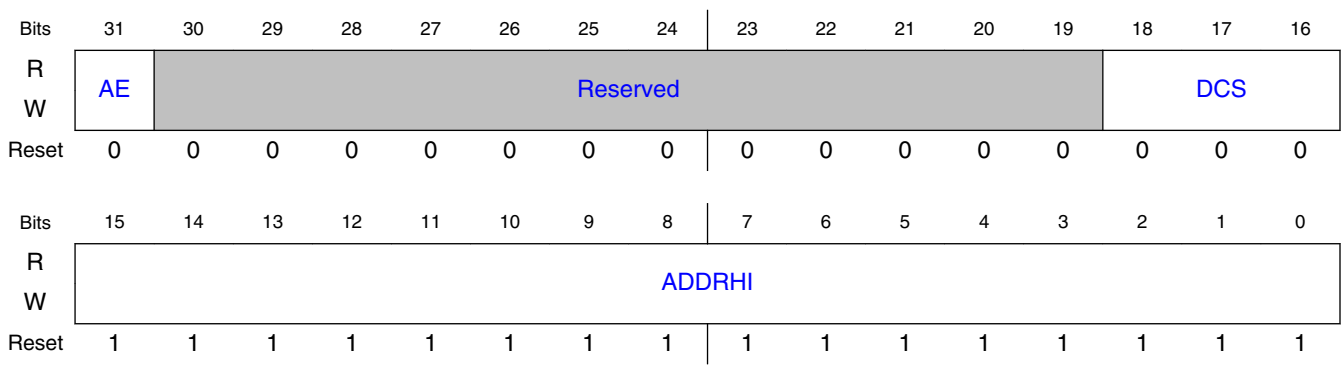
61.7.1.114.2 Function

The MAC ADDRESS33 High register holds the upper 16 bits of the 34th 6-byte MAC address of the station.

The MAC ADDRESS33 High register holds the upper 16 bits of the 34th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.114.3 Diagram



61.7.1.114.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 34th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 34th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS33 content is routed.
15-0 ADDRHI	MAC ADDRESS33 [47:32] This field contains the upper 16 bits (47:32) of the 34th 6-byte MAC address.

61.7.1.115 MAC Address33 Low (MAC_ADDRESS33_LOW)

61.7.1.115.1 Offset

Register	Offset
MAC_ADDRESS33_LOW	40Ch

61.7.1.115.2 Function

The MAC ADDRESS33 Low register holds the lower 16 bits of the 34th 6-byte MAC address of the station.

61.7.1.115.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.115.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS33 [31:0] This field contains the lower 32 bits of the 34th 6-byte MAC address. MAC ADDRESS33 [31:0] This field contains the lower 32 bits of the 34th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.116 MAC Address34 High (MAC_ADDRESS34_HIGH)

61.7.1.116.1 Offset

Register	Offset
MAC_ADDRESS34_HIGH	410h

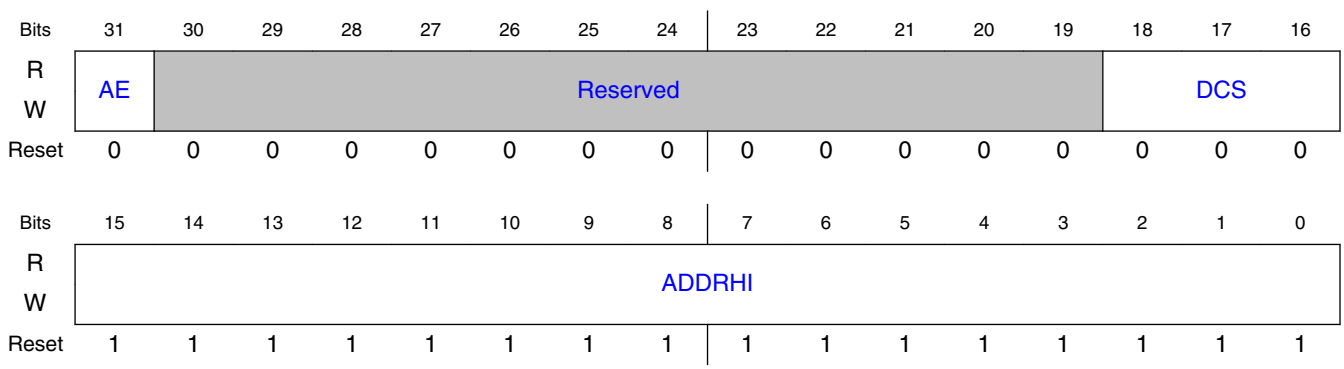
61.7.1.116.2 Function

The MAC ADDRESS34 High register holds the upper 16 bits of the 35th 6-byte MAC address of the station.

The MAC ADDRESS34 High register holds the upper 16 bits of the 35th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.116.3 Diagram



61.7.1.116.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 35th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 35th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS34 content is routed.
15-0 ADDRHI	MAC ADDRESS34 [47:32] This field contains the upper 16 bits (47:32) of the 35th 6-byte MAC address.

61.7.1.117 MAC Address34 Low (MAC_ADDRESS34_LOW)

61.7.1.117.1 Offset

Register	Offset
MAC_ADDRESS34_LOW	414h

61.7.1.117.2 Function

The MAC ADDRESS34 Low register holds the lower 16 bits of the 35th 6-byte MAC address of the station.

61.7.1.117.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.117.4 Fields

Field	Function
31-0	MAC ADDRESS34 [31:0] This field contains the lower 32 bits of the 35th 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS34 [31:0] This field contains the lower 32 bits of the 35th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.118 MAC Address35 High (MAC_ADDRESS35_HIGH)

61.7.1.118.1 Offset

Register	Offset
MAC_ADDRESS35_HIGH	418h

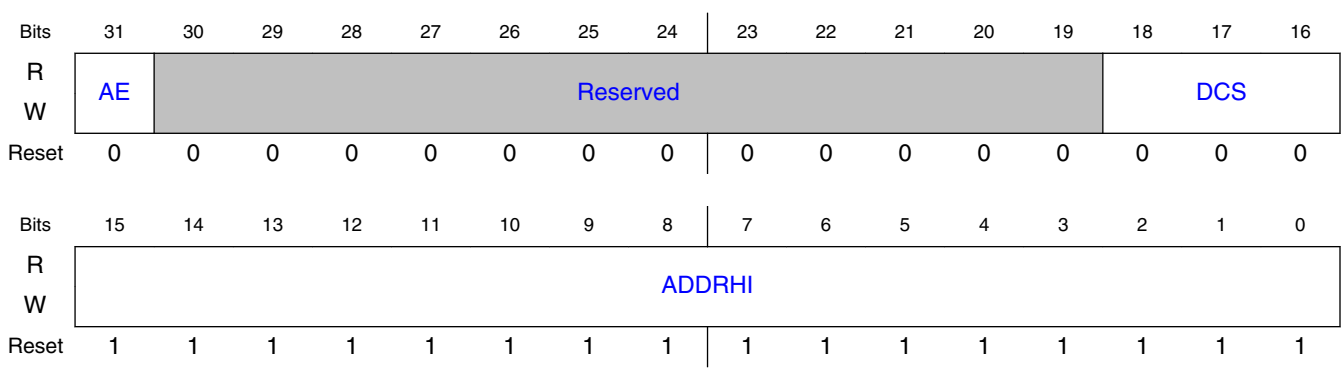
61.7.1.118.2 Function

The MAC ADDRESS35 High register holds the upper 16 bits of the 36th 6-byte MAC address of the station.

The MAC ADDRESS35 High register holds the upper 16 bits of the 36th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.118.3 Diagram



61.7.1.118.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 36th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 36th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS35 content is routed.
15-0 ADDRHI	MAC ADDRESS35 [47:32] This field contains the upper 16 bits (47:32) of the 36th 6-byte MAC address.

61.7.1.119 MAC Address35 Low (MAC_ADDRESS35_LOW)

61.7.1.119.1 Offset

Register	Offset
MAC_ADDRESS35_LOW	41Ch

61.7.1.119.2 Function

The MAC ADDRESS35 Low register holds the lower 16 bits of the 36th 6-byte MAC address of the station.

61.7.1.119.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.119.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS35 [31:0] This field contains the lower 32 bits of the 36th 6-byte MAC address. MAC ADDRESS35 [31:0] This field contains the lower 32 bits of the 36th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.120 MAC Address36 High (MAC_ADDRESS36_HIGH)

61.7.1.120.1 Offset

Register	Offset
MAC_ADDRESS36_HIGH	420h

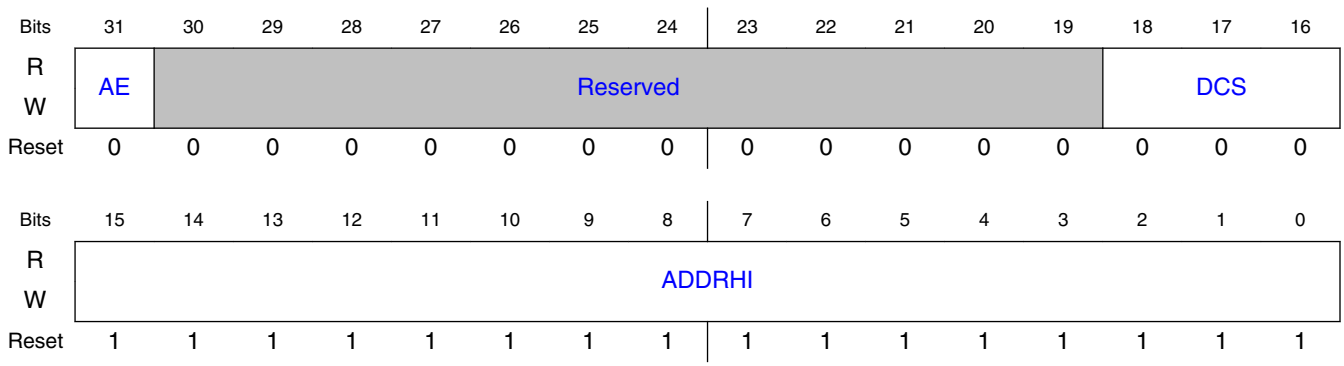
61.7.1.120.2 Function

The MAC ADDRESS36 High register holds the upper 16 bits of the 37th 6-byte MAC address of the station.

The MAC ADDRESS36 High register holds the upper 16 bits of the 37th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.120.3 Diagram



61.7.1.120.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 37th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 37th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS36 content is routed.
15-0 ADDRHI	MAC ADDRESS36 [47:32] This field contains the upper 16 bits (47:32) of the 37th 6-byte MAC address.

61.7.1.121 MAC Address36 Low (MAC_ADDRESS36_LOW)

61.7.1.121.1 Offset

Register	Offset
MAC_ADDRESS36_LOW	424h

61.7.1.121.2 Function

The MAC ADDRESS36 Low register holds the lower 16 bits of the 37th 6-byte MAC address of the station.

61.7.1.121.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.121.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS36 [31:0] This field contains the lower 32 bits of the 37th 6-byte MAC address. MAC ADDRESS36 [31:0] This field contains the lower 32 bits of the 37th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.122 MAC Address37 High (MAC_ADDRESS37_HIGH)

61.7.1.122.1 Offset

Register	Offset
MAC_ADDRESS37_HIGH	428h

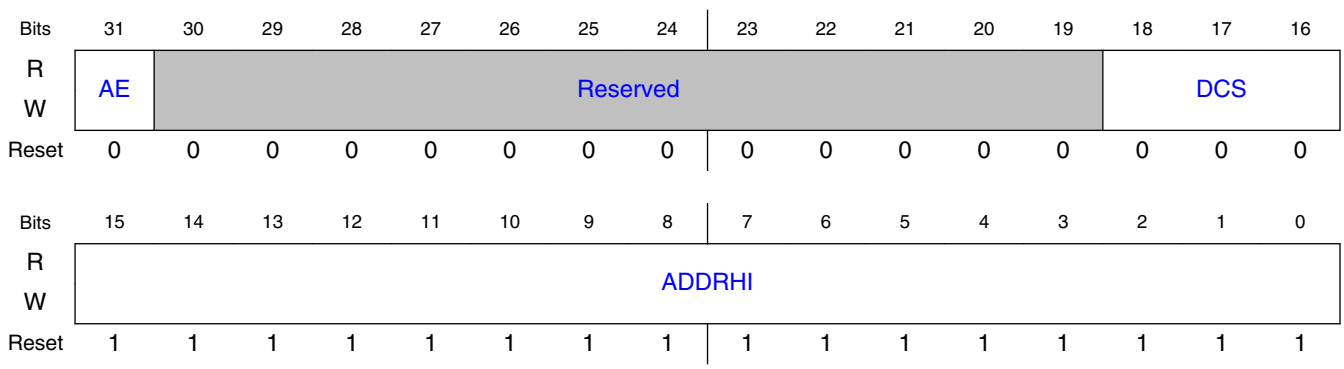
61.7.1.122.2 Function

The MAC ADDRESS37 High register holds the upper 16 bits of the 38th 6-byte MAC address of the station.

The MAC ADDRESS37 High register holds the upper 16 bits of the 38th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.122.3 Diagram



61.7.1.122.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 38th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 38th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS37 content is routed.
15-0 ADDRHI	MAC ADDRESS37 [47:32] This field contains the upper 16 bits (47:32) of the 38th 6-byte MAC address.

61.7.1.123 MAC Address37 Low (MAC_ADDRESS37_LOW)

61.7.1.123.1 Offset

Register	Offset
MAC_ADDRESS37_LOW	42Ch

61.7.1.123.2 Function

The MAC ADDRESS37 Low register holds the lower 16 bits of the 38th 6-byte MAC address of the station.

61.7.1.123.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.123.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS37 [31:0] This field contains the lower 32 bits of the 38th 6-byte MAC address. MAC ADDRESS37 [31:0] This field contains the lower 32 bits of the 38th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.124 MAC Address38 High (MAC_ADDRESS38_HIGH)

61.7.1.124.1 Offset

Register	Offset
MAC_ADDRESS38_HIGH	430h

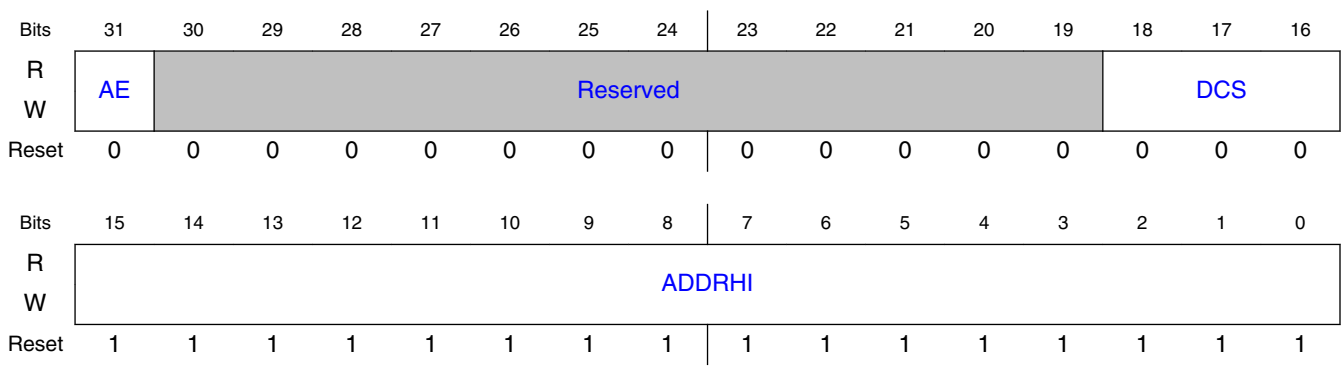
61.7.1.124.2 Function

The MAC ADDRESS38 High register holds the upper 16 bits of the 39th 6-byte MAC address of the station.

The MAC ADDRESS38 High register holds the upper 16 bits of the 39th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.124.3 Diagram



61.7.1.124.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 39th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 39th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS38 content is routed.
15-0 ADDRHI	MAC ADDRESS38 [47:32] This field contains the upper 16 bits (47:32) of the 39th 6-byte MAC address.

61.7.1.125 MAC Address38 Low (MAC_ADDRESS38_LOW)

61.7.1.125.1 Offset

Register	Offset
MAC_ADDRESS38_LOW	434h

61.7.1.125.2 Function

The MAC ADDRESS38 Low register holds the lower 16 bits of the 39th 6-byte MAC address of the station.

61.7.1.125.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.125.4 Fields

Field	Function
31-0	MAC ADDRESS38 [31:0] This field contains the lower 32 bits of the 39th 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS38 [31:0] This field contains the lower 32 bits of the 39th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.126 MAC Address39 High (MAC_ADDRESS39_HIGH)

61.7.1.126.1 Offset

Register	Offset
MAC_ADDRESS39_HIGH	438h

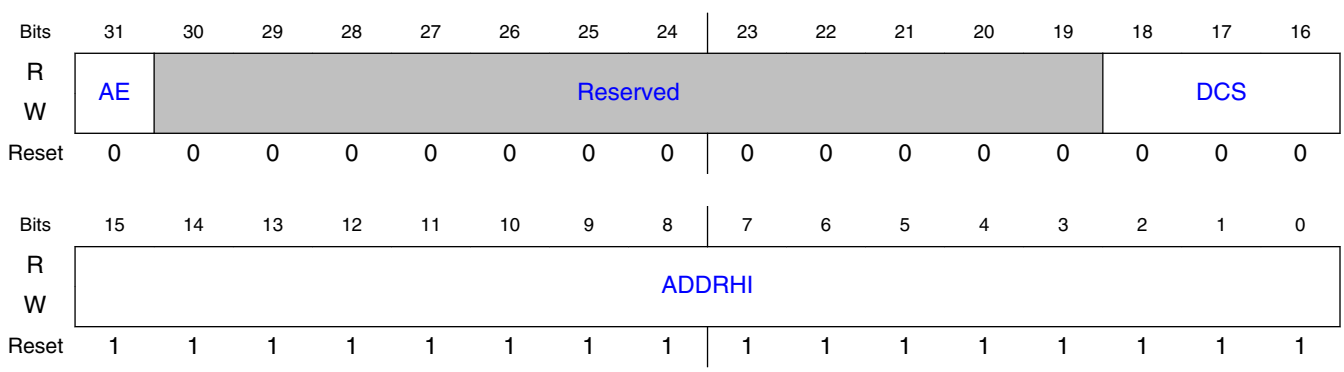
61.7.1.126.2 Function

The MAC ADDRESS39 High register holds the upper 16 bits of the 40th 6-byte MAC address of the station.

The MAC ADDRESS39 High register holds the upper 16 bits of the 40th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.126.3 Diagram



61.7.1.126.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 40th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 40th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS39 content is routed.
15-0 ADDRHI	MAC ADDRESS39 [47:32] This field contains the upper 16 bits (47:32) of the 40th 6-byte MAC address.

61.7.1.127 MAC Address39 Low (MAC_ADDRESS39_LOW)

61.7.1.127.1 Offset

Register	Offset
MAC_ADDRESS39_LOW	43Ch

61.7.1.127.2 Function

The MAC ADDRESS39 Low register holds the lower 16 bits of the 40th 6-byte MAC address of the station.

61.7.1.127.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.127.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS39 [31:0] This field contains the lower 32 bits of the 40th 6-byte MAC address. MAC ADDRESS39 [31:0] This field contains the lower 32 bits of the 40th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.128 MAC Address40 High (MAC_ADDRESS40_HIGH)

61.7.1.128.1 Offset

Register	Offset
MAC_ADDRESS40_HIGH	440h

61.7.1.128.2 Function

The MAC ADDRESS40 High register holds the upper 16 bits of the 41st 6-byte MAC address of the station.

The MAC ADDRESS40 High register holds the upper 16 bits of the 41st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.128.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.128.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 41st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 41st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS40 content is routed.
15-0 ADDRHI	MAC ADDRESS40 [47:32] This field contains the upper 16 bits (47:32) of the 41st 6-byte MAC address.

61.7.1.129 MAC Address40 Low (MAC_ADDRESS40_LOW)

61.7.1.129.1 Offset

Register	Offset
MAC_ADDRESS40_LOW	444h

61.7.1.129.2 Function

The MAC ADDRESS40 Low register holds the lower 16 bits of the 41st 6-byte MAC address of the station.

61.7.1.129.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.129.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS40 [31:0] This field contains the lower 32 bits of the 41st 6-byte MAC address. MAC ADDRESS40 [31:0] This field contains the lower 32 bits of the 41st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.130 MAC Address41 High (MAC_ADDRESS41_HIGH)

61.7.1.130.1 Offset

Register	Offset
MAC_ADDRESS41_HIGH	448h

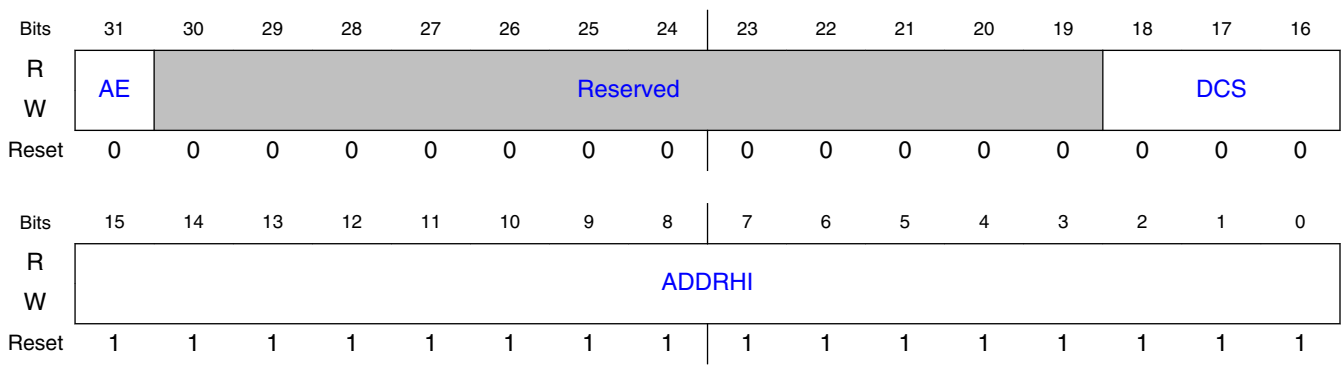
61.7.1.130.2 Function

The MAC ADDRESS41 High register holds the upper 16 bits of the 42nd 6-byte MAC address of the station.

The MAC ADDRESS41 High register holds the upper 16 bits of the 42nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.130.3 Diagram



61.7.1.130.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 42nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 42nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS41 content is routed.
15-0 ADDRHI	MAC ADDRESS41 [47:32] This field contains the upper 16 bits (47:32) of the 42nd 6-byte MAC address.

61.7.1.131 MAC Address41 Low (MAC_ADDRESS41_LOW)

61.7.1.131.1 Offset

Register	Offset
MAC_ADDRESS41_LOW	44Ch

61.7.1.131.2 Function

The MAC ADDRESS41 Low register holds the lower 16 bits of the 42nd 6-byte MAC address of the station.

61.7.1.131.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.131.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS41 [31:0] This field contains the lower 32 bits of the 42nd 6-byte MAC address. MAC ADDRESS41 [31:0] This field contains the lower 32 bits of the 42nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.132 MAC Address42 High (MAC_ADDRESS42_HIGH)

61.7.1.132.1 Offset

Register	Offset
MAC_ADDRESS42_HIGH	450h

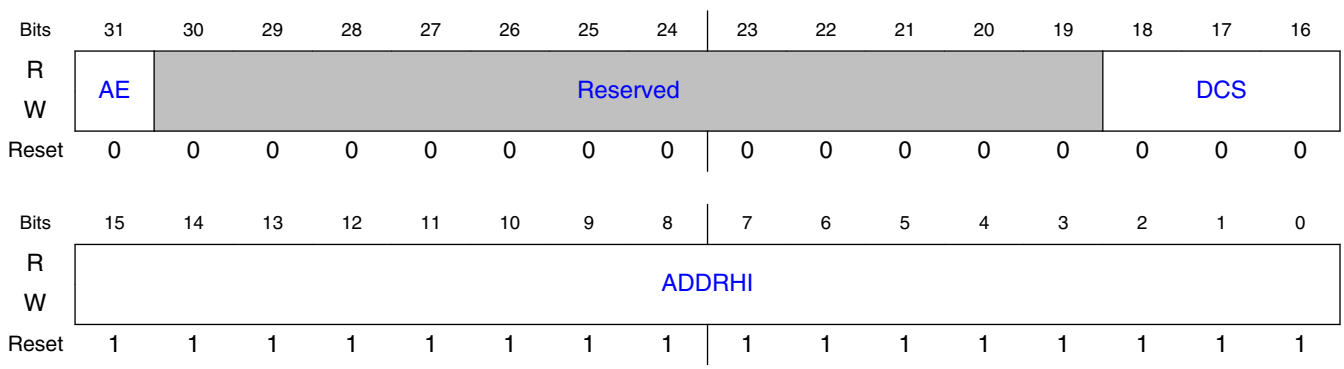
61.7.1.132.2 Function

The MAC ADDRESS42 High register holds the upper 16 bits of the 43rd 6-byte MAC address of the station.

The MAC ADDRESS42 High register holds the upper 16 bits of the 43rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.132.3 Diagram



61.7.1.132.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 43rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 43rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS42 content is routed.
15-0 ADDRHI	MAC ADDRESS42 [47:32] This field contains the upper 16 bits (47:32) of the 43rd 6-byte MAC address.

61.7.1.133 MAC Address42 Low (MAC_ADDRESS42_LOW)

61.7.1.133.1 Offset

Register	Offset
MAC_ADDRESS42_LOW	454h

61.7.1.133.2 Function

The MAC ADDRESS42 Low register holds the lower 16 bits of the 43rd 6-byte MAC address of the station.

61.7.1.133.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.133.4 Fields

Field	Function
31-0	MAC ADDRESS42 [31:0] This field contains the lower 32 bits of the 43rd 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS42 [31:0] This field contains the lower 32 bits of the 43rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.134 MAC Address43 High (MAC_ADDRESS43_HIGH)

61.7.1.134.1 Offset

Register	Offset
MAC_ADDRESS43_HIGH	458h

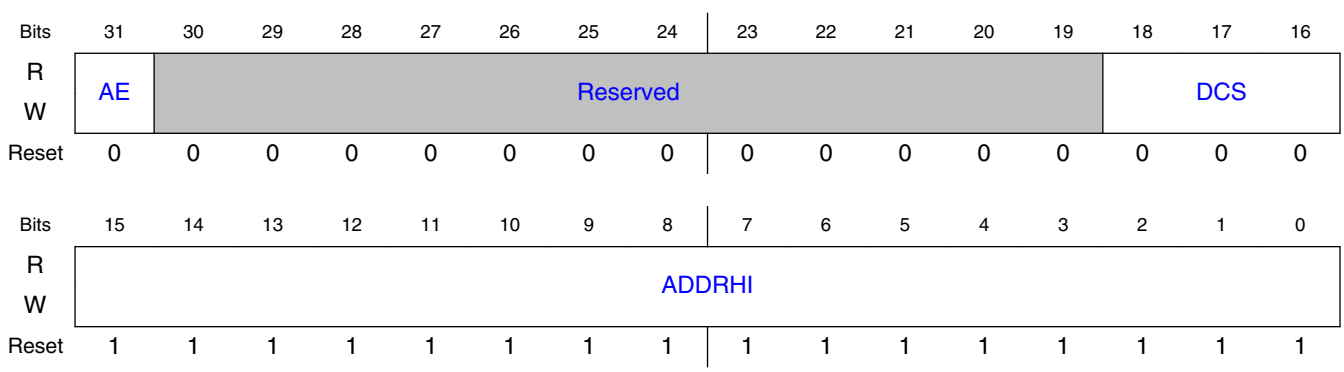
61.7.1.134.2 Function

The MAC ADDRESS43 High register holds the upper 16 bits of the 44th 6-byte MAC address of the station.

The MAC ADDRESS43 High register holds the upper 16 bits of the 44th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.134.3 Diagram



61.7.1.134.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 44th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 44th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS43 content is routed.
15-0 ADDRHI	MAC ADDRESS43 [47:32] This field contains the upper 16 bits (47:32) of the 44th 6-byte MAC address.

61.7.1.135 MAC Address43 Low (MAC_ADDRESS43_LOW)

61.7.1.135.1 Offset

Register	Offset
MAC_ADDRESS43_LOW	45Ch

61.7.1.135.2 Function

The MAC ADDRESS43 Low register holds the lower 16 bits of the 44th 6-byte MAC address of the station.

61.7.1.135.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.135.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS43 [31:0] This field contains the lower 32 bits of the 44th 6-byte MAC address. MAC ADDRESS43 [31:0] This field contains the lower 32 bits of the 44th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.136 MAC Address44 High (MAC_ADDRESS44_HIGH)

61.7.1.136.1 Offset

Register	Offset
MAC_ADDRESS44_HIGH	460h

61.7.1.136.2 Function

The MAC ADDRESS44 High register holds the upper 16 bits of the 45th 6-byte MAC address of the station.

The MAC ADDRESS44 High register holds the upper 16 bits of the 45th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.136.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.136.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 45th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 45th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS44 content is routed.
15-0 ADDRHI	MAC ADDRESS44 [47:32] This field contains the upper 16 bits (47:32) of the 45th 6-byte MAC address.

61.7.1.137 MAC Address44 Low (MAC_ADDRESS44_LOW)

61.7.1.137.1 Offset

Register	Offset
MAC_ADDRESS44_LOW	464h

61.7.1.137.2 Function

The MAC ADDRESS44 Low register holds the lower 16 bits of the 45th 6-byte MAC address of the station.

61.7.1.137.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.137.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS44 [31:0] This field contains the lower 32 bits of the 45th 6-byte MAC address. MAC ADDRESS44 [31:0] This field contains the lower 32 bits of the 45th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.138 MAC Address45 High (MAC_ADDRESS45_HIGH)

61.7.1.138.1 Offset

Register	Offset
MAC_ADDRESS45_HIGH	468h

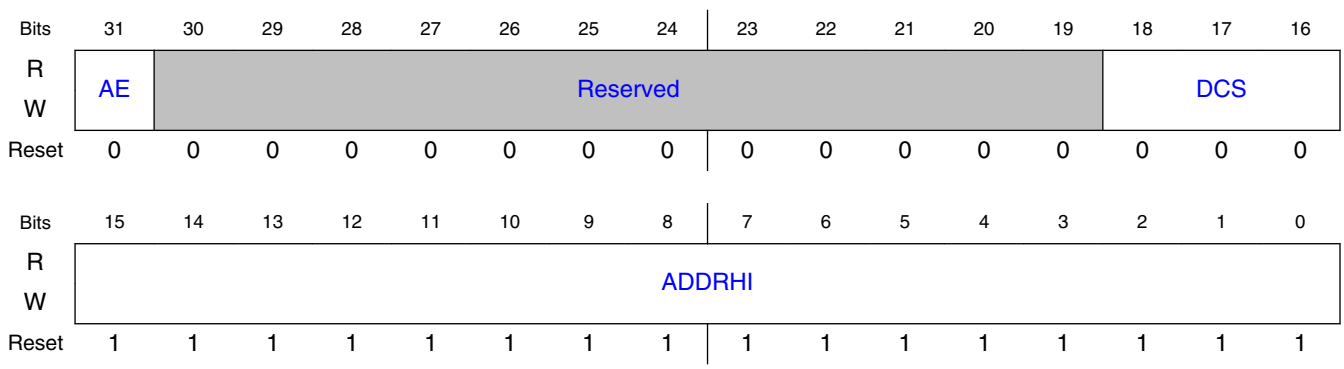
61.7.1.138.2 Function

The MAC ADDRESS45 High register holds the upper 16 bits of the 46th 6-byte MAC address of the station.

The MAC ADDRESS45 High register holds the upper 16 bits of the 46th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.138.3 Diagram



61.7.1.138.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 46th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 46th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS45 content is routed.
15-0 ADDRHI	MAC ADDRESS45 [47:32] This field contains the upper 16 bits (47:32) of the 46th 6-byte MAC address.

61.7.1.139 MAC Address45 Low (MAC_ADDRESS45_LOW)

61.7.1.139.1 Offset

Register	Offset
MAC_ADDRESS45_LOW	46Ch

61.7.1.139.2 Function

The MAC ADDRESS45 Low register holds the lower 16 bits of the 46th 6-byte MAC address of the station.

61.7.1.139.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.139.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS45 [31:0] This field contains the lower 32 bits of the 46th 6-byte MAC address. MAC ADDRESS45 [31:0] This field contains the lower 32 bits of the 46th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.140 MAC Address46 High (MAC_ADDRESS46_HIGH)

61.7.1.140.1 Offset

Register	Offset
MAC_ADDRESS46_HIGH	470h

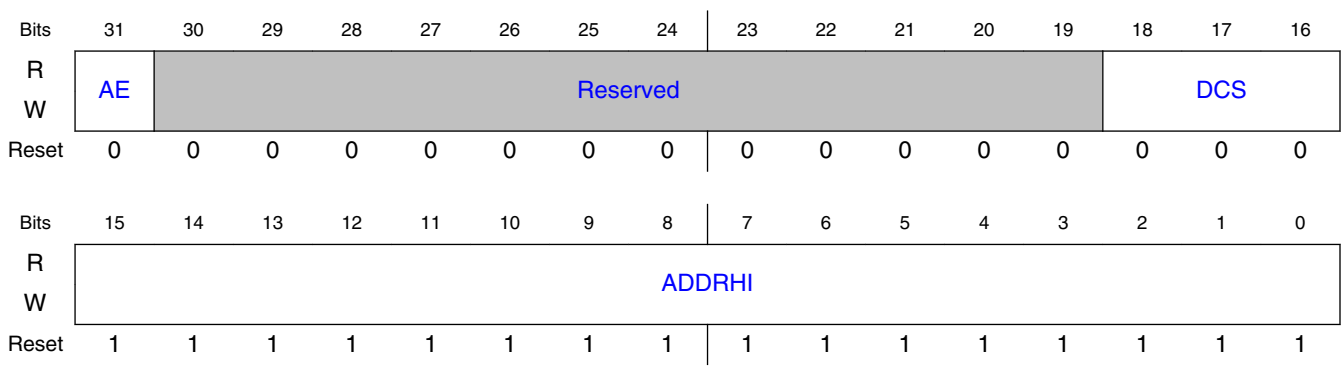
61.7.1.140.2 Function

The MAC ADDRESS46 High register holds the upper 16 bits of the 47th 6-byte MAC address of the station.

The MAC ADDRESS46 High register holds the upper 16 bits of the 47th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.140.3 Diagram



61.7.1.140.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 47th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 47th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS46 content is routed.
15-0 ADDRHI	MAC ADDRESS46 [47:32] This field contains the upper 16 bits (47:32) of the 47th 6-byte MAC address.

61.7.1.141 MAC Address46 Low (MAC_ADDRESS46_LOW)

61.7.1.141.1 Offset

Register	Offset
MAC_ADDRESS46_LOW	474h

61.7.1.141.2 Function

The MAC ADDRESS46 Low register holds the lower 16 bits of the 47th 6-byte MAC address of the station.

61.7.1.141.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.141.4 Fields

Field	Function
31-0	MAC ADDRESS46 [31:0] This field contains the lower 32 bits of the 47th 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS46 [31:0] This field contains the lower 32 bits of the 47th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.142 MAC Address47 High (MAC_ADDRESS47_HIGH)

61.7.1.142.1 Offset

Register	Offset
MAC_ADDRESS47_HIGH	478h

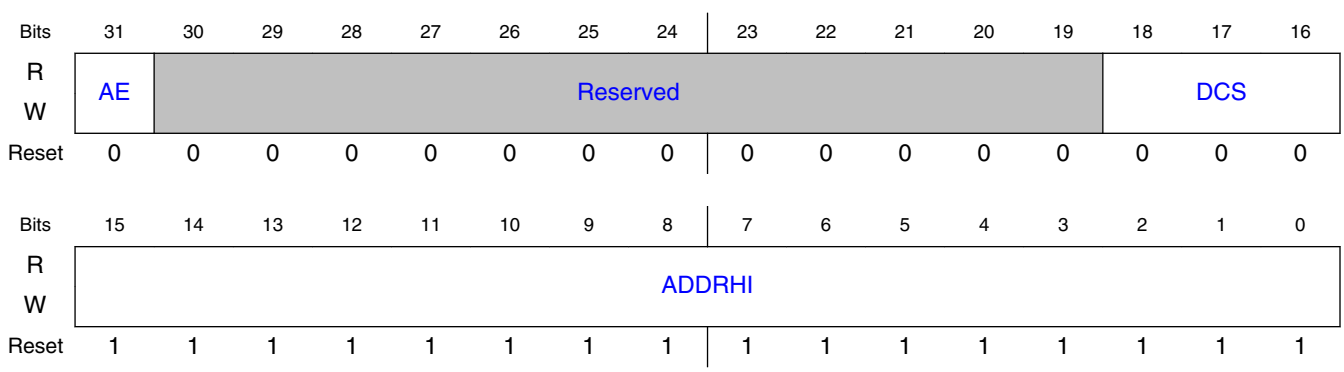
61.7.1.142.2 Function

The MAC ADDRESS47 High register holds the upper 16 bits of the 48th 6-byte MAC address of the station.

The MAC ADDRESS47 High register holds the upper 16 bits of the 48th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.142.3 Diagram



61.7.1.142.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 48th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 48th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS47 content is routed.
15-0 ADDRHI	MAC ADDRESS47 [47:32] This field contains the upper 16 bits (47:32) of the 48th 6-byte MAC address.

61.7.1.143 MAC Address47 Low (MAC_ADDRESS47_LOW)

61.7.1.143.1 Offset

Register	Offset
MAC_ADDRESS47_LOW	47Ch

61.7.1.143.2 Function

The MAC ADDRESS47 Low register holds the lower 16 bits of the 48th 6-byte MAC address of the station.

61.7.1.143.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.143.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS47 [31:0] This field contains the lower 32 bits of the 48th 6-byte MAC address. MAC ADDRESS47 [31:0] This field contains the lower 32 bits of the 48th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.144 MAC Address48 High (MAC_ADDRESS48_HIGH)

61.7.1.144.1 Offset

Register	Offset
MAC_ADDRESS48_HIGH	480h

61.7.1.144.2 Function

The MAC ADDRESS48 High register holds the upper 16 bits of the 49th 6-byte MAC address of the station.

The MAC ADDRESS48 High register holds the upper 16 bits of the 49th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.144.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.144.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 49th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 49th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS48 content is routed.
15-0 ADDRHI	MAC ADDRESS48 [47:32] This field contains the upper 16 bits (47:32) of the 49th 6-byte MAC address.

61.7.1.145 MAC Address48 Low (MAC_ADDRESS48_LOW)

61.7.1.145.1 Offset

Register	Offset
MAC_ADDRESS48_LOW	484h

61.7.1.145.2 Function

The MAC ADDRESS48 Low register holds the lower 16 bits of the 49th 6-byte MAC address of the station.

61.7.1.145.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.145.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS48 [31:0] This field contains the lower 32 bits of the 49th 6-byte MAC address. MAC ADDRESS48 [31:0] This field contains the lower 32 bits of the 49th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.146 MAC Address49 High (MAC_ADDRESS49_HIGH)

61.7.1.146.1 Offset

Register	Offset
MAC_ADDRESS49_HIGH	488h

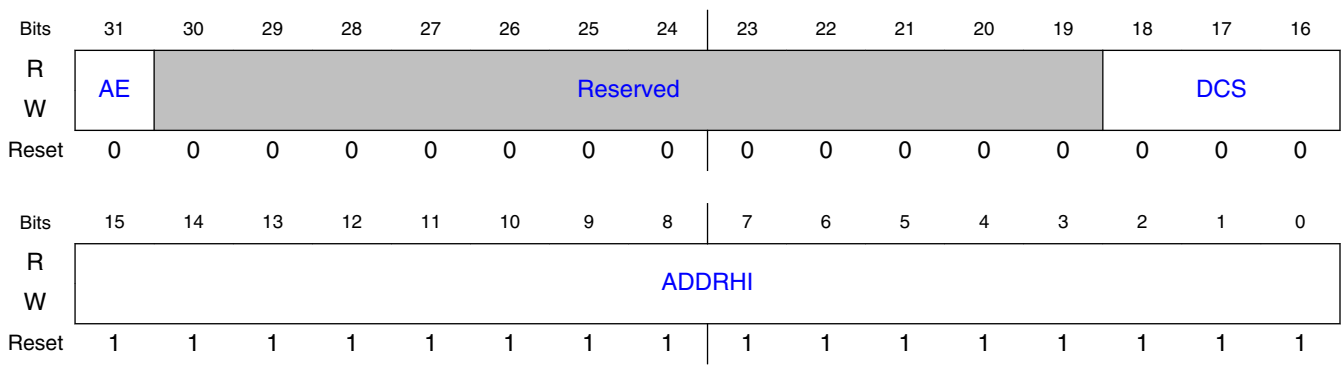
61.7.1.146.2 Function

The MAC ADDRESS49 High register holds the upper 16 bits of the 50th 6-byte MAC address of the station.

The MAC ADDRESS49 High register holds the upper 16 bits of the 50th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.146.3 Diagram



61.7.1.146.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 50th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 50th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS49 content is routed.
15-0 ADDRHI	MAC ADDRESS49 [47:32] This field contains the upper 16 bits (47:32) of the 50th 6-byte MAC address.

61.7.1.147 MAC Address49 Low (MAC_ADDRESS49_LOW)

61.7.1.147.1 Offset

Register	Offset
MAC_ADDRESS49_LOW	48Ch

61.7.1.147.2 Function

The MAC ADDRESS49 Low register holds the lower 16 bits of the 50th 6-byte MAC address of the station.

61.7.1.147.3 Diagram

61.7.1.147.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS49 [31:0] This field contains the lower 32 bits of the 50th 6-byte MAC address. MAC ADDRESS49 [31:0] This field contains the lower 32 bits of the 50th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.148 MAC Address50 High (MAC_ADDRESS50_HIGH)

61.7.1.148.1 Offset

Register	Offset
MAC_ADDRESS50_HIGH	490h

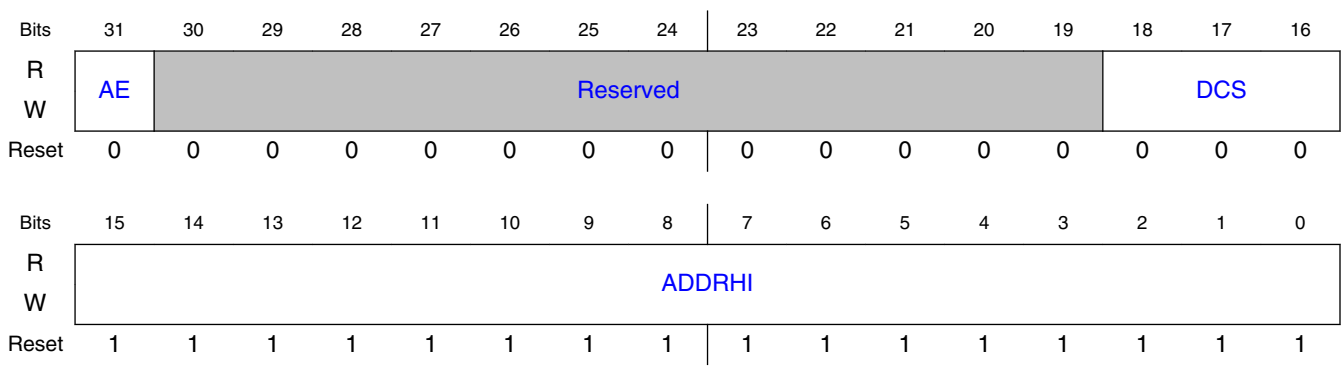
61.7.1.148.2 Function

The MAC ADDRESS50 High register holds the upper 16 bits of the 51st 6-byte MAC address of the station.

The MAC ADDRESS50 High register holds the upper 16 bits of the 51st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.148.3 Diagram



61.7.1.148.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 51st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 51st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS50 content is routed.
15-0 ADDRHI	MAC ADDRESS50 [47:32] This field contains the upper 16 bits (47:32) of the 51st 6-byte MAC address.

61.7.1.149 MAC Address50 Low (MAC_ADDRESS50_LOW)

61.7.1.149.1 Offset

Register	Offset
MAC_ADDRESS50_LOW	494h

61.7.1.149.2 Function

The MAC ADDRESS50 Low register holds the lower 16 bits of the 51st 6-byte MAC address of the station.

61.7.1.149.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.149.4 Fields

Field	Function
31-0	MAC ADDRESS50 [31:0] This field contains the lower 32 bits of the 51st 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS50 [31:0] This field contains the lower 32 bits of the 51st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.150 MAC Address51 High (MAC_ADDRESS51_HIGH)

61.7.1.150.1 Offset

Register	Offset
MAC_ADDRESS51_HIGH	498h

61.7.1.150.2 Function

The MAC ADDRESS51 High register holds the upper 16 bits of the 52nd 6-byte MAC address of the station.

The MAC ADDRESS51 High register holds the upper 16 bits of the 52nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.150.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.150.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 52nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 52nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS51 content is routed.
15-0 ADDRHI	MAC ADDRESS51 [47:32] This field contains the upper 16 bits (47:32) of the 52nd 6-byte MAC address.

61.7.1.151 MAC Address51 Low (MAC_ADDRESS51_LOW)

61.7.1.151.1 Offset

Register	Offset
MAC_ADDRESS51_LOW	49Ch

61.7.1.151.2 Function

The MAC ADDRESS51 Low register holds the lower 16 bits of the 52nd 6-byte MAC address of the station.

61.7.1.151.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.151.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS51 [31:0] This field contains the lower 32 bits of the 52nd 6-byte MAC address. MAC ADDRESS51 [31:0] This field contains the lower 32 bits of the 52nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.152 MAC Address52 High (MAC_ADDRESS52_HIGH)

61.7.1.152.1 Offset

Register	Offset
MAC_ADDRESS52_HIGH	4A0h

61.7.1.152.2 Function

The MAC ADDRESS52 High register holds the upper 16 bits of the 53rd 6-byte MAC address of the station.

The MAC ADDRESS52 High register holds the upper 16 bits of the 53rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.152.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.152.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 53rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 53rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS52 content is routed.
15-0 ADDRHI	MAC ADDRESS52 [47:32] This field contains the upper 16 bits (47:32) of the 53rd 6-byte MAC address.

61.7.1.153 MAC Address52 Low (MAC_ADDRESS52_LOW)

61.7.1.153.1 Offset

Register	Offset
MAC_ADDRESS52_LOW	4A4h

61.7.1.153.2 Function

The MAC ADDRESS52 Low register holds the lower 16 bits of the 53rd 6-byte MAC address of the station.

61.7.1.153.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.153.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS52 [31:0] This field contains the lower 32 bits of the 53rd 6-byte MAC address. MAC ADDRESS52 [31:0] This field contains the lower 32 bits of the 53rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.154 MAC Address53 High (MAC_ADDRESS53_HIGH)

61.7.1.154.1 Offset

Register	Offset
MAC_ADDRESS53_HIGH	4A8h

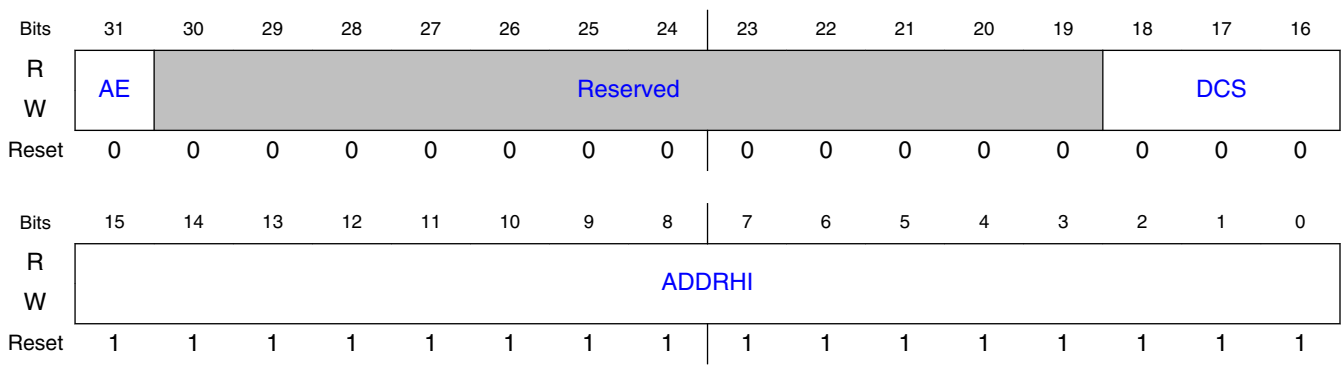
61.7.1.154.2 Function

The MAC ADDRESS53 High register holds the upper 16 bits of the 54th 6-byte MAC address of the station.

The MAC ADDRESS53 High register holds the upper 16 bits of the 54th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.154.3 Diagram



61.7.1.154.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 54th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 54th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS53 content is routed.
15-0 ADDRHI	MAC ADDRESS53 [47:32] This field contains the upper 16 bits (47:32) of the 54th 6-byte MAC address.

61.7.1.155 MAC Address53 Low (MAC_ADDRESS53_LOW)

61.7.1.155.1 Offset

Register	Offset
MAC_ADDRESS53_LOW	4ACh

61.7.1.155.2 Function

The MAC ADDRESS53 Low register holds the lower 16 bits of the 54th 6-byte MAC address of the station.

61.7.1.155.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.155.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS53 [31:0] This field contains the lower 32 bits of the 54th 6-byte MAC address. MAC ADDRESS53 [31:0] This field contains the lower 32 bits of the 54th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.156 MAC Address54 High (MAC_ADDRESS54_HIGH)

61.7.1.156.1 Offset

Register	Offset
MAC_ADDRESS54_HIGH	4B0h

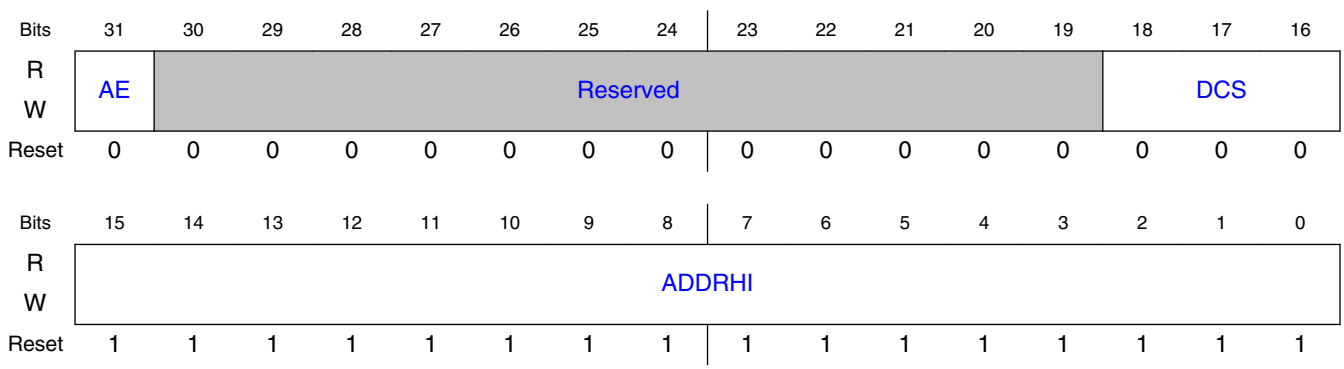
61.7.1.156.2 Function

The MAC ADDRESS54 High register holds the upper 16 bits of the 55th 6-byte MAC address of the station.

The MAC ADDRESS54 High register holds the upper 16 bits of the 55th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.156.3 Diagram



61.7.1.156.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 55th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 55th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS54 content is routed.
15-0 ADDRHI	MAC ADDRESS54 [47:32] This field contains the upper 16 bits (47:32) of the 55th 6-byte MAC address.

61.7.1.157 MAC Address54 Low (MAC_ADDRESS54_LOW)

61.7.1.157.1 Offset

Register	Offset
MAC_ADDRESS54_LOW	4B4h

61.7.1.157.2 Function

The MAC ADDRESS54 Low register holds the lower 16 bits of the 55th 6-byte MAC address of the station.

61.7.1.157.3 Diagram

61.7.1.157.4 Fields

Field	Function
31-0	MAC ADDRESS54 [31:0] This field contains the lower 32 bits of the 55th 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS54 [31:0] This field contains the lower 32 bits of the 55th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.158 MAC Address55 High (MAC_ADDRESS55_HIGH)

61.7.1.158.1 Offset

Register	Offset
MAC_ADDRESS55_HIGH	4B8h

61.7.1.158.2 Function

The MAC ADDRESS55 High register holds the upper 16 bits of the 56th 6-byte MAC address of the station.

The MAC ADDRESS55 High register holds the upper 16 bits of the 56th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.158.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.158.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 56th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 56th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS55 content is routed.
15-0 ADDRHI	MAC ADDRESS55 [47:32] This field contains the upper 16 bits (47:32) of the 56th 6-byte MAC address.

61.7.1.159 MAC Address55 Low (MAC_ADDRESS55_LOW)

61.7.1.159.1 Offset

Register	Offset
MAC_ADDRESS55_LOW	4BCh

61.7.1.159.2 Function

The MAC ADDRESS55 Low register holds the lower 16 bits of the 56th 6-byte MAC address of the station.

61.7.1.159.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.159.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS55 [31:0] This field contains the lower 32 bits of the 56th 6-byte MAC address. MAC ADDRESS55 [31:0] This field contains the lower 32 bits of the 56th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.160 MAC Address56 High (MAC_ADDRESS56_HIGH)

61.7.1.160.1 Offset

Register	Offset
MAC_ADDRESS56_HIGH	4C0h

61.7.1.160.2 Function

The MAC ADDRESS56 High register holds the upper 16 bits of the 57th 6-byte MAC address of the station.

The MAC ADDRESS56 High register holds the upper 16 bits of the 57th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.160.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.160.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 57th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 57th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS56 content is routed.
15-0 ADDRHI	MAC ADDRESS56 [47:32] This field contains the upper 16 bits (47:32) of the 57th 6-byte MAC address.

61.7.1.161 MAC Address56 Low (MAC_ADDRESS56_LOW)

61.7.1.161.1 Offset

Register	Offset
MAC_ADDRESS56_LOW	4C4h

61.7.1.161.2 Function

The MAC ADDRESS56 Low register holds the lower 16 bits of the 57th 6-byte MAC address of the station.

61.7.1.161.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.161.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS56 [31:0] This field contains the lower 32 bits of the 57th 6-byte MAC address. MAC ADDRESS56 [31:0] This field contains the lower 32 bits of the 57th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.162 MAC Address57 High (MAC_ADDRESS57_HIGH)

61.7.1.162.1 Offset

Register	Offset
MAC_ADDRESS57_HIGH	4C8h

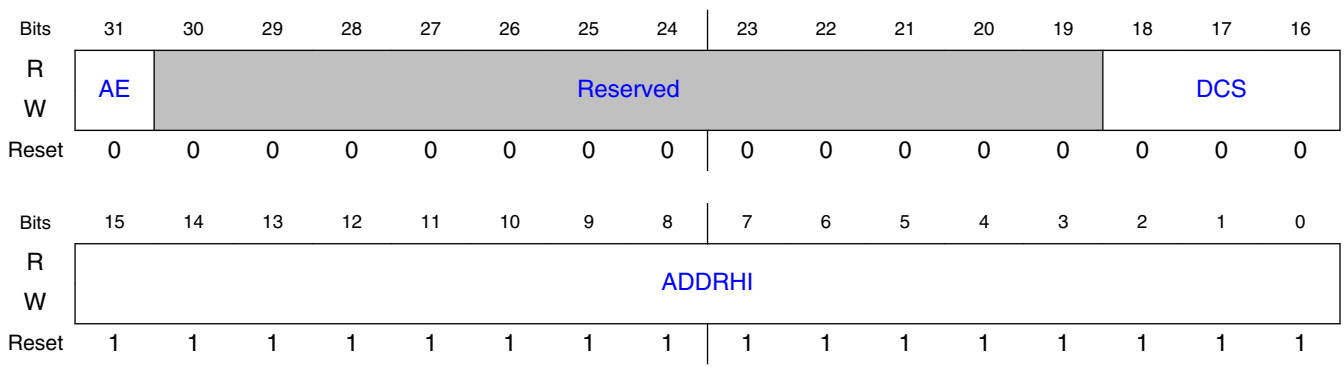
61.7.1.162.2 Function

The MAC ADDRESS57 High register holds the upper 16 bits of the 58th 6-byte MAC address of the station.

The MAC ADDRESS57 High register holds the upper 16 bits of the 58th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.162.3 Diagram



61.7.1.162.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 58th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 58th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS57 content is routed.
15-0 ADDRHI	MAC ADDRESS57 [47:32] This field contains the upper 16 bits (47:32) of the 58th 6-byte MAC address.

61.7.1.163 MAC Address57 Low (MAC_ADDRESS57_LOW)

61.7.1.163.1 Offset

Register	Offset
MAC_ADDRESS57_LOW	4CCh

61.7.1.163.2 Function

The MAC ADDRESS57 Low register holds the lower 16 bits of the 58th 6-byte MAC address of the station.

61.7.1.163.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.163.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS57 [31:0] This field contains the lower 32 bits of the 58th 6-byte MAC address. MAC ADDRESS57 [31:0] This field contains the lower 32 bits of the 58th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.164 MAC Address58 High (MAC_ADDRESS58_HIGH)

61.7.1.164.1 Offset

Register	Offset
MAC_ADDRESS58_HIGH	4D0h

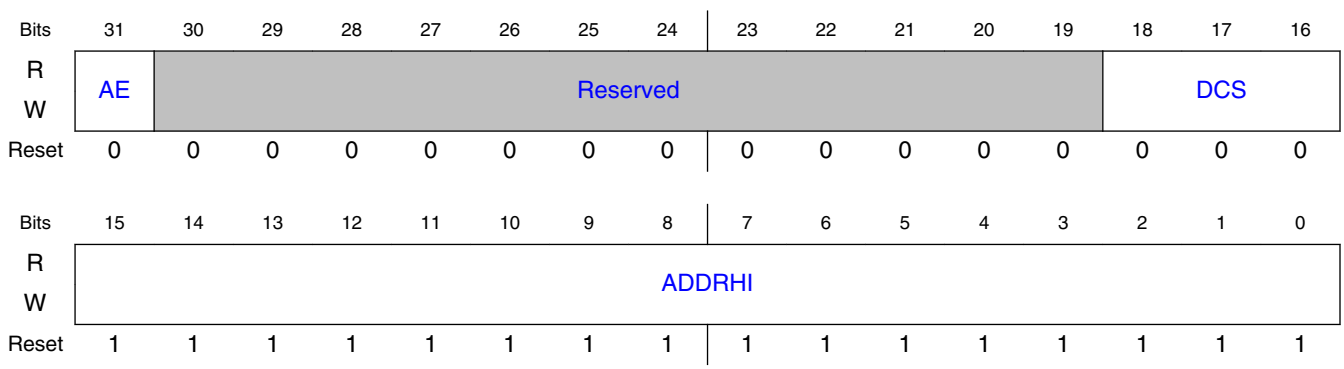
61.7.1.164.2 Function

The MAC ADDRESS58 High register holds the upper 16 bits of the 59th 6-byte MAC address of the station.

The MAC ADDRESS58 High register holds the upper 16 bits of the 59th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.164.3 Diagram



61.7.1.164.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 59th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 59th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS58 content is routed.
15-0 ADDRHI	MAC ADDRESS58 [47:32] This field contains the upper 16 bits (47:32) of the 59th 6-byte MAC address.

61.7.1.165 MAC Address58 Low (MAC_ADDRESS58_LOW)

61.7.1.165.1 Offset

Register	Offset
MAC_ADDRESS58_LOW	4D4h

61.7.1.165.2 Function

The MAC ADDRESS58 Low register holds the lower 16 bits of the 59th 6-byte MAC address of the station.

61.7.1.165.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.165.4 Fields

Field	Function
31-0	MAC ADDRESS58 [31:0] This field contains the lower 32 bits of the 59th 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS58 [31:0] This field contains the lower 32 bits of the 59th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.166 MAC Address59 High (MAC_ADDRESS59_HIGH)

61.7.1.166.1 Offset

Register	Offset
MAC_ADDRESS59_HIGH	4D8h

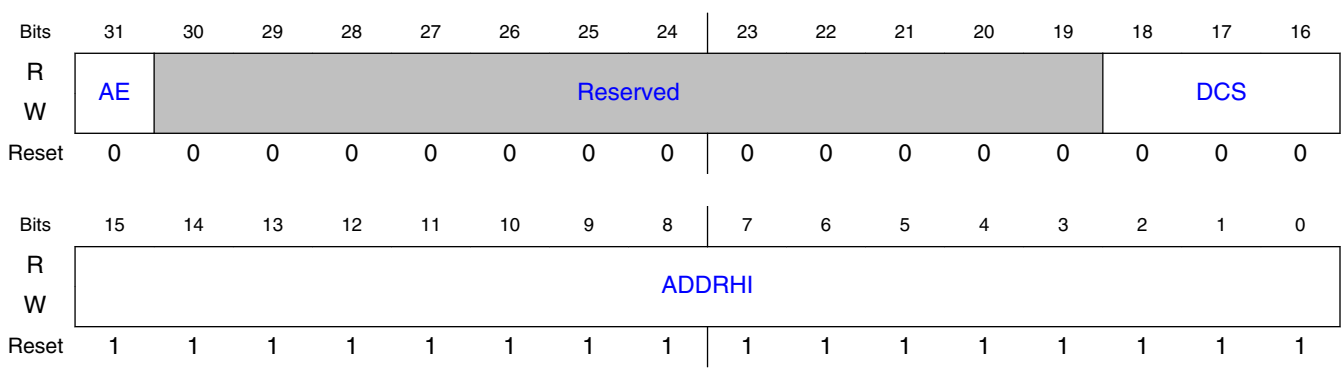
61.7.1.166.2 Function

The MAC ADDRESS59 High register holds the upper 16 bits of the 60th 6-byte MAC address of the station.

The MAC ADDRESS59 High register holds the upper 16 bits of the 60th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.166.3 Diagram



61.7.1.166.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 60th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 60th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS59 content is routed.
15-0 ADDRHI	MAC ADDRESS59 [47:32] This field contains the upper 16 bits (47:32) of the 60th 6-byte MAC address.

61.7.1.167 MAC Address59 Low (MAC_ADDRESS59_LOW)

61.7.1.167.1 Offset

Register	Offset
MAC_ADDRESS59_LOW	4DCh

61.7.1.167.2 Function

The MAC ADDRESS59 Low register holds the lower 16 bits of the 60th 6-byte MAC address of the station.

61.7.1.167.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.167.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS59 [31:0] This field contains the lower 32 bits of the 60th 6-byte MAC address. MAC ADDRESS59 [31:0] This field contains the lower 32 bits of the 60th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.168 MAC Address60 High (MAC_ADDRESS60_HIGH)

61.7.1.168.1 Offset

Register	Offset
MAC_ADDRESS60_HIGH	4E0h

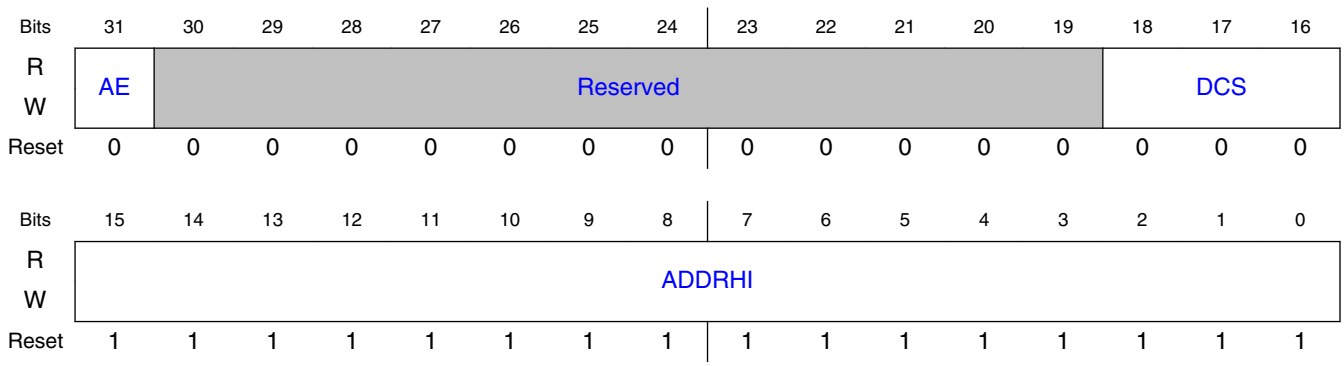
61.7.1.168.2 Function

The MAC ADDRESS60 High register holds the upper 16 bits of the 61st 6-byte MAC address of the station.

The MAC ADDRESS60 High register holds the upper 16 bits of the 61st 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.168.3 Diagram



61.7.1.168.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 61st MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 61st MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS60 content is routed.
15-0 ADDRHI	MAC ADDRESS60 [47:32] This field contains the upper 16 bits (47:32) of the 61st 6-byte MAC address.

61.7.1.169 MAC Address60 Low (MAC_ADDRESS60_LOW)

61.7.1.169.1 Offset

Register	Offset
MAC_ADDRESS60_LOW	4E4h

61.7.1.169.2 Function

The MAC ADDRESS60 Low register holds the lower 16 bits of the 61st 6-byte MAC address of the station.

61.7.1.169.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.169.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS60 [31:0] This field contains the lower 32 bits of the 61st 6-byte MAC address. MAC ADDRESS60 [31:0] This field contains the lower 32 bits of the 61st 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.170 MAC Address61 High (MAC_ADDRESS61_HIGH)

61.7.1.170.1 Offset

Register	Offset
MAC_ADDRESS61_HIGH	4E8h

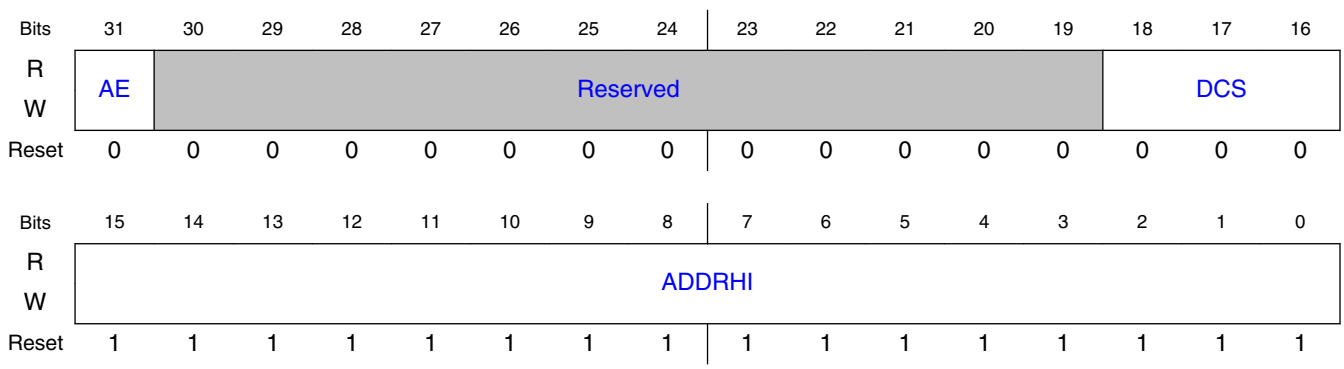
61.7.1.170.2 Function

The MAC ADDRESS61 High register holds the upper 16 bits of the 62nd 6-byte MAC address of the station.

The MAC ADDRESS61 High register holds the upper 16 bits of the 62nd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.170.3 Diagram



61.7.1.170.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 62nd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 62nd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS61 content is routed.
15-0 ADDRHI	MAC ADDRESS61 [47:32] This field contains the upper 16 bits (47:32) of the 62nd 6-byte MAC address.

61.7.1.171 MAC Address61 Low (MAC_ADDRESS61_LOW)

61.7.1.171.1 Offset

Register	Offset
MAC_ADDRESS61_LOW	4ECh

61.7.1.171.2 Function

The MAC ADDRESS61 Low register holds the lower 16 bits of the 62nd 6-byte MAC address of the station.

61.7.1.171.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.171.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS61 [31:0] This field contains the lower 32 bits of the 62nd 6-byte MAC address. MAC ADDRESS61 [31:0] This field contains the lower 32 bits of the 62nd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.172 MAC Address62 High (MAC_ADDRESS62_HIGH)

61.7.1.172.1 Offset

Register	Offset
MAC_ADDRESS62_HIGH	4F0h

61.7.1.172.2 Function

The MAC ADDRESS62 High register holds the upper 16 bits of the 63rd 6-byte MAC address of the station.

The MAC ADDRESS62 High register holds the upper 16 bits of the 63rd 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.172.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.172.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 63rd MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 63rd MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled

Table continues on the next page...

Field	Function
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS62 content is routed.
15-0 ADDRHI	MAC ADDRESS62 [47:32] This field contains the upper 16 bits (47:32) of the 63rd 6-byte MAC address.

61.7.1.173 MAC Address62 Low (MAC_ADDRESS62_LOW)

61.7.1.173.1 Offset

Register	Offset
MAC_ADDRESS62_LOW	4F4h

61.7.1.173.2 Function

The MAC ADDRESS62 Low register holds the lower 16 bits of the 63rd 6-byte MAC address of the station.

61.7.1.173.3 Diagram

61.7.1.173.4 Fields

Field	Function
31-0	MAC ADDRESS62 [31:0] This field contains the lower 32 bits of the 63rd 6-byte MAC address.

Field	Function
ADDRLO	MAC ADDRESS62 [31:0] This field contains the lower 32 bits of the 63rd 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.174 MAC Address63 High (MAC_ADDRESS63_HIGH)

61.7.1.174.1 Offset

Register	Offset
MAC_ADDRESS63_HIGH	4F8h

61.7.1.174.2 Function

The MAC ADDRESS63 High register holds the upper 16 bits of the 64th 6-byte MAC address of the station.

The MAC ADDRESS63 High register holds the upper 16 bits of the 64th 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address Low Register are written. For proper synchronization updates, you should perform the consecutive writes to the MAC Address Low Register after at least four clock cycles of the destination clock.

61.7.1.174.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved												DCS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRHI															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.174.4 Fields

Field	Function
31 AE	Address Enable When this bit is set, the Address filter module uses the 64th MAC address for perfect filtering. Address Enable When this bit is set, the Address filter module uses the 64th MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering. 0b - Address is ignored 1b - Address is enabled
30-19 —	Reserved
18-16 DCS	DMA Channel Select This field contains the DMA Channel number to which an Rx packet whose DA matches the MAC ADDRESS63 content is routed.
15-0 ADDRHI	MAC ADDRESS63 [47:32] This field contains the upper 16 bits (47:32) of the 64th 6-byte MAC address.

61.7.1.175 MAC Address63 Low (MAC_ADDRESS63_LOW)

61.7.1.175.1 Offset

Register	Offset
MAC_ADDRESS63_LOW	4FCh

61.7.1.175.2 Function

The MAC ADDRESS63 Low register holds the lower 16 bits of the 64th 6-byte MAC address of the station.

61.7.1.175.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRLO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

61.7.1.175.4 Fields

Field	Function
31-0 ADDRLO	MAC ADDRESS63 [31:0] This field contains the lower 32 bits of the 64th 6-byte MAC address. MAC ADDRESS63 [31:0] This field contains the lower 32 bits of the 64th 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.

61.7.1.176 MMC Control (MAC_MMC_CONTROL)

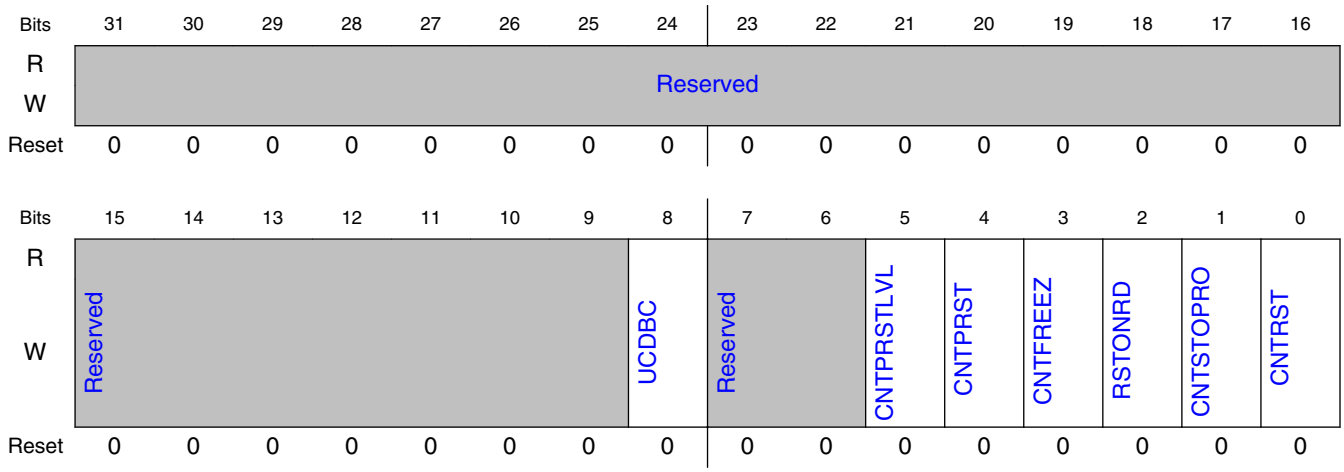
61.7.1.176.1 Offset

Register	Offset
MAC_MMC_CONTROL	700h

61.7.1.176.2 Function

This register establishes the operating mode of MMC.

61.7.1.176.3 Diagram



61.7.1.176.4 Fields

Field	Function
31-9 —	Reserved
8 UCDBC	Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. 0b - Update MMC Counters for Dropped Broadcast Packets is disabled 1b - Update MMC Counters for Dropped Broadcast Packets is enabled
7-6 —	Reserved
5 CNTPRSTLVL	Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. 0b - Full-Half Preset is disabled

Table continues on the next page...

Field	Function
	1b - Full-Half Preset is enabled
4 CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0b - Counters Preset is disabled 1b - Counters Preset is enabled
3 CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. 0b - MMC Counter Freeze is disabled 1b - MMC Counter Freeze is enabled
2 RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read. 0b - Reset on Read is disabled 1b - Reset on Read is enabled
1 CNTSTOPRO	Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. 0b - Counter Stop Rollover is disabled 1b - Counter Stop Rollover is enabled
0 CNTRST	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0b - Counters are not reset 1b - All counters are reset

61.7.1.177 MMC Rx Interrupt (MAC_MMC_RX_INTERRUPT)

61.7.1.177.1 Offset

Register	Offset
MAC_MMC_RX_INTERRUPT	704h

61.7.1.177.2 Function

This register maintains the interrupts generated from all Receive statistics counters.

This register maintains the interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt register maintains the interrupts that are generated when the following occur:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter).

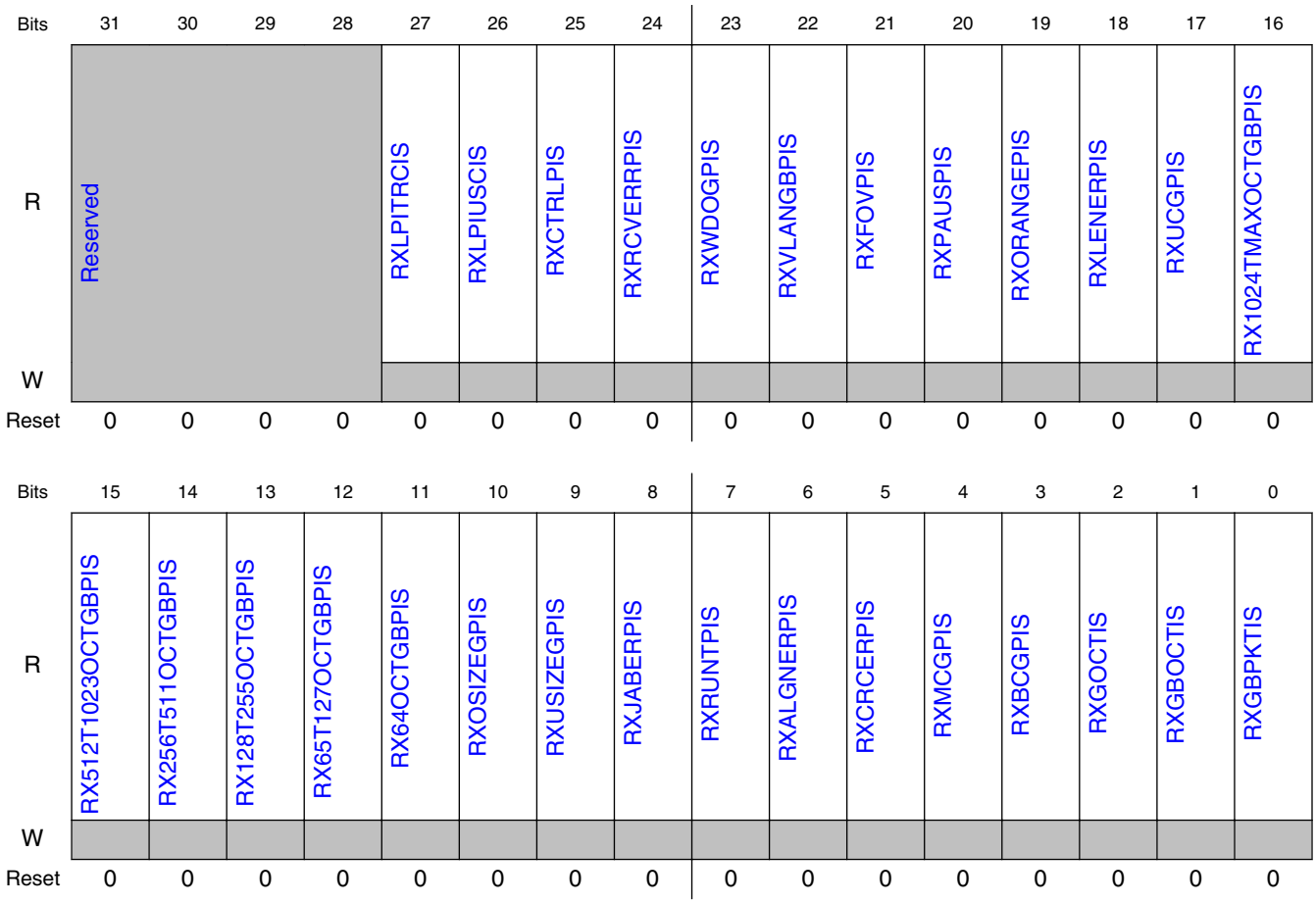
- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter).

When the Counter Stop Rollover is set, interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register

is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

Note: R_SS_RC means that this register bit is set internally, and it is cleared when the Counter register is read.

61.7.1.177.3 Diagram



61.7.1.177.4 Fields

Field	Function
31-28 —	Reserved
27 RXLPITRCIS	MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI transition counter interrupt status This bit is set when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive LPI transition Counter Interrupt Status not detected 1b - MMC Receive LPI transition Counter Interrupt Status detected
26 RXLPIUSCIS	MMC Receive LPI microsecond counter interrupt status This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Receive LPI microsecond counter interrupt status

Table continues on the next page...

Field	Function
	This bit is set when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive LPI microsecond Counter Interrupt Status not detected 1b - MMC Receive LPI microsecond Counter Interrupt Status detected
25 RXCTRLPIS	MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Control Packet Counter Interrupt Status This bit is set when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Control Packet Counter Interrupt Status not detected 1b - MMC Receive Control Packet Counter Interrupt Status detected
24 RXRCVERRPIS	MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. MMC Receive Error Packet Counter Interrupt Status This bit is set when the rxrcverror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Error Packet Counter Interrupt Status not detected 1b - MMC Receive Error Packet Counter Interrupt Status detected
23 RXWDOGPIS	MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. MMC Receive Watchdog Error Packet Counter Interrupt Status This bit is set when the rxwatchdog error counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Watchdog Error Packet Counter Interrupt Status not detected 1b - MMC Receive Watchdog Error Packet Counter Interrupt Status detected
22 RXVLANGBPIS	MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. MMC Receive VLAN Good Bad Packet Counter Interrupt Status This bit is set when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive VLAN Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive VLAN Good Bad Packet Counter Interrupt Status detected
21 RXFOVPIS	MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected 1b - MMC Receive FIFO Overflow Packet Counter Interrupt Status detected

Table continues on the next page...

Field	Function
20 RXPAUSPIS	MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Pause Packet Counter Interrupt Status not detected 1b - MMC Receive Pause Packet Counter Interrupt Status detected
19 RXORANGEPIS	MMC Receive Out Of Range Error Packet Counter Interrupt Status. MMC Receive Out Of Range Error Packet Counter Interrupt Status. This bit is set when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Out Of Range Error Packet Counter Interrupt Status not detected 1b - MMC Receive Out Of Range Error Packet Counter Interrupt Status detected
18 RXLENERPIS	MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Length Error Packet Counter Interrupt Status not detected 1b - MMC Receive Length Error Packet Counter Interrupt Status detected
17 RXUCGPIS	MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Unicast Good Packet Counter Interrupt Status This bit is set when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Unicast Good Packet Counter Interrupt Status not detected 1b - MMC Receive Unicast Good Packet Counter Interrupt Status detected
16 RX1024TMAXOCTGBPIS	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
15 RX512T1023OCTGBPIS	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event.

Table continues on the next page...

Field	Function
	0b - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
14 RX256T511OCTGBPIS	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
13 RX128T255OCTGBPIS	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
12 RX65T127OCTGBPIS	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected
11 RX64OCTGBPIS	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the rx64octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Status detected
10 RXOSIZEGPIS	MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. MMC Receive Oversize Good Packet Counter Interrupt Status This bit is set when the rxoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Oversize Good Packet Counter Interrupt Status not detected 1b - MMC Receive Oversize Good Packet Counter Interrupt Status detected
9 RXUSIZEGPIS	MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value. MMC Receive Undersize Good Packet Counter Interrupt Status This bit is set when the rxundersize_g counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Function
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Undersize Good Packet Counter Interrupt Status not detected 1b - MMC Receive Undersize Good Packet Counter Interrupt Status detected
8 RXJABERPIS	MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. MMC Receive Jabber Error Packet Counter Interrupt Status This bit is set when the rxjabbererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Jabber Error Packet Counter Interrupt Status not detected 1b - MMC Receive Jabber Error Packet Counter Interrupt Status detected
7 RXRUNTPIS	MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. MMC Receive Runt Packet Counter Interrupt Status This bit is set when the rxrunterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Runt Packet Counter Interrupt Status not detected 1b - MMC Receive Runt Packet Counter Interrupt Status detected
6 RXALGNERPIS	MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. MMC Receive Alignment Error Packet Counter Interrupt Status This bit is set when the rxalignmenterror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Alignment Error Packet Counter Interrupt Status not detected 1b - MMC Receive Alignment Error Packet Counter Interrupt Status detected
5 RXCRCERPIS	MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive CRC Error Packet Counter Interrupt Status not detected 1b - MMC Receive CRC Error Packet Counter Interrupt Status detected
4 RXMCGPIS	MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1b - MMC Receive Multicast Good Packet Counter Interrupt Status detected
3 RXBCGPIS	MMC Receive Broadcast Good Packet Counter Interrupt Status This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Receive Broadcast Good Packet Counter Interrupt Status

Table continues on the next page...

Field	Function
	This bit is set when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Broadcast Good Packet Counter Interrupt Status not detected 1b - MMC Receive Broadcast Good Packet Counter Interrupt Status detected
2 RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Good Octet Counter Interrupt Status not detected 1b - MMC Receive Good Octet Counter Interrupt Status detected
1 RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Good Bad Octet Counter Interrupt Status not detected 1b - MMC Receive Good Bad Octet Counter Interrupt Status detected
0 RXGBPKTIS	MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive Good Bad Packet Counter Interrupt Status not detected 1b - MMC Receive Good Bad Packet Counter Interrupt Status detected

61.7.1.178 MMC Tx Interrupt (MAC_MMC_TX_INTERRUPT)

61.7.1.178.1 Offset

Register	Offset
MAC_MMC_TX_INTERRUPT	708h

61.7.1.178.2 Function

This register maintains the interrupts generated from all Transmit statistics counters.

This register maintains the interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values

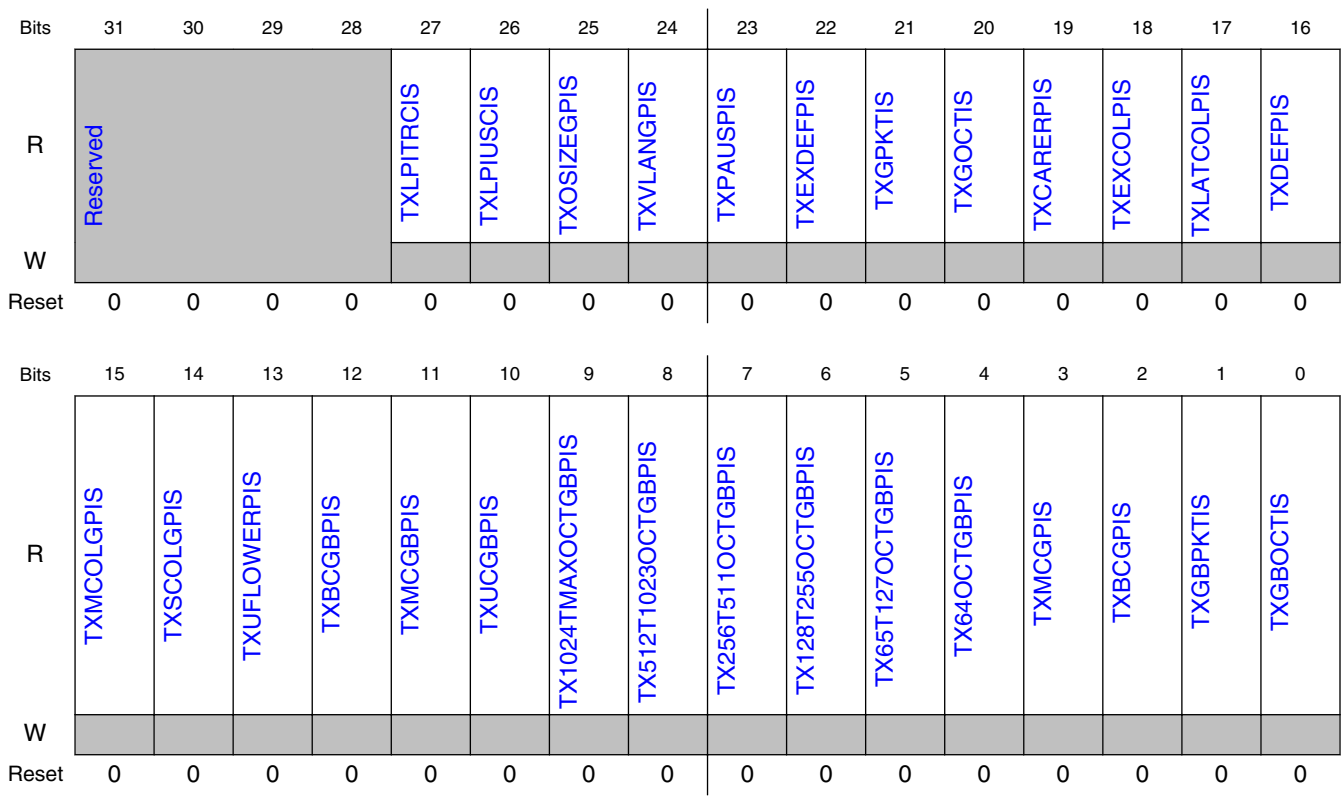
(0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read.

The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

61.7.1.178.3 Diagram



61.7.1.178.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27 TXLPITRCIS	MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI transition counter interrupt status This bit is set when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit LPI transition Counter Interrupt Status not detected 1b - MMC Transmit LPI transition Counter Interrupt Status detected
26 TXLPIUSCIS	MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit LPI microsecond counter interrupt status This bit is set when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit LPI microsecond Counter Interrupt Status not detected 1b - MMC Transmit LPI microsecond Counter Interrupt Status detected
25 TXOSIZEGPIS	MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. MMC Transmit Oversize Good Packet Counter Interrupt Status This bit is set when the txoversize_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Oversize Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Oversize Good Packet Counter Interrupt Status detected
24 TXVLANGPIS	MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit VLAN Good Packet Counter Interrupt Status This bit is set when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit VLAN Good Packet Counter Interrupt Status not detected 1b - MMC Transmit VLAN Good Packet Counter Interrupt Status detected
23 TXPAUSPIS	MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Pause Packet Counter Interrupt Status not detected 1b - MMC Transmit Pause Packet Counter Interrupt Status detected
22 TXEXDEFPIS	MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Deferral Packet Counter Interrupt Status This bit is set when the txexcessdef counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Function
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Excessive Deferral Packet Counter Interrupt Status not detected 1b - MMC Transmit Excessive Deferral Packet Counter Interrupt Status detected
21 TXGPKTIS	MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Good Packet Counter Interrupt Status detected
20 TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Good Octet Counter Interrupt Status not detected 1b - MMC Transmit Good Octet Counter Interrupt Status detected
19 TXCARERPIS	MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1b - MMC Transmit Carrier Error Packet Counter Interrupt Status detected
18 TXEXCOLPIS	MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. MMC Transmit Excessive Collision Packet Counter Interrupt Status This bit is set when the txexesscol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Excessive Collision Packet Counter Interrupt Status not detected 1b - MMC Transmit Excessive Collision Packet Counter Interrupt Status detected
17 TXLATCOLPIS	MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. MMC Transmit Late Collision Packet Counter Interrupt Status This bit is set when the txlatecol counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Late Collision Packet Counter Interrupt Status not detected 1b - MMC Transmit Late Collision Packet Counter Interrupt Status detected
16 TXDEFPIS	MMC Transmit Deferred Packet Counter Interrupt Status This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. MMC Transmit Deferred Packet Counter Interrupt Status

Table continues on the next page...

Field	Function
	This bit is set when the txdeferred counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Deferred Packet Counter Interrupt Status not detected 1b - MMC Transmit Deferred Packet Counter Interrupt Status detected
15 TXMCOLGPIS	MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. MMC Transmit Multiple Collision Good Packet Counter Interrupt Status This bit is set when the txmulticol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Multiple Collision Good Packet Counter Interrupt Status detected
14 TXSCOLGPIS	MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. MMC Transmit Single Collision Good Packet Counter Interrupt Status This bit is set when the txsinglecol_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Single Collision Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Single Collision Good Packet Counter Interrupt Status detected
13 TXUFLOWERPIS	MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1b - MMC Transmit Underflow Error Packet Counter Interrupt Status detected
12 TXBCGBPIS	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Status detected
11 TXMCGBPIS	MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Bad Packet Counter Interrupt Status The bit is set when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit Multicast Good Bad Packet Counter Interrupt Status detected
10	MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Function
TXUCGBPIS	MMC Transmit Unicast Good Bad Packet Counter Interrupt Status This bit is set when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit Unicast Good Bad Packet Counter Interrupt Status detected
9 TX1024TMAXOCTGBPIS	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Status detected
8 TX512T1023OCTGBPIS	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Status detected
7 TX256T511OCTGBPIS	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Status detected
6 TX128T255OCTGBPIS	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Status detected
5 TX65T127OCTGBPIS	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value. MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx65to127octets_gb counter reaches half the maximum value, and also when it reaches the maximum value.

Table continues on the next page...

Field	Function
	Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Status detected
4 TX64OCTGBPIS	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status This bit is set when the tx64octets_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Status detected
3 TXMCGPIS	MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Multicast Good Packet Counter Interrupt Status This bit is set when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Multicast Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Multicast Good Packet Counter Interrupt Status detected
2 TXBCGPIS	MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. MMC Transmit Broadcast Good Packet Counter Interrupt Status This bit is set when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Broadcast Good Packet Counter Interrupt Status not detected 1b - MMC Transmit Broadcast Good Packet Counter Interrupt Status detected
1 TXGBPKTIS	MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1b - MMC Transmit Good Bad Packet Counter Interrupt Status detected
0 TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1b - MMC Transmit Good Bad Octet Counter Interrupt Status detected

61.7.1.179 MMC Rx Interrupt Mask (MAC_MMC_RX_INTERRUPT_MASK)

61.7.1.179.1 Offset

Register	Offset
MAC_MMC_RX_INTERRUPT_MASK	70Ch

61.7.1.179.2 Function

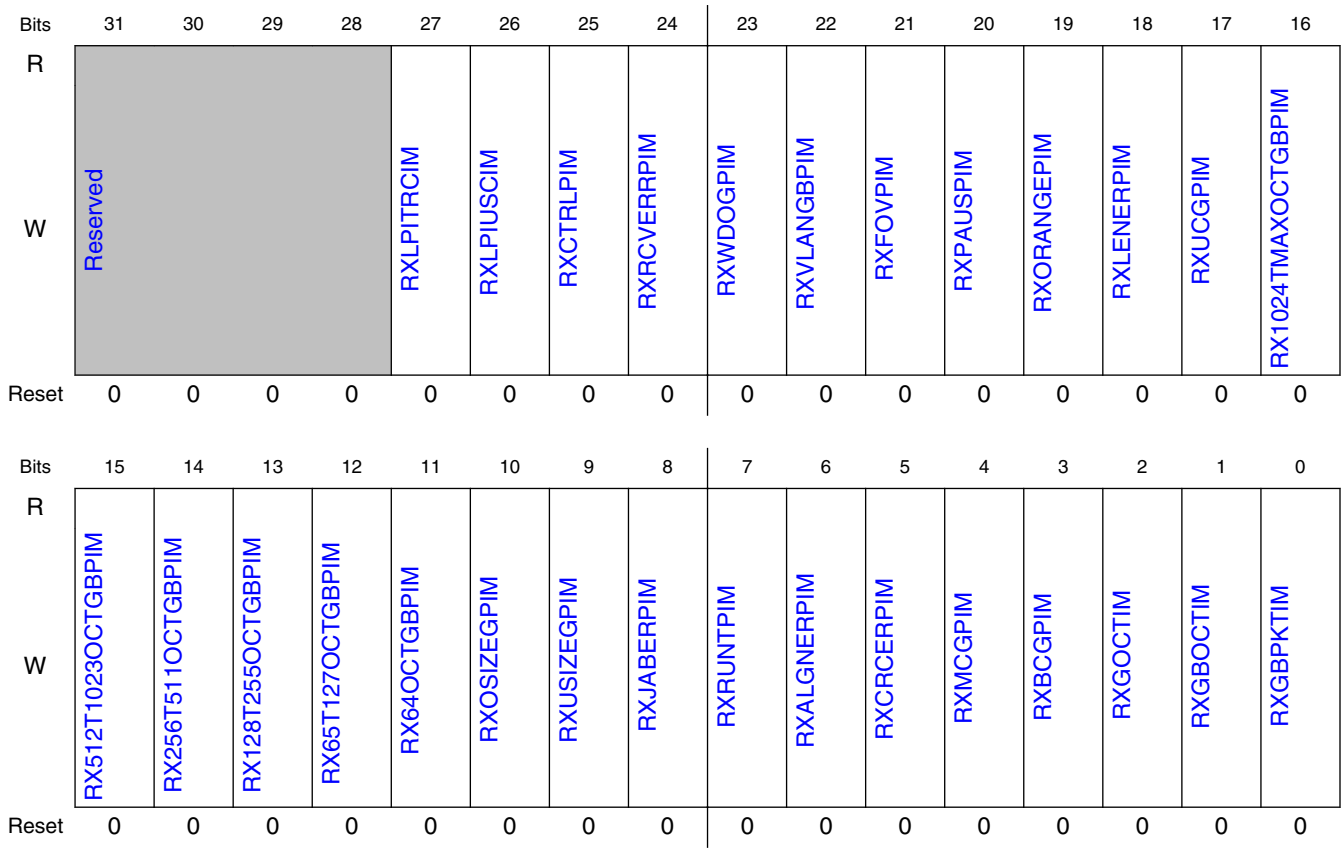
This register maintains the masks for interrupts generated from all Receive statistics counters.

This register maintains the masks for interrupts generated from all Receive statistics counters.

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values.

This register is 32 bit wide.

61.7.1.179.3 Diagram



61.7.1.179.4 Fields

Field	Function
31-28 —	Reserved
27 RXLPITRCIM	MMC Receive LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0b - MMC Receive LPI transition counter interrupt Mask is disabled 1b - MMC Receive LPI transition counter interrupt Mask is enabled
26 RXLPIUSCIM	MMC Receive LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Rx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0b - MMC Receive LPI microsecond counter interrupt Mask is disabled 1b - MMC Receive LPI microsecond counter interrupt Mask is enabled
25 RXCTRLPIM	MMC Receive Control Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxctrlpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Control Packet Counter Interrupt Mask is disabled 1b - MMC Receive Control Packet Counter Interrupt Mask is enabled
24 RXRCVERRPIM	MMC Receive Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrcverror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Error Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
23 RXWDOGPIM	MMC Receive Watchdog Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxwatchdog counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Watchdog Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Watchdog Error Packet Counter Interrupt Mask is enabled
22 RXVLANGBPIM	MMC Receive VLAN Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxvlanpackets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive VLAN Good Bad Packet Counter Interrupt Mask is enabled
21 RXFOVPIM	MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. 0b - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1b - MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled
20 RXPAUSPIM	MMC Receive Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpausepackets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Pause Packet Counter Interrupt Mask is disabled 1b - MMC Receive Pause Packet Counter Interrupt Mask is enabled
19 RXORANGEPIM	MMC Receive Out Of Range Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoutofrangetype counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Out Of Range Error Packet Counter Interrupt Mask is enabled
18 RXLENERPIM	MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Length Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Length Error Packet Counter Interrupt Mask is enabled
17 RXUCGPIM	MMC Receive Unicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxunicastpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Unicast Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive Unicast Good Packet Counter Interrupt Mask is enabled
16 RX1024TMAXOCTGBPIM	MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask. Setting this bit masks the interrupt when the rx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
15 RX512T1023OCTGBPIM	MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled
14 RX256T511OCTGBPIM	MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
13 RX128T255OCTGBPIM	MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
12 RX65T127OCTGBPIM	MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
11 RX64OCTGBPIM	MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rx64octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Receive 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
10 RXOSIZEGPIM	MMC Receive Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoversize_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Oversize Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive Oversize Good Packet Counter Interrupt Mask is enabled
9 RXUSIZEGPIM	MMC Receive Undersize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxundersize_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Undersize Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive Undersize Good Packet Counter Interrupt Mask is enabled
8 RXJABERPIM	MMC Receive Jabber Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxjabbererror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Jabber Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Jabber Error Packet Counter Interrupt Mask is enabled
7 RXRUNTPIM	MMC Receive Runt Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxrunterror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Runt Packet Counter Interrupt Mask is disabled 1b - MMC Receive Runt Packet Counter Interrupt Mask is enabled
6 RXALGNERPIM	MMC Receive Alignment Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxalignmenterror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Alignment Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive Alignment Error Packet Counter Interrupt Mask is enabled
5 RXCRCERPIM	MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. 0b - MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive CRC Error Packet Counter Interrupt Mask is enabled
4 RXMCGPIM	MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled
3 RXBCGPIM	MMC Receive Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Broadcast Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive Broadcast Good Packet Counter Interrupt Mask is enabled
2 RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive Good Octet Counter Interrupt Mask is enabled
1 RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1b - MMC Receive Good Bad Octet Counter Interrupt Mask is enabled
0 RXGBPKTIM	MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Receive Good Bad Packet Counter Interrupt Mask is disabled

Field	Function
	1b - MMC Receive Good Bad Packet Counter Interrupt Mask is enabled

61.7.1.180 MMC Tx Interrupt Mask (MAC_MMC_TX_INTERRUPT_MASK)

61.7.1.180.1 Offset

Register	Offset
MAC_MMC_TX_INTERRUPT_MASK	710h

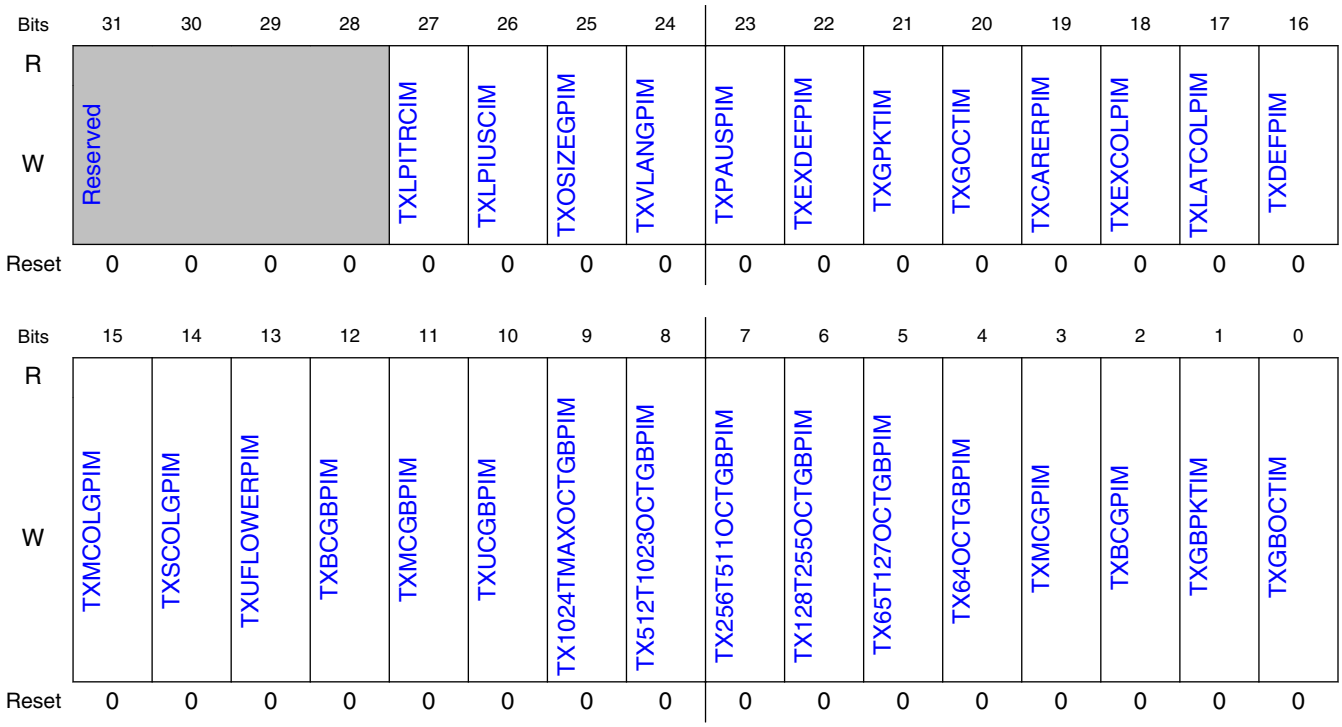
61.7.1.180.2 Function

This register maintains the masks for interrupts generated from all Transmit statistics counters.

This register maintains the masks for interrupts generated from all Transmit statistics counters.

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

61.7.1.180.3 Diagram



61.7.1.180.4 Fields

Field	Function
31-28 —	Reserved
27 TXLPITRCIM	MMC Transmit LPI transition counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_Tran_Cntr counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit LPI transition counter interrupt Mask is disabled 1b - MMC Transmit LPI transition counter interrupt Mask is enabled
26 TXLPIUSCIM	MMC Transmit LPI microsecond counter interrupt Mask Setting this bit masks the interrupt when the Tx_LPI_USEC_Cntr counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit LPI microsecond counter interrupt Mask is disabled 1b - MMC Transmit LPI microsecond counter interrupt Mask is enabled
25 TXOSIZEGPIM	MMC Transmit Oversize Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txoversize_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Oversize Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Oversize Good Packet Counter Interrupt Mask is enabled
24 TXVLANGPIM	MMC Transmit VLAN Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txvlanpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit VLAN Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit VLAN Good Packet Counter Interrupt Mask is enabled
23 TXPAUSPIM	MMC Transmit Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpausepackets counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Pause Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Pause Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
22 TXEXDEFPIM	MMC Transmit Excessive Deferral Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcessdef counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Excessive Deferral Packet Counter Interrupt Mask is enabled
21 TXGPKTIM	MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Good Packet Counter Interrupt Mask is enabled
20 TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Good Octet Counter Interrupt Mask is disabled 1b - MMC Transmit Good Octet Counter Interrupt Mask is enabled
19 TXCARERPIM	MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled
18 TXEXCOLPIM	MMC Transmit Excessive Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txexcesscol counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Excessive Collision Packet Counter Interrupt Mask is enabled
17 TXLATCOLPIM	MMC Transmit Late Collision Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txlatecol counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Late Collision Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Late Collision Packet Counter Interrupt Mask is enabled
16 TXDEFPIM	MMC Transmit Deferred Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txdeferred counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Deferred Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Deferred Packet Counter Interrupt Mask is enabled
15 TXMCOLGPIM	MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticol_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask is enabled
14 TXSCOLGPIM	MMC Transmit Single Collision Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txsinglecol_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Single Collision Good Packet Counter Interrupt Mask is enabled
13 TXUFLOWERPIM	MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled
12 TXBCGBPIM	MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Broadcast Good Bad Packet Counter Interrupt Mask is enabled
11 TXMCGBPIM	MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Multicast Good Bad Packet Counter Interrupt Mask is enabled
10 TXUCGBPIM	MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunicastpackets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Unicast Good Bad Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
9 TX1024TMAXOCTGBPIM	MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx1024tomaxoctets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 1024 to Maximum Octet Good Bad Packet Counter Interrupt Mask is enabled
8 TX512T1023OCTGBPIM	MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx512to1023octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 512 to 1023 Octet Good Bad Packet Counter Interrupt Mask is enabled
7 TX256T511OCTGBPIM	MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx256to511octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 256 to 511 Octet Good Bad Packet Counter Interrupt Mask is enabled
6 TX128T255OCTGBPIM	MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx128to255octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 128 to 255 Octet Good Bad Packet Counter Interrupt Mask is enabled
5 TX65T127OCTGBPIM	MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx65to127octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 65 to 127 Octet Good Bad Packet Counter Interrupt Mask is enabled
4 TX64OCTGBPIM	MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the tx64octets_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit 64 Octet Good Bad Packet Counter Interrupt Mask is enabled
3 TXMCGPIM	MMC Transmit Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txmulticastpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Multicast Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Multicast Good Packet Counter Interrupt Mask is enabled
2 TXBCGPIM	MMC Transmit Broadcast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txbroadcastpackets_g counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Broadcast Good Packet Counter Interrupt Mask is enabled
1 TXGBPKTIM	MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1b - MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled
0 TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. 0b - MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1b - MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled

61.7.1.181 Tx Octet Count Good and Bad (MAC_TX_OCTET_COUNT_GOOD_BAD)

61.7.1.181.1 Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD_BAD	714h

61.7.1.181.2 Function

This register provides the number of bytes transmitted by the DWC_ether_qos, exclusive of preamble and retried bytes, in good and bad packets.

61.7.1.181.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.181.4 Fields

Field	Function
31-0 TXOCTGB	Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.

61.7.1.182 Tx Packet Count Good and Bad (MAC_TX_PACKET_COUNT_GOOD_BAD)

61.7.1.182.1 Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD_BAD	718h

61.7.1.182.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos, exclusive of retried packets.

61.7.1.182.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.182.4 Fields

Field	Function
31-0 TXPKTGB	Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.

61.7.1.183 Tx Broadcast Packets Good (MAC_TX_BROADCAST_PACKETS_GOOD)

61.7.1.183.1 Offset

Register	Offset
MAC_TX_BROADCAST_PACKETS_GOOD	71Ch

61.7.1.183.2 Function

This register provides the number of good broadcast packets transmitted by DWC_ether_qos.

61.7.1.183.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.183.4 Fields

Field	Function
31-0 TXBCASTG	Tx Broadcast Packets Good This field indicates the number of good broadcast packets transmitted.

61.7.1.184 Tx Multicast Packets Good (MAC_TX_MULTICAST_PACKETS_GOOD)

61.7.1.184.1 Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD	720h

61.7.1.184.2 Function

This register provides the number of good multicast packets transmitted by DWC_ether_qos.

61.7.1.184.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.184.4 Fields

Field	Function
31-0 TXMCASTG	Tx Multicast Packets Good This field indicates the number of good multicast packets transmitted.

61.7.1.185 Tx Good and Bad 64-Byte Packets (MAC_TX_64OCTETS_PACKETS_GOOD_BAD)

61.7.1.185.1 Offset

Register	Offset
MAC_TX_64OCTETS_PACKETS_GOOD_BAD	724h

61.7.1.185.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 64 bytes, exclusive of preamble and retried packets.

61.7.1.185.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.185.4 Fields

Field	Function
31-0 TX64OCTGB	Tx 64Octets Packets Good_Bad This field indicates the number of good and bad packets transmitted with length 64 bytes, exclusive of preamble and retried packets.

61.7.1.186 Tx Good and Bad 65 to 127-Byte Packets (MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD)

61.7.1.186.1 Offset

Register	Offset
MAC_TX_65TO127OCTETS_PACKETS_GOOD_BAD	728h

61.7.1.186.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.186.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.186.4 Fields

Field	Function
31-0 TX65_127OCTGB	Tx 65To127Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.187 Tx Good and Bad 128 to 255-Byte Packets (MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD)

61.7.1.187.1 Offset

Register	Offset
MAC_TX_128TO255OCTETS_PACKETS_GOOD_BAD	72Ch

61.7.1.187.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 128 to 255 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.187.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.187.4 Fields

Field	Function
31-0 TX128_255OCTGB	Tx 128To255Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.188 Tx Good and Bad 256 to 511-Byte Packets (MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD)

61.7.1.188.1 Offset

Register	Offset
MAC_TX_256TO511OCTETS_PACKETS_GOOD_BAD	730h

61.7.1.188.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length between 256 to 511 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.188.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.188.4 Fields

Field	Function
31-0 TX256_511OCTGB	Tx 256To511Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.189 Tx Good and Bad 512 to 1023-Byte Packets (MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD)

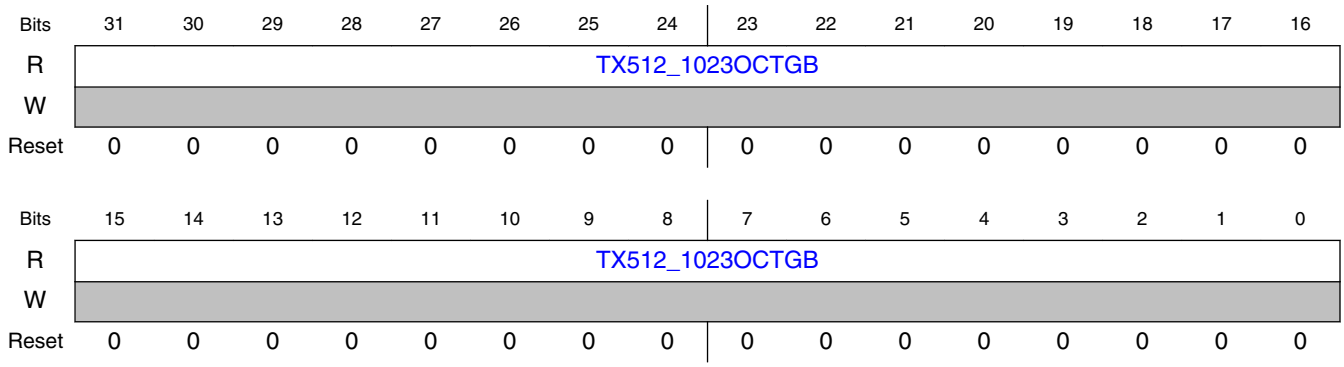
61.7.1.189.1 Offset

Register	Offset
MAC_TX_512TO1023OCTETS_PACKETS_GOOD_BAD	734h

61.7.1.189.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 512 to 1023 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.189.3 Diagram



61.7.1.189.4 Fields

Field	Function
31-0 TX512_1023OCTGB	Tx 512To1023Octets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 512 and 1023 (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.190 Tx Good and Bad 1024 to Max-Byte Packets (MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

61.7.1.190.1 Offset

Register	Offset
MAC_TX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	738h

61.7.1.190.2 Function

This register provides the number of good and bad packets transmitted by DWC_ether_qos with length 1024 to maxsize (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.190.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.190.4 Fields

Field	Function
31-0 TX1024_MAXOCTGB	Tx 1024ToMaxOctets Packets Good Bad This field indicates the number of good and bad packets transmitted with length between 1024 and maxsize (inclusive) bytes, exclusive of preamble and retried packets.

61.7.1.191 Good and Bad Unicast Packets Transmitted (MAC_TX_UNICAST_PACKETS_GOOD_BAD)

61.7.1.191.1 Offset

Register	Offset
MAC_TX_UNICAST_PACKETS_GOOD_BAD	73Ch

61.7.1.191.2 Function

This register provides the number of good and bad unicast packets transmitted by DWC_ether_qos.

61.7.1.191.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.191.4 Fields

Field	Function
31-0 TXUCASTGB	Tx Unicast Packets Good Bad This field indicates the number of good and bad unicast packets transmitted.

61.7.1.192 Good and Bad Multicast Packets Transmitted (MAC_TX_MULTICAST_PACKETS_GOOD_BAD)

61.7.1.192.1 Offset

Register	Offset
MAC_TX_MULTICAST_ PACKETS_GOOD_BAD	740h

61.7.1.192.2 Function

This register provides the number of good and bad multicast packets transmitted by DWC_ether_qos.

61.7.1.192.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXMCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXMCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.192.4 Fields

Field	Function
31-0 TXMCASTGB	Tx Multicast Packets Good Bad This field indicates the number of good and bad multicast packets transmitted.

61.7.1.193 Good and Bad Broadcast Packets Transmitted (MAC_TX_BROADCAST_PACKETS_GOOD_BAD)

61.7.1.193.1 Offset

Register	Offset
MAC_TX_BROADCAST_PACKETS_GOOD_BAD	744h

61.7.1.193.2 Function

This register provides the number of good and bad broadcast packets transmitted by DWC_ether_qos.

61.7.1.193.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXBCASTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.193.4 Fields

Field	Function
31-0 TXBCASTGB	Tx Broadcast Packets Good Bad This field indicates the number of good and bad broadcast packets transmitted.

61.7.1.194 Tx Packets Aborted By Underflow Error (MAC_TX_UNDERFLOW_ERROR_PACKETS)

61.7.1.194.1 Offset

Register	Offset
MAC_TX_UNDERFLOW_ERROR_PACKETS	748h

61.7.1.194.2 Function

This register provides the number of packets aborted by DWC_ether_qos because of packets underflow error.

61.7.1.194.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXUNDRFLW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.194.4 Fields

Field	Function
31-0 TXUNDRFLW	Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.

61.7.1.195 Single Collision Good Packets Transmitted (MAC_TX_SINGLE_COLLISION_GOOD_PACKETS)

61.7.1.195.1 Offset

Register	Offset
MAC_TX_SINGLE_COLLISION_GOOD_PACKETS	74Ch

61.7.1.195.2 Function

This register provides the number of successfully transmitted packets by DWC_ether_qos after a single collision in the half-duplex mode.

61.7.1.195.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXSNGLCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXSNGLCOLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.195.4 Fields

Field	Function
31-0 TXSNGLCOLG	Tx Single Collision Good Packets This field indicates the number of successfully transmitted packets after a single collision in the half-duplex mode.

61.7.1.196 Multiple Collision Good Packets Transmitted (MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS)

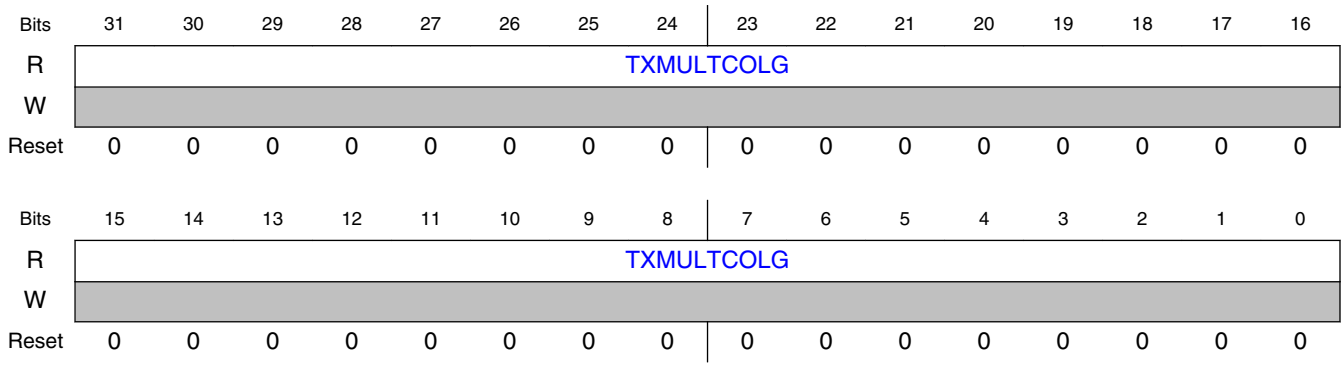
61.7.1.196.1 Offset

Register	Offset
MAC_TX_MULTIPLE_COLLISION_GOOD_PACKETS	750h

61.7.1.196.2 Function

This register provides the number of successfully transmitted packets by DWC_ether_qos after multiple collisions in the half-duplex mode.

61.7.1.196.3 Diagram



61.7.1.196.4 Fields

Field	Function
31-0 TXMULTCOLG	Tx Multiple Collision Good Packets This field indicates the number of successfully transmitted packets after multiple collisions in the half-duplex mode.

61.7.1.197 Deferred Packets Transmitted (MAC_TX_DEFERRED_PACKETS)

61.7.1.197.1 Offset

Register	Offset
MAC_TX_DEFERRED_PACKETS	754h

61.7.1.197.2 Function

This register provides the number of successfully transmitted by DWC_ether_qos after a deferral in the half-duplex mode.

61.7.1.197.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXDEFRD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXDEFRD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.197.4 Fields

Field	Function
31-0 TXDEFRD	Tx Deferred Packets This field indicates the number of successfully transmitted after a deferral in the half-duplex mode.

61.7.1.198 Late Collision Packets Transmitted (MAC_TX_LATE_COLLISION_PACKETS)

61.7.1.198.1 Offset

Register	Offset
MAC_TX_LATE_COLLISION_PACKETS	758h

61.7.1.198.2 Function

This register provides the number of packets aborted by DWC_ether_qos because of late collision error.

61.7.1.198.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLATECOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLATECOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.198.4 Fields

Field	Function
31-0 TXLATECOL	Tx Late Collision Packets This field indicates the number of packets aborted because of late collision error.

61.7.1.199 Excessive Collision Packets Transmitted (MAC_TX_EXCESSIVE_COLLISION_PACKETS)

61.7.1.199.1 Offset

Register	Offset
MAC_TX_EXCESSIVE_ COLLISION_PACKETS	75Ch

61.7.1.199.2 Function

This register provides the number of packets aborted by DWC_ether_qos because of excessive (16) collision errors.

61.7.1.199.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXEXSCOL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.199.4 Fields

Field	Function
31-0 TXEXSCOL	Tx Excessive Collision Packets This field indicates the number of packets aborted because of excessive (16) collision errors.

61.7.1.200 Carrier Error Packets Transmitted (MAC_TX_CARRIER_ERROR_PACKETS)

61.7.1.200.1 Offset

Register	Offset
MAC_TX_CARRIER_ERROR_PACKETS	760h

61.7.1.200.2 Function

This register provides the number of packets aborted by DWC_ether_qos because of carrier sense error (no carrier or loss of carrier).

61.7.1.200.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXCARR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCARR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.200.4 Fields

Field	Function
31-0 TXCARR	Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).

61.7.1.201 Bytes Transmitted in Good Packets (MAC_TX_OCTET_COUNT_GOOD)

61.7.1.201.1 Offset

Register	Offset
MAC_TX_OCTET_COUNT_GOOD	764h

61.7.1.201.2 Function

This register provides the number of bytes transmitted by DWC_ether_qos, exclusive of preamble, only in good packets.

61.7.1.201.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.201.4 Fields

Field	Function
31-0 TXOCTG	Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.

61.7.1.202 Good Packets Transmitted (MAC_TX_PACKET_COUNT_GOOD)

61.7.1.202.1 Offset

Register	Offset
MAC_TX_PACKET_COUNT_GOOD	768h

61.7.1.202.2 Function

This register provides the number of good packets transmitted by DWC_ether_qos.

61.7.1.202.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXPKTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.202.4 Fields

Field	Function
31-0 TXPKTG	Tx Packet Count Good This field indicates the number of good packets transmitted.

61.7.1.203 Packets Aborted By Excessive Deferral Error (MAC_TX_EXCESSIVE_DEFERRAL_ERROR)

61.7.1.203.1 Offset

Register	Offset
MAC_TX_EXCESSIVE_DEFERRAL_ERROR	76Ch

61.7.1.203.2 Function

This register provides the number of packets aborted by DWC_ether_qos because of excessive deferral error (deferred for more than two max-sized packet times).

61.7.1.203.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXEXSDEF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXEXSDEF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.203.4 Fields

Field	Function
31-0 TXEXSDEF	Tx Excessive Deferral Error This field indicates the number of packets aborted because of excessive deferral error (deferred for more than two max-sized packet times).

61.7.1.204 Pause Packets Transmitted (MAC_TX_PAUSE_PACKETS)

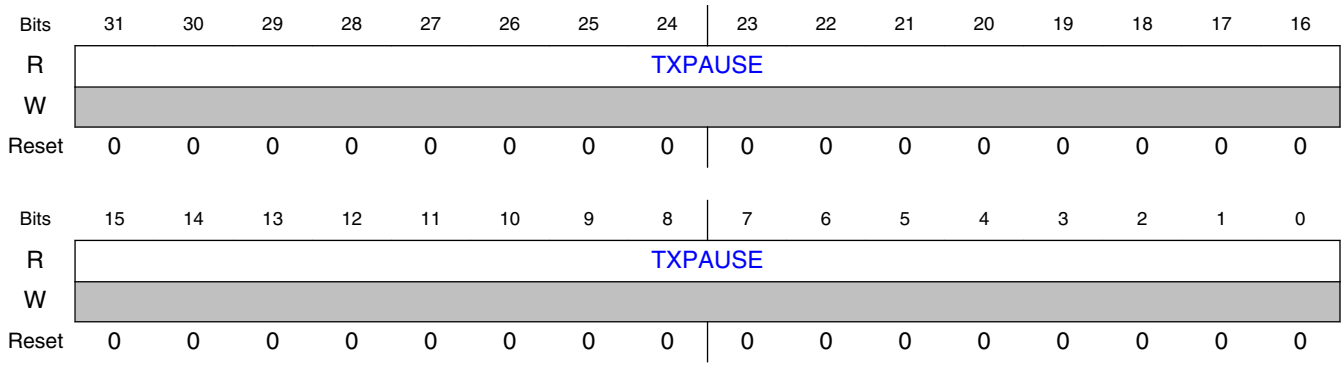
61.7.1.204.1 Offset

Register	Offset
MAC_TX_PAUSE_PACKETS	770h

61.7.1.204.2 Function

This register provides the number of good Pause packets transmitted by DWC_ether_qos.

61.7.1.204.3 Diagram



61.7.1.204.4 Fields

Field	Function
31-0 TXPAUSE	Tx Pause Packets This field indicates the number of good Pause packets transmitted.

61.7.1.205 Good VLAN Packets Transmitted (MAC_TX_VLAN_PACKETS_GOOD)

61.7.1.205.1 Offset

Register	Offset
MAC_TX_VLAN_PACKETS_GOOD	774h

61.7.1.205.2 Function

This register provides the number of good VLAN packets transmitted by DWC_ether_qos.

61.7.1.205.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXVLANG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.205.4 Fields

Field	Function
31-0 TXVLANG	Tx VLAN Packets Good This field provides the number of good VLAN packets transmitted.

61.7.1.206 Good Oversize Packets Transmitted (MAC_TX_OSIZE_PACKETS_GOOD)

61.7.1.206.1 Offset

Register	Offset
MAC_TX_OSIZE_PACKETS_GOOD	778h

61.7.1.206.2 Function

This register provides the number of packets transmitted by DWC_ether_qos without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

61.7.1.206.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXOSIZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXOSIZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.206.4 Fields

Field	Function
31-0 TXOSIZG	Tx OSize Packets Good This field indicates the number of packets transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in S2KP bit of the CONFIGURATION register).

61.7.1.207 Good and Bad Packets Received (MAC_RX_PACKETS_COUNT_GOOD_BAD)

61.7.1.207.1 Offset

Register	Offset
MAC_RX_PACKETS_COUNT_GOOD_BAD	780h

61.7.1.207.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos.

61.7.1.207.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.207.4 Fields

Field	Function
31-0 RXPKTGB	Rx Packets Count Good Bad This field indicates the number of good and bad packets received.

61.7.1.208 Bytes in Good and Bad Packets Received (MAC_RX_OCTET_COUNT_GOOD_BAD)

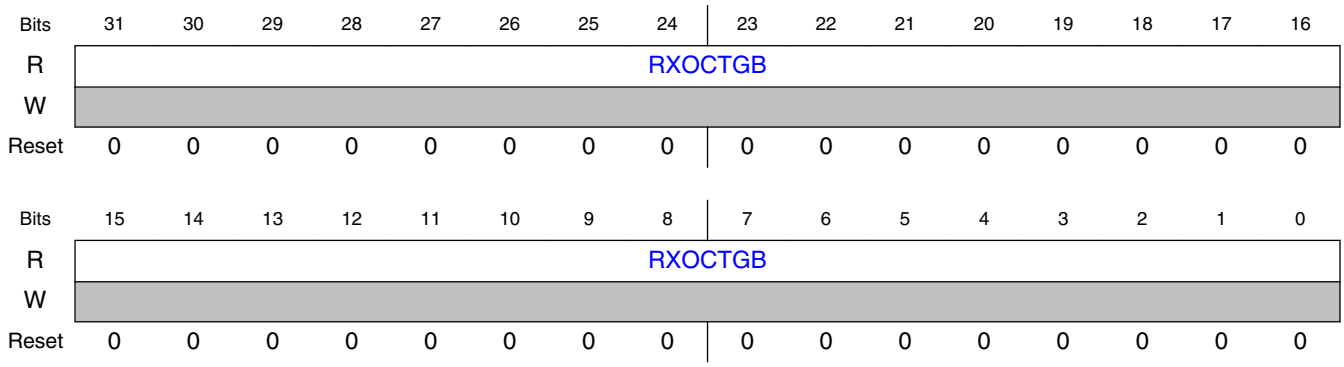
61.7.1.208.1 Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD_BAD	784h

61.7.1.208.2 Function

This register provides the number of bytes received by DWC_ther_qos, exclusive of preamble, in good and bad packets.

61.7.1.208.3 Diagram



61.7.1.208.4 Fields

Field	Function
31-0 RXOCTGB	Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.

61.7.1.209 Bytes in Good Packets Received (MAC_RX_OCTET_COUNT_GOOD)

61.7.1.209.1 Offset

Register	Offset
MAC_RX_OCTET_COUNT_GOOD	788h

61.7.1.209.2 Function

This register provides the number of bytes received by DWC_ether_qos, exclusive of preamble, only in good packets.

61.7.1.209.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOCTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.209.4 Fields

Field	Function
31-0 RXOCTG	Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.

61.7.1.210 Good Broadcast Packets Received (MAC_RX_BROADCAST_PACKETS_GOOD)

61.7.1.210.1 Offset

Register	Offset
MAC_RX_BROADCAST_PACKETS_GOOD	78Ch

61.7.1.210.2 Function

This register provides the number of good broadcast packets received by DWC_ether_qos.

61.7.1.210.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXBCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.210.4 Fields

Field	Function
31-0 RXBCASTG	Rx Broadcast Packets Good This field indicates the number of good broadcast packets received.

61.7.1.211 Good Multicast Packets Received (MAC_RX_MULTICAST_PACKETS_GOOD)

61.7.1.211.1 Offset

Register	Offset
MAC_RX_MULTICAST_PACKETS_GOOD	790h

61.7.1.211.2 Function

This register provides the number of good multicast packets received by DWC_ether_qos.

61.7.1.211.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXMCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.211.4 Fields

Field	Function
31-0 RXMCASTG	Rx Multicast Packets Good This field indicates the number of good multicast packets received.

61.7.1.212 CRC Error Packets Received (MAC_RX_CRC_ERROR_PACKETS)

61.7.1.212.1 Offset

Register	Offset
MAC_RX_CRC_ERROR_PACKETS	794h

61.7.1.212.2 Function

This register provides the number of packets received by DWC_ether_qos with CRC error.

61.7.1.212.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCRCERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.212.4 Fields

Field	Function
31-0 RXCRCERR	Rx CRC Error Packets This field indicates the number of packets received with CRC error.

61.7.1.213 Alignment Error Packets Received (MAC_RX_ALIGNMENT_ERROR_PACKETS)

61.7.1.213.1 Offset

Register	Offset
MAC_RX_ALIGNMENT_ERROR_PACKETS	798h

61.7.1.213.2 Function

This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error.

This register provides the number of packets received by DWC_ether_qos with alignment (dribble) error. It is valid only in 10/100 mode.

61.7.1.213.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXALGNERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXALGNERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.213.4 Fields

Field	Function
31-0 RXALGNERR	Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. Rx Alignment Error Packets This field indicates the number of packets received with alignment (dribble) error. It is valid only in 10/100 mode.

61.7.1.214 Runt Error Packets Received (MAC_RX_RUNT_ERROR_PACKETS)

61.7.1.214.1 Offset

Register	Offset
MAC_RX_RUNT_ERROR_PACKETS	79Ch

61.7.1.214.2 Function

This register provides the number of packets received by DWC_ether_qos with runt (length less than 64 bytes and CRC error) error.

61.7.1.214.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXRUNTERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXRUNTERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.214.4 Fields

Field	Function
31-0 RXRUNTERR	Rx Runt Error Packets This field indicates the number of packets received with runt (length less than 64 bytes and CRC error) error.

61.7.1.215 Jabber Error Packets Received (MAC_RX_JABBER_ERROR_PACKETS)

61.7.1.215.1 Offset

Register	Offset
MAC_RX_JABBER_ERROR_PACKETS	7A0h

61.7.1.215.2 Function

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error.

This register provides the number of giant packets received by DWC_ether_qos with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

61.7.1.215.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXJABERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.215.4 Fields

Field	Function
31-0 RXJABERR	Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. Rx Jabber Error Packets This field indicates the number of giant packets received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Packet mode is enabled, packets of length greater than 9,018 bytes (9,022 bytes for VLAN tagged) are considered as giant packets.

61.7.1.216 Good Undersize Packets Received (MAC_RX_UNDERSIZE_PACKETS_GOOD)

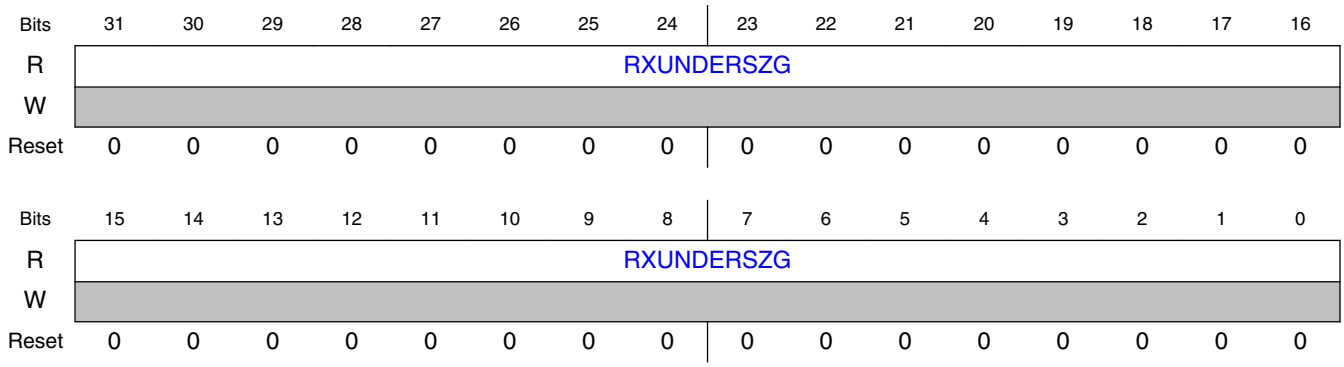
61.7.1.216.1 Offset

Register	Offset
MAC_RX_UNDERSIZE_PACKETS_GOOD	7A4h

61.7.1.216.2 Function

This register provides the number of packets received by DWC_ether_qos with length less than 64 bytes, without any errors.

61.7.1.216.3 Diagram



61.7.1.216.4 Fields

Field	Function
31-0 RXUNDERSZG	Rx Undersize Packets Good This field indicates the number of packets received with length less than 64 bytes, without any errors.

61.7.1.217 Good Oversize Packets Received (MAC_RX_OVERSIZE_PACKETS_GOOD)

61.7.1.217.1 Offset

Register	Offset
MAC_RX_OVERSIZE_PACKETS_GOOD	7A8h

61.7.1.217.2 Function

This register provides the number of packets received by DWC_ether_qos without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

61.7.1.217.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOVERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOVERSZG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.217.4 Fields

Field	Function
31-0 RXOVERSZG	Rx Oversize Packets Good This field indicates the number of packets received without errors, with length greater than the maxsize (1,518 bytes or 1,522 bytes for VLAN tagged packets; 2000 bytes if enabled in the S2KP bit of the MAC_CONFIGURATION register).

61.7.1.218 Good and Bad 64-Byte Packets Received (MAC_RX_64OCTETS_PACKETS_GOOD_BAD)

61.7.1.218.1 Offset

Register	Offset
MAC_RX_64OCTETS_PACKETS_GOOD_BAD	7ACh

61.7.1.218.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length 64 bytes, exclusive of the preamble.

61.7.1.218.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX64OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.218.4 Fields

Field	Function
31-0 RX64OCTGB	Rx 64 Octets Packets Good Bad This field indicates the number of good and bad packets received with length 64 bytes, exclusive of the preamble.

61.7.1.219 Good and Bad 64-to-127 Byte Packets Received (MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD)

61.7.1.219.1 Offset

Register	Offset
MAC_RX_65TO127OCTETS_PACKETS_GOOD_BAD	7B0h

61.7.1.219.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

61.7.1.219.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX65_127OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.219.4 Fields

Field	Function
31-0 RX65_127OCTGB	Rx 65-127 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 65 and 127 (inclusive) bytes, exclusive of the preamble.

61.7.1.220 Good and Bad 128-to-255 Byte Packets Received (MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD)

61.7.1.220.1 Offset

Register	Offset
MAC_RX_128TO255OCTETS_PACKETS_GOOD_BAD	7B4h

61.7.1.220.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

61.7.1.220.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX128_255OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.220.4 Fields

Field	Function
31-0 RX128_255OCTGB	Rx 128-255 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 128 and 255 (inclusive) bytes, exclusive of the preamble.

61.7.1.221 Good and Bad 256-to-511 Byte Packets Received (MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD)

61.7.1.221.1 Offset

Register	Offset
MAC_RX_256TO511OCTETS_PACKETS_GOOD_BAD	7B8h

61.7.1.221.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

61.7.1.221.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX256_511OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.221.4 Fields

Field	Function
31-0 RX256_511OCTGB	Rx 256-511 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 256 and 511 (inclusive) bytes, exclusive of the preamble.

61.7.1.222 Good and Bad 512-to-1023 Byte Packets Received (MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD)

61.7.1.222.1 Offset

Register	Offset
MAC_RX_512TO1023OCTETS_PACKETS_GOOD_BAD	7BCh

61.7.1.222.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

61.7.1.222.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX512_1023OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX512_1023OCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.222.4 Fields

Field	Function
31-0 RX512_1023OCTGB	RX 512-1023 Octets Packets Good Bad This field indicates the number of good and bad packets received with length between 512 and 1023 (inclusive) bytes, exclusive of the preamble.

61.7.1.223 Good and Bad 1024-to-Max Byte Packets Received (MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD)

61.7.1.223.1 Offset

Register	Offset
MAC_RX_1024TOMAXOCTETS_PACKETS_GOOD_BAD	7C0h

61.7.1.223.2 Function

This register provides the number of good and bad packets received by DWC_ether_qos with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

61.7.1.223.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX1024_MAXOCTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.223.4 Fields

Field	Function
31-0 RX1024_MAXOCTGB	Rx 1024-Max Octets Good Bad This field indicates the number of good and bad packets received with length between 1024 and maxsize (inclusive) bytes, exclusive of the preamble.

61.7.1.224 Good Unicast Packets Received (MAC_RX_UNICAST_PACKETS_GOOD)

61.7.1.224.1 Offset

Register	Offset
MAC_RX_UNICAST_PACKETS_GOOD	7C4h

61.7.1.224.2 Function

This register provides the number of good unicast packets received by DWC_ether_qos.

61.7.1.224.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXUCASTG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.224.4 Fields

Field	Function
31-0 RXUCASTG	Rx Unicast Packets Good This field indicates the number of good unicast packets received.

61.7.1.225 Length Error Packets Received (MAC_RX_LENGTH_ERROR_PACKETS)

61.7.1.225.1 Offset

Register	Offset
MAC_RX_LENGTH_ERROR_PACKETS	7C8h

61.7.1.225.2 Function

This register provides the number of packets received by DWC_ether_qos with length error (Length Type field not equal to packet size), for all packets with valid length field.

61.7.1.225.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLENERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLENERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.225.4 Fields

Field	Function
31-0 RXLENERR	Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.

61.7.1.226 Out-of-range Type Packets Received (MAC_RX_OUT_OF_RANGE_TYPE_PACKETS)

61.7.1.226.1 Offset

Register	Offset
MAC_RX_OUT_OF_RANGE_TYPE_PACKETS	7CCh

61.7.1.226.2 Function

This register provides the number of packets received by DWC_ether_qos with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

61.7.1.226.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXOUTOFRNG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXOUTOFRNG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.226.4 Fields

Field	Function
31-0 RXOUTOFRNG	Rx Out of Range Type Packet This field indicates the number of packets received with length field not equal to the valid packet size (greater than 1,500 but less than 1,536).

61.7.1.227 Pause Packets Received (MAC_RX_PAUSE_PACKETS)

61.7.1.227.1 Offset

Register	Offset
MAC_RX_PAUSE_PACKETS	7D0h

61.7.1.227.2 Function

This register provides the number of good and valid Pause packets received by DWC_ether_qos.

61.7.1.227.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPAUSEPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.227.4 Fields

Field	Function
31-0 RXPAUSEPKT	Rx Pause Packets This field indicates the number of good and valid Pause packets received.

61.7.1.228 Missed Packets Due to FIFO Overflow (MAC_RX_FIFO_OVERFLOW_PACKETS)

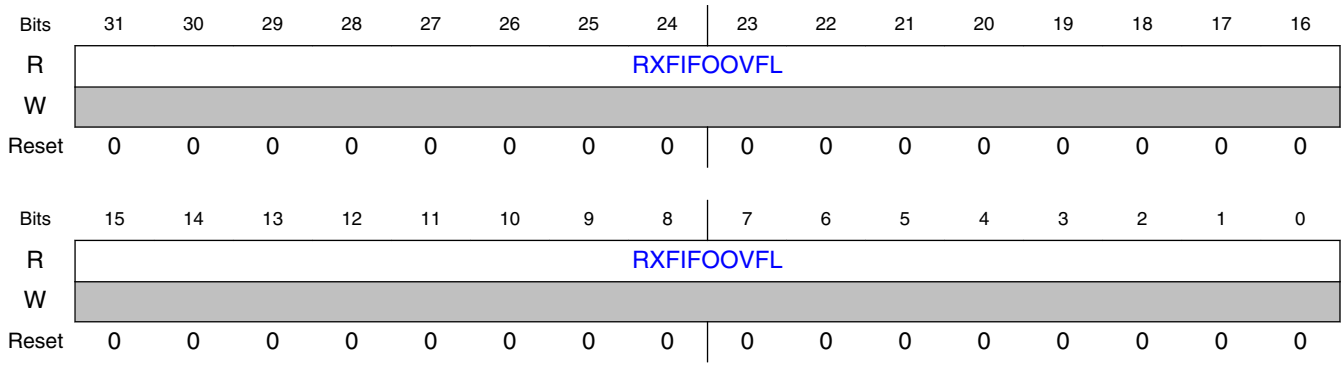
61.7.1.228.1 Offset

Register	Offset
MAC_RX_FIFO_OVERFLOW_PACKETS	7D4h

61.7.1.228.2 Function

This register provides the number of missed received packets because of FIFO overflow in DWC_ether_qos.

61.7.1.228.3 Diagram



61.7.1.228.4 Fields

Field	Function
31-0 RXFIFOOVFL	Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.

61.7.1.229 Good and Bad VLAN Packets Received (MAC_RX_VLAN_PACKETS_GOOD_BAD)

61.7.1.229.1 Offset

Register	Offset
MAC_RX_VLAN_PACKETS_GOOD_BAD	7D8h

61.7.1.229.2 Function

This register provides the number of good and bad VLAN packets received by DWC_ether_qos.

61.7.1.229.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXVLANPKTGB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.229.4 Fields

Field	Function
31-0 RXVLANPKTGB	Rx VLAN Packets Good Bad This field indicates the number of good and bad VLAN packets received.

61.7.1.230 Watchdog Error Packets Received (MAC_RX_WATCHDOG_ERROR_PACKETS)

61.7.1.230.1 Offset

Register	Offset
MAC_RX_WATCHDOG_ERROR_PACKETS	7DCh

61.7.1.230.2 Function

This register provides the number of packets received by DWC_ether_qos with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

61.7.1.230.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXWDGERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXWDGERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.230.4 Fields

Field	Function
31-0 RXWDGERR	Rx Watchdog Error Packets This field indicates the number of packets received with error because of watchdog timeout error (packets with a data load larger than 2,048 bytes (when JE and WD bits are reset in MAC_CONFIGURATION register), 10,240 bytes (when JE bit is set and WD bit is reset in MAC_CONFIGURATION register), 16,384 bytes (when WD bit is set in MAC_CONFIGURATION register) or the value programmed in the MAC_WATCHDOG_TIMEOUT register).

61.7.1.231 Receive Error Packets Received (MAC_RX_RECEIVE_ERROR_PACKETS)

61.7.1.231.1 Offset

Register	Offset
MAC_RX_RECEIVE_ERROR_PACKETS	7E0h

61.7.1.231.2 Function

This register provides the number of packets received by DWC_ether_qos with Receive error or Packet Extension error on the GMII or MII interface.

61.7.1.231.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXRCVERR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.231.4 Fields

Field	Function
31-0 RXRCVERR	Rx Receive Error Packets This field indicates the number of packets received with Receive error or Packet Extension error on the GMII or MII interface.

61.7.1.232 Good Control Packets Received (MAC_RX_CONTROL_PACKETS_GOOD)

61.7.1.232.1 Offset

Register	Offset
MAC_RX_CONTROL_PACKETS_GOOD	7E4h

61.7.1.232.2 Function

This register provides the number of good control packets received by DWC_ether_qos.

61.7.1.232.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXCTRLG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.232.4 Fields

Field	Function
31-0 RXCTRLG	Rx Control Packets Good This field indicates the number of good control packets received.

61.7.1.233 Microseconds Tx LPI Asserted (MAC_TX_LPI_USEC_CNTR)

61.7.1.233.1 Offset

Register	Offset
MAC_TX_LPI_USEC_CNTR	7ECh

61.7.1.233.2 Function

This register provides the number of microseconds Tx LPI is asserted by DWC_ether_qos.

61.7.1.233.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.233.4 Fields

Field	Function
31-0 TXLPIUSC	Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. Tx LPI Microseconds Counter This field indicates the number of microseconds Tx LPI is asserted. For every Tx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

61.7.1.234 Number of Times Tx LPI Asserted (MAC_TX_LPI_TRAN_CNTR)

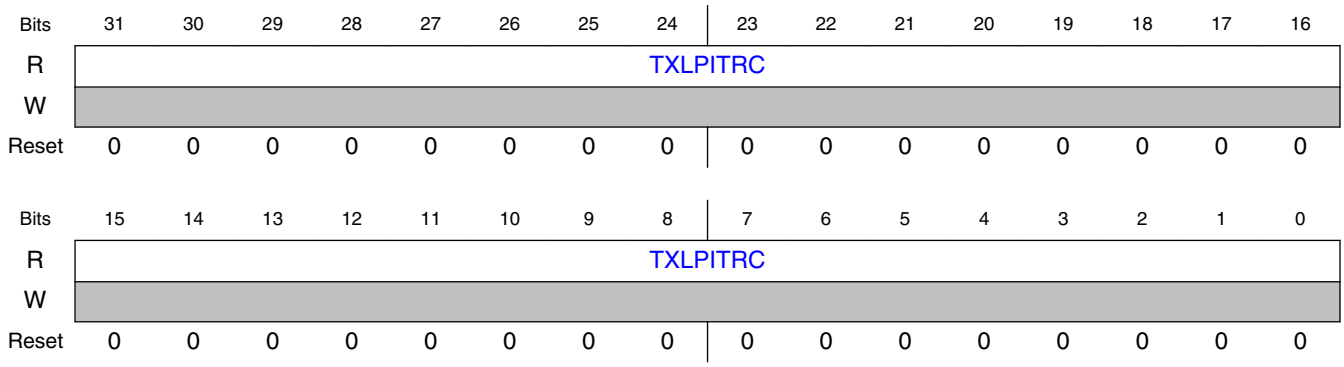
61.7.1.234.1 Offset

Register	Offset
MAC_TX_LPI_TRAN_CNTR	7F0h

61.7.1.234.2 Function

This register provides the number of times DWC_ether_qos has entered Tx LPI.

61.7.1.234.3 Diagram



61.7.1.234.4 Fields

Field	Function
31-0 TXLPITRC	Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Tx LPI Transition counter This field indicates the number of times Tx LPI Entry has occurred. Even if Tx LPI Entry occurs in Automate Mode (because of LPITXA bit set in the LPI Control and Status register), the counter will increment.

61.7.1.235 Microseconds Rx LPI Sampled (MAC_RX_LPI_USEC_CNTR)

61.7.1.235.1 Offset

Register	Offset
MAC_RX_LPI_USEC_CNTR	7F4h

61.7.1.235.2 Function

This register provides the number of microseconds Rx LPI is sampled by DWC_ether_qos.

61.7.1.235.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPIUSC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.235.4 Fields

Field	Function
31-0 RXLPIUSC	Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. Rx LPI Microseconds Counter This field indicates the number of microseconds Rx LPI is asserted. For every Rx LPI Entry and Exit, the Timer value can have an error of +/- 1 microsecond.

61.7.1.236 Number of Times Rx LPI Entered (MAC_RX_LPI_TRAN_CNTR)

61.7.1.236.1 Offset

Register	Offset
MAC_RX_LPI_TRAN_CNTR	7F8h

61.7.1.236.2 Function

This register provides the number of times DWC_ether_qos has entered Rx LPI.

61.7.1.236.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXLPITRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.236.4 Fields

Field	Function
31-0 RXLPITRC	Rx LPI Transition counter This field indicates the number of times Rx LPI Entry has occurred.

61.7.1.237 MMC IPC Receive Interrupt Mask (MAC_MMC_IPC_RX_INTERRUPT_MASK)

61.7.1.237.1 Offset

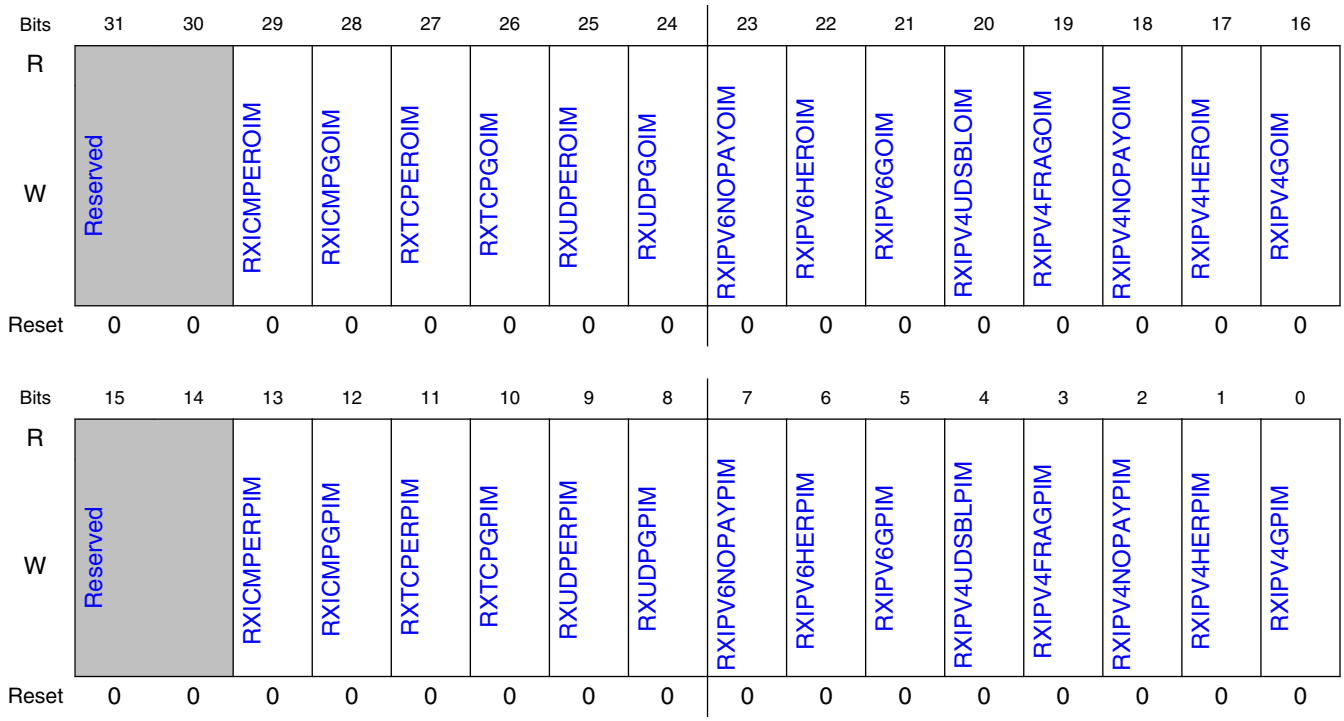
Register	Offset
MAC_MMC_IPC_RX_INTERRUPT_MASK	800h

61.7.1.237.2 Function

This register maintains the mask for the interrupt generated from the receive IPC statistic counters.

The MMC Receive Checksum Off load Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Off load) statistic counters reach half their maximum value, and when they reach their maximum values. This register is 32 bits wide.

61.7.1.237.3 Diagram



61.7.1.237.4 Fields

Field	Function
31-30 —	Reserved
29 RXICMPEROIM	MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled 1b - MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled
28 RXICMPGOIM	MMC Receive ICMP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive ICMP Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive ICMP Good Octet Counter Interrupt Mask is enabled
27 RXTCPEROIM	MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1b - MMC Receive TCP Error Octet Counter Interrupt Mask is enabled
26 RXTCPGOIM	MMC Receive TCP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive TCP Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive TCP Good Octet Counter Interrupt Mask is enabled
25 RXUDPEROIM	MMC Receive UDP Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive UDP Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive UDP Good Octet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
24 RXUDPGOIM	MMC Receive IPV6 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 No Payload Octet Counter Interrupt Mask is enabled
23 RXIPV6NOPAYOIM	MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled
22 RXIPV6HEROIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
21 RXIPV6GOIM	MMC Receive IPV6 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 Good Octet Counter Interrupt Mask is enabled
20 RXIPV4UDSBLOIM	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Mask is enabled
19 RXIPV4FRAGOIM	MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Fragmented Octet Counter Interrupt Mask is enabled
18 RXIPV4NOPAYOIM	MMC Receive IPV4 No Payload Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 No Payload Octet Counter Interrupt Mask is enabled
17 RXIPV4HEROIM	MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled
16 RXIPV4GOIM	MMC Receive IPV4 Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Good Octet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Good Octet Counter Interrupt Mask is enabled
15-14 —	Reserved
13 RXICMPERPIM	MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled
12 RXICMPGPIM	MMC Receive ICMP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive ICMP Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive ICMP Good Packet Counter Interrupt Mask is enabled
11 RXTCPERPIM	MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive TCP Error Packet Counter Interrupt Mask is enabled

Table continues on the next page...

Field	Function
10 RXTCPGPIM	MMC Receive TCP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive TCP Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive TCP Good Packet Counter Interrupt Mask is enabled
9 RXUDPERPIM	MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive UDP Error Packet Counter Interrupt Mask is enabled
8 RXUDPGPIM	MMC Receive UDP Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive UDP Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive UDP Good Packet Counter Interrupt Mask is enabled
7 RXIPV6NOPAYPIM	MMC Receive IPV6 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 No Payload Packet Counter Interrupt Mask is enabled
6 RXIPV6HERPIM	MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled
5 RXIPV6GPIM	MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled
4 RXIPV4UDSBLPIM	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Mask is enabled
3 RXIPV4FRAGPIM	MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Fragmented Packet Counter Interrupt Mask is enabled
2 RXIPV4NOPAYPIM	MMC Receive IPV4 No Payload Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 No Payload Packet Counter Interrupt Mask is enabled
1 RXIPV4HERPIM	MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled
0 RXIPV4GPIM	MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 Good Packet Counter Interrupt Mask is disabled 1b - MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled

61.7.1.238 MMC IPC Receive Interrupt (MAC_MMC_IPC_RX_INTERRUPT)

61.7.1.238.1 Offset

Register	Offset
MAC_MMC_IPC_RX_INTERRUPT	808h

61.7.1.238.2 Function

This register maintains the interrupt that the receive IPC statistic counters generate.

The MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32 bit counter and 0xFFFF for 16 bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones.

The MMC Receive Checksum Offload Interrupt register is 32 bit wide. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counter's least-significant byte lane (Bits[7:0]) must be read to clear the interrupt bit.

61.7.1.238.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		RXICMPEROIS	RXICMPGOIS	RXTCPEROIS	RXTCPGOIS	RXUDPEROIS	RXUDPGOIS	RXIPV6NOPAYOIS	RXIPV6HEROIS	RXIPV6GOIS	RXIPV4UDSBLOIS	RXIPV4FRAGOIS	RXIPV4NOPAYOIS	RXIPV4HEROIS	RXIPV4GOIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		RXICMPERPIS	RXICMPGPIS	RXTCPERPIS	RXTCPGPIS	RXUDPERPIS	RXUDPGPIS	RXIPV6NOPAYPIS	RXIPV6HERPIS	RXIPV6GPIS	RXIPV4UDSBLPIS	RXIPV4FRAGPIS	RXIPV4NOPAYPIS	RXIPV4HERPIS	RXIPV4GPIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.238.4 Fields

Field	Function
31-30 —	Reserved
29 RXICMPEROIS	MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1b - MMC Receive ICMP Error Octet Counter Interrupt Status detected
28 RXICMPGOIS	MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Octet Counter Interrupt Status This bit is set when the rxicmp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive ICMP Good Octet Counter Interrupt Status not detected 1b - MMC Receive ICMP Good Octet Counter Interrupt Status detected
27 RXTCPEROIS	MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive TCP Error Octet Counter Interrupt Status not detected 1b - MMC Receive TCP Error Octet Counter Interrupt Status detected
26 RXTCPGOIS	MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Octet Counter Interrupt Status This bit is set when the rxtcp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive TCP Good Octet Counter Interrupt Status not detected 1b - MMC Receive TCP Good Octet Counter Interrupt Status detected
25 RXUDPEROIS	MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive UDP Error Octet Counter Interrupt Status not detected

Table continues on the next page...

Field	Function
	1b - MMC Receive UDP Error Octet Counter Interrupt Status detected
24 RXUDPGOIS	MMC Receive UDP Good Octet Counter Interrupt Status This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive UDP Good Octet Counter Interrupt Status This bit is set when the rxudp_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive UDP Good Octet Counter Interrupt Status not detected 1b - MMC Receive UDP Good Octet Counter Interrupt Status detected
23 RXIPV6NOPAYOIS	MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Octet Counter Interrupt Status This bit is set when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 No Payload Octet Counter Interrupt Status not detected 1b - MMC Receive IPV6 No Payload Octet Counter Interrupt Status detected
22 RXIPV6HEROIS	MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1b - MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected
21 RXIPV6GOIS	MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Good Octet Counter Interrupt Status This bit is set when the rxipv6_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 Good Octet Counter Interrupt Status not detected 1b - MMC Receive IPV6 Good Octet Counter Interrupt Status detected
20 RXIPV4UDSBLOIS	MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status This bit is set when the rxipv4_udsbl_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status not detected 1b - MMC Receive IPV4 UDP Checksum Disabled Octet Counter Interrupt Status detected
19	MMC Receive IPV4 Fragmented Octet Counter Interrupt Status This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value.

Table continues on the next page...

Field	Function
RXIPV4FRAGOIS	MMC Receive IPV4 Fragmented Octet Counter Interrupt Status This bit is set when the rxipv4_frag_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Fragmented Octet Counter Interrupt Status detected
18 RXIPV4NOPAYOIS	MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Octet Counter Interrupt Status This bit is set when the rxipv4_nopay_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 No Payload Octet Counter Interrupt Status not detected 1b - MMC Receive IPV4 No Payload Octet Counter Interrupt Status detected
17 RXIPV4HEROIS	MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected
16 RXIPV4GOIS	MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Good Octet Counter Interrupt Status This bit is set when the rxipv4_gd_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Good Octet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Good Octet Counter Interrupt Status detected
15-14 —	Reserved
13 RXICMPERPIS	MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1b - MMC Receive ICMP Error Packet Counter Interrupt Status detected
12 RXICMPGPIS	MMC Receive ICMP Good Packet Counter Interrupt Status This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive ICMP Good Packet Counter Interrupt Status

Table continues on the next page...

Field	Function
	This bit is set when the rxicmp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive ICMP Good Packet Counter Interrupt Status not detected 1b - MMC Receive ICMP Good Packet Counter Interrupt Status detected
11 RXTCPERPIS	MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive TCP Error Packet Counter Interrupt Status not detected 1b - MMC Receive TCP Error Packet Counter Interrupt Status detected
10 RXTCPGPIS	MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive TCP Good Packet Counter Interrupt Status This bit is set when the rxtcp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive TCP Good Packet Counter Interrupt Status not detected 1b - MMC Receive TCP Good Packet Counter Interrupt Status detected
9 RXUDPERPIS	MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive UDP Error Packet Counter Interrupt Status not detected 1b - MMC Receive UDP Error Packet Counter Interrupt Status detected
8 RXUDPGPIS	MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. MC Receive UDP Good Packet Counter Interrupt Status This bit is set when the rxudp_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive UDP Good Packet Counter Interrupt Status not detected 1b - MMC Receive UDP Good Packet Counter Interrupt Status detected
7 RXIPV6NOPAYPIS	MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 No Payload Packet Counter Interrupt Status This bit is set when the rxipv6_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 No Payload Packet Counter Interrupt Status not detected 1b - MMC Receive IPV6 No Payload Packet Counter Interrupt Status detected

Table continues on the next page...

Field	Function
6 RXIPV6HERPIS	MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1b - MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected
5 RXIPV6GPIS	MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1b - MMC Receive IPV6 Good Packet Counter Interrupt Status detected
4 RXIPV4UDSBLPIS	MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status This bit is set when the rxipv4_udsbl_pkts counter reaches half of the maximum value or the maximum value. 0b - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status not detected 1b - MMC Receive IPV4 UDP Checksum Disabled Packet Counter Interrupt Status detected
3 RXIPV4FRAGPIS	MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Fragmented Packet Counter Interrupt Status This bit is set when the rxipv4_frag_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Fragmented Packet Counter Interrupt Status detected
2 RXIPV4NOPAYPIS	MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 No Payload Packet Counter Interrupt Status This bit is set when the rxipv4_nopay_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 No Payload Packet Counter Interrupt Status not detected 1b - MMC Receive IPV4 No Payload Packet Counter Interrupt Status detected
1 RXIPV4HERPIS	MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected

Table continues on the next page...

Field	Function
0 RXIPV4GPIS	MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1b - MMC Receive IPV4 Good Packet Counter Interrupt Status detected

61.7.1.239 Good IPv4 Datagrams Received (MAC_RXIPV4_GOOD_PACKETS)

61.7.1.239.1 Offset

Register	Offset
MAC_RXIPV4_GOOD_PACKETS	810h

61.7.1.239.2 Function

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

61.7.1.239.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.239.4 Fields

Field	Function
31-0 RXIPV4GDPKT	RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.

61.7.1.240 IPv4 Datagrams Received with Header Errors (MAC_RXIPV4_HEADER_ERROR_PACKETS)

61.7.1.240.1 Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_PACKETS	814h

61.7.1.240.2 Function

RxIPv4 Header Error Packets

This register provides the number of IPv4 datagrams received by DWC_ether_qos with header (checksum, length, or version mismatch) errors.

61.7.1.240.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.240.4 Fields

Field	Function
31-0 RXIPV4HDRERRPKT	RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.

61.7.1.241 IPv4 Datagrams Received with No Payload (MAC_RXIPV4_NO_PAYLOAD_PACKETS)

61.7.1.241.1 Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_PACKETS	818h

61.7.1.241.2 Function

This register provides the number of IPv4 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

61.7.1.241.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4NOPAYPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.241.4 Fields

Field	Function
31-0	RxIPv4 Payload Packets This field indicates the number of IPv4 datagram packets received that did not have a TCP, UDP, or ICMP payload.

Field	Function
RXIPV4NOPAYPKT	

61.7.1.242 IPv4 Datagrams Received with Fragmentation (MAC_RXIPV4_FRAGMENTED_PACKETS)

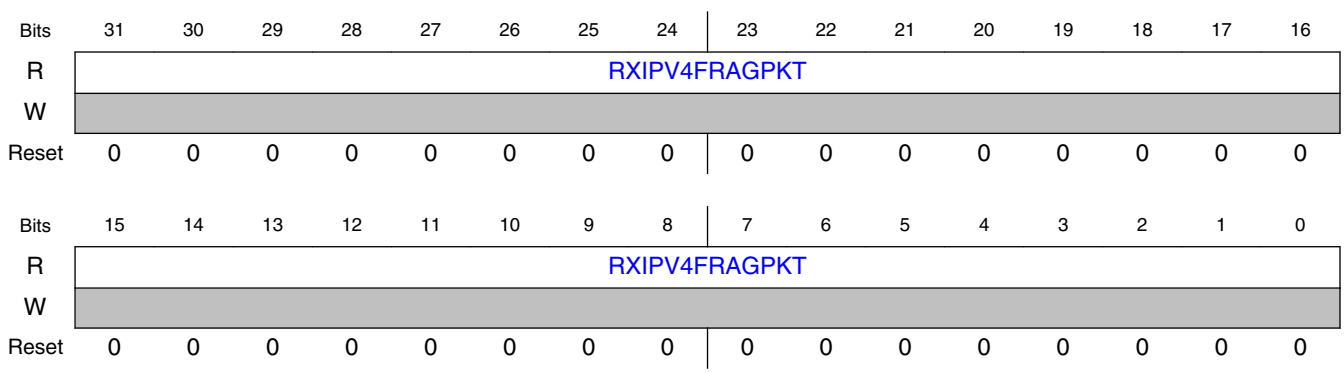
61.7.1.242.1 Offset

Register	Offset
MAC_RXIPV4_FRAGMENTED_PACKETS	81Ch

61.7.1.242.2 Function

This register provides the number of good IPv4 datagrams received by DWC_ether_qos with fragmentation.

61.7.1.242.3 Diagram



61.7.1.242.4 Fields

Field	Function
31-0 RXIPV4FRAGPKT	RxIPv4 Fragmented Packets This field indicates the number of good IPv4 datagrams received with fragmentation.

61.7.1.243 IPv4 Datagrams Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS)

61.7.1.243.1 Offset

Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLED_PACKETS	820h

61.7.1.243.2 Function

This register provides the number of good IPv4 datagrams received by DWC_ether_qos that had a UDP payload with checksum disabled.

61.7.1.243.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4UDSBLPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4UDSBLPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.243.4 Fields

Field	Function
31-0 RXIPV4UDSBLPKT	RxIPv4 UDP Checksum Disabled Packets This field indicates the number of good IPv4 datagrams received that had a UDP payload with checksum disabled.

61.7.1.244 Good IPv6 Datagrams Received (MAC_RXIPV6_GOOD_PACKETS)

61.7.1.244.1 Offset

Register	Offset
MAC_RXIPV6_GOOD_PACKETS	824h

61.7.1.244.2 Function

This register provides the number of good IPv6 datagrams received by DWC_ether_qos with the TCP, UDP, or ICMP payload.

61.7.1.244.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6GDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.244.4 Fields

Field	Function
31-0 RXIPV6GDPKT	RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.

61.7.1.245 IPv6 Datagrams Received with Header Errors (MAC_RXIPV6_HEADER_ERROR_PACKETS)

61.7.1.245.1 Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_PACKETS	828h

61.7.1.245.2 Function

This register provides the number of IPv6 datagrams received by DWC_ether_qos with header (length or version mismatch) errors.

61.7.1.245.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6HDRERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.245.4 Fields

Field	Function
31-0 RXIPV6HDRERRPKT	RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.

61.7.1.246 IPv6 Datagrams Received with No Payload (MAC_RXIPV6_NO_PAYLOAD_PACKETS)

61.7.1.246.1 Offset

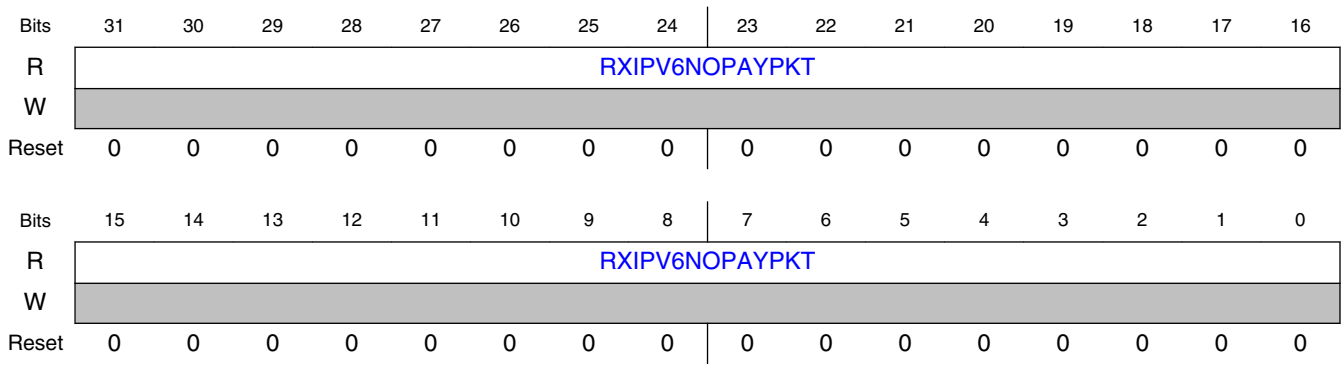
Register	Offset
MAC_RXIPV6_NO_PAYLOAD_PACKETS	82Ch

61.7.1.246.2 Function

This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload.

This register provides the number of IPv6 datagram packets received by DWC_ether_qos that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

61.7.1.246.3 Diagram



61.7.1.246.4 Fields

Field	Function
31-0 RXIPV6NOPAYPKT	RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Packets This field indicates the number of IPv6 datagram packets received that did not have a TCP, UDP, or ICMP payload. This includes all IPv6 datagrams with fragmentation or security extension headers.

61.7.1.247 IPv6 Datagrams Received with Good UDP (MAC_RXUDP_GOOD_PACKETS)

61.7.1.247.1 Offset

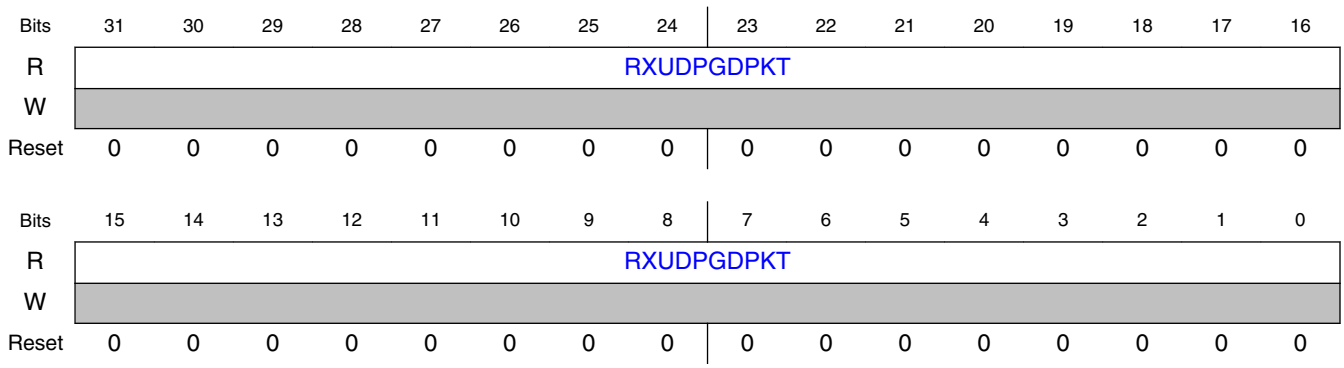
Register	Offset
MAC_RXUDP_GOOD_PACKETS	830h

61.7.1.247.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload.

This register provides the number of good IP datagrams received by DWC_ether_qos with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

61.7.1.247.3 Diagram



61.7.1.247.4 Fields

Field	Function
31-0 RXUDPGDPKT	RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. RxUDP Good Packets This field indicates the number of good IP datagrams received with a good UDP payload. This counter is not updated when the RxIPv4_UDP_Checksum_Disabled_Packets counter is incremented.

61.7.1.248 IPv6 Datagrams Received with UDP Checksum Error (MAC_RXUDP_ERROR_PACKETS)

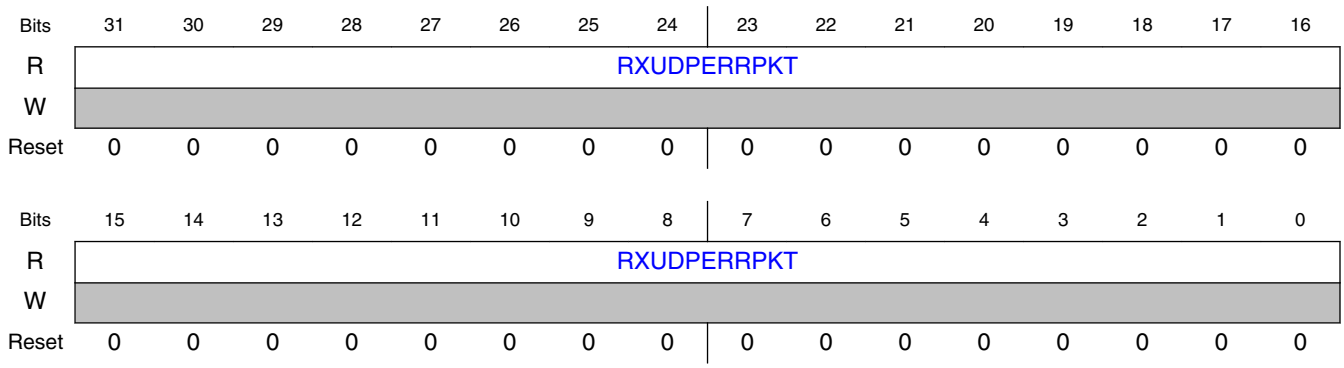
61.7.1.248.1 Offset

Register	Offset
MAC_RXUDP_ERROR_PACKETS	834h

61.7.1.248.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos whose UDP payload has a checksum error.

61.7.1.248.3 Diagram



61.7.1.248.4 Fields

Field	Function
31-0 RXUDPERRPKT	RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.

61.7.1.249 IPv6 Datagrams Received with Good TCP Payload (MAC_RXTCP_GOOD_PACKETS)

61.7.1.249.1 Offset

Register	Offset
MAC_RXTCP_GOOD_PACKETS	838h

61.7.1.249.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos with a good TCP payload.

61.7.1.249.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXTCPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXTCPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.249.4 Fields

Field	Function
31-0 RXTCPGDPKT	RxTCP Good Packets This field indicates the number of good IP datagrams received with a good TCP payload.

61.7.1.250 IPv6 Datagrams Received with TCP Checksum Error (MAC_RXTCP_ERROR_PACKETS)

61.7.1.250.1 Offset

Register	Offset
MAC_RXTCP_ERROR_PACKETS	83Ch

61.7.1.250.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos whose TCP payload has a checksum error.

61.7.1.250.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXTCPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXTCPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.250.4 Fields

Field	Function
31-0 RXTCPERRPKT	RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.

61.7.1.251 IPv6 Datagrams Received with Good ICMP Payload (MAC_RXICMP_GOOD_PACKETS)

61.7.1.251.1 Offset

Register	Offset
MAC_RXICMP_GOOD_PACKETS	840h

61.7.1.251.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos with a good ICMP payload.

61.7.1.251.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPGDPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.251.4 Fields

Field	Function
31-0 RXICMPGDPKT	RxICMP Good Packets This field indicates the number of good IP datagrams received with a good ICMP payload.

61.7.1.252 IPv6 Datagrams Received with ICMP Checksum Error (MAC_RXICMP_ERROR_PACKETS)

61.7.1.252.1 Offset

Register	Offset
MAC_RXICMP_ERROR_PACKETS	844h

61.7.1.252.2 Function

This register provides the number of good IP datagrams received by DWC_ether_qos whose ICMP payload has a checksum error.

61.7.1.252.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXICMPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXICMPERRPKT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.252.4 Fields

Field	Function
31-0 RXICMPERRPKT	RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.

61.7.1.253 Good Bytes Received in IPv4 Datagrams (MAC_RXIPV4_GOOD_OCTETS)

61.7.1.253.1 Offset

Register	Offset
MAC_RXIPV4_GOOD_OCTETS	850h

61.7.1.253.2 Function

This register provides the number of bytes received by DWC_ether_qos in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.253.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4GDOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.253.4 Fields

Field	Function
31-0 RXIPV4GDOCT	RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv4 Good Octets This field indicates the number of bytes received in good IPv4 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.254 Bytes Received in IPv4 Datagrams with Header Errors (MAC_RXIPV4_HEADER_ERROR_OCTETS)

61.7.1.254.1 Offset

Register	Offset
MAC_RXIPV4_HEADER_ERROR_OCTETS	854h

61.7.1.254.2 Function

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.254.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4HDRERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4HDRERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.254.4 Fields

Field	Function
31-0 RXIPV4HDRERROCT	RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). RxIPv4 Header Error Octets This field indicates the number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.255 Bytes Received in IPv4 Datagrams with No Payload (MAC_RXIPV4_NO_PAYLOAD_OCTETS)

61.7.1.255.1 Offset

Register	Offset
MAC_RXIPV4_NO_PAYLOAD_OCTETS	858h

61.7.1.255.2 Function

This register provides the number of bytes received by DWC_ether_qos in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.255.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV4NOPAYOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV4NOPAYOCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.255.4 Fields

Field	Function
31-0 RXIPV4NOPAYOCT	RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv4 Payload Octets This field indicates the number of bytes received in IPv4 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.256 Bytes Received in Fragmented IPv4 Datagrams (MAC_RXIPV4_FRAGMENTED_OCTETS)

61.7.1.256.1 Offset

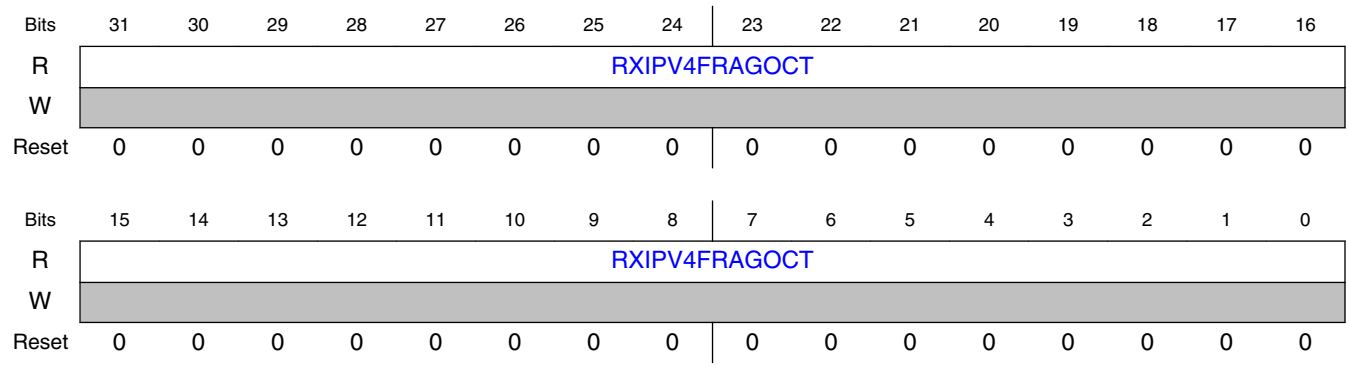
Register	Offset
MAC_RXIPV4_FRAGMENTED_OCTETS	85Ch

61.7.1.256.2 Function

This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams.

This register provides the number of bytes received by DWC_ether_qos in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.256.3 Diagram



61.7.1.256.4 Fields

Field	Function
31-0 RXIPV4FRAGOCT	RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. RxIPv4 Fragmented Octets This field indicates the number of bytes received in fragmented IPv4 datagrams. The value in the Length field of IPv4 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.257 Bytes Received with UDP Checksum Disabled (MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS)

61.7.1.257.1 Offset

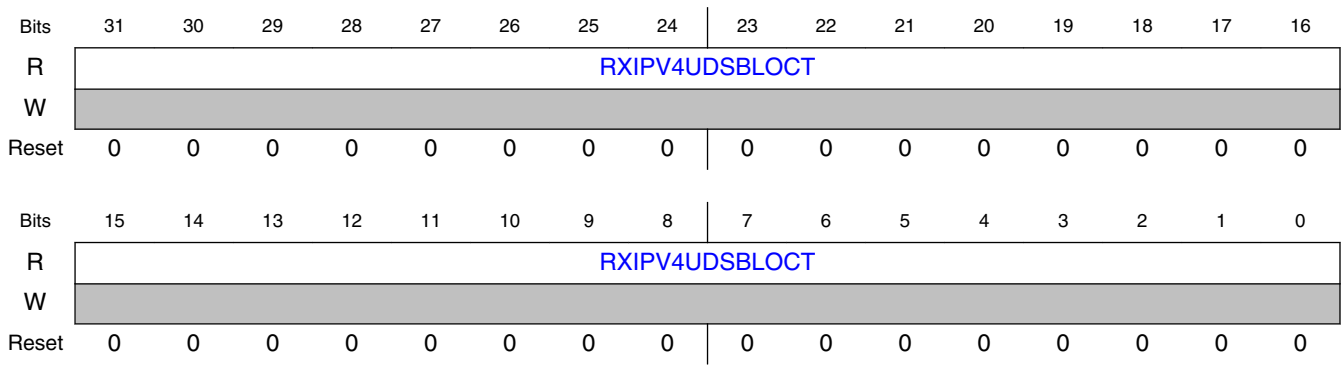
Register	Offset
MAC_RXIPV4_UDP_CHECKSUM_DISABLE_OCTETS	860h

61.7.1.257.2 Function

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.257.3 Diagram



61.7.1.257.4 Fields

Field	Function
31-0 RXIPV4UDSBLOCT	RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. RxIPv4 UDP Checksum Disable Octets This field indicates the number of bytes received in a UDP segment that had the UDP checksum disabled. This counter does not count IP Header bytes. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.258 Bytes Received in Good IPv6 Datagrams (MAC_RXIPV6_GOOD_OCTETS)

61.7.1.258.1 Offset

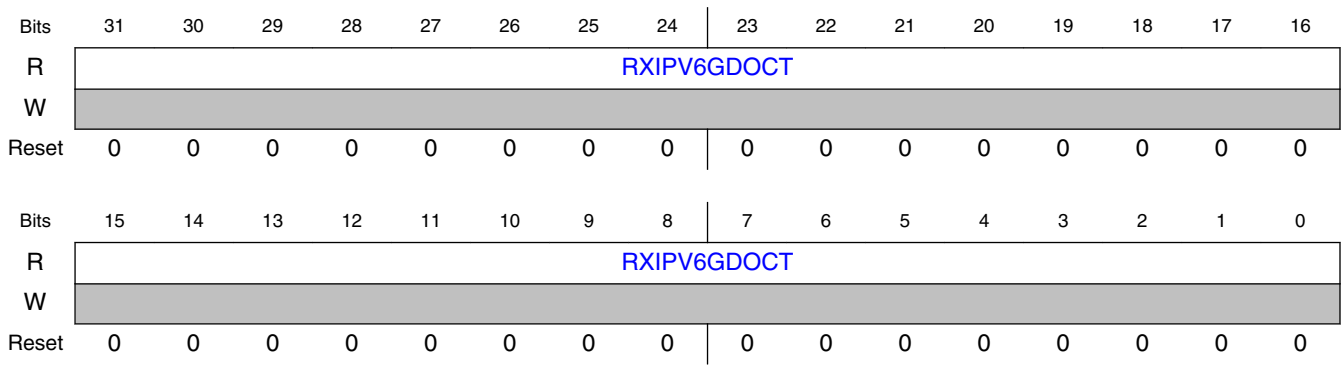
Register	Offset
MAC_RXIPV6_GOOD_OCTETS	864h

61.7.1.258.2 Function

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data.

This register provides the number of bytes received by DWC_ether_qos in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.258.3 Diagram



61.7.1.258.4 Fields

Field	Function
31-0 RXIPV6GDOCT	RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. RxIPv6 Good Octets This field indicates the number of bytes received in good IPv6 datagrams encapsulating TCP, UDP, or ICMP data. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.259 Bytes Received in IPv6 Datagrams with Data Errors (MAC_RXIPV6_HEADER_ERROR_OCTETS)

61.7.1.259.1 Offset

Register	Offset
MAC_RXIPV6_HEADER_ERROR_OCTETS	868h

61.7.1.259.2 Function

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch).

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.259.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXIPV6HDRERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXIPV6HDRERROCT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.259.4 Fields

Field	Function
31-0 RXIPV6HDRERROCT	RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.260 Bytes Received in IPv6 Datagrams with No Payload (MAC_RXIPV6_NO_PAYLOAD_OCTETS)

61.7.1.260.1 Offset

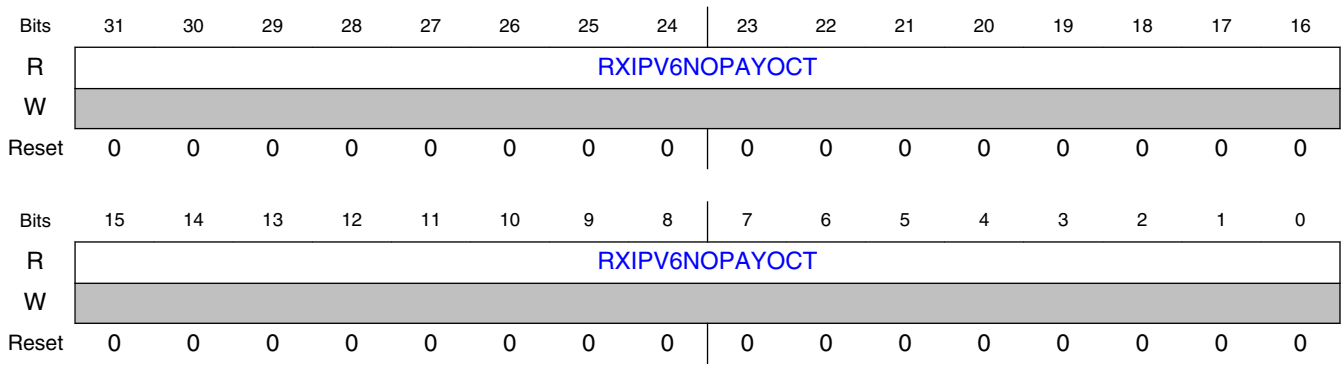
Register	Offset
MAC_RXIPV6_NO_PAYLOAD_OCTETS	86Ch

61.7.1.260.2 Function

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload.

This register provides the number of bytes received by DWC_ether_qos in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.260.3 Diagram



61.7.1.260.4 Fields

Field	Function
31-0 RXIPV6NOPAYOCT	RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. RxIPv6 Payload Octets This field indicates the number of bytes received in IPv6 datagrams that did not have a TCP, UDP, or ICMP payload. The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

61.7.1.261 Bytes Received in Good UDP Segment (MAC_RXUDP_GOOD_OCTETS)

61.7.1.261.1 Offset

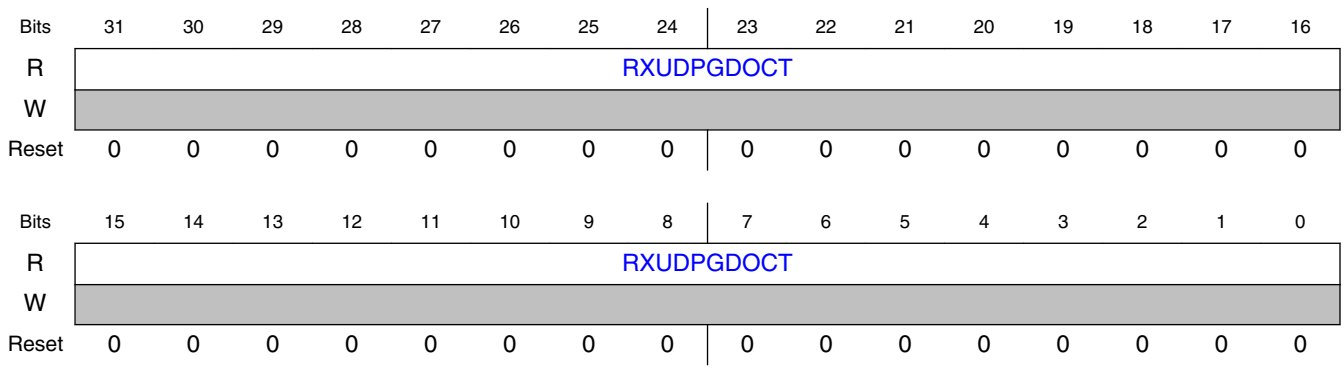
Register	Offset
MAC_RXUDP_GOOD_OCTETS	870h

61.7.1.261.2 Function

This register provides the number of bytes received by DWC_ether_qos in a good UDP segment.

This register provides the number of bytes received by DWC_ether_qos in a good UDP segment. This counter does not count IP header bytes.

61.7.1.261.3 Diagram



61.7.1.261.4 Fields

Field	Function
31-0 RXUDPGDOCT	RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. RxUDP Good Octets This field indicates the number of bytes received in a good UDP segment. This counter does not count IP header bytes.

61.7.1.262 Bytes Received in UDP Segment with Checksum Errors (MAC_RXUDP_ERROR_OCTETS)

61.7.1.262.1 Offset

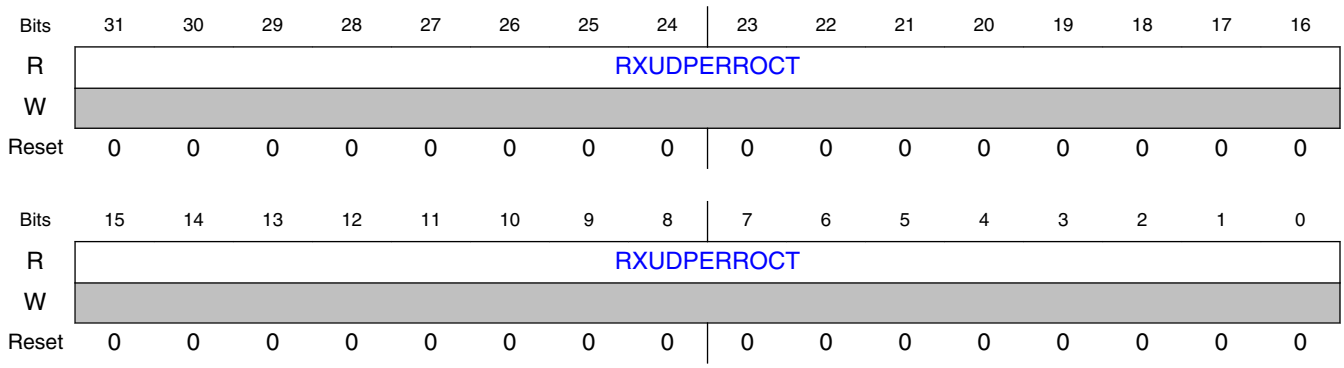
Register	Offset
MAC_RXUDP_ERROR_OCTETS	874h

61.7.1.262.2 Function

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a UDP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.262.3 Diagram



61.7.1.262.4 Fields

Field	Function
31-0 RXUDPERROCT	RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.263 Bytes Received in Good TCP Segment (MAC_RXTCP_GOOD_OCTETS)

61.7.1.263.1 Offset

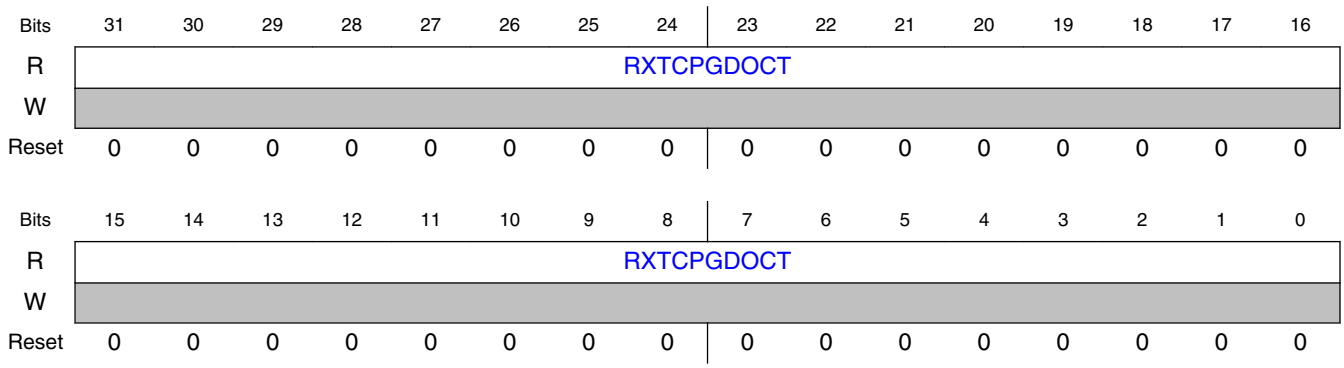
Register	Offset
MAC_RXTCP_GOOD_OCTETS	878h

61.7.1.263.2 Function

This register provides the number of bytes received by DWC_ether_qos in a good TCP segment.

This register provides the number of bytes received by DWC_ether_qos in a good TCP segment. This counter does not count IP header bytes.

61.7.1.263.3 Diagram



61.7.1.263.4 Fields

Field	Function
31-0 RXTCPGDOCT	RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. RxTCP Good Octets This field indicates the number of bytes received in a good TCP segment. This counter does not count IP header bytes.

61.7.1.264 Bytes Received in TCP Segment with Checksum Errors (MAC_RXTCP_ERROR_OCTETS)

61.7.1.264.1 Offset

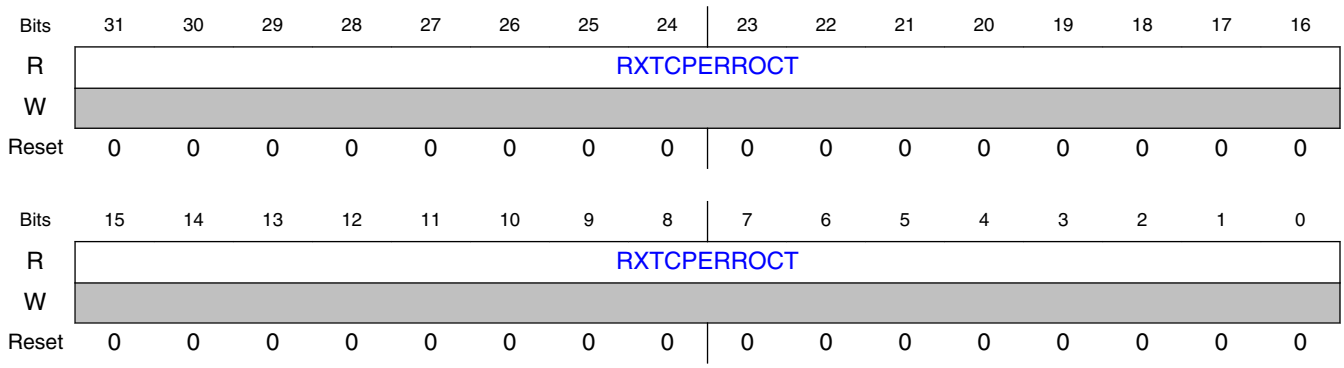
Register	Offset
MAC_RXTCP_ERROR_OCTETS	87Ch

61.7.1.264.2 Function

This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a TCP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.264.3 Diagram



61.7.1.264.4 Fields

Field	Function
31-0 RXTCPERROCT	RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.265 Bytes Received in Good ICMP Segment (MAC_RXICMP_GOOD_OCTETS)

61.7.1.265.1 Offset

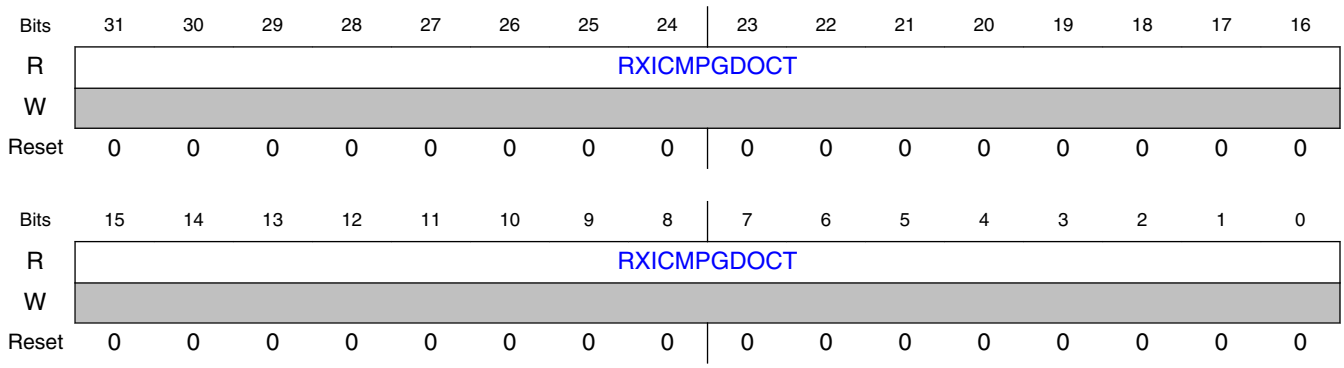
Register	Offset
MAC_RXICMP_GOOD_OCTETS	880h

61.7.1.265.2 Function

This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment.

This register provides the number of bytes received by DWC_ether_qos in a good ICMP segment. This counter does not count IP header bytes.

61.7.1.265.3 Diagram



61.7.1.265.4 Fields

Field	Function
31-0 RXICMPGDOCT	RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. RxICMP Good Octets This field indicates the number of bytes received in a good ICMP segment. This counter does not count IP header bytes.

61.7.1.266 Bytes Received in ICMP Segment with Checksum Errors (MAC_RXICMP_ERROR_OCTETS)

61.7.1.266.1 Offset

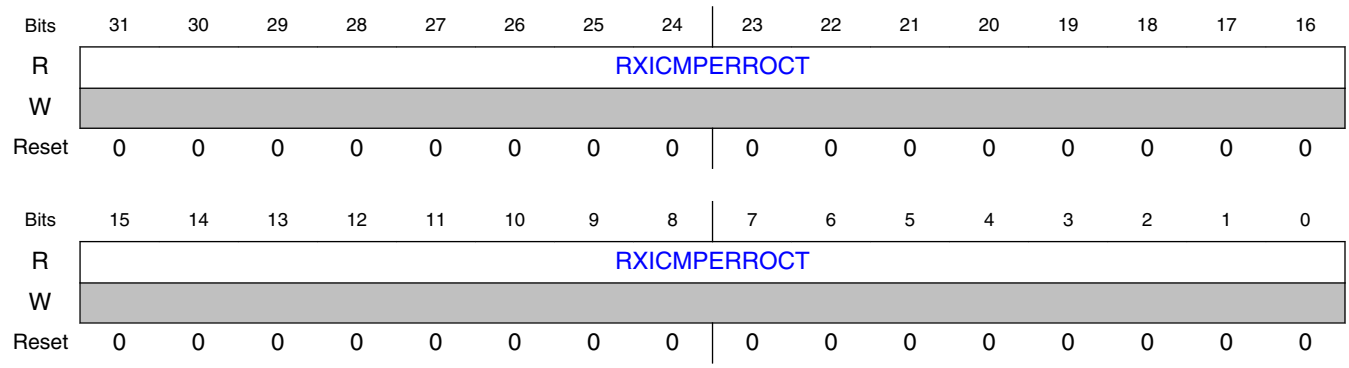
Register	Offset
MAC_RXICMP_ERROR_OCTETS	884h

61.7.1.266.2 Function

This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors.

This register provides the number of bytes received by DWC_ether_qos in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.266.3 Diagram



61.7.1.266.4 Fields

Field	Function
31-0 RXICMPERROCT	RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

61.7.1.267 MMC FPE Transmit Interrupt (MAC_MMC_FPE_TX_INTERRUPT)

61.7.1.267.1 Offset

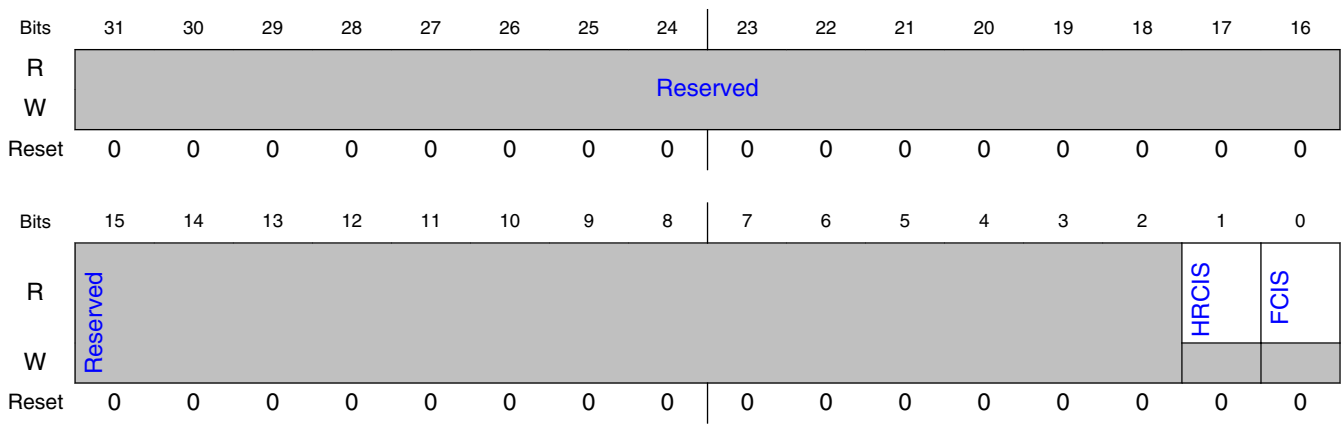
Register	Offset
MAC_MMC_FPE_TX_INTERRUPT	8A0h

61.7.1.267.2 Function

This register maintains the interrupts generated from all FPE related Transmit statistics counters.

This register maintains the interrupts generated from all FPE related Transmit statistics counters. The MMC FPE Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Transmit Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

61.7.1.267.3 Diagram



61.7.1.267.4 Fields

Field	Function
31-2 —	Reserved
1 HRCIS	MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx Hold Request Counter Interrupt Status This bit is set when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0b - MMC Tx Hold Request Counter Interrupt Status not detected

Table continues on the next page...

Field	Function
	1b - MMC Tx Hold Request Counter Interrupt Status detected
0 FCIS	MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Tx FPE Fragment Counter Interrupt status This bit is set when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Tx FPE Fragment Counter Interrupt status not detected 1b - MMC Tx FPE Fragment Counter Interrupt status detected

61.7.1.268 MMC FPE Transmit Mask Interrupt (MAC_MMC_FPE_TX_INTERRUPT_MASK)

61.7.1.268.1 Offset

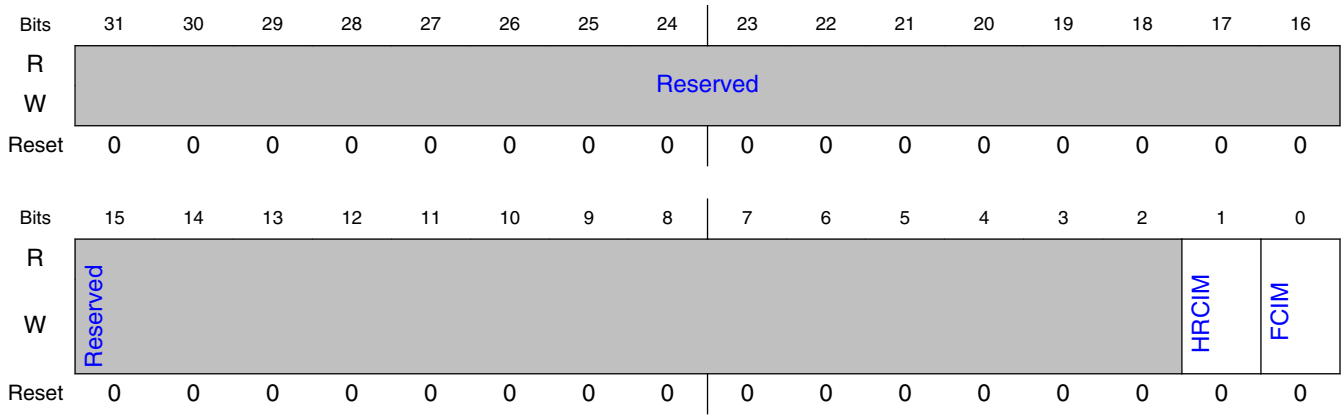
Register	Offset
MAC_MMC_FPE_TX_INTERRUPT_MASK	8A4h

61.7.1.268.2 Function

This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters.

This register maintains the masks for interrupts generated from all FPE related Transmit statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

61.7.1.268.3 Diagram



61.7.1.268.4 Fields

Field	Function
31-2 —	Reserved
1 HRCIM	MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Hold Request Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_Hold_Req_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration. 0b - MMC Transmit Hold Request Counter Interrupt Mask is disabled 1b - MMC Transmit Hold Request Counter Interrupt Mask is enabled
0 FCIM	MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Transmit Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Transmit Fragment Counter Interrupt Mask is disabled 1b - MMC Transmit Fragment Counter Interrupt Mask is enabled

61.7.1.269 MMC FPE Transmitted Fragment Counter (MAC_MMC_TX_FPE_FRAGMENT_CNTR)

61.7.1.269.1 Offset

Register	Offset
MAC_MMC_TX_FPE_FRAGMENT_CNTR	8A8h

61.7.1.269.2 Function

This register provides the number of additional mPackets transmitted due to preemption.

61.7.1.269.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXFFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXFFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.269.4 Fields

Field	Function
31-0 TXFFC	Tx FPE Fragment counter This field indicates the number of additional mPackets that has been transmitted due to preemption Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

61.7.1.270 MMC FPE Transmitted Hold Request Counter (MAC_MMC_TX_HOLD_REQ_CNTR)

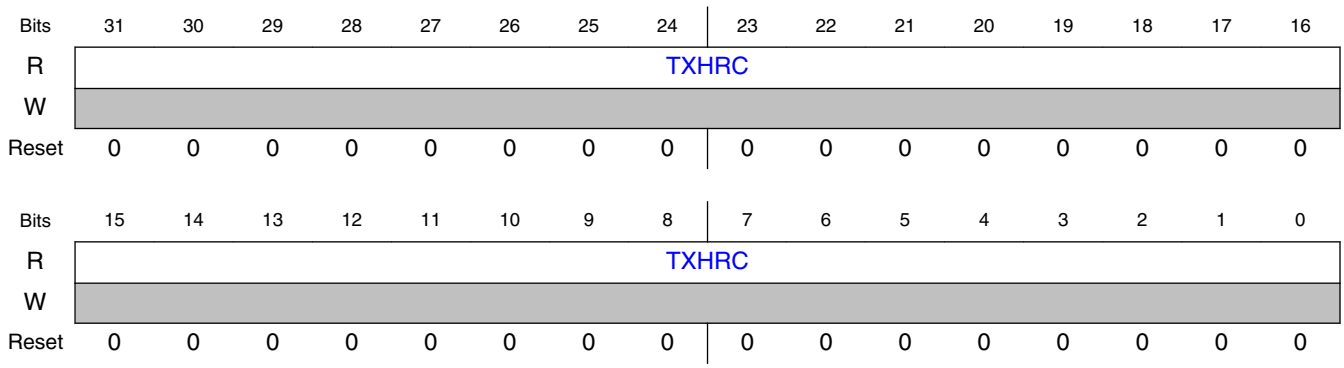
61.7.1.270.1 Offset

Register	Offset
MAC_MMC_TX_HOLD_REQ_CNTR	8ACh

61.7.1.270.2 Function

This register provides the count of number of times a hold request is given to MAC

61.7.1.270.3 Diagram



61.7.1.270.4 Fields

Field	Function
31-0 TXHRC	Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Tx Hold Request Counter This field indicates count of number of a hold request is given to MAC. Exists when any one of the RX/TX MMC counters are enabled during FPE with AV_EST Enabled configuration.

61.7.1.271 MMC FPE Receive Interrupt (MAC_MMC_FPE_RX_INTERRUPT)

61.7.1.271.1 Offset

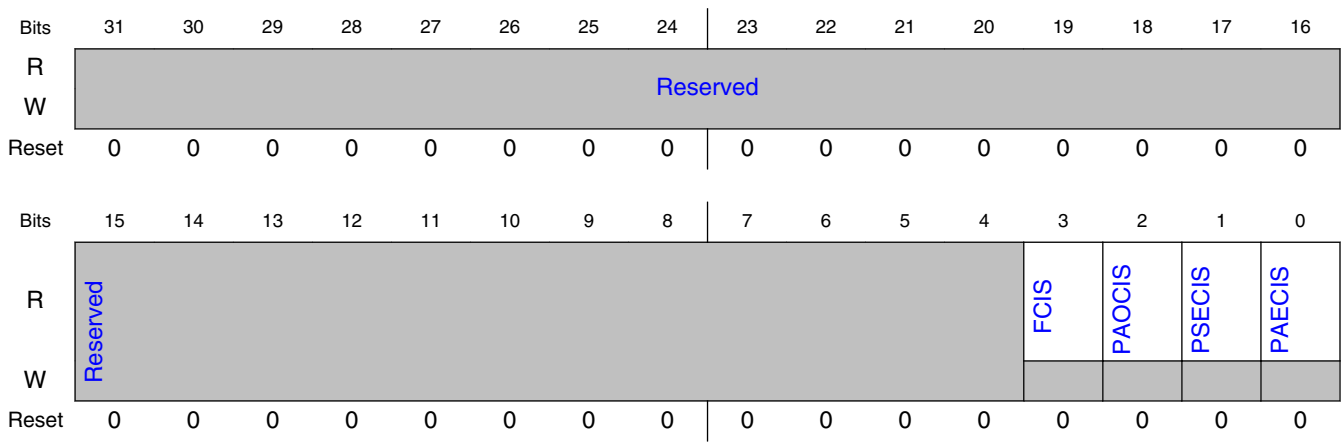
Register	Offset
MAC_MMC_FPE_RX_INTERRUPT	8C0h

61.7.1.271.2 Function

This register maintains the interrupts generated from all FPE related Receive statistics counters.

This register maintains the interrupts generated from all FPE related Receive statistics counters. The MMC FPE Receive Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32 bit counter and 0x8000 for 16 bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, the interrupts are set but the counter remains at all-ones. The MMC FPE Receive Interrupt register is a 32 bit register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (Bits[7:0]) of the respective counter must be read to clear the interrupt bit.

61.7.1.271.3 Diagram



61.7.1.271.4 Fields

Field	Function
31-4 —	Reserved
3 FCIS	MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Status This bit is set when the Rx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx FPE Fragment Counter Interrupt Status not detected 1b - MMC Rx FPE Fragment Counter Interrupt Status detected
2 PAOCIS	MMC Rx Packet Assembly OK Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Status

Table continues on the next page...

Field	Function
	This bit is set when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet Assembly OK Counter Interrupt Status not detected 1b - MMC Rx Packet Assembly OK Counter Interrupt Status detected
1 PSECIS	MMC Rx Packet SMD Error Counter Interrupt Status This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Status This bit is set when the Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet SMD Error Counter Interrupt Status not detected 1b - MMC Rx Packet SMD Error Counter Interrupt Status detected
0 PAECIS	MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Status This bit is set when the Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet Assembly Error Counter Interrupt Status not detected 1b - MMC Rx Packet Assembly Error Counter Interrupt Status detected

61.7.1.272 MMC FPE Receive Interrupt Mask (MAC_MMC_FPE_RX_INTERRUPT_MASK)

61.7.1.272.1 Offset

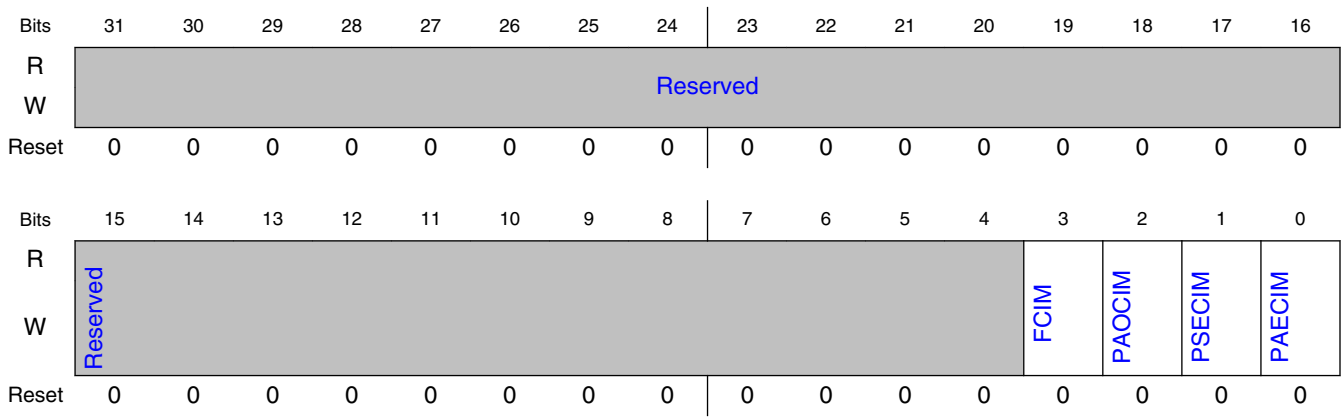
Register	Offset
MAC_MMC_FPE_RX_INTERRUPT_MASK	8C4h

61.7.1.272.2 Function

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters.

This register maintains the masks for interrupts generated from all FPE related Receive statistics counters. The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when FPE related receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bit wide.

61.7.1.272.3 Diagram



61.7.1.272.4 Fields

Field	Function
31-4 —	Reserved
3 FCIM	MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx FPE Fragment Counter Interrupt Mask Setting this bit masks the interrupt when the Tx_FPE_Fragment_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx FPE Fragment Counter Interrupt Mask is disabled 1b - MMC Rx FPE Fragment Counter Interrupt Mask is enabled
2 PAOCIM	MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly OK Counter Interrupt Mask Setting this bit masks the interrupt when the Rx_Packet_Assemble_Ok_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet Assembly OK Counter Interrupt Mask is disabled 1b - MMC Rx Packet Assembly OK Counter Interrupt Mask is enabled
1 PSECIM	MMC Rx Packet SMD Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet SMD Error Counter Interrupt Mask

Table continues on the next page...

Field	Function
	Setting this bit masks the interrupt when the R Rx_Packet_SMD_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet SMD Error Counter Interrupt Mask is disabled 1b - MMC Rx Packet SMD Error Counter Interrupt Mask is enabled
0 PAECIM	MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. MMC Rx Packet Assembly Error Counter Interrupt Mask Setting this bit masks the interrupt when the R Rx_Packet_Assemble_Err_Cntr counter reaches half of the maximum value or the maximum value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration. 0b - MMC Rx Packet Assembly Error Counter Interrupt Mask is disabled 1b - MMC Rx Packet Assembly Error Counter Interrupt Mask is enabled

61.7.1.273 MMC Receive Packet Reassembly Error Counter (MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR)

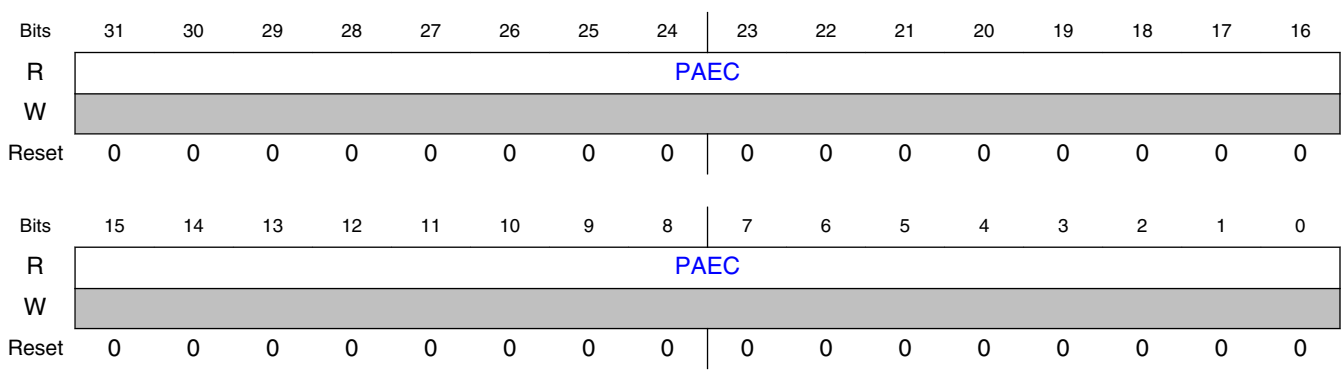
61.7.1.273.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_ERR_CNTR	8C8h

61.7.1.273.2 Function

This register provides the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value.

61.7.1.273.3 Diagram



61.7.1.273.4 Fields

Field	Function
31-0 PAEC	Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value. Rx Packet Assembly Error Counter This field indicates the number of MAC frames with reassembly errors on the Receiver, due to mismatch in the Fragment Count value. Exists when any one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

61.7.1.274 MMC Receive Packet SMD Error Counter (MAC_MMC_RX_PACKET_SMD_ERR_CNTR)

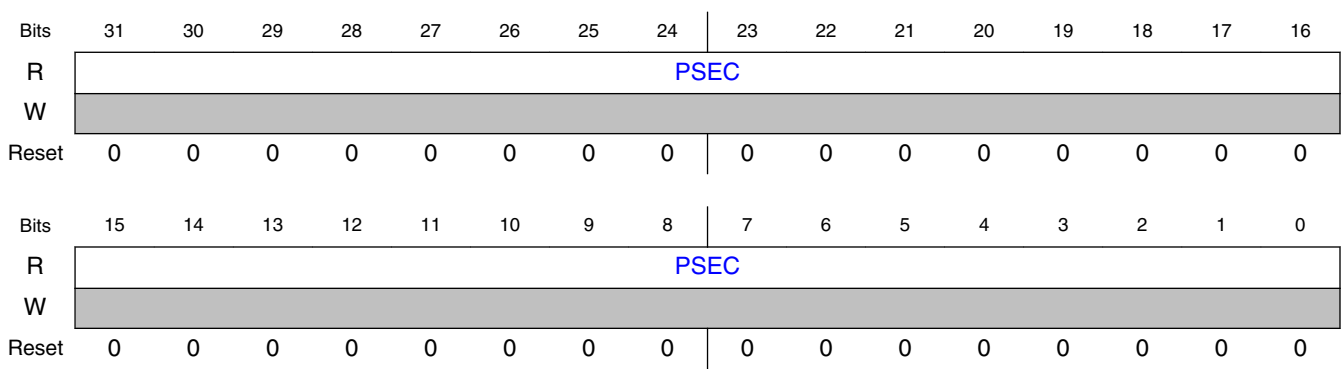
61.7.1.274.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_SMD_ERR_CNTR	8CCh

61.7.1.274.2 Function

This register provides the number of received MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame.

61.7.1.274.3 Diagram



61.7.1.274.4 Fields

Field	Function
31-0 PSEC	Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame. Rx Packet SMD Error Counter This field indicates the number of MAC frames rejected due to unknown SMD value and MAC frame fragments rejected due to arriving with an SMD-C when there was no preceding preempted frame. Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

61.7.1.275 MMC Receive Packet Successful Reassembly Counter (MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR)

61.7.1.275.1 Offset

Register	Offset
MAC_MMC_RX_PACKET_ASSEMBLY_OK_CNTR	8D0h

61.7.1.275.2 Function

This register provides the number of MAC frames that were successfully reassembled and delivered to MAC.

61.7.1.275.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAOC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.275.4 Fields

Field	Function
31-0 PAOC	Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Rx Packet Assembly OK Counter This field indicates the number of MAC frames that were successfully reassembled and delivered to MAC. Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

61.7.1.276 MMC FPE Received Fragment Counter (MAC_MMC_RX_FPE_FRAGMENT_CNTR)

61.7.1.276.1 Offset

Register	Offset
MAC_MMC_RX_FPE_FRAGMENT_CNTR	8D4h

61.7.1.276.2 Function

This register provides the number of additional mPackets received due to preemption.

61.7.1.276.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FFC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.276.4 Fields

Field	Function
31-0 FFC	Rx FPE Fragment Counter This field indicates the number of additional mPackets received due to preemption Exists when at least one of the RX/TX MMC counters are enabled during FPE Enabled configuration.

61.7.1.277 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL0)

61.7.1.277.1 Offset

Register	Offset
MAC_L3_L4_CONTROL0	900h

61.7.1.277.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

61.7.1.277.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN0	Reserved	DMCHN0			Reserved		L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Reserved	L4PEN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM0					L3HSBM0					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Reserved	L3PEN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.277.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN0	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN0	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM0	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM0	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM0	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled

Table continues on the next page...

Field	Function
	1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM0	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN0	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM0	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM0	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match

Table continues on the next page...

Field	Function
	IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM0	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM0	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM0	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM0	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering.

Table continues on the next page...

Field	Function
	0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN0	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.278 Layer 4 Address 0 (MAC_LAYER4_ADDRESS0)

61.7.1.278.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS0	904h

61.7.1.278.2 Function

The MAC_LAYER4_ADDRESS0, MAC_L3_L4_CONTROL0, MAC_LAYER3_ADDR0_Reg0, MAC_LAYER3_ADDR1_Reg0, MAC_LAYER3_ADDR2_Reg0 and MAC_LAYER3_ADDR3_Reg0 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.278.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.278.4 Fields

Field	Function
31-16 L4DP0	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP0	Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.279 Layer 3 Address 0 Register 0 (MAC_LAYER3_ADDR0_REG0)

61.7.1.279.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG0	910h

61.7.1.279.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field.

For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.279.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.279.4 Fields

Field	Function
31-0 L3A00	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.280 Layer 3 Address 1 Register 0 (MAC_LAYER3_ADDR1_REG0)

61.7.1.280.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG0	914h

61.7.1.280.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field.

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.280.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.280.4 Fields

Field	Function
31-0 L3A10	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.281 Layer 3 Address 2 Register 0 (MAC_LAYER3_ADDR2_REG0)

61.7.1.281.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG0	918h

61.7.1.281.2 Function

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.281.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.281.4 Fields

Field	Function
31-0 L3A20	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.282 Layer 3 Address 3 Register 0 (MAC_LAYER3_ADDR3_REG0)

61.7.1.282.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG0	91Ch

61.7.1.282.2 Function

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets.

The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.282.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A30															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A30															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.282.4 Fields

Field	Function
31-0 L3A30	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets.

Field	Function
	When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.283 Layer 3 and Layer 4 Control of Filter 1 (MAC_L3_L4_CONTROL1)

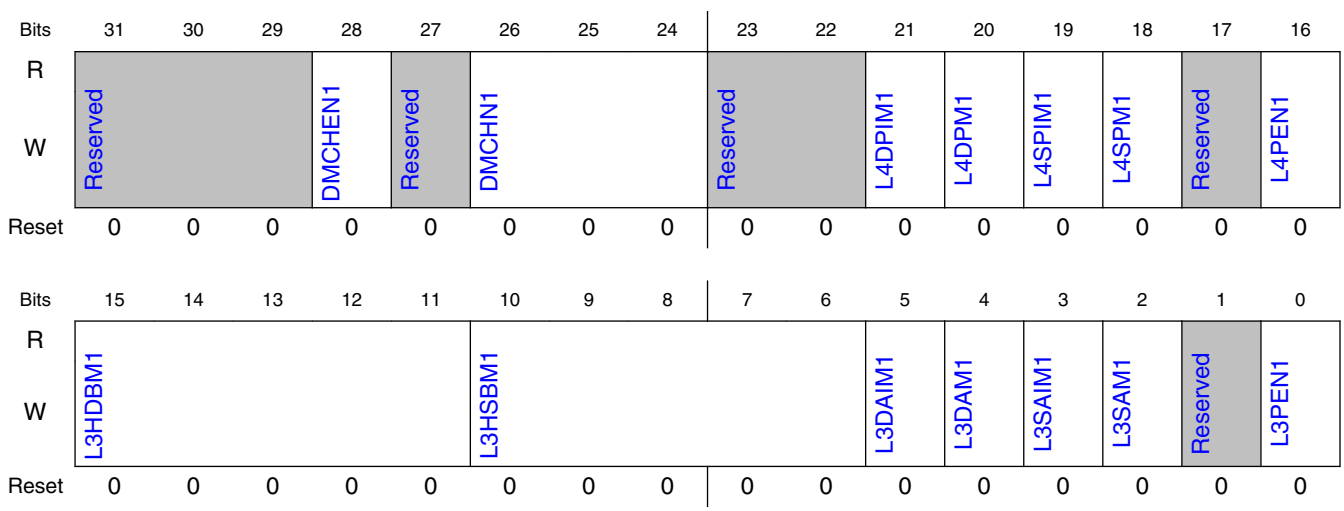
61.7.1.283.1 Offset

Register	Offset
MAC_L3_L4_CONTROL1	930h

61.7.1.283.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 1 of Layer 3 and Layer 4.

61.7.1.283.3 Diagram



61.7.1.283.4 Fields

Field	Function
31-29 —	Reserved

Table continues on the next page...

Field	Function
28 DMCHEN1	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN1	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM1	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM1 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM1	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM1	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM1	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable

Table continues on the next page...

Field	Function
	When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled
17 —	Reserved
16 L4PEN1	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM1 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM1	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM1	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked.

Table continues on the next page...

Field	Function
	- 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM1	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM1	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM1	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM1	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved

Table continues on the next page...

Field	Function
0 L3PEN1	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.284 Layer 4 Address 0 (MAC_LAYER4_ADDRESS1)

61.7.1.284.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS1	934h

61.7.1.284.2 Function

The MAC_LAYER4_ADDRESS1, MAC_L3_L4_CONTROL1, MAC_LAYER3_ADDR0_Reg1, MAC_LAYER3_ADDR1_Reg1, MAC_LAYER3_ADDR2_Reg1 and MAC_LAYER3_ADDR3_Reg1 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.284.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.284.4 Fields

Field	Function
31-16 L4DP1	Layer 4 Destination Port Number Field When the L4PEN0 bit is reset and the L4DPM1 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN1 bit is reset and the L4DPM1 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN1 and L4DPM1 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP1	Layer 4 Source Port Number Field When the L4PEN1 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN1 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN1 and L4SPM1 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.285 Layer 3 Address 0 Register 1 (MAC_LAYER3_ADDR0_REG1)

61.7.1.285.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG1	940h

61.7.1.285.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 1 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.285.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.285.4 Fields

Field	Function
31-0 L3A01	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.286 Layer 3 Address 1 Register 1 (MAC_LAYER3_ADDR1_REG1)

61.7.1.286.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG1	944h

61.7.1.286.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.286.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.286.4 Fields

Field	Function
31-0 L3A11	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.287 Layer 3 Address 2 Register 1 (MAC_LAYER3_ADDR2_REG1)

61.7.1.287.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG1	948h

61.7.1.287.2 Function

The Layer 3 Address 2 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.287.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.287.4 Fields

Field	Function
31-0 L3A21	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.288 Layer 3 Address 3 Register 1 (MAC_LAYER3_ADDR3_REG1)

61.7.1.288.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG1	94Ch

61.7.1.288.2 Function

The Layer 3 Address 3 Register 1 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.288.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A31															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.288.4 Fields

Field	Function
31-0 L3A31	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.289 Layer 3 and Layer 4 Control of Filter 2 (MAC_L3_L4_CONTROL2)

61.7.1.289.1 Offset

Register	Offset
MAC_L3_L4_CONTROL2	960h

61.7.1.289.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 2 of Layer 3 and Layer 4.

61.7.1.289.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN2	Reserved	DMCHN2			Reserved		L4DPIM2	L4DPM2	L4SPIM2	L4SPM2	Reserved	L4PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM2					L3HSBM2					L3DAIM2	L3DAM2	L3SAIM2	L3SAM2	Reserved	L3PEN2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.289.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN2	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN2	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM2	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM2 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM2	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM2	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM2	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN2	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM2 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM2	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM2	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM2	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM2	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM2	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM2	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN2	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.290 Layer 4 Address 2 (MAC_LAYER4_ADDRESS2)

61.7.1.290.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS2	964h

61.7.1.290.2 Function

The MAC_LAYER4_ADDRESS2, MAC_L3_L4_CONTROL2), MAC_LAYER3_ADDR0_Reg2), MAC_LAYER3_ADDR1_Reg2, MAC_LAYER3_ADDR2_Reg2 and MAC_LAYER3_ADDR3_Reg2 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.290.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.290.4 Fields

Field	Function
31-16 L4DP2	Layer 4 Destination Port Number Field When the L4PEN2 bit is reset and the L4DPM2 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN2 bit is reset and the L4DPM2 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN2 and L4DPM2 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP2	Layer 4 Source Port Number Field When the L4PEN2 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN2 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN2 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.291 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR0_REG2)

61.7.1.291.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG2	970h

61.7.1.291.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 2 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.291.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.291.4 Fields

Field	Function
31-0 L3A02	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.292 Layer 3 Address 0 Register 2 (MAC_LAYER3_ADDR1_REG2)

61.7.1.292.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG2	974h

61.7.1.292.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 2 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.292.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.292.4 Fields

Field	Function
31-0 L3A12	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.293 Layer 3 Address 2 Register 2 (MAC_LAYER3_ADDR2_REG2)

61.7.1.293.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG2	978h

61.7.1.293.2 Function

The Layer 3 Address 2 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.293.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.293.4 Fields

Field	Function
31-0 L3A22	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.294 Layer 3 Address 3 Register 2 (MAC_LAYER3_ADDR3_REG2)

61.7.1.294.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG2	97Ch

61.7.1.294.2 Function

The Layer 3 Address 3 Register 2 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.294.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A32															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.294.4 Fields

Field	Function
31-0 L3A32	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.295 Layer 3 and Layer 4 Control of Filter 3 (MAC_L3_L4_CONTROL3)

61.7.1.295.1 Offset

Register	Offset
MAC_L3_L4_CONTROL3	990h

61.7.1.295.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 3 of Layer 3 and Layer 4.

61.7.1.295.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN3	Reserved	DMCHN3			Reserved		L4DPIM3	L4DPM3	L4SPIM3	L4SPM3	Reserved	L4PEN3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM3					L3HSBM3					L3DAIM3	L3DAM3	L3SAIM3	L3SAM3	Reserved	L3PEN3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.295.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN3	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN3	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM3	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM3 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM3	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM3	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM3	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN3	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM3 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM3	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM3	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM3	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM3	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM3	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM3	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN3	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.296 Layer 4 Address 3 (MAC_LAYER4_ADDRESS3)

61.7.1.296.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS3	994h

61.7.1.296.2 Function

The MAC_LAYER4_ADDRESS4, MAC_L3_L4_CONTROL4, MAC_LAYER3_ADDR0_Reg4, MAC_LAYER3_ADDR1_Reg4, MAC_LAYER3_ADDR2_Reg4 and MAC_LAYER3_ADDR3_Reg4 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.296.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.296.4 Fields

Field	Function
31-16 L4DP3	Layer 4 Destination Port Number Field When the L4PEN3 bit is reset and the L4DPM3 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN3 bit is reset and the L4DPM3 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN3 and L4DPM3 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP3	Layer 4 Source Port Number Field When the L4PEN3 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN3 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN3 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.297 Layer 3 Address 0 Register 3 (MAC_LAYER3_ADDR0_REG3)

61.7.1.297.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG3	9A0h

61.7.1.297.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 3 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.297.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A03															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.297.4 Fields

Field	Function
31-0 L3A03	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.298 Layer 3 Address 1 Register 3 (MAC_LAYER3_ADDR1_REG3)

61.7.1.298.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG3	9A4h

61.7.1.298.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 3 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.298.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A13															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A13															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.298.4 Fields

Field	Function
31-0 L3A13	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.299 Layer 3 Address 2 Register 3 (MAC_LAYER3_ADDR2_REG3)

61.7.1.299.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG3	9A8h

61.7.1.299.2 Function

The Layer 3 Address 2 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.299.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A23															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.299.4 Fields

Field	Function
31-0 L3A23	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.300 Layer 3 Address 3 Register 3 (MAC_LAYER3_ADDR3_REG3)

61.7.1.300.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG3	9ACh

61.7.1.300.2 Function

The Layer 3 Address 3 Register 3 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.300.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A33															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A33															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.300.4 Fields

Field	Function
31-0 L3A33	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.301 Layer 3 and Layer 4 Control of Filter 4 (MAC_L3_L4_CONTROL4)

61.7.1.301.1 Offset

Register	Offset
MAC_L3_L4_CONTROL4	9C0h

61.7.1.301.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 3 of Layer 3 and Layer 4.

61.7.1.301.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN4	Reserved	DMCHN4			Reserved		L4DPIM4	L4DPM4	L4SPIM4	L4SPM4	Reserved	L4PEN4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM4					L3HSBM4					L3DAIM4	L3DAM4	L3SAIM4	L3SAM4	Reserved	L3PEN4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.301.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN4	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN4	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM4	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM4 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM4	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM4	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM4	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN4	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM4bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM4	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM4	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM4	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM4	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM4	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM4	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN4	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.302 Layer 4 Address 4 (MAC_LAYER4_ADDRESS4)

61.7.1.302.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS4	9C4h

61.7.1.302.2 Function

The MAC_LAYER4_ADDRESS4, MAC_L3_L4_CONTROL4, MAC_LAYER3_ADDR0_Reg4, MAC_LAYER3_ADDR1_Reg4, MAC_LAYER3_ADDR2_Reg4 and MAC_LAYER3_ADDR3_Reg4) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.302.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.302.4 Fields

Field	Function
31-16 L4DP4	Layer 4 Destination Port Number Field When the L4PEN4 bit is reset and the L4DPM4 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN4 bit is reset and the L4DPM4 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN4 and L4DPM4 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP4	Layer 4 Source Port Number Field When the L4PEN4 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN4 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN4 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.303 Layer 3 Address 0 Register 4 (MAC_LAYER3_ADDR0_REG4)

61.7.1.303.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG4	9D0h

61.7.1.303.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 4 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.303.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A04															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A04															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.303.4 Fields

Field	Function
31-0 L3A04	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.304 Layer 3 Address 1 Register 4 (MAC_LAYER3_ADDR1_REG4)

61.7.1.304.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG4	9D4h

61.7.1.304.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 4 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.304.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A14															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A14															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.304.4 Fields

Field	Function
31-0 L3A14	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.305 Layer 3 Address 2 Register 4 (MAC_LAYER3_ADDR2_REG4)

61.7.1.305.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG4	9D8h

61.7.1.305.2 Function

The Layer 3 Address 2 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.305.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A24															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.305.4 Fields

Field	Function
31-0 L3A24	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.306 Layer 3 Address 3 Register 4 (MAC_LAYER3_ADDR3_REG4)

61.7.1.306.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG4	9DCh

61.7.1.306.2 Function

The Layer 3 Address 3 Register 4 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.306.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A34															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A34															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.306.4 Fields

Field	Function
31-0 L3A34	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.307 Layer 3 and Layer 4 Control of Filter 5 (MAC_L3_L4_CONTROL5)

61.7.1.307.1 Offset

Register	Offset
MAC_L3_L4_CONTROL5	9F0h

61.7.1.307.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 5 of Layer 3 and Layer 4.

61.7.1.307.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN5	Reserved	DMCHN5			Reserved		L4DPIM5	L4DPM5	L4SPIM5	L4SPM5	Reserved	L4PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM5					L3HSBM5					L3DAIM5	L3DAM5	L3SAIM5	L3SAM5	Reserved	L3PEN5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.307.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN5	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN5	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM5	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM5 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM5	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM5	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM5	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN5	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM5 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM5	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM5	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM5	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM5	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM5	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM5	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN5	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.308 Layer 4 Address 5 (MAC_LAYER4_ADDRESS5)

61.7.1.308.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS5	9F4h

61.7.1.308.2 Function

The MAC_LAYER4_ADDRESS5, MAC_L3_L4_CONTROL5, MAC_LAYER3_ADDR0_Reg5, MAC_LAYER3_ADDR1_Reg5, MAC_LAYER3_ADDR2_Reg5 and MAC_LAYER3_ADDR3_Reg5 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.308.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.308.4 Fields

Field	Function
31-16 L4DP5	Layer 4 Destination Port Number Field When the L4PEN5 bit is reset and the L4DPM5 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN5 bit is reset and the L4DPM5 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN5 and L4DPM5 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP5	Layer 4 Source Port Number Field When the L4PEN5 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN5 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN5 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.309 Layer 3 Address 0 Register 5 (MAC_LAYER3_ADDR0_REG5)

61.7.1.309.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG5	A00h

61.7.1.309.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 5 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.309.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A05															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.309.4 Fields

Field	Function
31-0 L3A05	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.310 Layer 3 Address 1 Register 5 (MAC_LAYER3_ADDR1_REG5)

61.7.1.310.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG5	A04h

61.7.1.310.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 5 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.310.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A15															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A15															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.310.4 Fields

Field	Function
31-0 L3A15	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.311 Layer 3 Address 2 Register 5 (MAC_LAYER3_ADDR2_REG5)

61.7.1.311.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG5	A08h

61.7.1.311.2 Function

The Layer 3 Address 2 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.311.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A25															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A25															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.311.4 Fields

Field	Function
31-0 L3A25	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.312 Layer 3 Address 3 Register 5 (MAC_LAYER3_ADDR3_REG5)

61.7.1.312.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG5	A0Ch

61.7.1.312.2 Function

The Layer 3 Address 3 Register 5 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.312.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A35															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A35															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.312.4 Fields

Field	Function
31-0 L3A35	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.313 Layer 3 and Layer 4 Control of Filter 6 (MAC_L3_L4_CONTROL6)

61.7.1.313.1 Offset

Register	Offset
MAC_L3_L4_CONTROL6	A20h

61.7.1.313.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 6 of Layer 3 and Layer 4.

61.7.1.313.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DMCHEN6	Reserved	DMCHN6			Reserved		L4DPIM6	L4DPM6	L4SPIM6	L4SPM6	Reserved	L4PEN6
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3HDBM6					L3HSBM6					L3DAIM6	L3DAM6	L3SAIM6	L3SAM6	Reserved	L3PEN6
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.313.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN6	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN6	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM6	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM6 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM6	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM6	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM6	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN6	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM6 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM6	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM6	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM6	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM6	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM6	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM6	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN6	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.314 Layer 4 Address 6 (MAC_LAYER4_ADDRESS6)

61.7.1.314.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS6	A24h

61.7.1.314.2 Function

The MAC_LAYER4_ADDRESS6), MAC_L3_L4_CONTROL6, MAC_LAYER3_ADDR0_Reg6, MAC_LAYER3_ADDR1_Reg6, MAC_LAYER3_ADDR2_Reg5 and MAC_LAYER3_ADDR3_Reg6 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.314.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.314.4 Fields

Field	Function
31-16 L4DP6	Layer 4 Destination Port Number Field When the L4PEN6 bit is reset and the L4DPM6 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN6 bit is reset and the L4DPM6 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN6 and L4DPM6 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP6	Layer 4 Source Port Number Field When the L4PEN6 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN6 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN6 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.315 Layer 3 Address 0 Register 6 (MAC_LAYER3_ADDR0_REG6)

61.7.1.315.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG6	A30h

61.7.1.315.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 6 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.315.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A06															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A06															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.315.4 Fields

Field	Function
31-0 L3A06	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.316 Layer 3 Address 1 Register 6 (MAC_LAYER3_ADDR1_REG6)

61.7.1.316.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG6	A34h

61.7.1.316.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 6 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.316.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.316.4 Fields

Field	Function
31-0 L3A16	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.317 Layer 3 Address 2 Register 6 (MAC_LAYER3_ADDR2_REG6)

61.7.1.317.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG6	A38h

61.7.1.317.2 Function

The Layer 3 Address 2 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.317.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A26															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A26															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.317.4 Fields

Field	Function
31-0 L3A26	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.318 Layer 3 Address 3 Register 6 (MAC_LAYER3_ADDR3_REG6)

61.7.1.318.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG6	A3Ch

61.7.1.318.2 Function

The Layer 3 Address 3 Register 6 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.318.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A36															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A36															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.318.4 Fields

Field	Function
31-0 L3A36	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.319 Layer 3 and Layer 4 Control of Filter 0 (MAC_L3_L4_CONTROL7)

61.7.1.319.1 Offset

Register	Offset
MAC_L3_L4_CONTROL7	A50h

61.7.1.319.2 Function

The Layer 3 and Layer 4 Control register controls the operations of filter 7 of Layer 3 and Layer 4.

61.7.1.319.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			DMCHEN7	Reserved	DMCHN7			Reserved		L4DPIM7	L4DPM7	L4SPIM7	L4SPM7	Reserved	L4PEN7
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L3HDBM7					L3HSBM7					L3DAIM7	L3DAM7	L3SAIM7	L3SAM7	Reserved	L3PEN7
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.319.4 Fields

Field	Function
31-29 —	Reserved
28 DMCHEN7	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. DMA Channel Select Enable

Table continues on the next page...

Field	Function
	When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. 0b - DMA Channel Select is disabled 1b - DMA Channel Select is enabled
27 —	Reserved
26-24 DMCHN7	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration.
23-22 —	Reserved
21 L4DPIM7	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM7 bit is set high. 0b - Layer 4 Destination Port Inverse Match is disabled 1b - Layer 4 Destination Port Inverse Match is enabled
20 L4DPM7	Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching. 0b - Layer 4 Destination Port Match is disabled 1b - Layer 4 Destination Port Match is enabled
19 L4SPIM7	Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4SPM0 bit is set high. 0b - Layer 4 Source Port Inverse Match is disabled 1b - Layer 4 Source Port Inverse Match is enabled
18 L4SPM7	Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching. 0b - Layer 4 Source Port Match is disabled 1b - Layer 4 Source Port Match is enabled

Table continues on the next page...

Field	Function
17 —	Reserved
16 L4PEN7	Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching. The Layer 4 matching is done only when the L4SPM0 or L4DPM7 bit is set. 0b - Layer 4 Protocol is disabled 1b - Layer 4 Protocol is enabled
15-11 L3HDBM7	Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. Layer 3 IP DA Higher Bits Match IPv4 Packets: This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - .. - 31: All bits except MSb are masked. IPv6 Packets: Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits: - 0: No bits are masked. - 1: LSb[0] is masked. - 2: Two LSbs [1:0] are masked - .. - 127: All bits except MSb are masked. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.
10-6 L3HSBM7	Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. Layer 3 IP SA Higher Bits Match IPv4 Packets: This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field: - 0: No bits are masked. - 1: LSb[0] is masked - 2: Two LSbs [1:0] are masked - ..

Table continues on the next page...

Field	Function
	- 31: All bits except MSb are masked. IPv6 Packets: This field contains Bits[4:0] of L3HSBM0. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.
5 L3DAIM7	Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high. 0b - Layer 3 IP DA Inverse Match is disabled 1b - Layer 3 IP DA Inverse Match is enabled
4 L3DAM7	Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering. 0b - Layer 3 IP DA Match is disabled 1b - Layer 3 IP DA Match is enabled
3 L3SAIM7	Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set. 0b - Layer 3 IP SA Inverse Match is disabled 1b - Layer 3 IP SA Inverse Match is enabled
2 L3SAM7	Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering. 0b - Layer 3 IP SA Match is disabled 1b - Layer 3 IP SA Match is enabled
1 —	Reserved
0 L3PEN7	Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. Layer 3 Protocol Enable

Field	Function
	When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set. 0b - Layer 3 Protocol is disabled 1b - Layer 3 Protocol is enabled

61.7.1.320 Layer 4 Address 7 (MAC_LAYER4_ADDRESS7)

61.7.1.320.1 Offset

Register	Offset
MAC_LAYER4_ADDRESS7	A54h

61.7.1.320.2 Function

The MAC_LAYER4_ADDRESS7, MAC_L3_L4_CONTROL7, MAC_LAYER3_ADDR0_Reg7, MAC_LAYER3_ADDR1_Reg7, MAC_LAYER3_ADDR2_Reg7 and MAC_LAYER3_ADDR3_Reg7 registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the core.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the core. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

61.7.1.320.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L4DP7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L4SP7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.320.4 Fields

Field	Function
31-16 L4DP7	Layer 4 Destination Port Number Field When the L4PEN7 bit is reset and the L4DPM7 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. Layer 4 Destination Port Number Field When the L4PEN7 bit is reset and the L4DPM7 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets. When the L4PEN7 and L4DPM7 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.
15-0 L4SP7	Layer 4 Source Port Number Field When the L4PEN7 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. Layer 4 Source Port Number Field When the L4PEN7 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets. When the L4PEN7 and L4SPM0 bits are set in MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.

61.7.1.321 Layer 3 Address 0 Register 7 (MAC_LAYER3_ADDR0_REG7)

61.7.1.321.1 Offset

Register	Offset
MAC_LAYER3_ADDR0_REG7	A60h

61.7.1.321.2 Function

For IPv4 packets, the Layer 3 Address 0 Register 7 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

61.7.1.321.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A07															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.321.4 Fields

Field	Function
31-0 L3A07	Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. Layer 3 Address 0 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.

61.7.1.322 Layer 3 Address 1 Register 7 (MAC_LAYER3_ADDR1_REG7)

61.7.1.322.1 Offset

Register	Offset
MAC_LAYER3_ADDR1_REG7	A64h

61.7.1.322.2 Function

For IPv4 packets, the Layer 3 Address 1 Register 7 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

61.7.1.322.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A17															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A17															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.322.4 Fields

Field	Function
31-0 L3A17	Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. Layer 3 Address 1 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.

61.7.1.323 Layer 3 Address 2 Register 7 (MAC_LAYER3_ADDR2_REG7)

61.7.1.323.1 Offset

Register	Offset
MAC_LAYER3_ADDR2_REG7	A68h

61.7.1.323.2 Function

The Layer 3 Address 2 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

61.7.1.323.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A27															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.323.4 Fields

Field	Function
31-0 L3A27	Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. Layer 3 Address 2 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.324 Layer 3 Address 3 Register 7 (MAC_LAYER3_ADDR3_REG7)

61.7.1.324.1 Offset

Register	Offset
MAC_LAYER3_ADDR3_REG7	A6Ch

61.7.1.324.2 Function

The Layer 3 Address 3 Register 7 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

61.7.1.324.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L3A37															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	L3A37															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.324.4 Fields

Field	Function
31-0 L3A37	Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. Layer 3 Address 3 Field When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets. When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_CONTROL0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets. When the L3PEN0 bit is reset in the MAC_L3_L4_CONTROL0 register, this field is not used.

61.7.1.325 MAC_INDIR_ACCESS_CTRL (MAC_INDIR_ACCESS_CTRL)

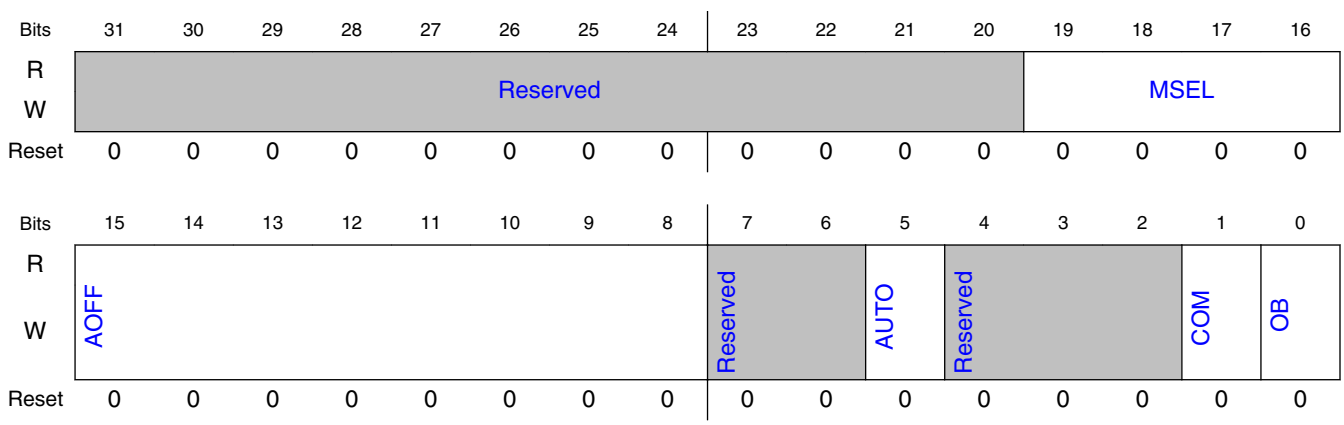
61.7.1.325.1 Offset

Register	Offset
MAC_INDIR_ACCESS_CTRL	A70h

61.7.1.325.2 Function

This register provides the Indirect Access control and status for MAC_<MSEl>_<AOFF> registers.

61.7.1.325.3 Diagram



61.7.1.325.4 Fields

Field	Function
31-20 —	Reserved
19-16 MSEL	Mode Select This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 0000: ESTTyp_RXQ_ (Type based RXQ mappping) 0001-1111: Reserved for future use.

Table continues on the next page...

Field	Function
15-8 AOFF	Address Offset This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 00000000: IndReg0(Indirect register 0) 00000001: IndReg1(Indirect register 1) ... 00000111: IndReg7(Indirect register 7) 00001000-11111111: Reserved for future use.
7-6 —	Reserved
5 AUTO	Auto increment enable 0b - AOFF is not incremented automatically. Software should program the correct Address Offset for each access. 1b - AOFF is incremented by 1. Software should ensure not to cause a wrap condition. Byte wise read/write is not supported when auto increment is enabled.
4-2 —	Reserved
1 COM	Command type. Indicates the register access type. 0b - Indicates a write operation 1b - Indicates a read operation
0 OB	Operation Busy This bit is set along with a read or write command for initiating the indirect access to MAC_<MSEL>_<AOFF> register. This bit is reset when the read or write command to MAC_<MSEL>_<AOFF> register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into MAC_<MSEL>_<AOFF> register. During a read operation, the data should be read from MAC_INDIR_ACCESS_DATA register only after this bit is reset.

61.7.1.326 MAC_INDIR_ACCESS_DATA (MAC_INDIR_ACCESS_DATA)

61.7.1.326.1 Offset

Register	Offset
MAC_INDIR_ACCESS_DATA	A74h

61.7.1.326.2 Function

This register holds the read/write data for Indirect Access of MAC_<MSEL_<AOFF> registers.

61.7.1.326.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.326.4 Fields

Field	Function
31-0 DATA	This field contains data to read/write for Indirect address access associated with MAC_INDIR_ACCESS_CTRL register.

61.7.1.327 Timestamp Control (MAC_TIMESTAMP_CONTROL)

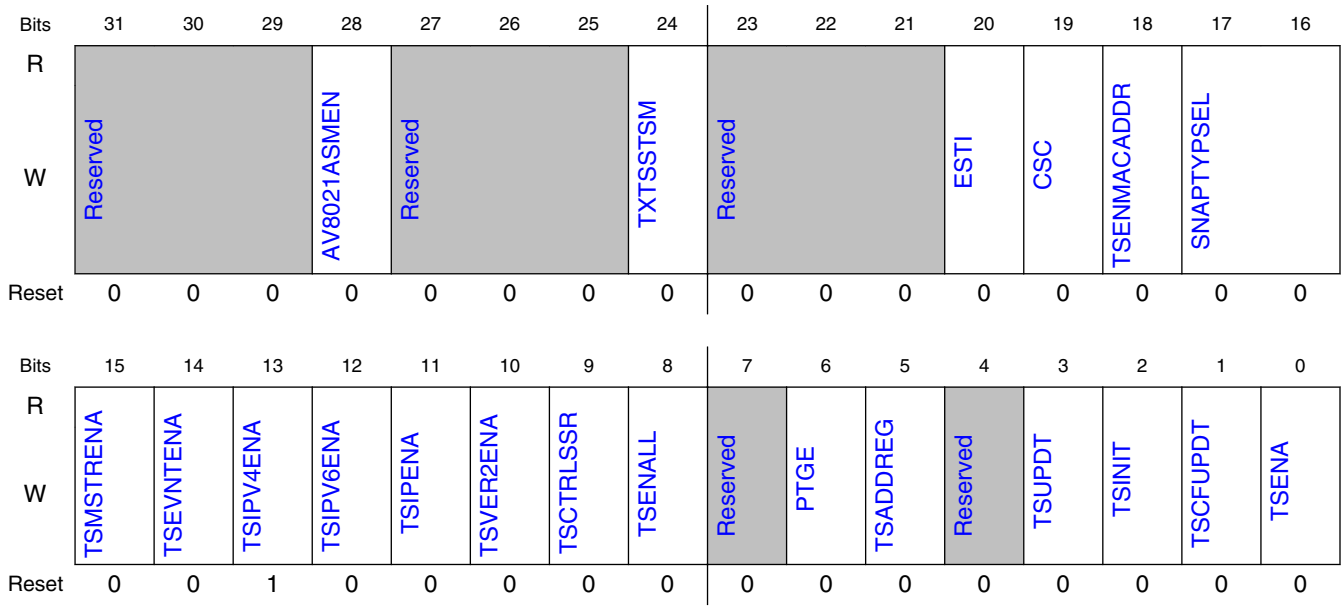
61.7.1.327.1 Offset

Register	Offset
MAC_TIMESTAMP_CONTROL	B00h

61.7.1.327.2 Function

This register controls the operation of the System Time generator and processing of PTP packets for timestamping in the Receiver.

61.7.1.327.3 Diagram



61.7.1.327.4 Fields

Field	Function
31-29 —	Reserved
28 AV8021ASMEN	AV 802. AV 802.1AS Mode Enable When this bit is set, the MAC processes only untagged PTP over Ethernet packets for providing PTP status and capturing timestamp snapshots, that is, IEEE 802.1AS mode of operation. When PTP offload feature is enabled, for the purpose of PTP offload, the transport specific field in the PTP header is generated and checked based on the value of this bit. 0b - AV 802.1AS Mode is disabled 1b - AV 802.1AS Mode is enabled
27-25 —	Reserved
24 TXTSSTSM	Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. Transmit Timestamp Status Mode When this bit is set, the MAC overwrites the earlier transmit timestamp status even if it is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. When this bit is reset, the MAC ignores the timestamp status of current packet if the timestamp status of previous packet is not read by the software. The MAC indicates this by setting the TXTSSMIS bit of the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register. 0b - Transmit Timestamp Status Mode is disabled

Table continues on the next page...

Field	Function
	1b - Transmit Timestamp Status Mode is enabled
23-21 —	Reserved
20 ESTI	External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. External System Time Input When this bit is set, the MAC uses the external 64-bit reference System Time input for the following: - To take the timestamp provided as status - To insert the timestamp in transmit PTP packets when One-step Timestamp or Timestamp Offload feature is enabled. When this bit is reset, the MAC uses the internal reference System Time. 0b - External System Time Input is disabled 1b - External System Time Input is enabled
19 CSC	Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. Enable checksum correction during OST for PTP over UDP/IPv4 packets When this bit is set, the last two bytes of PTP message sent over UDP/IPv4 is updated to keep the UDP checksum correct, for changes made to origin timestamp and/or correction field as part of one step timestamp operation. The application shall form the packet with these two dummy bytes. When reset, no updates are done to keep the UDP checksum correct. The application shall form the packet with UDP checksum set to 0. 0b - checksum correction during OST for PTP over UDP/IPv4 packets is disabled 1b - checksum correction during OST for PTP over UDP/IPv4 packets is enabled
18 TSENMACADDR	Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. Enable MAC Address for PTP Packet Filtering When this bit is set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP packets when PTP is directly sent over Ethernet. When this bit is set, received PTP packets with DA containing a special multicast or unicast address that matches the one programmed in MAC address registers are considered for processing as indicated below, when PTP is directly sent over Ethernet. For normal time stamping operation, MAC address registers 0 to 31 is considered for unicast destination address matching. For PTP offload, only MAC address register 0 is considered for unicast destination address matching. 0b - MAC Address for PTP Packet Filtering is disabled 1b - MAC Address for PTP Packet Filtering is enabled
17-16 SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. Select PTP packets for Taking Snapshots These bits, along with Bits 15 and 14, decide the set of PTP packet types for which snapshot needs to be taken. The encoding is given in Timestamp Snapshot Dependency on Register Bits Table.
15 TSMSTRENA	Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node.

Table continues on the next page...

Field	Function
	Enable Snapshot for Messages Relevant to Master When this bit is set, the snapshot is taken only for the messages that are relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node. 0b - Snapshot for Messages Relevant to Master is disabled 1b - Snapshot for Messages Relevant to Master is enabled
14 TSEVNTENA	Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). Enable Timestamp Snapshot for Event Messages When this bit is set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When this bit is reset, the snapshot is taken for all messages except Announce, Management, and Signaling. For more information about the timestamp snapshots, see Timestamp Snapshot Dependency on Register Bits Table. 0b - Timestamp Snapshot for Event Messages is disabled 1b - Timestamp Snapshot for Event Messages is enabled
13 TSIPV4ENA	Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. Enable Processing of PTP Packets Sent over IPv4-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv4-UDP packets. When this bit is reset, the MAC ignores the PTP transported over IPv4-UDP packets. This bit is set by default. 0b - Processing of PTP Packets Sent over IPv4-UDP is disabled 1b - Processing of PTP Packets Sent over IPv4-UDP is enabled
12 TSIPV6ENA	Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. Enable Processing of PTP Packets Sent over IPv6-UDP When this bit is set, the MAC receiver processes the PTP packets encapsulated in IPv6-UDP packets. When this bit is clear, the MAC ignores the PTP transported over IPv6-UDP packets. 0b - Processing of PTP Packets Sent over IPv6-UDP is disabled 1b - Processing of PTP Packets Sent over IPv6-UDP is enabled
11 TSIPENA	Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. Enable Processing of PTP over Ethernet Packets When this bit is set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet packets. When this bit is reset, the MAC ignores the PTP over Ethernet packets. 0b - Processing of PTP over Ethernet Packets is disabled 1b - Processing of PTP over Ethernet Packets is enabled
10 TSVER2ENA	Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. Enable PTP Packet Processing for Version 2 Format When this bit is set, the IEEE 1588 version 2 format is used to process the PTP packets. When this bit is reset, the IEEE 1588 version 1 format is used to process the PTP packets. The IEEE 1588 formats are described in 'PTP Processing and Control'. 0b - PTP Packet Processing for Version 2 Format is disabled 1b - PTP Packet Processing for Version 2 Format is enabled

Table continues on the next page...

Field	Function
9 TSCTRLSSR	Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. Timestamp Digital or Binary Rollover Control When this bit is set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When this bit is reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit. 0b - Timestamp Digital or Binary Rollover Control is disabled 1b - Timestamp Digital or Binary Rollover Control is enabled
8 TSENALL	Enable Timestamp for All Packets When this bit is set, the timestamp snapshot is enabled for all packets received by the MAC. 0b - Timestamp for All Packets disabled 1b - Timestamp for All Packets enabled
7 —	Reserved
6 PTGE	Presentation Time Generation Enable When this bit is set the Presentation Time generation will be enabled. 0b - Presentation Time Generation is disabled 1b - Presentation Time Generation is enabled
5 TSADDREG	Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. Update Addend Register When this bit is set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This bit is cleared when the update is complete. This bit should be zero before it is set. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Addend Register is not updated 1b - Addend Register is updated
4 —	Reserved
3 TSUPDT	Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. Update Timestamp When this bit is set, the system time is updated (added or subtracted) with the value specified in MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before updating it. This bit is reset when the update is complete in hardware. The Timestamp Higher Word register (if enabled during core configuration) is not updated. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Timestamp is not updated 1b - Timestamp is updated
2 TSINIT	Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers.

Table continues on the next page...

Field	Function
	Initialize Timestamp When this bit is set, the system time is initialized (overwritten) with the value specified in the MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update registers. This bit should be zero before it is updated. This bit is reset when the initialization is complete. The Timestamp Higher Word register (if enabled during core configuration) can only be initialized. When Media Clock Generation and Recovery is configured (DWC_EQOS_FLEXI_PPS_OUT_EN) and enabled MAC_Presn_Time_Updt should also be updated before setting this field. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Timestamp is not initialized 1b - Timestamp is initialized
1 TSCFUPDT	Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. Fine or Coarse Timestamp Update When this bit is set, the Fine method is used to update system timestamp. When this bit is reset, Coarse method is used to update the system timestamp. 0b - Coarse method is used to update system timestamp 1b - Fine method is used to update system timestamp
0 TSENA	Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. Enable Timestamp When this bit is set, the timestamp is added for Transmit and Receive packets. When disabled, timestamp is not added for transmit and receive packets and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the Receive side, the MAC processes the 1588 packets only if this bit is set. 0b - Timestamp is disabled 1b - Timestamp is enabled

61.7.1.328 Subsecond Increment (MAC_SUB_SECOND_INCREMENT)

61.7.1.328.1 Offset

Register	Offset
MAC_SUB_SECOND_INCREMENT	B04h

61.7.1.328.2 Function

This register specifies the value to be added to the internal system time register every cycle of clk_ptp_ref_i clock.

61.7.1.328.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								SSINC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SNSINC								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.328.4 Fields

Field	Function
31-24 —	Reserved
23-16 SSINC	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when the PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in TIMESTAMP_CONTROL]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) which is derived by 20 ns/0.465.
15-8 SNSINC	Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. Sub-nanosecond Increment Value This field contains the sub-nanosecond increment value, represented in nanoseconds multiplied by 2^8. This value is accumulated with the sub-nanoseconds field of the subsecond register. For example, when TSCTRLSSR field in the TIMESTAMP_CONTROL register is set. and if the required increment is 5.3ns, then SSINC should be 0x05 and SNSINC should be 0x4C.
7-0 —	Reserved

61.7.1.329 System Time Seconds (MAC_SYSTEM_TIME_SECONDS)

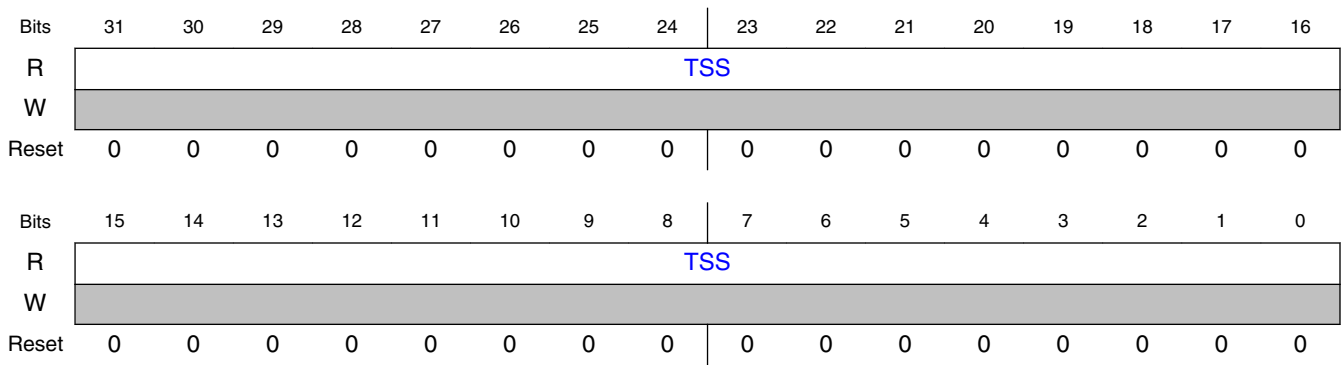
61.7.1.329.1 Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS	B08h

61.7.1.329.2 Function

The System Time Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the MAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from clk_ptp_ref_i to CSR clock).

61.7.1.329.3 Diagram



61.7.1.329.4 Fields

Field	Function
31-0 TSS	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.

61.7.1.330 System Time Nanoseconds (MAC_SYSTEM_TIME_NANOSECONDS)

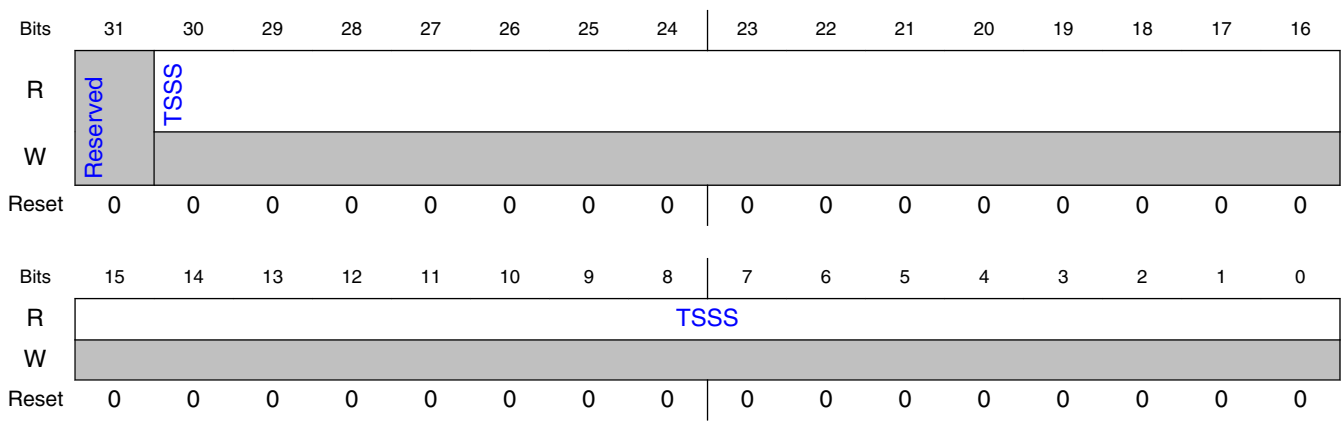
61.7.1.330.1 Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS	B0Ch

61.7.1.330.2 Function

The System Time Nanoseconds register, along with System Time Seconds register, indicates the current value of the system time maintained by the MAC.

61.7.1.330.3 Diagram



61.7.1.330.4 Fields

Field	Function
31 —	Reserved
30-0 TSSS	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0. Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When Bit 9 is set in TIMESTAMP_CONTROL, each bit represents 1 ns. The maximum value is 0x3B9A_C9FF after which it rolls-over to zero.

61.7.1.331 System Time Seconds Update (MAC_SYSTEM_TIME_SECONDS_UPDATE)

61.7.1.331.1 Offset

Register	Offset
MAC_SYSTEM_TIME_SECONDS_UPDATE	B10h

61.7.1.331.2 Function

The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the MAC. You must write both registers before setting the TSINIT or TSUPDT bits in DWC_eqos_top_map/EQOS_MAC/MAC_TIMESTAMP_CONTROL.

61.7.1.331.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.331.4 Fields

Field	Function
31-0 TSS	Timestamp Seconds The value in this field is the seconds part of the update. Timestamp Seconds The value in this field is the seconds part of the update. When ADDSUB is reset, this field must be programmed with the seconds part of the update value. When ADDSUB is set, this field must be programmed with the complement of the seconds part of the update value. For example, if 2.000000001 seconds need to be subtracted from the system time, the TSS field in the MAC_Timestamp_Seconds_Update register must be 0xFFFF_FFFE (that is, 2^32 - 2).

61.7.1.332 System Time Nanoseconds Update (MAC_SYSTEM_TIME_NANOSECONDS_UPDATE)

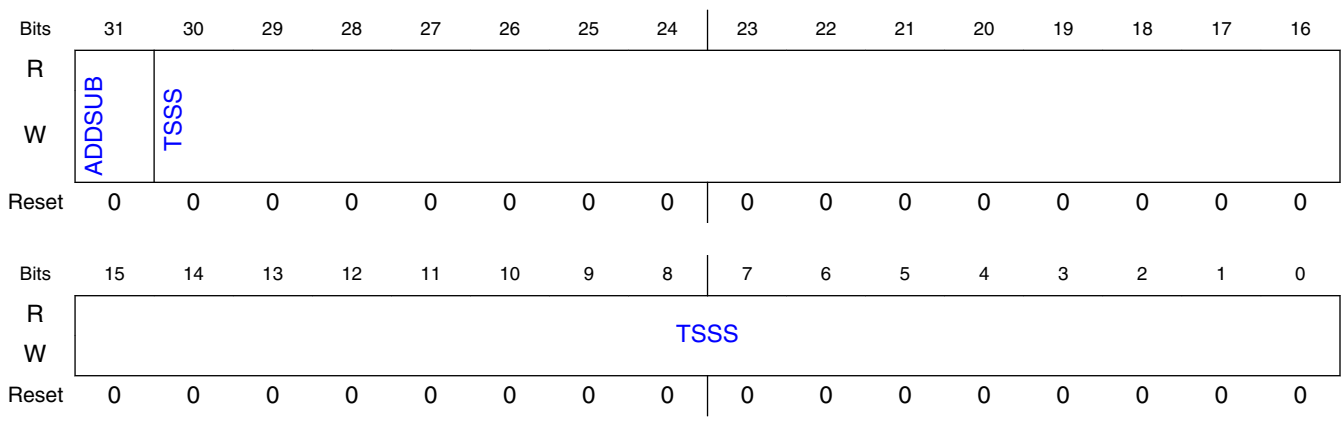
61.7.1.332.1 Offset

Register	Offset
MAC_SYSTEM_TIME_NANOSECONDS_UPDATE	B14h

61.7.1.332.2 Function

MAC System Time Nanoseconds Update register.

61.7.1.332.3 Diagram



61.7.1.332.4 Fields

Field	Function
31 ADDSUB	Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register. 0b - Add time 1b - Subtract time
30-0	Timestamp Sub Seconds The value in this field is the sub-seconds part of the update.

Field	Function
TSSS	Timestamp Sub Seconds The value in this field is the sub-seconds part of the update. When ADDSUB is reset, this field must be programmed with the sub-seconds part of the update value, with an accuracy based on the TSCTRLSSR bit of the TIMESTAMP_CONTROL register. When ADDSUB is set, this field must be programmed with the complement of the sub-seconds part of the update value as described below. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set, the programmed value must be 10^9 - <sub-second value>. When TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset, the programmed value must be 2^31 - <sub-second_value>. When the TSCTRLSSR bit is reset in the TMAC_IMESTAMP_CONTROL register, each bit represents an accuracy of 0.46 ns. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF. For example, if 2.000000001 seconds need to be subtracted from the system time, then the TSSS field in the MAC_Timestamp_Nanoseconds_Update register must be 0x7FFF_FFFF (that is, 2^31 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is reset and 0x3B9A_C9FF (that is, 10^9 - 1), when TSCTRLSSR bit in MAC_TIMESTAMP_CONTROL is set.

61.7.1.333 Timestamp Addend (MAC_TIMESTAMP_ADDEND)

61.7.1.333.1 Offset

Register	Offset
MAC_TIMESTAMP_ADDEND	B18h

61.7.1.333.2 Function

Timestamp Addend register. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in the MAC_TIMESTAMP_CONTROL register). The content of this register is added to a 32-bit accumulator in every clock cycle (of clk_ptp_ref_i) and the system time is updated whenever the accumulator overflows.

61.7.1.333.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.333.4 Fields

Field	Function
31-0 TSAR	Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization.

61.7.1.334 System Time - Higher Word Seconds (MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS)

61.7.1.334.1 Offset

Register	Offset
MAC_SYSTEM_TIME_HIGHER_WORD_SECONDS	B1Ch

61.7.1.334.2 Function

System Time - Higher Word Seconds register.

61.7.1.334.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSHWR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.334.4 Fields

Field	Function
31-16 —	Reserved
15-0 TSHWR	Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. Timestamp Higher Word Register This field contains the most-significant 16-bits of timestamp seconds value. This register is optional. You can add this register by selecting the Add IEEE 1588 Higher Word Register option. This register is directly written to initialize the value and it is incremented when there is an overflow from 32-bits of the System Time Seconds register. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears.

61.7.1.335 Timestamp Status (MAC_TIMESTAMP_STATUS)

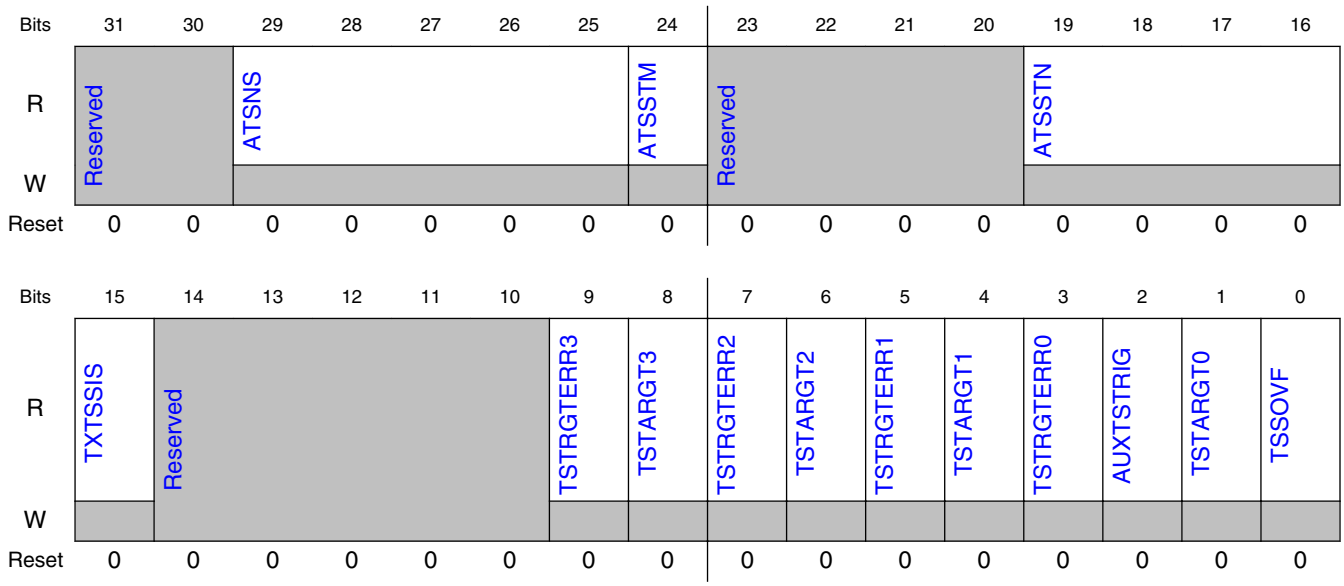
61.7.1.335.1 Offset

Register	Offset
MAC_TIMESTAMP_STATUS	B20h

61.7.1.335.2 Function

Timestamp Status register. All bits except Bits[27:25] gets cleared when the application reads this register.

61.7.1.335.3 Diagram



61.7.1.335.4 Fields

Field	Function
31-30 —	Reserved
29-25 ATSNS	Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. Number of Auxiliary Timestamp Snapshots This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected.
24 ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. 0b - Auxiliary Timestamp Snapshot Trigger Missed status not detected 1b - Auxiliary Timestamp Snapshot Trigger Missed status detected
23-20 —	Reserved
19-16 ATSSTN	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. Auxiliary Timestamp Snapshot Trigger Identifier

Table continues on the next page...

Field	Function
	These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: - Bit 16: Auxiliary trigger 0 - Bit 17: Auxiliary trigger 1 - Bit 18: Auxiliary trigger 2 - Bit 19: Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15 TXTSSIS	Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers. Tx Timestamp Status Interrupt Status In non-EQOS_CORE configurations when drop transmit status is enabled in MTL, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and TMAC_X_TIMESTAMP_STATUS_SECONDS registers. When PTP offload feature is enabled, this bit is set when the captured transmit timestamp is updated in the MAC_TX_TIMESTAMP_STATUS_NANOSECONDS and MAC_TX_TIMESTAMP_STATUS_SECONDS registers, for PTO generated Delay Request and Pdelay request packets. This bit is cleared when the MAC_TX_TIMESTAMP_STATUS_SECONDS register is read (or write to MAC_TX_TIMESTAMP_STATUS_SECONDS register when RCWE bit of CSR_SW_CTRL register is set). 0b - Tx Timestamp Status Interrupt status not detected 1b - Tx Timestamp Status Interrupt status detected
14-10 —	Reserved
9 TSTRGTERR3	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Error status not detected 1b - Timestamp Target Time Error status detected
8 TSTARGT3	Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS3 When this bit is set, it indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS registers.

Table continues on the next page...

Field	Function
	Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Reached for Target Time PPS3 status not detected 1b - Timestamp Target Time Reached for Target Time PPS3 status detected
7 TSTRGTERR2	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Error status not detected 1b - Timestamp Target Time Error status detected
6 TSTARGT2	Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Reached for Target Time PPS2 status not detected 1b - Timestamp Target Time Reached for Target Time PPS2 status detected
5 TSTRGTERR1	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Error status not detected 1b - Timestamp Target Time Error status detected
4 TSTARGT1	Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Reached for Target Time PPS1 status not detected

Table continues on the next page...

Field	Function
	1b - Timestamp Target Time Reached for Target Time PPS1 status detected
3 TSTRGTERR0	Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers elapses. Timestamp Target Time Error This bit is set when the latest target time programmed in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_TARGET_TIME_NANOSECONDS registers elapses. This bit is cleared when the application reads this bit. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in MAC_CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Error status not detected 1b - Timestamp Target Time Error status detected
2 AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Auxiliary Timestamp Trigger Snapshot status not detected 1b - Auxiliary Timestamp Trigger Snapshot status detected
1 TSTARGT0	Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds registers. Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS registers. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Target Time Reached status not detected 1b - Timestamp Target Time Reached status detected
0 TSSOVF	Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Timestamp Seconds Overflow When this bit is set, it indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF. Access restriction applies. Clears on read (or this bit is written to 1 when RCWE bit in CSR_SW_CTRL register is set). Self-set to 1 on internal event. 0b - Timestamp Seconds Overflow status not detected 1b - Timestamp Seconds Overflow status detected

61.7.1.336 Transmit Timestamp Status Nanoseconds (MAC_TX_TIMESTAMP_STATUS_NANOSECONDS)

61.7.1.336.1 Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_NANOSECONDS	B30h

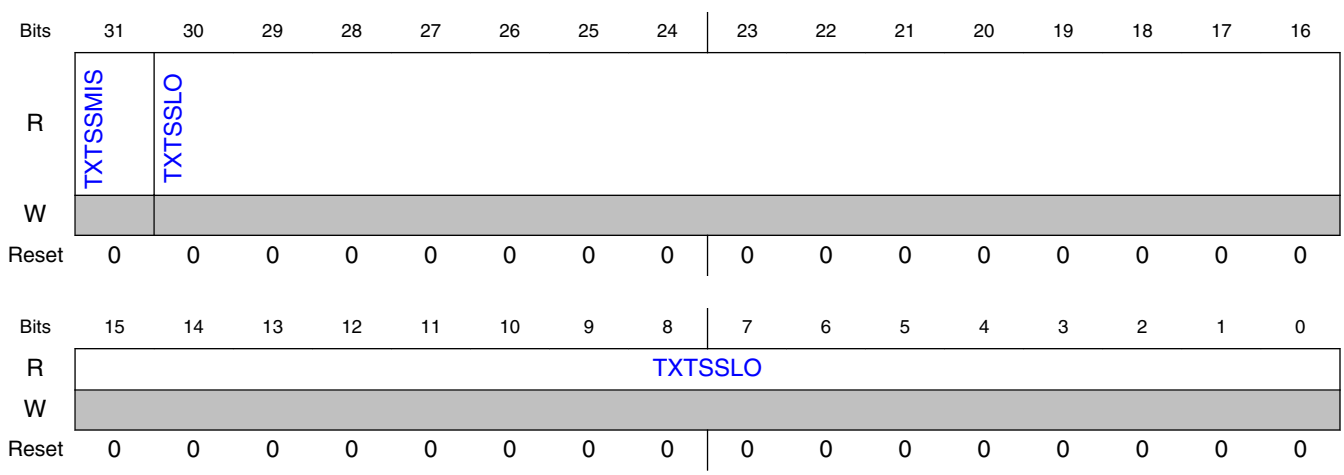
61.7.1.336.2 Function

This register contains the nanosecond part of timestamp captured for Transmit packets when Tx status is disabled.

The MAC_TX_TIMESTAMP_STATUS_NANOSECONDS register, along with MAC_TX_TIMESTAMP_STATUS_SECONDS, gives the 64-bit timestamp captured for the PTP packet successfully transmitted by the MAC. This value is considered to be read by the application when the last byte of MAC_TX_TIMESTAMP_STATUS_NANOSECONDS is read. In the little-endian mode, this means when bits[31:24] are read; in big-endian mode, bits[7:0] are read.

If the application does not read these registers and timestamp of another packet is captured, then either the current timestamp is lost (overwritten) or the new timestamp is lost (dropped), depending on the setting of the TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register. The status bit TXTSC bit [15] in MAC_TIMESTAMP_STATUS register is set whenever the MAC transmitter captures the timestamp.

61.7.1.336.3 Diagram



61.7.1.336.4 Fields

Field	Function
31 TXTSSMIS	Transmit Timestamp Status Missed When this bit is set, it indicates one of the following: - The timestamp of the current packet is ignored if TXTSSTSM bit of the TIMESTAMP_CONTROL register is reset - The timestamp of the previous packet is overwritten with timestamp of the current packet if TXTSSTSM bit of the MAC_TIMESTAMP_CONTROL register is set. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Transmit Timestamp Status Missed status not detected 1b - Transmit Timestamp Status Missed status detected
30-0 TXTSSLO	Transmit Timestamp Status Low This field contains the 31 bits of the Nanoseconds field of the Transmit packet's captured timestamp.

61.7.1.337 Transmit Timestamp Status Seconds (MAC_TX_TIMESTAMP_STATUS_SECONDS)

61.7.1.337.1 Offset

Register	Offset
MAC_TX_TIMESTAMP_STATUS_SECONDS	B34h

61.7.1.337.2 Function

The register contains the higher 32 bits of the timestamp (in seconds) captured when a PTP packet is transmitted.

61.7.1.337.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXTSSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXTSSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.337.4 Fields

Field	Function
31-0 TXTSSHI	Transmit Timestamp Status High This field contains the lower 32 bits of the Seconds field of Transmit packet's captured timestamp.

61.7.1.338 Auxiliary Timestamp Control (MAC_AUXILIARY_CONTROL)

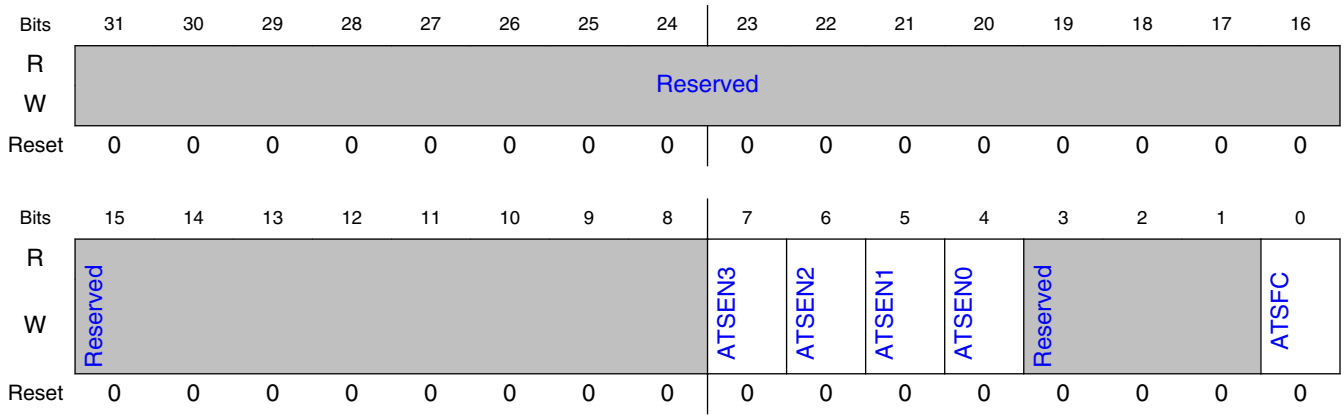
61.7.1.338.1 Offset

Register	Offset
MAC_AUXILIARY_CONTROL	B40h

61.7.1.338.2 Function

The Auxiliary Timestamp Control register controls the Auxiliary Timestamp snapshot.

61.7.1.338.3 Diagram



61.7.1.338.4 Fields

Field	Function
31-8 —	Reserved
7 ATSEN3	Auxiliary Snapshot 3 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 3. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[3] input is enabled. When this bit is reset, the events on this input are ignored. 0b - Auxiliary Snapshot $i is disabled 1b - Auxiliary Snapshot $i is enabled
6 ATSEN2	Auxiliary Snapshot 2 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 2. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[2] input is enabled. When this bit is reset, the events on this input are ignored. 0b - Auxiliary Snapshot $i is disabled 1b - Auxiliary Snapshot $i is enabled
5 ATSEN1	Auxiliary Snapshot 1 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 1. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[1] input is enabled. When this bit is reset, the events on this input are ignored. 0b - Auxiliary Snapshot $i is disabled 1b - Auxiliary Snapshot $i is enabled
4 ATSEN0	Auxiliary Snapshot 0 Enable This bit controls the capturing of Auxiliary Snapshot Trigger 0. When this bit is set, the auxiliary snapshot of the event on ptp_aux_trig_i[0] input is enabled. When this bit is reset, the events on this input are ignored. 0b - Auxiliary Snapshot $i is disabled 1b - Auxiliary Snapshot $i is enabled
3-1 —	Reserved

Table continues on the next page...

Field	Function
0 ATSFC	Auxiliary Snapshot FIFO Clear When set, this bit resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, the auxiliary snapshots are stored in the FIFO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Auxiliary Snapshot FIFO Clear is disabled 1b - Auxiliary Snapshot FIFO Clear is enabled

61.7.1.339 Auxiliary Timestamp Nanoseconds (MAC_AUXILIARY_TIMESTAMP_NANOSECONDS)

61.7.1.339.1 Offset

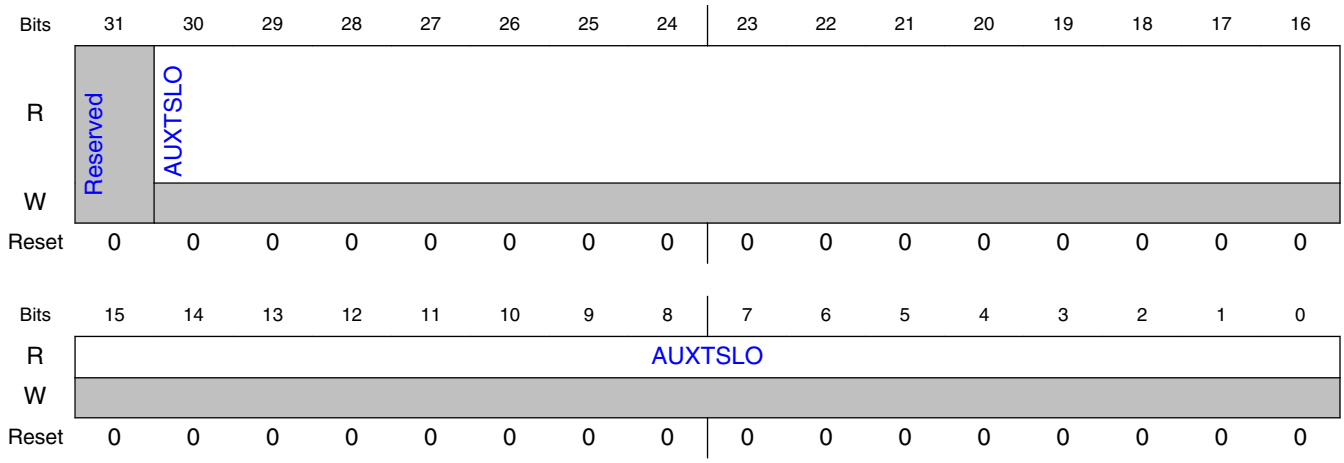
Register	Offset
MAC_AUXILIARY_TIMESTAMP_NANOSECONDS	B48h

61.7.1.339.2 Function

The Auxiliary Timestamp Nanoseconds register, along with MAC_Auxiliary_Timestamp_Seconds, gives the 64-bit timestamp stored as auxiliary snapshot. These two registers form the read port of a 64-bit wide FIFO with a depth of 4, 8, or 16 as selected while configuring the core.

You can store multiple snapshots in this FIFO. Bits[29:25] in MAC_TIMESTAMP_STATUS indicate the fill-level of the FIFO. The top of the FIFO is removed only when the last byte of MAC_AUXILIARY_TIMESTAMP_SECONDS register is read. In the little-endian mode, this means when Bits[31:24] are read and in big-endian mode, Bits[7:0] are read.

61.7.1.339.3 Diagram



61.7.1.339.4 Fields

Field	Function
31 —	Reserved
30-0 AUXTSLO	Auxiliary Timestamp Contains the lower 31 bits (nanoseconds field) of the auxiliary timestamp.

61.7.1.340 Auxiliary Timestamp Seconds (MAC_AUXILIARY_TIMESTAMP_SECONDS)

61.7.1.340.1 Offset

Register	Offset
MAC_AUXILIARY_TIMESTAMP_SECONDS	B4Ch

61.7.1.340.2 Function

The Auxiliary Timestamp - Seconds register contains the lower 32 bits of the Seconds field of the auxiliary timestamp register.

61.7.1.340.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUXTSHI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.340.4 Fields

Field	Function
31-0 AUXTSHI	Auxiliary Timestamp Contains the lower 32 bits of the Seconds field of the auxiliary timestamp.

61.7.1.341 Timestamp Ingress Asymmetry Correction (MAC_TIMESTAMP_INGRESS_ASYM_CORR)

61.7.1.341.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_ASYM_CORR	B50h

61.7.1.341.2 Function

The MAC Timestamp Ingress Asymmetry Correction register contains the Ingress Asymmetry Correction value to be used while updating correction field in PDelay_Resp PTP messages.

61.7.1.341.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSTIAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSTIAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.341.4 Fields

Field	Function
31-0 OSTIAC	One-Step Timestamp Ingress Asymmetry Correction This field contains the ingress path asymmetry value to be added to correctionField of Pdelay_Resp PTP packet. The programmed value should be in units of nanoseconds and multiplied by 2^16. For example, 2.5 ns is represented as 0x00028000. The value can also be negative, which is represented in 2's complement form with bit 31 representing the sign bit.

61.7.1.342 Timestamp Egress Asymmetry Correction (MAC_TIMESTAMP_EGRESS_ASYM_CORR)

61.7.1.342.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_ASYM_CORR	B54h

61.7.1.342.2 Function

The MAC Timestamp Egress Asymmetry Correction register contains the Egress Asymmetry Correction value to be used while updating the correction field in PDelay_Req PTP messages.

61.7.1.342.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OSTEAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSTEAC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.342.4 Fields

Field	Function
31-0 OSTEAC	One-Step Timestamp Egress Asymmetry Correction This field contains the egress path asymmetry value to be subtracted from correctionField of Pdelay_Resp PTP packet. The programmed value must be the negated value in units of nanoseconds multiplied by 2^16. For example, if the required correction is +2.5 ns, the programmed value must be 0xFFFD_8000, which is the 2's complement of 0x0002_8000(2.5 * 216). Similarly, if the required correction is -3.3 ns, the programmed value is 0x0003_4CCC (3.3 * 216).

61.7.1.343 Timestamp Ingress Correction Nanosecond (MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND)

61.7.1.343.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_NANOSECOND	B58h

61.7.1.343.2 Function

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the ingress path.

61.7.1.343.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSIC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.343.4 Fields

Field	Function
31-0 TSIC	Timestamp Ingress Correction This field contains the ingress path correction value as defined by the Ingress Correction expression.

61.7.1.344 Timestamp Egress Correction Nanosecond (MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND)

61.7.1.344.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_NANOSECOND	B5Ch

61.7.1.344.2 Function

This register contains the correction value in nanoseconds to be used with the captured timestamp value in the egress path.

61.7.1.344.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.344.4 Fields

Field	Function
31-0 TSEC	Timestamp Egress Correction This field contains the nanoseconds part of the egress path correction value as defined by the Egress Correction expression.

61.7.1.345 Timestamp Ingress Correction Subnanosecond (MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC)

61.7.1.345.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_CORR_SUBNANOSEC	B60h

61.7.1.345.2 Function

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for ingress direction.

61.7.1.345.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSICSNS								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.345.4 Fields

Field	Function
31-16 —	Reserved
15-8 TSICSNS	Timestamp Ingress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the ingress path correction value as defined by the "Ingress Correction" expression.
7-0 —	Reserved

61.7.1.346 Timestamp Egress Correction Subnanosecond (MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC)

61.7.1.346.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_CORR_SUBNANOSEC	B64h

61.7.1.346.2 Function

This register contains the sub-nanosecond part of the correction value to be used with the captured timestamp value, for egress direction.

61.7.1.346.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSECSNS								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.346.4 Fields

Field	Function
31-16 —	Reserved
15-8 TSECSNS	Timestamp Egress Correction, sub-nanoseconds This field contains the sub-nanoseconds part of the egress path correction value as defined by the "Egress Correction" expression.
7-0 —	Reserved

61.7.1.347 Timestamp Ingress Latency (MAC_TIMESTAMP_INGRESS_LATENCY)

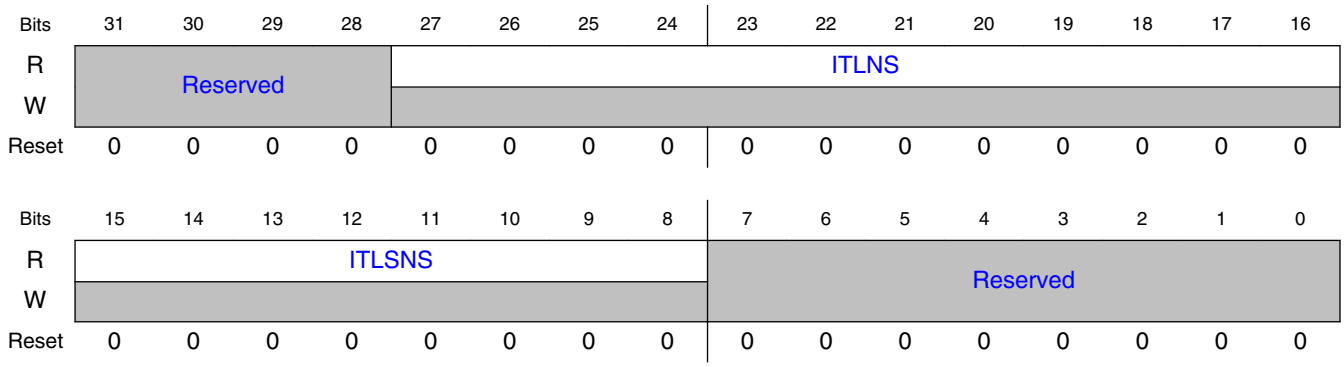
61.7.1.347.1 Offset

Register	Offset
MAC_TIMESTAMP_INGRESS_LATENCY	B68h

61.7.1.347.2 Function

This register holds the Ingress MAC latency.

61.7.1.347.3 Diagram



61.7.1.347.4 Fields

Field	Function
31-28 —	Reserved
27-16 ITLNS	Ingress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII/MII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
15-8 ITLSNS	Ingress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the input ports (phy_rxd_i) of MAC and the actual point (GMII/MII) where the ingress timestamp is taken. Ingress correction value is computed as described in the section 7.1.2.4.1 of QoS Databook.
7-0 —	Reserved

61.7.1.348 Timestamp Egress Latency (MAC_TIMESTAMP_EGRESS_LATENCY)

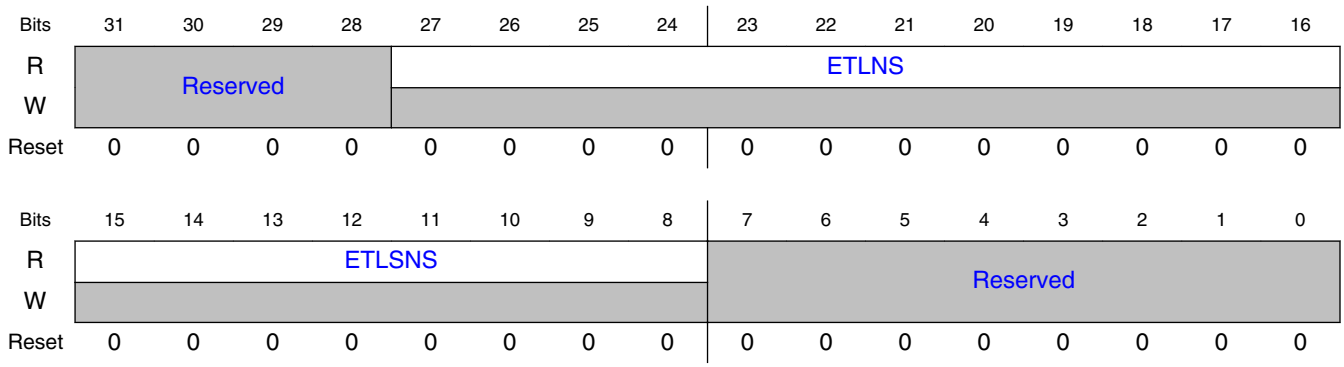
61.7.1.348.1 Offset

Register	Offset
MAC_TIMESTAMP_EGRESS_LATENCY	B6Ch

61.7.1.348.2 Function

This register holds the Egress MAC latency.

61.7.1.348.3 Diagram



61.7.1.348.4 Fields

Field	Function
31-28 —	Reserved
27-16 ETLNS	Egress Timestamp Latency, in nanoseconds This register holds the average latency in nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
15-8 ETLSNS	Egress Timestamp Latency, in sub-nanoseconds This register holds the average latency in sub-nanoseconds between the actual point (GMII/MII) where the egress timestamp is taken and the output ports (phy_txd_o) of the MAC. Ingress correction value is computed as described in the section 7.1.2.4.2 of QoS Databook.
7-0 —	Reserved

61.7.1.349 PPS Control (MAC_PPS_CONTROL)

61.7.1.349.1 Offset

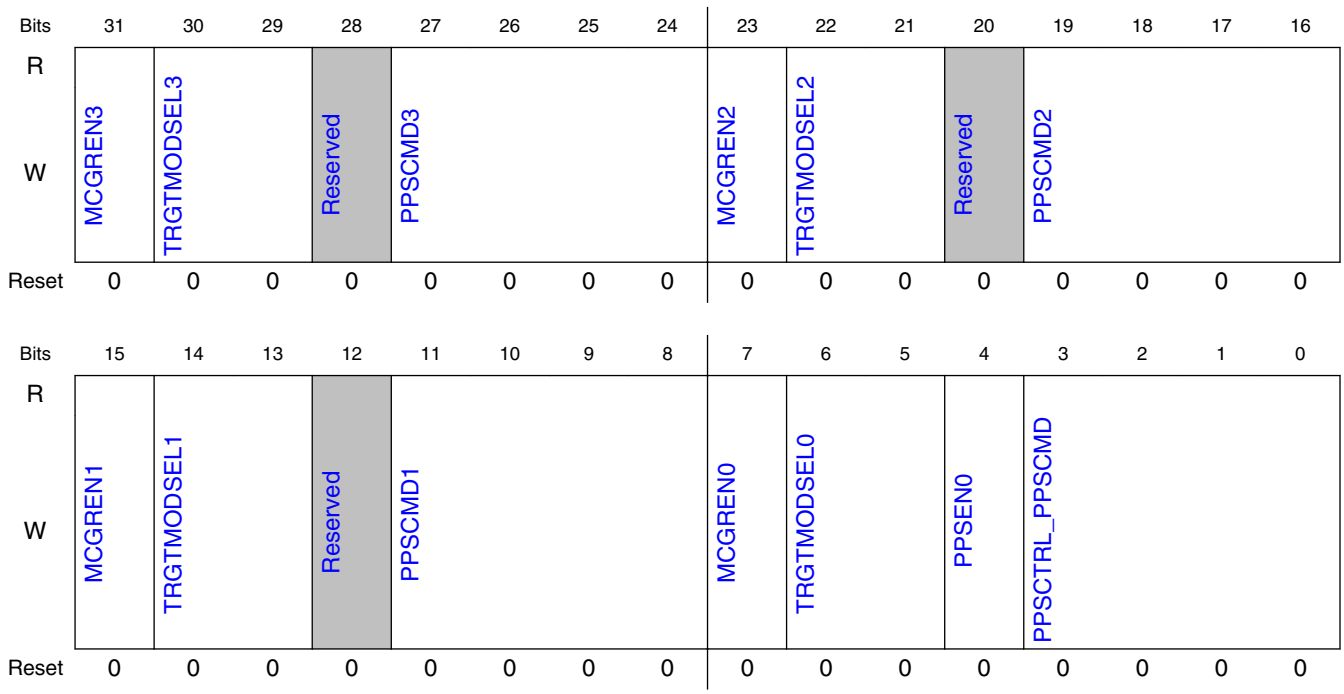
Register	Offset
MAC_PPS_CONTROL	B70h

61.7.1.349.2 Function

PPS Control register.

Bits[30:24] of this register are valid only when four Flexible PPS outputs are selected. Bits[22:16] are valid only when three or more Flexible PPS outputs are selected. Bits[14:8] are valid only when two or more Flexible PPS outputs are selected. Bits[6:4] are valid only when Flexible PPS feature is selected.

61.7.1.349.3 Diagram



61.7.1.349.4 Fields

Field	Function
31 MCGREN3	MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS3 Output This field enables the 3rd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode.
30-29 TRGTMODSEL3	Target Time Register Mode for PPS3 Output This field indicates the Target Time registers (MAC_PPS3_TARGET_TIME_SECONDS and MAC_PPS3_TARGET_TIME_NANOSECONDS) mode for PPS3 output signal. 00b - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01b - Reserved 10b - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11b - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted

Table continues on the next page...

Field	Function
28 —	Reserved
27-24 PPSCMD3	Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. Flexible PPS3 Output Control This field controls the flexible PPS3 output (ptp_pps_o[3]) signal. This field is similar to the PPSCMD0[2:0] field. If MCGREN3 is set, then PPSCMD3 indicated by these 4 bits [27:24] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 3. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN3 is not set then only 3 bits from [26:24] is used as PPSCMD3 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
23 MCGREN2	MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS2 Output This field enables the 2nd PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0b - 2nd PPS instance is disabled to operate in PPS or MCGR mode 1b - 2nd PPS instance is enabled to operate in PPS or MCGR mode
22-21 TRGTMODSEL2	Target Time Register Mode for PPS2 Output This field indicates the Target Time registers (MAC_PPS2_TARGET_TIME_SECONDS and MAC_PPS2_TARGET_TIME_NANOSECONDS) mode for PPS2 output signal. 00b - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01b - Reserved 10b - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11b - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
20 —	Reserved
19-16 PPSCMD2	Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. Flexible PPS2 Output Control This field controls the flexible PPS2 output (ptp_pps_o[2]) signal. This field is similar to the PPSCMD0 field. If MCGREN2 is set, then PPSCMD2 indicated by these 4 bits [19:16] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 2. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN2 is not set then only 3 bits from [18:16] is used as PPSCMD2 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
15 MCGREN1	MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS1 Output This field enables the 1st PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0b - 1st PPS instance is disabled to operate in PPS or MCGR mode 1b - 1st PPS instance is enabled to operate in PPS or MCGR mode

Table continues on the next page...

Field	Function
14-13 TRGTMODSEL1	Target Time Register Mode for PPS1 Output This field indicates the Target Time registers (MAC_PPS1_TARGET_TIME_SECONDS and MAC_PPS1_TARGET_TIME_NANOSECONDS) mode for PPS1 output signal. 00b - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01b - Reserved 10b - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11b - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
12 —	Reserved
11-8 PPSCMD1	Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. Flexible PPS1 Output Control This field controls the flexible PPS1 output (ptp_pps_o[1]) signal. This field is similar to the PPSCMD0 field. If MCGREN1 is set, then PPSCMD1 indicated by these 4 bits [11:8] are taken as Presentation Time Control bits for media clock generation and recovery for comparator instance 1. This field is similar to the PPSCMD0 Presentation Time Control bits. If MCGREN1 is not set then only 3 bits from [10:8] is used as PPSCMD1 and the 4th bit is to be set as 0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.
7 MCGREN0	MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. MCGR Mode Enable for PPS0 Output This field enables the 0th PPS instance to operate in PPS or MCGR mode. When set it operates in MCGR mode and on reset it operates in PPS mode. 0b - 0th PPS instance is enabled to operate in PPS mode 1b - 0th PPS instance is enabled to operate in MCGR mode
6-5 TRGTMODSEL0	Target Time Register Mode for PPS0 Output This field indicates the Target Time registers (MAC_PPS0_TARGET_TIME_SECONDS and MAC_PPS0_TARGET_TIME_NANOSECONDS) mode for PPS0 output signal: 00b - Target Time registers are programmed only for generating the interrupt event. The Flexible PPS function must not be enabled in this mode, otherwise spurious transitions may be observed on the corresponding ptp_pps_o output port 01b - Reserved 10b - Target Time registers are programmed for generating the interrupt event and starting or stopping the PPS0 output signal generation 11b - Target Time registers are programmed only for starting or stopping the PPS0 output signal generation. No interrupt is asserted
4 PPSEN0	Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. Flexible PPS Output Mode Enable When this bit is set, Bits[3:0] function as PPSCMD. When this bit is reset, Bits[3:0] function as PPSCTRL (Fixed PPS mode). 0b - Flexible PPS Output Mode is disabled 1b - Flexible PPS Output Mode is enabled
3-0 PPSCTRL_PPSCMD	PPS Output Frequency Control This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. PPS Output Frequency Control

Field	Function
	This field controls the frequency of the PPS0 output (ptp_pps_o[0]) signal. The default value of PPSCTRL is 0000, and the PPS output is 1 pulse (of width clk_ptp_i) every second. For other values of PPSCTRL, the PPS output becomes a generated clock of following frequencies: - 0001: The binary rollover is 2 Hz, and the digital rollover is 1 Hz. - 0010: The binary rollover is 4 Hz, and the digital rollover is 2 Hz. - 0011: The binary rollover is 8 Hz, and the digital rollover is 4 Hz. - 0100: The binary rollover is 16 Hz, and the digital rollover is 8 Hz. - .. - 1111: The binary rollover is 32.768 KHz and the digital rollover is 16.384 KHz. Note: In the binary rollover mode, the PPS output (ptp_pps_o) has a duty cycle of 50 percent with these frequencies. In the digital rollover mode, the PPS output frequency is an average number. The actual clock is of different frequency that gets synchronized every second. For example: - When PPSCTRL = 0001, the PPS (1 Hz) has a low period of 537 ms and a high period of 463 ms - When PPSCTRL = 0010, the PPS (2 Hz) is a sequence of One clock of 50 percent duty cycle and 537 ms period Second clock of 463 ms period (268 ms low and 195 ms high) - When PPSCTRL = 0011, the PPS (4 Hz) is a sequence of Three clocks of 50 percent duty cycle and 268 ms period Fourth clock of 195 ms period (134 ms low and 61 ms high) This behavior is because of the non-linear toggling of bits in the digital rollover mode in the MAC_System_Time_Nanoseconds register. or Flexible PPS Output (ptp_pps_o[0]) Control Programming these bits with a non-zero value instructs the MAC to initiate an event. When the command is transferred or synchronized to the PTP clock domain, these bits get cleared automatically. The software should ensure that these bits are programmed only when they are 'all-zero'. The following list describes the values of PPSCMD0: - 0000: No Command - 0001: START Single Pulse This command generates single pulse rising at the start point defined in MAC_PPS0_Target_Time_Seconds and MAC_PPS0_Target_Time_Nanoseconds register and of a duration defined in the PPS0 Width Register. - 0010: START Pulse Train This command generates the train of pulses rising at the start point defined in the Target Time Registers and of a duration defined in the PPS0 Width Register and repeated at interval defined in the PPS Interval Register. By default, the PPS pulse train is free-running unless stopped by the 'Stop Pulse train at time' or 'Stop Pulse Train immediately' commands. - 0011: Cancel START This command cancels the START Single Pulse and START Pulse Train commands if the system time has not crossed the programmed start time. - 0100: STOP Pulse train at time

Field	Function
	This command stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010) after the time programmed in the Target Time registers elapses. - 0101: STOP Pulse Train immediately This command immediately stops the train of pulses initiated by the START Pulse Train command (PPSCMD = 0010). - 110: Cancel STOP Pulse train This command cancels the STOP pulse train at time command if the programmed stop time has not elapsed. The PPS pulse train becomes free-running on the successful execution of this command. - 0111-1111: Reserved or Presentation Time Control If MCGREN0 is set then these bits are treated as Presentation time control bits. The following list describes the values of PPSCMD0: - 0000: No Operation related to MCGR will be carried out. If set to this value in the mid of clock recovery or generation all the processing inputs will be flushed. - 0001: Capture the Presentation time at rising edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0010: Capture the Presentation time at falling edge of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0011: Capture the Presentation time at both edges of mcg_pst_trig_i[0] into the MAC_PPS0_Target_Time_Seconds register - 0100-1000: Reserved - 1001: Toggle output on compare - 1010: Pulse output low on compare for one PTP-clock cycle - 1011: Pulse output high on compare for one PTP-clock cycle - 1100-1111: Reserved

61.7.1.350 PPS0 Target Time Seconds (MAC_PPS0_TARGET_TIME_SECONDS)

61.7.1.350.1 Offset

Register	Offset
MAC_PPS0_TARGET_TIME_SECONDS	B80h

61.7.1.350.2 Function

The PPS0 Target Time Seconds register, along with PPS0 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of MAC_TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

61.7.1.350.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.350.4 Fields

Field	Function
31-0 TSTRH0	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the MAC_PPS_CONTROL register.

61.7.1.351 PPS0 Target Time Nanoseconds (MAC_PPS0_TARGET_TIME_NANOSECONDS)

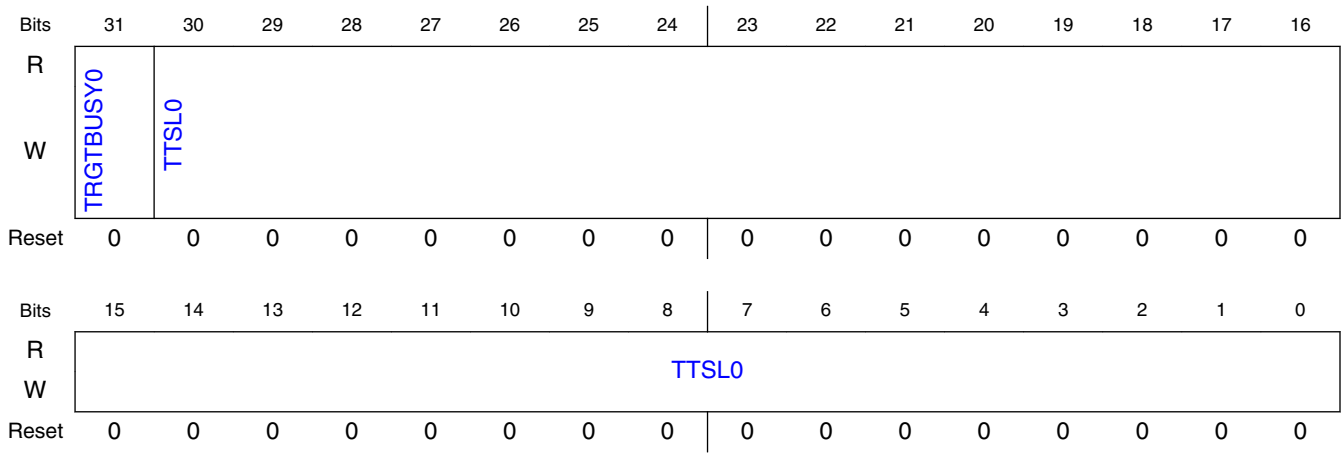
61.7.1.351.1 Offset

Register	Offset
MAC_PPS0_TARGET_TIME_NANOSECONDS	B84h

61.7.1.351.2 Function

PPS0 Target Time Nanoseconds register.

61.7.1.351.3 Diagram



61.7.1.351.4 Fields

Field	Function
31 TRGTBUSY0	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the MAC_PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0b - PPS Target Time Register Busy status is not detected 1b - PPS Target Time Register Busy is detected
30-0 TTSL0	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the MAC_TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the MAC_TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

61.7.1.352 PPS0 Interval (MAC_PPS0_INTERVAL)

61.7.1.352.1 Offset

Register	Offset
MAC_PPS0_INTERVAL	B88h

61.7.1.352.2 Function

The PPS0 Interval register contains the number of units of sub-second increment value between the rising edges of PPS0 signal output (ptp_pps_o[0]).

61.7.1.352.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.352.4 Fields

Field	Function
31-0 PPSINT0	PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS0 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS0 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

61.7.1.353 PPS0 Width (MAC_PPS0_WIDTH)

61.7.1.353.1 Offset

Register	Offset
MAC_PPS0_WIDTH	B8Ch

61.7.1.353.2 Function

The PPS0 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS0 signal output (ptp_pps_o[0]).

61.7.1.353.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.353.4 Fields

Field	Function
31-0 PPSWIDTH0	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

61.7.1.354 PPS1 Target Time Seconds (MAC_PPS1_TARGET_TIME_SECONDS)

61.7.1.354.1 Offset

Register	Offset
MAC_PPS1_TARGET_TIME_SECONDS	B90h

61.7.1.354.2 Function

The PPS1 Target Time Seconds register, along with PPS1 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

61.7.1.354.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.354.4 Fields

Field	Function
31-0 TSTRH1	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register. If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

61.7.1.355 PPS1 Target Time Nanoseconds (MAC_PPS1_TARGET_TIME_NANOSECONDS)

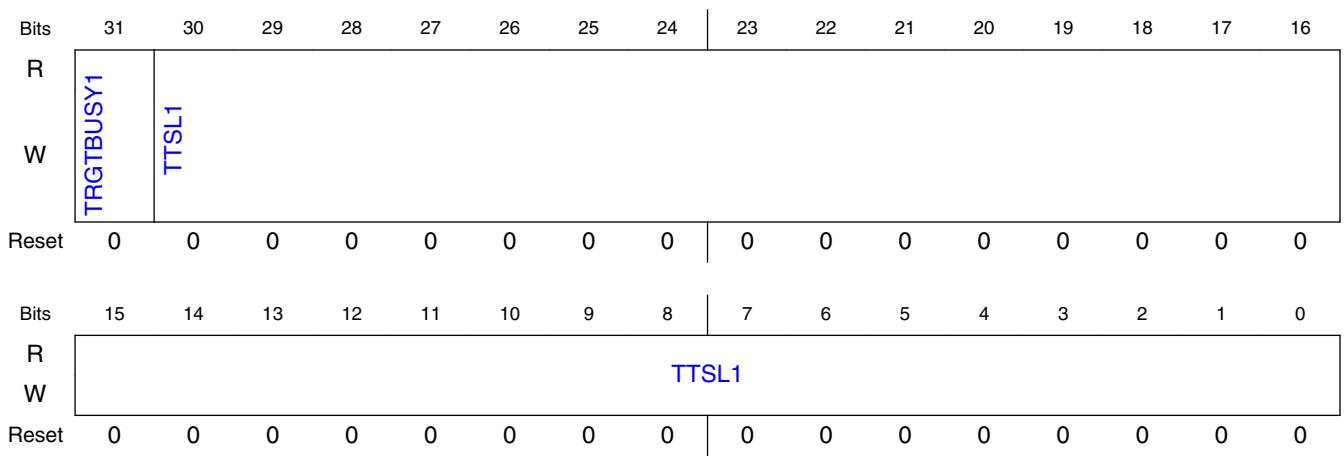
61.7.1.355.1 Offset

Register	Offset
MAC_PPS1_TARGET_TIME_NANOSECONDS	B94h

61.7.1.355.2 Function

PPS1 Target Time Nanoseconds register.

61.7.1.355.3 Diagram



61.7.1.355.4 Fields

Field	Function
31 TRGTBUSY1	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted.

Table continues on the next page...

Field	Function
	0b - PPS Target Time Register Busy status is not detected 1b - PPS Target Time Register Busy is detected
30-0 TTSL1	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

61.7.1.356 PPS1 Interval (MAC_PPS1_INTERVAL)

61.7.1.356.1 Offset

Register	Offset
MAC_PPS1_INTERVAL	B98h

61.7.1.356.2 Function

The PPS1 Interval register contains the number of units of sub-second increment value between the rising edges of PPS1 signal output (ptp_pps_o[1]).

61.7.1.356.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.356.4 Fields

Field	Function
31-0 PPSINT1	PPS Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS1 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS1 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

61.7.1.357 PPS1 Width (MAC_PPS1_WIDTH)

61.7.1.357.1 Offset

Register	Offset
MAC_PPS1_WIDTH	B9Ch

61.7.1.357.2 Function

The PPS1 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS1 signal output (ptp_pps_o[1]).

61.7.1.357.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.357.4 Fields

Field	Function
31-0 PPSWIDTH1	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

61.7.1.358 PPS2 Target Time Seconds (MAC_PPS2_TARGET_TIME_SECONDS)

61.7.1.358.1 Offset

Register	Offset
MAC_PPS2_TARGET_TIME_SECONDS	BA0h

61.7.1.358.2 Function

The PPS2 Target Time Seconds register, along with PPS2 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

61.7.1.358.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.358.4 Fields

Field	Function
31-0 TSTRH2	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register. If DWC_EQOS_FLEXI_PPS_OUT_EN is enabled in the configuration and PTGE field of TIMESTAMP_CONTROL Register is set with Presentation time control set in recovery mode, then these bits indicate the TPT being programmed by the application and in generation mode it indicates the CPT generated at the sampled trigger.

61.7.1.359 PPS2 Target Time Nanoseconds (MAC_PPS2_TARGET_TIME_NANOSECONDS)

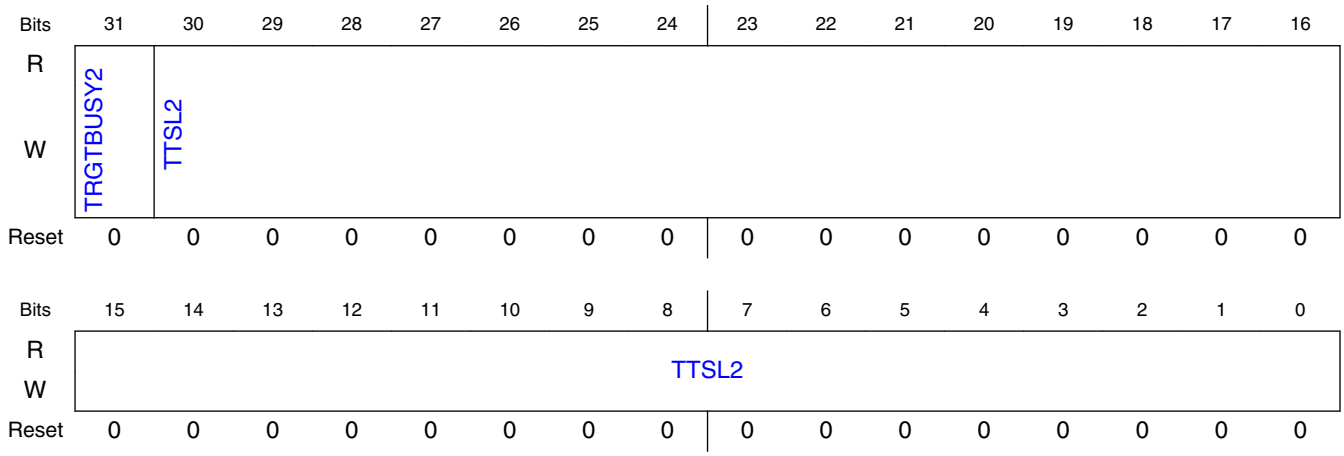
61.7.1.359.1 Offset

Register	Offset
MAC_PPS2_TARGET_TIME_NANOSECONDS	BA4h

61.7.1.359.2 Function

PPS2 Target Time Nanoseconds register.

61.7.1.359.3 Diagram



61.7.1.359.4 Fields

Field	Function
31 TRGTBUSY2	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0b - PPS Target Time Register Busy status is not detected 1b - PPS Target Time Register Busy is detected
30-0 TTSL2	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

61.7.1.360 PPS2 Interval (MAC_PPS2_INTERVAL)

61.7.1.360.1 Offset

Register	Offset
MAC_PPS2_INTERVAL	BA8h

61.7.1.360.2 Function

The PPS2 Interval register contains the number of units of sub-second increment value between the rising edges of PPS2 signal output (ptp_pps_o[2]).

61.7.1.360.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.360.4 Fields

Field	Function
31-0 PPSINT2	PPS Output Signal Interval These bits store the interval between the rising edges of PPS2 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS2 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS2 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

61.7.1.361 PPS2 Width (MAC_PPS2_WIDTH)

61.7.1.361.1 Offset

Register	Offset
MAC_PPS2_WIDTH	BACh

61.7.1.361.2 Function

The PPS2 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS2 signal output (ptp_pps_o[2]).

61.7.1.361.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.361.4 Fields

Field	Function
31-0 PPSWIDTH2	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

61.7.1.362 PPS3 Target Time Seconds (MAC_PPS3_TARGET_TIME_SECONDS)

61.7.1.362.1 Offset

Register	Offset
MAC_PPS3_TARGET_TIME_SECONDS	BB0h

61.7.1.362.2 Function

The PPS3 Target Time Seconds register, along with PPS3 Target Time Nanoseconds register, is used to schedule an interrupt event [Bit 1 of TIMESTAMP_STATUS] when the system time exceeds the value programmed in these registers.

61.7.1.362.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTRH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTRH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.362.4 Fields

Field	Function
31-0 TSTRH3	PPS Target Time Seconds Register This field stores the time in seconds. PPS Target Time Seconds Register This field stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on Target Time mode selected for the corresponding PPS output in the PPS_CONTROL register.

61.7.1.363 PPS3 Target Time Nanoseconds (MAC_PPS3_TARGET_TIME_NANOSECONDS)

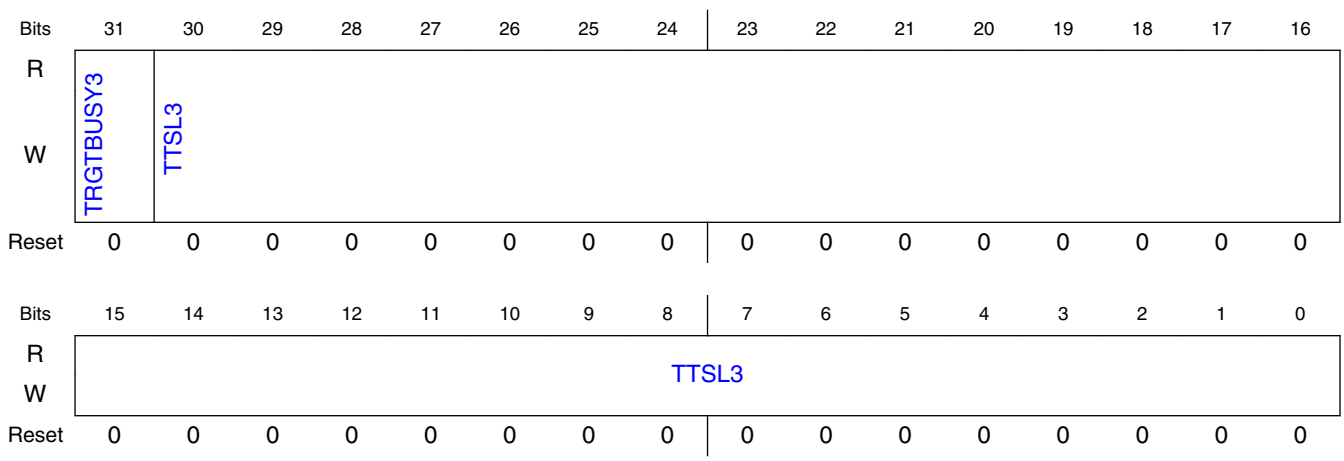
61.7.1.363.1 Offset

Register	Offset
MAC_PPS3_TARGET_TIME_NANOSECONDS	BB4h

61.7.1.363.2 Function

PPS3 Target Time Nanoseconds register.

61.7.1.363.3 Diagram



61.7.1.363.4 Fields

Field	Function
31 TRGTBUSY3	PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. PPS Target Time Register Busy The MAC sets this bit when the PPSCMD0 field in the PPS_CONTROL register is programmed to 010 or 011. Programming the PPSCMD0 field to 010 or 011 instructs the MAC to synchronize the Target Time Registers to the PTP clock domain. The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. 0b - PPS Target Time Register Busy status is not detected 1b - PPS Target Time Register Busy is detected

Table continues on the next page...

Field	Function
30-0 TTSL3	Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. Target Time Low for PPS Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the value in both Target Timestamp registers, the MAC starts or stops the PPS signal output and generates an interrupt (if enabled) based on the TRGTMODSEL0 field (Bits [6:5]) in PPS_CONTROL. When the TSCTRLSSR bit is reset in the TIMESTAMP_CONTROL register, this value should be (time in ns / 0.465). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. When the TSCTRLSSR bit is set in the TIMESTAMP_CONTROL register, this value should not exceed 0x3B9A_C9FF. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.

61.7.1.364 PPS3 Interval (MAC_PPS3_INTERVAL)

61.7.1.364.1 Offset

Register	Offset
MAC_PPS3_INTERVAL	BB8h

61.7.1.364.2 Function

The PPS3 Interval register contains the number of units of sub-second increment value between the rising edges of PPS3 signal output (ptp_pps_o[3]).

61.7.1.364.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSINT3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSINT3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.364.4 Fields

Field	Function
31-0 PPSINT3	PPS Output Signal Interval These bits store the interval between the rising edges of PPS3 signal output. PPS Output Signal Interval These bits store the interval between the rising edges of PPS3 signal output. The interval is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if the PTP reference clock is 50 MHz (period of 20 ns), and desired interval between the rising edges of PPS3 signal output is 100 ns (that is, 5 units of sub-second increment value), you should program value 4 (5-1) in this register.

61.7.1.365 PPS3 Width (MAC_PPS3_WIDTH)

61.7.1.365.1 Offset

Register	Offset
MAC_PPS3_WIDTH	BBCh

61.7.1.365.2 Function

The PPS3 Width register contains the number of units of sub-second increment value between the rising and corresponding falling edges of PPS3 signal output (ptp_pps_o[3]).

61.7.1.365.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPSWIDTH3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.365.4 Fields

Field	Function
31-0 PPSWIDTH3	PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. PPS Output Signal Width These bits store the width between the rising edge and corresponding falling edge of PPS0 signal output. The width is stored in terms of number of units of sub-second increment value. You need to program one value less than the required interval. For example, if PTP reference clock is 50 MHz (period of 20 ns), and width between the rising and corresponding falling edges of PPS0 signal output is 80 ns (that is, four units of sub-second increment value), you should program value 3 (4-1) in this register. Note: The value programmed in this register must be lesser than the value programmed in MAC_PPS0_Interval.

61.7.1.366 PTP Offload Engine Control (MAC_PTO_CONTROL)

61.7.1.366.1 Offset

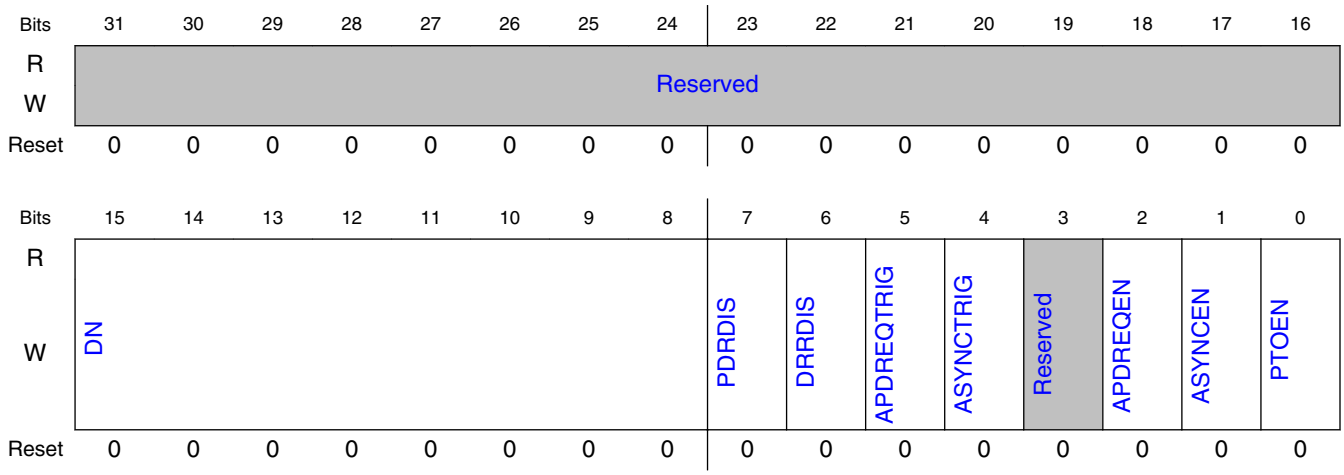
Register	Offset
MAC_PTO_CONTROL	BC0h

61.7.1.366.2 Function

This register controls the PTP Offload Engine operation.

This register controls the PTP Offload Engine operation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

61.7.1.366.3 Diagram



61.7.1.366.4 Fields

Field	Function
31-16 —	Reserved
15-8 DN	Domain Number This field indicates the domain Number in which the PTP node is operating. Domain Number This field indicates the domain Number in which the PTP node is operating.
7 PDRDIS	Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Disable Peer Delay Response response generation When this bit is set, the Peer Delay Response (Pdelay_Resp) response is not be generated for received Peer Delay Request (Pdelay_Req) request packet, as required by the programmed mode. Note: Setting this bit to 1 affects the normal PTP Offload operation and the time synchronization. So, this bit must be set only if there is problem with Pdelay_Resp generation in Hardware and/or Pdelay_Resp generation is handled by Software. 0b - Peer Delay Response response generation is enabled 1b - Peer Delay Response response generation is disabled
6 DRRDIS	Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. Disable PTO Delay Request/Response response generation When this bit is set, the Delay Request and Delay response is not generated for received SYNC and Delay request packet respectively, as required by the programmed mode. 0b - PTO Delay Request/Response response generation is enabled 1b - PTO Delay Request/Response response generation is disabled
5 APDREQTRIG	Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted.

Table continues on the next page...

Field	Function
	Automatic PTP Pdelay_Req message Trigger When this bit is set, one PTP Pdelay_Req message is transmitted. This bit is automatically cleared after the PTP Pdelay_Req message is transmitted. The application should set the APDREQEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Automatic PTP Pdelay_Req message Trigger is disabled 1b - Automatic PTP Pdelay_Req message Trigger is enabled
4 ASYNCTRIG	Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. Automatic PTP SYNC message Trigger When this bit is set, one PTP SYNC message is transmitted. This bit is automatically cleared after the PTP SYNC message is transmitted. The application should set the ASYNCEN bit for this operation. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Automatic PTP SYNC message Trigger is disabled 1b - Automatic PTP SYNC message Trigger is enabled
3 —	Reserved
2 APDREQEN	Automatic PTP Pdelay_Req message Enable When this bit is set, PTP Pdelay_Req message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Peer-to-Peer Transparent mode. 0b - Automatic PTP Pdelay_Req message is disabled 1b - Automatic PTP Pdelay_Req message is enabled
1 ASYNCEN	Automatic PTP SYNC message Enable When this bit is set, PTP SYNC message is generated periodically based on interval programmed or trigger from application, when the MAC is programmed to be in Clock Master mode. 0b - Automatic PTP SYNC message is disabled 1b - Automatic PTP SYNC message is enabled
0 PTOEN	PTP Offload Enable When this bit is set, the PTP Offload feature is enabled. 0b - PTP Offload feature is disabled 1b - PTP Offload feature is enabled

61.7.1.367 Source Port Identity 0 (MAC_SOURCE_PORT_IDENTITY0)

61.7.1.367.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY0	BC4h

61.7.1.367.2 Function

This register contains Bits[31:0] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

61.7.1.367.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.367.4 Fields

Field	Function
31-0 SPI0	Source Port Identity 0 This field indicates bits [31:0] of sourcePortIdentity of PTP node.

61.7.1.368 Source Port Identity 1 (MAC_SOURCE_PORT_IDENTITY1)

61.7.1.368.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY1	BC8h

61.7.1.368.2 Function

This register contains Bits[63:32] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

61.7.1.368.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.368.4 Fields

Field	Function
31-0 SPI1	Source Port Identity 1 This field indicates bits [63:32] of sourcePortIdentity of PTP node.

61.7.1.369 Source Port Identity 2 (MAC_SOURCE_PORT_IDENTITY2)

61.7.1.369.1 Offset

Register	Offset
MAC_SOURCE_PORT_IDENTITY2	BCCh

61.7.1.369.2 Function

This register contains Bits[79:64] of the 80-bit Source Port Identity of the PTP node. This register is available only when the Enable PTP Timestamp Offload feature is selected.

61.7.1.369.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPI2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.369.4 Fields

Field	Function
31-16 —	Reserved
15-0 SPI2	Source Port Identity 2 This field indicates bits [79:64] of sourcePortIdentity of PTP node.

61.7.1.370 Log Message Interval (MAC_LOG_MESSAGE_INTERVAL)

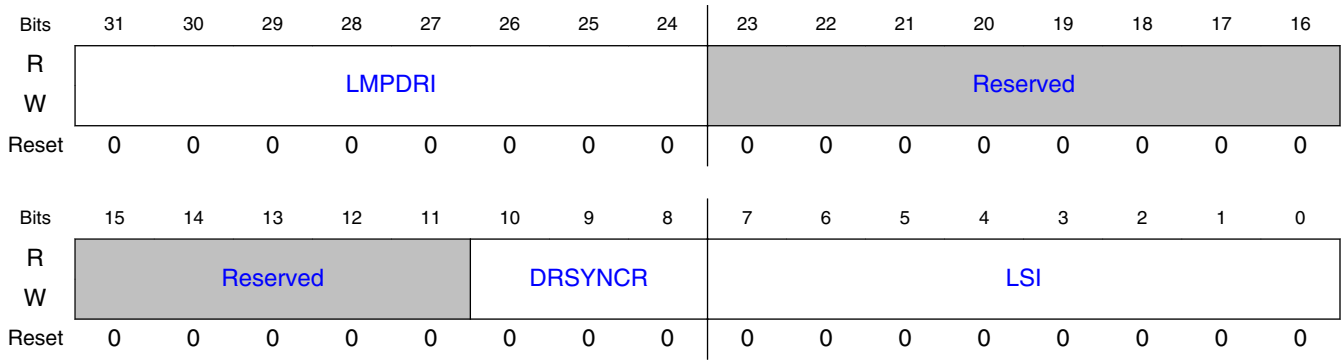
61.7.1.370.1 Offset

Register	Offset
MAC_LOG_MESSAGE_INTERVAL	BD0h

61.7.1.370.2 Function

This register contains the periodic intervals for automatic PTP packet generation. This register is available only when the Enable PTP Timestamp Offload feature is selected.

61.7.1.370.3 Diagram



61.7.1.370.4 Fields

Field	Function
31-24 LMPDRI	Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. Log Min Pdelay_Req Interval This field indicates logMinPdelayReqInterval of PTP node. This is used to schedule the periodic Pdelay request packet transmission. Allowed values are -15 to 15.Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.
23-11 —	Reserved
10-8 DRSYNCR	Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. Delay_Req to SYNC Ratio In Slave mode, it is used for controlling frequency of Delay_Req messages transmitted. - 0: DelayReq generated for every received SYNC - 1: DelayReq generated every alternate reception of SYNC - 2: for every 4 SYNC messages - 3: for every 8 SYNC messages - 4: for every 16 SYNC messages - 5: for every 32 SYNC messages - 6-7: Reserved The master sends this information (logMinDelayReqInterval) in the DelayResp PTP messages to the slave. The DWC_ether_qos Receiver processes this value from the received DelayResp messages and updates this field accordingly. In the Slave mode, the host must not write/update this register unless it has to override the received value. In Master mode, the sum of this field and logSyncInterval (LSI) field is provided in the logMinDelayReqInterval field of the generated multicast Delay_Resp PTP message. Access restriction applies. Updated based on the event. Setting 1 sets. Setting 0 clears. 000b - DelayReq generated for every received SYNC 001b - DelayReq generated every alternate reception of SYNC 010b - for every 4 SYNC messages 011b - for every 8 SYNC messages

Table continues on the next page...

Field	Function
	100b - for every 16 SYNC messages 101b - for every 32 SYNC messages 110b - Reserved
7-0 LSI	Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Log Sync Interval This field indicates the periodicity of the automatically generated SYNC message when the PTP node is Master. Allowed values are -15 to 15. Negative value must be represented in 2's-complement form. For example, if the required value is -1, the value programmed must be 0xFF.

61.7.1.371 MTL Operation Mode (MTL_OPERATION_MODE)

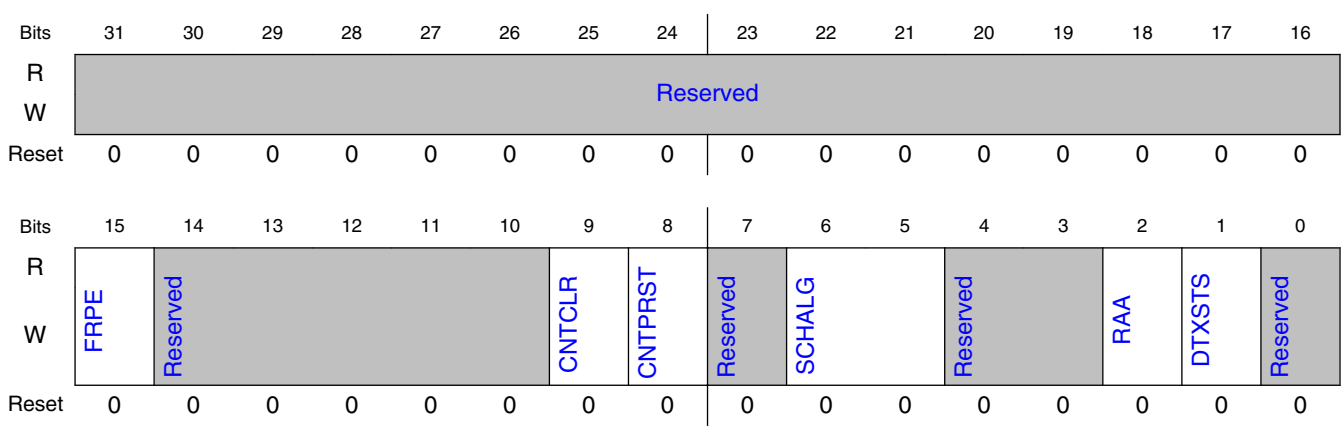
61.7.1.371.1 Offset

Register	Offset
MTL_OPERATION_MODE	C00h

61.7.1.371.2 Function

The Operation Mode register establishes the Transmit and Receive operating modes and commands.

61.7.1.371.3 Diagram



61.7.1.371.4 Fields

Field	Function
31-16 —	Reserved
15 FRPE	Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. Flexible Rx parser Enable When this bit is set to 1, the Programmable Rx Parser functionality is enabled. When the Rx parser is disabled and if the Rx parser is in the middle of the parsing then it gets disabled only after completing the current packet parsing. When the Rx parser is enabled from disabled state then the Rx parser gets activated for the next immediate packet. 0b - Flexible Rx parser is disabled 1b - Flexible Rx parser is enabled
14-10 —	Reserved
9 CNTCLR	Counters Reset When this bit is set, all counters are reset. Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Counters are not reset 1b - All counters are reset
8 CNTPRST	Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0. Counters Preset When this bit is set, - MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0. - Missed Packet and Overflow Packet counters in MTL_RxQ[0-7]_Missed_Packet_Overflow_Cnt register is initialized/preset to 12'h7F0. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Counters Preset is disabled 1b - Counters Preset is enabled
7 —	Reserved
6-5 SCHALG	Tx Scheduling Algorithm This field indicates the algorithm for Tx scheduling: 00b - WRR algorithm 01b - WFQ algorithm when DCB feature is selected.Otherwise, Reserved 10b - DWRR algorithm when DCB feature is selected.Otherwise, Reserved 11b - Strict priority algorithm
4-3 —	Reserved
2 RAA	Receive Arbitration Algorithm This field is used to select the arbitration algorithm for the Rx side. Receive Arbitration Algorithm

Table continues on the next page...

Field	Function
	This field is used to select the arbitration algorithm for the Rx side. - 0: Strict priority (SP) Queue 0 has the lowest priority and the last queue has the highest priority. - 1: Weighted Strict Priority (WSP) 0b - Strict priority (SP) 1b - Weighted Strict Priority (WSP)
1 DTXSTS	Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. When this bit is reset, the Tx packet status received from the MAC is forwarded to the application. 0b - Drop Transmit Status is disabled 1b - Drop Transmit Status is enabled
0 —	Reserved

61.7.1.372 FIFO Debug Access Control and Status (MTL_DBG_CTL)

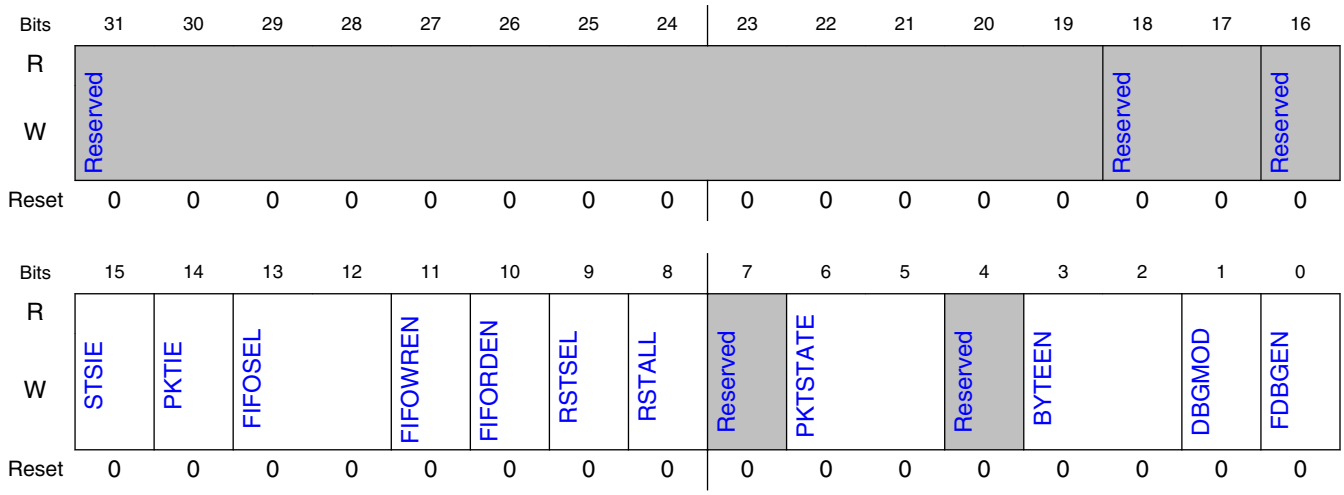
61.7.1.372.1 Offset

Register	Offset
MTL_DBG_CTL	C08h

61.7.1.372.2 Function

The FIFO Debug Access Control and Status register controls the operation mode of FIFO debug access.

61.7.1.372.3 Diagram



61.7.1.372.4 Fields

Field	Function
31-19 —	Reserved
18-17 —	Reserved
16 —	Reserved
15 STSIE	Transmit Status Available Interrupt Status Enable When this bit is set, an interrupt is generated when Transmit status is available in slave mode. 0b - Transmit Packet Available Interrupt Status is disabled 1b - Transmit Packet Available Interrupt Status is enabled
14 PKTIE	Receive Packet Available Interrupt Status Enable When this bit is set, an interrupt is generated when EOP of received packet is written to the Rx FIFO. 0b - Receive Packet Available Interrupt Status is disabled 1b - Receive Packet Available Interrupt Status is enabled
13-12 FIFOSEL	FIFO Selected for Access This field indicates the FIFO selected for debug access: 00b - Tx FIFO 01b - Tx Status FIFO (only read access when SLVMOD is set) 10b - TSO FIFO (cannot be accessed when SLVMOD is set) 11b - Rx FIFO
11 FIFOWREN	FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets.

Table continues on the next page...

Field	Function
	0b - FIFO Write is disabled 1b - FIFO Write is enabled
10 FIFORDEN	FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0b - FIFO Read is disabled 1b - FIFO Read is enabled
9 RSTSEL	Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0b - Reset Pointers of Selected FIFO is disabled 1b - Reset Pointers of Selected FIFO is enabled
8 RSTALL	Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. 0b - Reset All Pointers is disabled 1b - Reset All Pointers is enabled
7 —	Reserved
6-5 PKTSTATE	Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO: - 00: Packet Data - 01: Normal Status - 10: Last Status

Table continues on the next page...

Field	Function
	- 11: EOP 00b - Packet Data 01b - Control Word/Normal Status 10b - SOP Data/Last Status 11b - EOP Data/EOP
4 —	Reserved
3-2 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00b - Byte 0 valid 01b - Byte 0 and Byte 1 are valid 10b - Byte 0, Byte 1, and Byte 2 are valid 11b - All four bytes are valid
1 DBGMOD	Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. In this mode, the following access types are allowed: - Read and Write access to Tx FIFO, TSO FIFO, and Rx FIFO - Read access is allowed to Tx Status FIFO. When this bit is reset, it indicates that the current access to the FIFO is slave access bypassing the DMA. In this mode, the following access are allowed: - Write access to the Tx FIFO - Read access to the Rx FIFO and Tx Status FIFO 0b - Debug Mode Access to FIFO is disabled 1b - Debug Mode Access to FIFO is enabled
0 FDBGEN	FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. When this bit is reset, it indicates that the FIFO can be accessed only through a master interface. 0b - FIFO Debug Access is disabled 1b - FIFO Debug Access is enabled

61.7.1.373 FIFO Debug Status (MTL_DBG_STS)

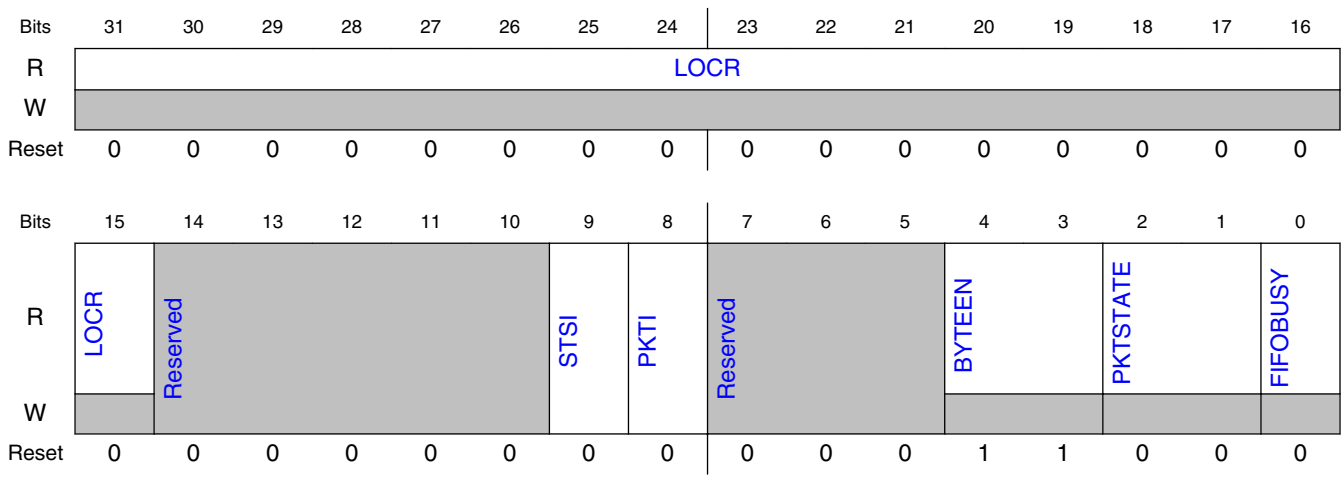
61.7.1.373.1 Offset

Register	Offset
MTL_DBG_STS	C0Ch

61.7.1.373.2 Function

The FIFO Debug Status register contains the status of FIFO debug access.

61.7.1.373.3 Diagram



61.7.1.373.4 Fields

Field	Function
31-15 LOCR	Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Debug Access Mode: This field contains the Write or Read pointer value of the selected FIFO during Write or Read operation, respectively. Reset: In single Tx Queue configurations, (DWC_EQOS_TXFIFO_SIZE/(DWC_EQOS_DATAWIDTH/8)), Otherwise 0000H
14-10 —	Reserved
9 STSI	Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO.

Table continues on the next page...

Field	Function
	Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. This bit is reset when 1 is written to this bit. 0b - Transmit Status Available Interrupt Status not detected 1b - Transmit Status Available Interrupt Status detected
8 PKTI	Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. This bit is reset when 1 is written to this bit. 0b - Receive Packet Available Interrupt Status not detected 1b - Receive Packet Available Interrupt Status detected
7-5 —	Reserved
4-3 BYTEEN	Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. 00b - Byte 0 valid 01b - Byte 0 and Byte 1 are valid 10b - Byte 0, Byte 1, and Byte 2 are valid 11b - All four bytes are valid
2-1 PKTSTATE	Encoded Packet State This field is used to get the control or status information of the selected FIFO. Encoded Packet State This field is used to get the control or status information of the selected FIFO. Tx FIFO: - 00: Packet Data - 01: Control Word - 10: SOP Data - 11: EOP Data Rx FIFO: - 00: Packet Data - 01: Normal Status - 10: Last Status - 11: EOP This field is applicable only for Tx FIFO and Rx FIFO during Read operation. 00b - Packet Data 01b - Control Word/Normal Status 10b - SOP Data/Last Status 11b - EOP Data/EOP
0 FIFOBUSY	FIFO Busy When set, this bit indicates that a FIFO operation is in progress in the MAC and content of the following fields is not valid: - All other fields of this register - All fields of the MTL_FIFO_DEBUG_DATA register

Field	Function
	0b - FIFO Busy not detected 1b - FIFO Busy detected

61.7.1.374 FIFO Debug Data (MTL_FIFO_DEBUG_DATA)

61.7.1.374.1 Offset

Register	Offset
MTL_FIFO_DEBUG_DATA	C10h

61.7.1.374.2 Function

The FIFO Debug Data register contains the data to be written to or read from the FIFOs.

61.7.1.374.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FDBGDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FDBGDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.374.4 Fields

Field	Function
31-0 FDBGDATA	FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. During debug or slave access read operation, this field contains the data read from the Tx FIFO, Rx FIFO, TSO FIFO, or Tx Status FIFO.

61.7.1.375 MTL Interrupt Status (MTL_INTERRUPT_STATUS)

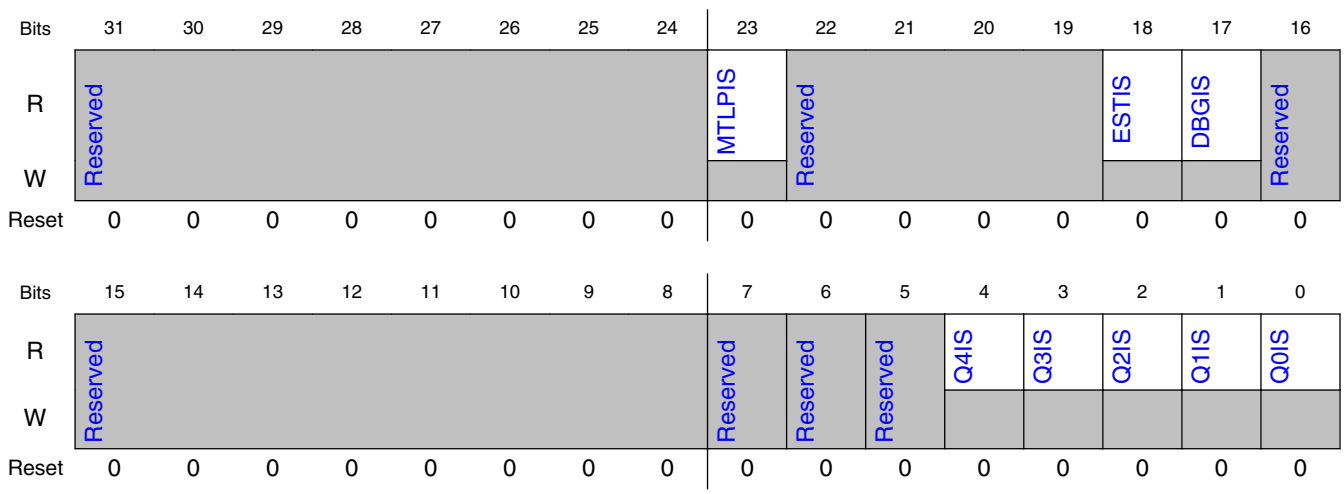
61.7.1.375.1 Offset

Register	Offset
MTL_INTERRUPT_STATUS	C20h

61.7.1.375.2 Function

The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

61.7.1.375.3 Diagram



61.7.1.375.4 Fields

Field	Function
31-24 —	Reserved
23 MTLPIS	MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. MTL Rx Parser Interrupt Status This bit indicates that there is an interrupt from Rx Parser Block. To reset this bit, the application must read the MTL_RXP_INTERRUPT_STATUS register to get the exact cause of the interrupt and clear its source.

Table continues on the next page...

Field	Function
	0b - MTL Rx Parser Interrupt status not detected 1b - MTL Rx Parser Interrupt status detected
22-19 —	Reserved
18 ESTIS	EST (TAS- 802. EST (TAS- 802.1Qbv) Interrupt Status This bit indicates an interrupt event during the operation of 802.1Qbv. To reset this bit, the application must clear the error/event that has caused the Interrupt. 0b - EST (TAS- 802.1Qbv) Interrupt status not detected 1b - EST (TAS- 802.1Qbv) Interrupt status detected
17 DBGIS	Debug Interrupt status This bit indicates an interrupt event during the slave access. Debug Interrupt status This bit indicates an interrupt event during the slave access. To reset this bit, the application must read the FIFO Debug Access Status register to get the exact cause of the interrupt and clear its source. 0b - Debug Interrupt status not detected 1b - Debug Interrupt status detected
16 —	Reserved
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 Q4IS	Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. Queue 4 Interrupt status This bit indicates that there is an interrupt from Queue 4. To reset this bit, the application must read the MTL_Q4_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0b - Queue 4 Interrupt status not detected 1b - Queue 4 Interrupt status detected
3 Q3IS	Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. Queue 3 Interrupt status This bit indicates that there is an interrupt from Queue 3. To reset this bit, the application must read the MTL_Q3_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0b - Queue 3 Interrupt status not detected 1b - Queue 3 Interrupt status detected
2 Q2IS	Queue 2 Interrupt status This bit indicates that there is an interrupt from Queue 2. Queue 2 Interrupt status

Table continues on the next page...

Field	Function
	This bit indicates that there is an interrupt from Queue 2. To reset this bit, the application must read the MTL_Q2_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0b - Queue 2 Interrupt status not detected 1b - Queue 2 Interrupt status detected
1 Q1IS	Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. Queue 1 Interrupt status This bit indicates that there is an interrupt from Queue 1. To reset this bit, the application must read the MTL_Q1_INTERRUPT_CONTROL_STATUS register to get the exact cause of the interrupt and clear its source. 0b - Queue 1 Interrupt status not detected 1b - Queue 1 Interrupt status detected
0 Q0IS	Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. 0b - Queue 0 Interrupt status not detected 1b - Queue 0 Interrupt status detected

61.7.1.376 Receive Queue and DMA Channel Mapping 0 (MTL_RXQ_DMA_MAP0)

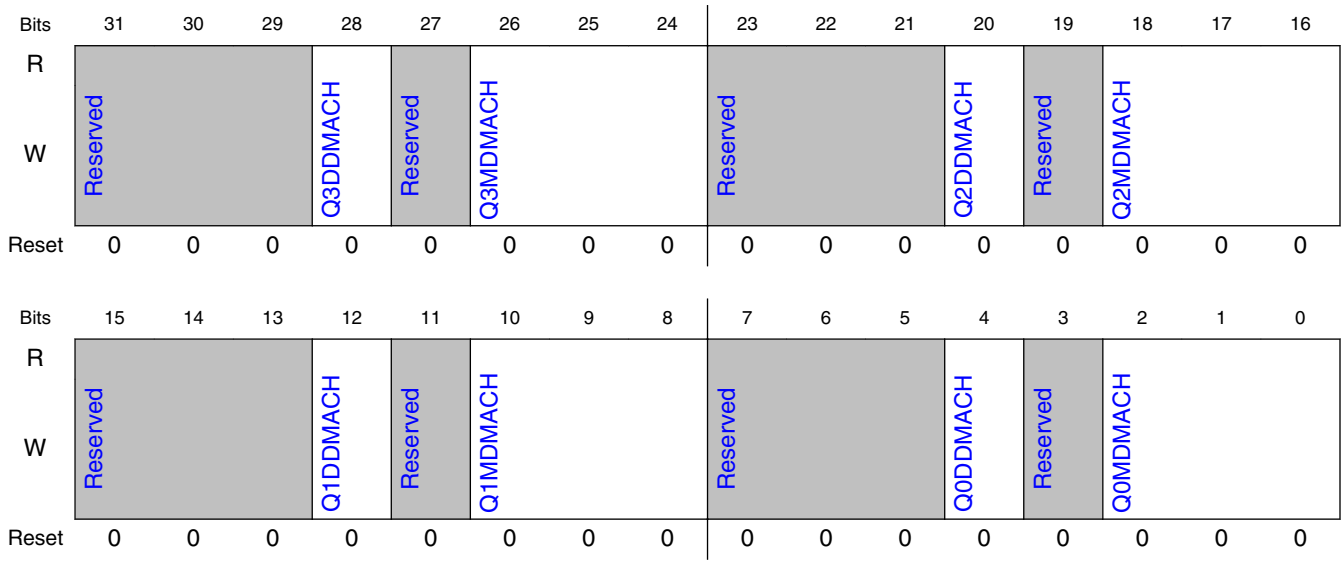
61.7.1.376.1 Offset

Register	Offset
MTL_RXQ_DMA_MAP0	C30h

61.7.1.376.2 Function

The Receive Queue and DMA Channel Mapping 0 register is reserved in EQOS-CORE and EQOS-MTL configurations.

61.7.1.376.3 Diagram



61.7.1.376.4 Fields

Field	Function
31-29 —	Reserved
28 Q3DDMACH	Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 3 are routed to the DMA Channel programmed in the Q3MDMACH field (Bits[26:24]). 0b - Queue 3 disabled for DA-based DMA Channel Selection 1b - Queue 3 enabled for DA-based DMA Channel Selection
27 —	Reserved
26-24 Q3MDMACH	Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Queue 3 Mapped to DMA Channel This field controls the routing of the received packet in Queue 3 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1

Table continues on the next page...

Field	Function
	- 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q3DDMACH field is reset. Note: The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the others are reserved
23-21 —	Reserved
20 Q2DDMACH	Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 2 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 2 are routed to the DMA Channel programmed in the Q2MDMACH field (Bits[18:16]). 0b - Queue 2 disabled for DA-based DMA Channel Selection 1b - Queue 2 enabled for DA-based DMA Channel Selection
19 —	Reserved
18-16 Q2MDMACH	Queue 2 Mapped to DMA Channel This field controls the routing of the received packet in Queue 2 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q2DDMACH field is reset.
15-13 —	Reserved
12 Q1DDMACH	Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 1 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 1 are routed to the DMA Channel programmed in the Q1MDMACH field (Bits[10:8]). 0b - Queue 1 disabled for DA-based DMA Channel Selection 1b - Queue 1 enabled for DA-based DMA Channel Selection
11 —	Reserved

Table continues on the next page...

Field	Function
10-8 Q1MDMACH	Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. Queue 1 Mapped to DMA Channel This field controls the routing of the received packet in Queue 1 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q1DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.
7-5 —	Reserved
4 Q0DDMACH	Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 0 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 0 are routed to the DMA Channel programmed in the Q0MDMACH field. 0b - Queue 0 disabled for DA-based DMA Channel Selection 1b - Queue 0 enabled for DA-based DMA Channel Selection
3 —	Reserved
2-0 Q0MDMACH	Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. Queue 0 Mapped to DMA Channel This field controls the routing of the packet received in Queue 0 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3

Field	Function
	- 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q0DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

61.7.1.377 Receive Queue and DMA Channel Mapping 1 (MTL_RXQ_DMA_MAP1)

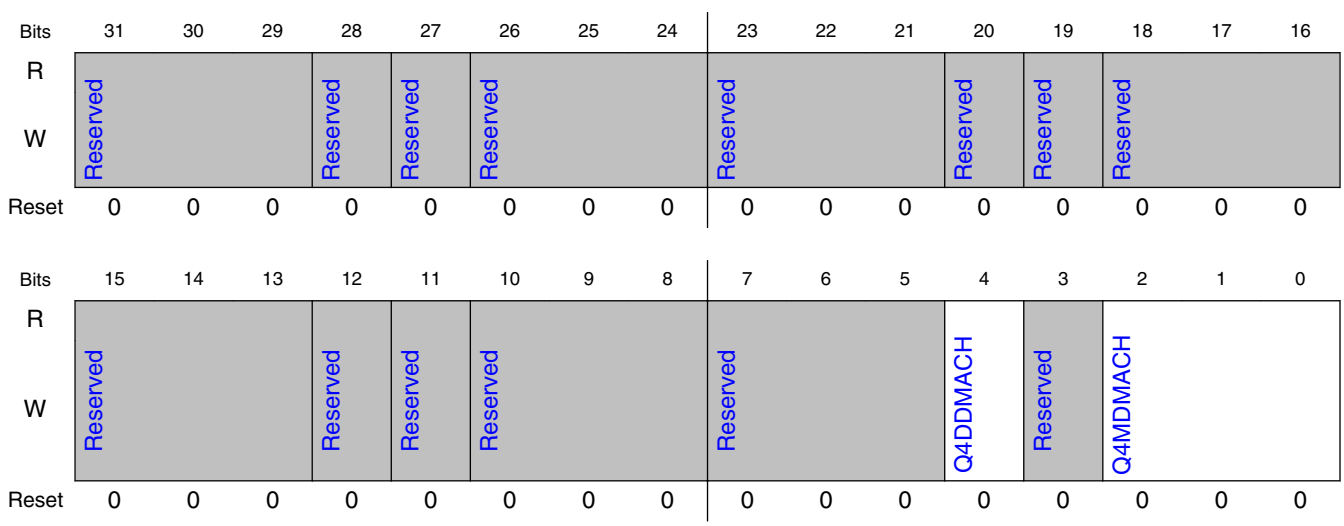
61.7.1.377.1 Offset

Register	Offset
MTL_RXQ_DMA_MAP1	C34h

61.7.1.377.2 Function

The Receive Queue and DMA Channel Mapping 1 register is reserved in EQOS-CORE and EQOS-MTL configurations.

61.7.1.377.3 Diagram



61.7.1.377.4 Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26-24 —	Reserved
23-21 —	Reserved
20 —	Reserved
19 —	Reserved
18-16 —	Reserved
15-13 —	Reserved
12 —	Reserved
11 —	Reserved
10-8 —	Reserved
7-5 —	Reserved
4 Q4DDMACH	Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. Queue 4 Enabled for DA-based DMA Channel Selection When set, this bit indicates that the packets received in Queue 4 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 4 are routed to the DMA Channel programmed in the Q4MDMACH field. 0b - Queue 4 disabled for DA-based DMA Channel Selection 1b - Queue 4 enabled for DA-based DMA Channel Selection
3 —	Reserved

Table continues on the next page...

Field	Function
2-0 Q4MDMACH	Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset. Queue 4 Mapped to DMA Channel This field controls the routing of the packet received in Queue 4 to the DMA channel: - 000: DMA Channel 0 - 001: DMA Channel 1 - 010: DMA Channel 2 - 011: DMA Channel 3 - 100: DMA Channel 4 - 101: Reserved - 110: Reserved - 111: Reserved This field is valid when the Q4DDMACH field is reset. The width of this field depends on the number of RX DMA channels and not all the values may be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.

61.7.1.378 Time Based Scheduling Control (MTL_TBS_CTRL)

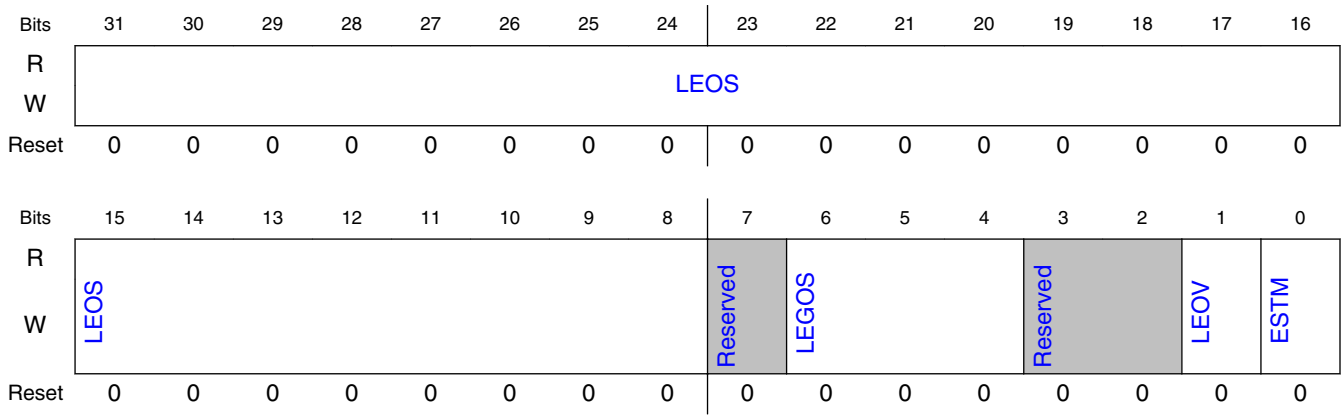
61.7.1.378.1 Offset

Register	Offset
MTL_TBS_CTRL	C40h

61.7.1.378.2 Function

This register controls the operation of Time Based Scheduling.

61.7.1.378.3 Diagram



61.7.1.378.4 Fields

Field	Function
31-8 LEOS	Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Launch Expiry Offset The value in units of 256 nanoseconds that has to be added to the Launch time to compute the Launch Expiry time. Value valid only when LEOV is set. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 LEGOS	Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Launch Expiry GSN Offset The number GSN slots that has to be added to the Launch GSN to compute the Launch Expiry time. Value valid only when LEOV is set.
3-2 —	Reserved
1 LEOV	Launch Expiry Offset Valid When set indicates the LEOS field is valid. Launch Expiry Offset Valid When set indicates the LEOS field is valid. When not set, indicates the Launch Expiry Offset is not valid and the MTL must not check for Launch expiry time. 0b - LEOS field is invalid 1b - LEOS field is valid
0 ESTM	EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list. EST offset Mode When this bit is set, the Launch Time value used in Time Based Scheduling is interpreted as an EST offset value and is added to the Base Time Register (BTR) of the current list.

Field	Function
	When reset, the Launch Time value is used as an absolute value that should be compared with the System time [39:8]. 0b - EST offset Mode is disabled 1b - EST offset Mode is enabled

61.7.1.379 Enhancements to Scheduled Transmission Control (MTL_EST_CONTROL)

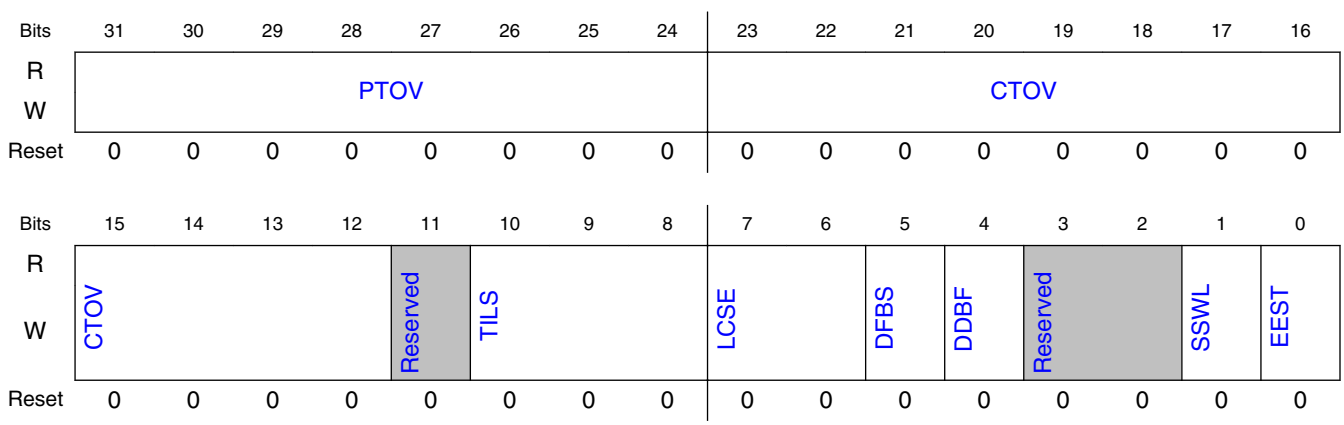
61.7.1.379.1 Offset

Register	Offset
MTL_EST_CONTROL	C50h

61.7.1.379.2 Function

This register controls the operation of Enhancements to Scheduled Transmission (IEEE802.1Qbv).

61.7.1.379.3 Diagram



61.7.1.379.4 Fields

Field	Function
31-24	PTP Time Offset Value The value of PTP Clock period multiplied by 6 in nanoseconds.
PTOV	PTP Time Offset Value

Table continues on the next page...

Field	Function
	The value of PTP Clock period multiplied by 6 in nanoseconds. This value is needed to avoid transmission overruns at the beginning of the installation of a new GCL.
23-12 CTOV	Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. Current Time Offset Value Provides a 12 bit time offset value in nano second that is added to the current time to compensate for all the implementation pipeline delays such as the CDC sync delay, buffering delays, data path delays etc. This offset helps to ensure that the impact of gate controls is visible on the line exactly at the pre-determined schedule (or as close to the schedule as possible).
11 —	Reserved
10-8 TILS	Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. Time Interval Left Shift Amount This field provides the left shift amount for the programmed Time Interval values used in the Gate Control Lists. - 000: No left shift needed (equal to x1ns) - 001: Left shift TI by 1 bit (equal to x2ns) - 010: Left shift TI by 2 bits (equal to x4ns) - . - . - 100: Left shift TI by 7 bits (equal to x128ns) Based on the configuration one or more bits of this field should be treated as Reserved/Read-Only.
7-6 LCSE	Loop Count to report Scheduling Error Programmable number of GCL list iterations before reporting an HLBS error defined in EST_STATUS register. 00b - 4 iterations 01b - 8 iterations 10b - 16 iterations 11b - 32 iterations
5 DFBS	Drop Frames causing Scheduling Error When set frames reported to cause HOL Blocking due to not getting scheduled (HLBS field of EST_STATUS register) after 4,8,16,32 (based on LCSE field of this register) GCL iterations are dropped. 0b - Do not Drop Frames causing Scheduling Error 1b - Drop Frames causing Scheduling Error
4 DDBF	Do not Drop frames during Frame Size Error When set, frames are not be dropped during Head-of-Line blocking due to Frame Size Error (HLBF field of MTL_EST_STATUS register). 0b - Drop frames during Frame Size Error 1b - Do not Drop frames during Frame Size Error
3-2 —	Reserved
1 SSWL	Switch to S/W owned list When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR. Switch to S/W owned list

Table continues on the next page...

Field	Function
	When set indicates that the software has programmed that list that it currently owns (SWOL) and the hardware should switch to the new list based on the new BTR. Hardware clears this bit when the switch to the SWOL happens to indicate the completion of the switch or when an BTR error (BTRE in Status register) is set. When BTRE is set this bit is cleared but SWOL is not updated as the switch was not successful. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Switch to S/W owned list is disabled 1b - Switch to S/W owned list is enabled
0 EEST	Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Enable EST When reset, the gate control list processing is halted and all gates are assumed to be in Open state. Should be set for the hardware to start processing the gate control lists. During the toggle from 0 to 1, the gate control list processing starts only after the SSWL bit it set. When DWC_EQOS_ASP_ECC is selected during the configuration, if any uncorrectable error is detected in the EST memory the hardware will reset this bit and disable the EST function. 0b - EST is disabled 1b - EST is enabled

61.7.1.380 MTL_EST_EXT_CONTROL (MTL_EST_EXT_CONTROL)

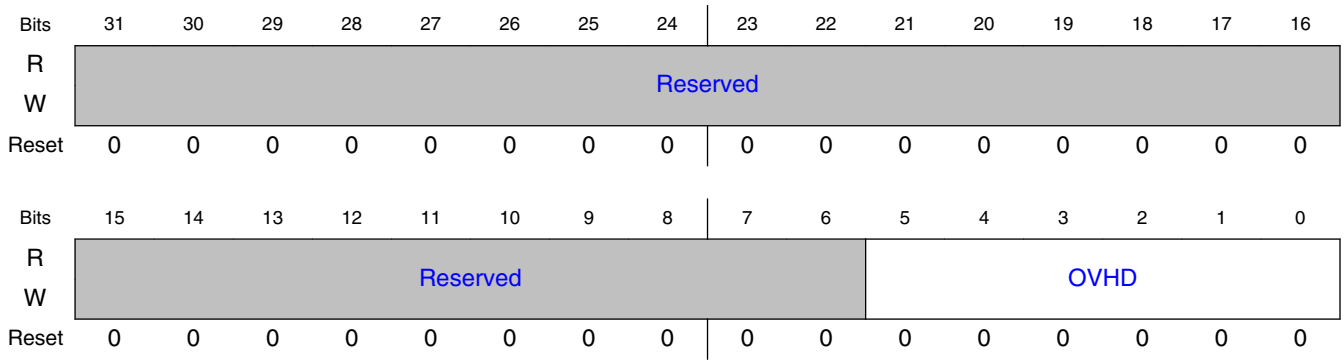
61.7.1.380.1 Offset

Register	Offset
MTL_EST_EXT_CONTROL	C54h

61.7.1.380.2 Function

This register indicates the number of overhead bytes for EST related scheduling.

61.7.1.380.3 Diagram



61.7.1.380.4 Fields

Field	Function
31-6 —	Reserved
5-0 OVHD	Overhead Bytes Value This field indicates the fixed overhead for every packet to account for EST Scheduler Delay, IPG or EIPG, and Preamble bytes.

61.7.1.381 Enhancements to Scheduled Transmission Status (MTL_EST_STATUS)

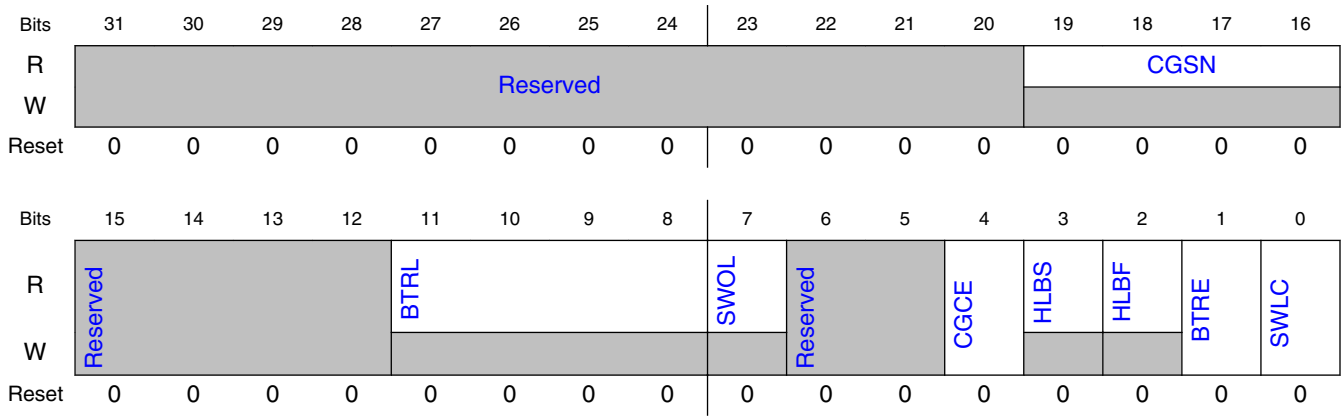
61.7.1.381.1 Offset

Register	Offset
MTL_EST_STATUS	C58h

61.7.1.381.2 Function

This register provides Status related to Enhancements to Scheduled Transmission (IEEE802.1Qbv).

61.7.1.381.3 Diagram



61.7.1.381.4 Fields

Field	Function
31-20 —	Reserved
19-16 CGSN	Current GCL Slot Number Indicates the slot number of the GCL list. Current GCL Slot Number Indicates the slot number of the GCL list. Slot number is a modulo 16 count of the GCL List loops executed so far. Even if a new GCL list is installed, the count is incremental.
15-12 —	Reserved
11-8 BTRL	BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. BTR Error Loop Count Provides the minimum count (N) for which the equation Current Time =< New BTR + (N * New Cycle Time) becomes true. N = "1111" indicates the iterations exceeded the value of 8 and the hardware was not able to update New BTR to be equal to or greater than Current Time. Software intervention is needed to update the New BTR. Value cleared when BTRE field of this register is cleared.
7 SWOL	S/W owned list When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. S/W owned list When '0' indicates Gate control list number "0" is owned by software and when "1" indicates the Gate Control list "1" is owned by the software. Any reads/writes by the software (using indirect access via GCL_Control) is directed to the list indicated by this value by default. The inverse of this value is treated as HWOL. R/W operations performed by hardware are directed to the list pointed by HWOL by default. 0b - Gate control list number "0" is owned by software 1b - Gate control list number "1" is owned by software
6-5	Reserved

Table continues on the next page...

Field	Function
—	
4 CGCE	Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). Constant Gate Control Error This error occurs when the list length (LLR) is 1 and the programmed Time Interval (TI) value after the optional Left Shifting is less than or equal to the Cycle Time (CTR). The above programming implies Gates are either always Closed or always Open based on the Gate Control values; the same effect can be achieved by other simpler (non TSN) programming mechanisms. Since the implementation does not support such a programming an error is reported. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Constant Gate Control Error not detected 1b - Constant Gate Control Error detected
3 HLBS	Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Head-Of-Line Blocking due to Scheduling Set when the frame is not able to win arbitration and get scheduled even after 4 iterations of the GCL. Indicates to software a potential programming error. The one hot encoded values of the Queue Numbers that are not able to make progress are indicated in the MTL_EST_SCH_ERROR register. Bit cleared when MTL_EST_SCH_ERROR register is all zeroes. 0b - Head-Of-Line Blocking due to Scheduling not detected 1b - Head-Of-Line Blocking due to Scheduling detected
2 HLBF	Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. Head-Of-Line Blocking due to Frame Size Set when HOL Blocking is noticed on one or more Queues as a result of none of the Time Intervals of gate open in the GCL being greater than or equal to the duration needed for frame size (or frame fragment size when preemption is enabled) transmission. The one hot encoded Queue numbers that are experiencing HLBF are indicated in the MTL_EST_FRM_SIZE_ERROR register. Additionally, the first Queue number that experienced HLBF along with the frame size is captured in MTL_EST_FRM_SIZE_CAPTURE register. Bit cleared when MTL_EST_FRAME_SIZE_ ERROR register is all zeros. 0b - Head-Of-Line Blocking due to Frame Size not detected 1b - Head-Of-Line Blocking due to Frame Size detected
1 BTRE	BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. BTR Error When "1" indicates a programming error in the BTR of SWOL where the programmed value is less than current time. If the BTRL = "1111", SWOL is not updated and Software should reprogram the BTR to a value greater than current time and then set SSWL to reinitiate the switch to SWOL. Else if the value of BTRL < "1111", SWOL is updated and this field indicates the number of iterations (of + CycleTime) taken by hardware to update the BTR to a value greater than Current Time. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - BTR Error not detected 1b - BTR Error detected
0 SWLC	Switch to S/W owned list Complete When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Switch to S/W owned list Complete

Field	Function
	When "1" indicates the hardware has successfully switched to the SWOL, and the SWOL bit has been updated to that effect. Cleared when the SSWL of MTL_EST_CONTROL register transitions from 0 to 1, or on a software write. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Switch to S/W owned list Complete not detected 1b - Switch to S/W owned list Complete detected

61.7.1.382 EST Scheduling Error (MTL_EST_SCH_ERROR)

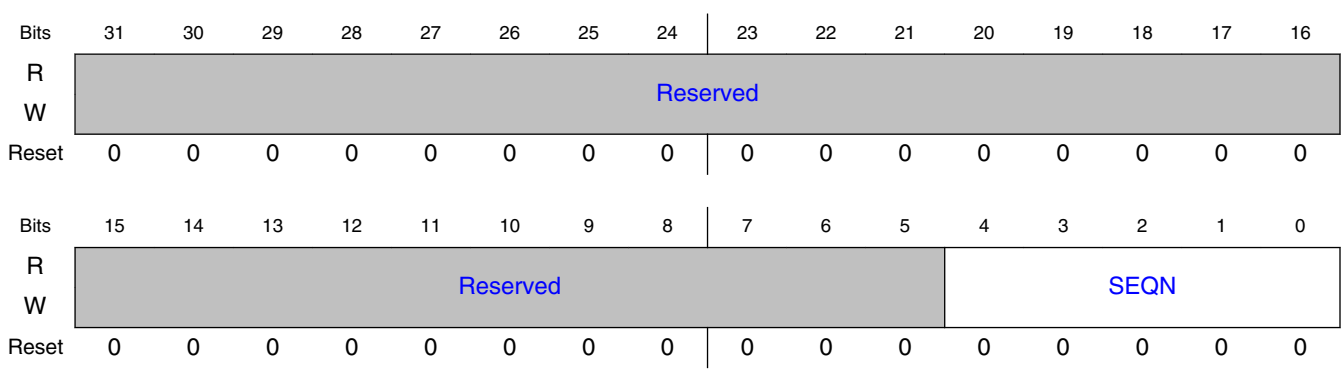
61.7.1.382.1 Offset

Register	Offset
MTL_EST_SCH_ERROR	C60h

61.7.1.382.2 Function

This register provides the One Hot encoded Queue Numbers that are having the Scheduling related error (timeout).

61.7.1.382.3 Diagram



61.7.1.382.4 Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Field	Function
4-0 SEQN	Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register. Schedule Error Queue Number The One Hot Encoded Queue Numbers that have experienced error/timeout described in HLBS field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

61.7.1.383 EST Frame Size Error (MTL_EST_FRM_SIZE_ERROR)

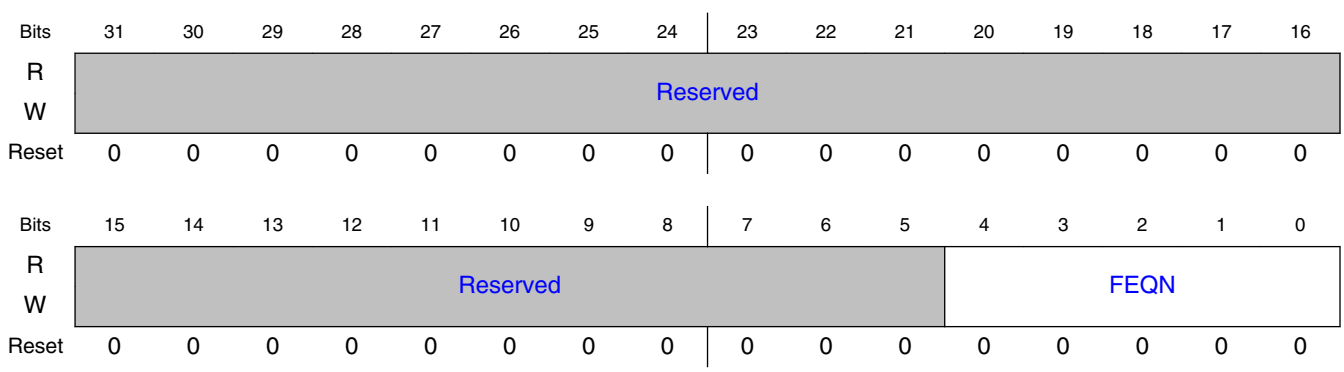
61.7.1.383.1 Offset

Register	Offset
MTL_EST_FRM_SIZE_ERROR	C64h

61.7.1.383.2 Function

This register provides the One Hot encoded Queue Numbers that are having the Frame Size related error.

61.7.1.383.3 Diagram



61.7.1.383.4 Fields

Field	Function
31-5	Reserved

Table continues on the next page...

Field	Function
—	
4-0 FEQN	Frame Size Error Queue Number The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Frame Size Error Queue Number The One Hot Encoded Queue Numbers that have experienced error described in HLBF field of status register. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.

61.7.1.384 EST Frame Size Capture (MTL_EST_FRM_SIZE_CAPTURE)

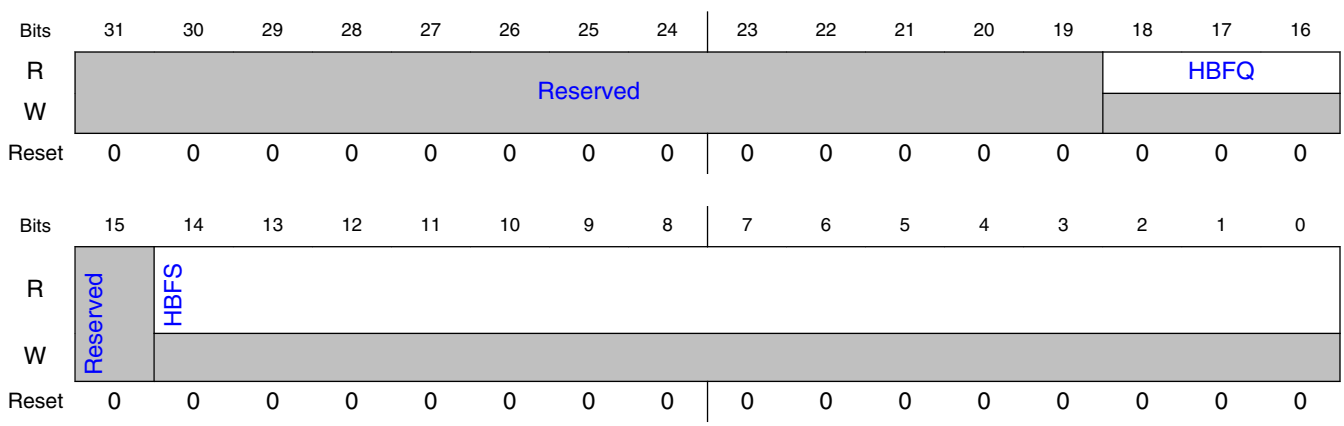
61.7.1.384.1 Offset

Register	Offset
MTL_EST_FRM_SIZE_CAPTURE	C68h

61.7.1.384.2 Function

This register captures the Frame Size and Queue Number of the first occurrence of the Frame Size related error. Up on clearing it captures the data of immediate next occurrence of a similar error.

61.7.1.384.3 Diagram



61.7.1.384.4 Fields

Field	Function
31-19 —	Reserved
18-16 HBFQ	Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Queue Number of HLBF Captures the binary value of the of the first Queue (number) experiencing HLBF error (see HLBF field of status register). Value once written is not altered by any subsequent queue errors of similar nature. Once cleared the queue number of the next occurring HLBF error is captured. Width is based on the number of Tx Queues configured; remaining bits are Read-Only. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.
15 —	Reserved
14-0 HBFS	Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Frame Size of HLBF Captures the Frame Size of the dropped frame related to queue number indicated in HBFQ field of this register. Contents of this register should be considered invalid, if this field is zero. Cleared when MTL_EST_FRM_SIZE_ERROR register is all zeros.

61.7.1.385 EST Interrupt Enable (MTL_EST_INTR_ENABLE)

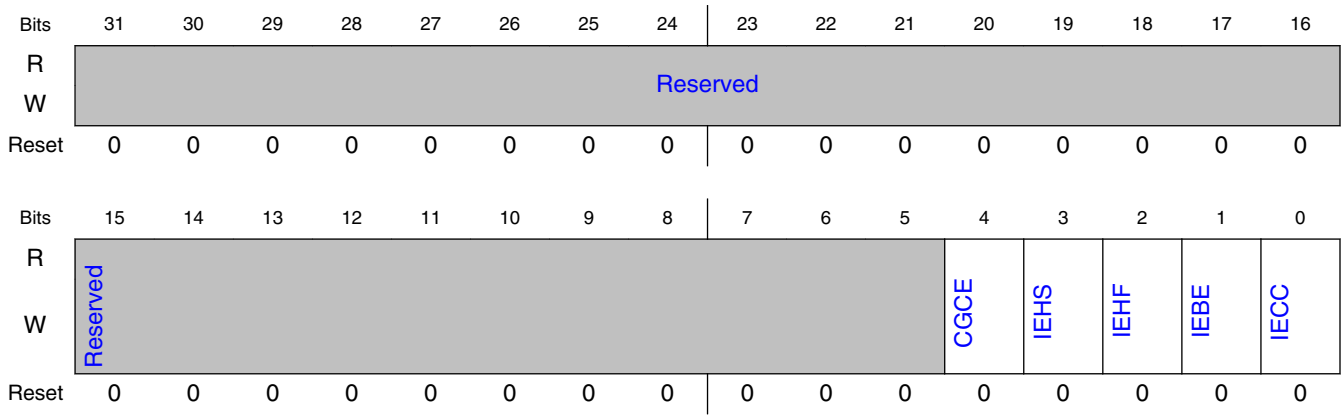
61.7.1.385.1 Offset

Register	Offset
MTL_EST_INTR_ENABLE	C70h

61.7.1.385.2 Function

This register implements the Interrupt Enable bits for the various events that generate an interrupt. Bit positions have a 1 to 1 correlation with the status bit positions in MTL_ETS_STATUS register.

61.7.1.385.3 Diagram



61.7.1.385.4 Fields

Field	Function
31-5 —	Reserved
4 CGCE	Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. Interrupt Enable for CGCE When set, generates interrupt when the Constant Gate Control Error occurs and is indicated in the status. When reset this event does not generate an interrupt 0b - Interrupt for CGCE is disabled 1b - Interrupt for CGCE is enabled
3 IEHS	Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. Interrupt Enable for HLBS When set, generates interrupt when the Head-of-Line Blocking due to Scheduling issue and is indicated in the status. When reset this event does not generate an interrupt. 0b - Interrupt for HLBS is disabled 1b - Interrupt for HLBS is enabled
2 IEHF	Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. Interrupt Enable for HLBF When set, generates interrupt when the Head-of-Line Blocking due to Frame Size error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0b - Interrupt for HLBF is disabled 1b - Interrupt for HLBF is enabled
1 IEBE	Interrupt Enable for BTR Error When set, generates interrupt when the BTR Error occurs and is indicated in the status. Interrupt Enable for BTR Error

Table continues on the next page...

Field	Function
	When set, generates interrupt when the BTR Error occurs and is indicated in the status. When reset this event does not generate an interrupt. 0b - Interrupt for BTR Error is disabled 1b - Interrupt for BTR Error is enabled
0 IECC	Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. Interrupt Enable for Switch List When set, generates interrupt when the configuration change is successful and the hardware has switched to the new list. When reset this event does not generate an interrupt. 0b - Interrupt for Switch List is disabled 1b - Interrupt for Switch List is enabled

61.7.1.386 EST GCL Control (MTL_EST_GCL_CONTROL)

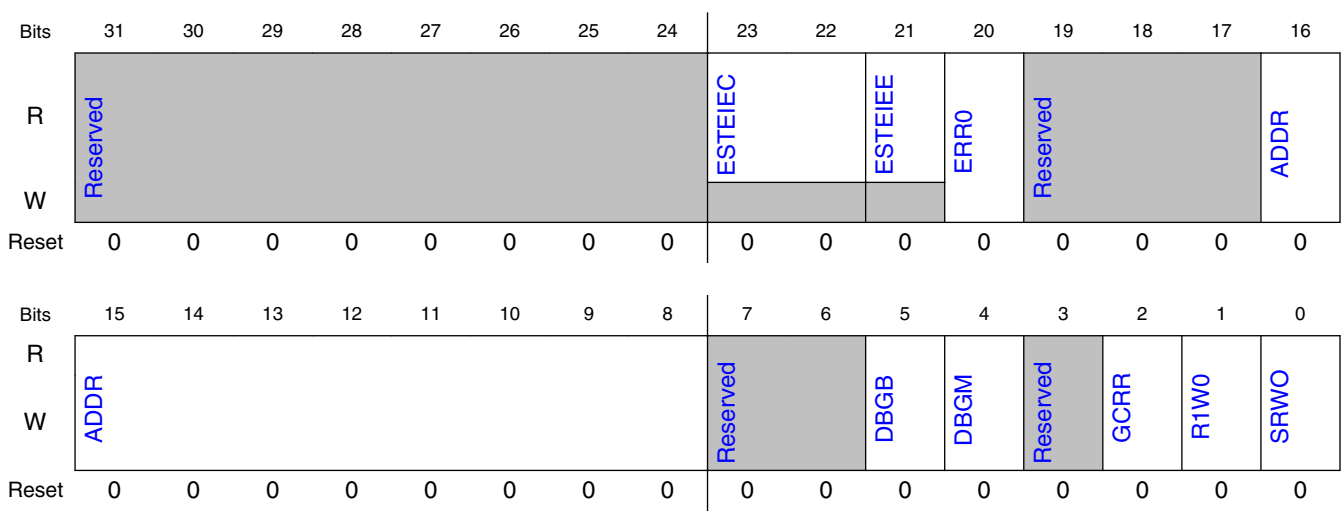
61.7.1.386.1 Offset

Register	Offset
MTL_EST_GCL_CONTROL	C80h

61.7.1.386.2 Function

This register provides the control information for reading/writing to the Gate Control lists.

61.7.1.386.3 Diagram



61.7.1.386.4 Fields

Field	Function
31-24 —	Reserved
23-22 ESTEIEC	ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. ECC Inject Error Control for EST Memory When EIEE bit of this register is set, following are the errors inserted based on the value encoded in this field. This filed will be valid only if DWC_EQOS_ASP_ECC feature is selected during the configuration, else it will be reserved. 00b - Insert 1 bit error 01b - Insert 2 bit errors 10b - Insert 3 bit errors 11b - Insert 1 bit error in address field
21 ESTEIEE	EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. EST ECC Inject Error Enable When set along with EEST bit of MTL_EST_CONTROL register, enables the ECC error injection feature. When reset, disables the ECC error injection feature. 0b - EST ECC Inject Error is disabled 1b - EST ECC Inject Error is enabled
20 ERR0	When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. When set indicates the last write operation was aborted as software writes to GCL and GCL registers is prohibited when SSWL bit of MTL_EST_CONTROL Register is set. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - ERR0 is disabled 1b - ERR1 is enabled
19-17 —	Reserved
16-8 ADDR	Gate Control List Address: (GCLA when GCRR is "0"). Gate Control List Address: (GCLA when GCRR is "0"). Provides the address (row number) of the Gate Control List at which the R/W operation has to be performed. By default the Gate Control List pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. The width of this field is dependent on DWC_EQOS_EST_DEP; unused bits should be treated as read only. Gate Control list Related Registers Address: (GCRA when GCRR is "1"). By default the GCL related register set pointed by SWOL of MTL_EST_STATUS is selected for R/W, however if the DBGM bit of this register is set, a debug mode access is given to R/W from DBGB. Lower 3 bits are only used in this mode, higher order bits are treated as dont cares. - 000: BTR Low (31:0) - 001: BTR High (63:31)

Table continues on the next page...

Field	Function
	- 010: CTR Low (31:0) - 011: CTR High (39:32) - 100: TER (31:0) - 101: LLR (n:0) (where n is (log{DWC_EQOS_EST_DEP} / log2)) - Others: Reserved
7-6 —	Reserved
5 DBGB	Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). Debug Mode Bank Select When set to "0" indicates R/W in debug mode should be directed to Bank 0 (GCL0 and corresponding Time related registers). When set to "1" indicates R/W in debug mode should be directed to Bank 1 (GCL1 and corresponding Time related registers). This value is used when DBGM is set and overrides by value of SWOL which is normally used. 0b - R/W in debug mode should be directed to Bank 0 1b - R/W in debug mode should be directed to Bank 1
4 DBGM	Debug Mode When set to "1" indicates R/W in debug mode where the memory bank (for GCL and Time related registers) is explicitly provided by DBGB value, when set to "0" SWOL bit is used to determine which bank to use. 0b - Debug Mode is disabled 1b - Debug Mode is enabled
3 —	Reserved
2 GCRR	Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. Gate Control Related Registers When set to "1" indicates the R/W access is for the GCL related registers (BTR, CTR, TER, LLR) whose address is provided by GCRA. When "0" indicates R/W should be directed to GCL from the address provided by GCLA. 0b - Gate Control Related Registers are disabled 1b - Gate Control Related Registers are enabled
1 R1W0	Read '1', Write '0': When set to '1': Read Operation When set to '0': Write Operation. 0b - Write Operation 1b - Read Operation
0 SRWO	Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. Start Read/Write Op When set indicates a Read/Write Op has started and is in progress. When reset by hardware indicates the R/W Op has completed or an error has occurred (when bit 20 is set) Reads: Data can be read from MTL_EST_GCL_DATA register after this bit is reset Writes: MTL_EST_GCL_DATA should be programmed with write data before setting SRWO. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Start Read/Write Op disabled 1b - Start Read/Write Op enabled

61.7.1.387 EST GCL Data (MTL_EST_GCL_DATA)

61.7.1.387.1 Offset

Register	Offset
MTL_EST_GCL_DATA	C84h

61.7.1.387.2 Function

This register holds the read data or write data in case of reads and writes respectively.

61.7.1.387.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GCD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GCD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.387.4 Fields

Field	Function
31-0 GCD	Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Gate Control Data The data corresponding to the address selected in the MTL_GCL_CONTROL register. Used for both Read and Write operations.

61.7.1.388 Frame Preemption Control and Status (MTL_FPE_CTRL_STS)

61.7.1.388.1 Offset

Register	Offset
MTL_FPE_CTRL_STS	C90h

61.7.1.388.2 Function

This register controls the operation of, and provides status for Frame Preemption (IEEE802.1Qbu/802.3br).

61.7.1.388.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			HRS	Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			PEC					Reserved						AFSZ	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.388.4 Fields

Field	Function
31-29 —	Reserved
28 HRS	Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. Hold/Release Status - 1: Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State. - 0: Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State. 0b - Indicates a Set-and-Release-MAC operation was last executed and the pMAC is in Release State 1b - Indicates a Set-and-Hold-MAC operation was last executed and the pMAC is in Hold State
27-16 —	Reserved
15-13 —	Reserved

Table continues on the next page...

Field	Function
12-8 PEC	Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. Preemption Classification When set indicates the corresponding Queue must be classified as preemptable, when '0' Queue is classified as express. When both EST (Qbv) and Preemption are enabled, Queue-0 is always assumed to be preemptable. When EST (Qbv) is enabled Queues categorized as preemptable here are always assumed to be in "Open" state in the Gate Control List.
7-2 —	Reserved
1-0 AFSZ	Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. Additional Fragment Size used to indicate, in units of 64 bytes, the minimum number of bytes over 64 bytes required in non-final fragments of preempted frames. The minimum non-final fragment size is (AFSZ +1) * 64 bytes

61.7.1.389 Frame Preemption Hold and Release Advance (MTL_FPE_ADVANCE)

61.7.1.389.1 Offset

Register	Offset
MTL_FPE_ADVANCE	C94h

61.7.1.389.2 Function

This register holds the Hold and Release Advance time.

61.7.1.389.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RADV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HADV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.389.4 Fields

Field	Function
31-16 RADV	Release Advance The maximum time in nanoseconds that can elapse between issuing a RELEASE to the MAC and the MAC being ready to resume transmission of preemptable frames, in the absence of there being any express frames available for transmission.
15-0 HADV	Hold Advance The maximum time in nanoseconds that can elapse between issuing a HOLD to the MAC and the MAC ceasing to transmit any preemptable frame that is in the process of transmission or any preemptable frames that are queued for transmission.

61.7.1.390 RXP Control Status (MTL_RXP_CONTROL_STATUS)

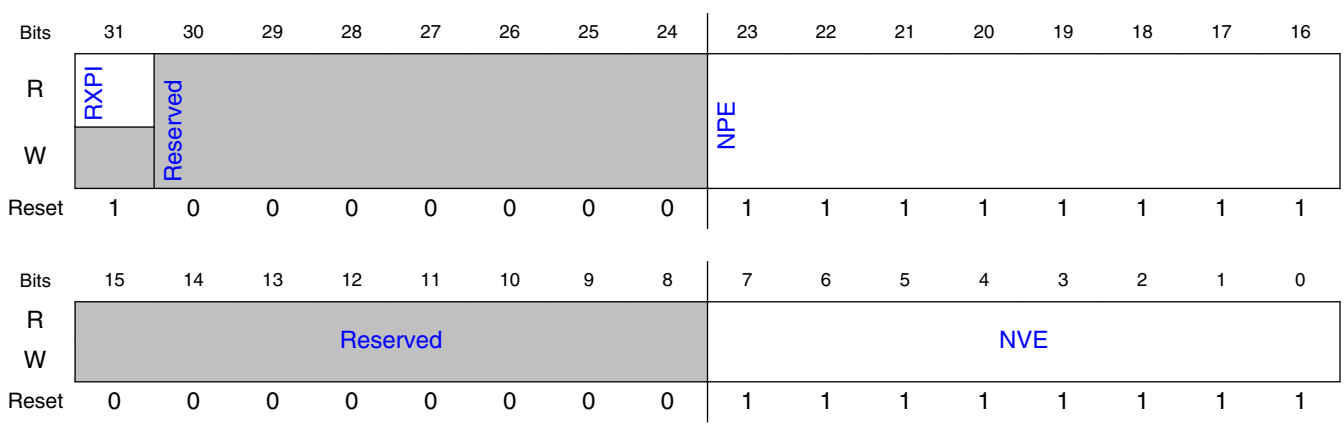
61.7.1.390.1 Offset

Register	Offset
MTL_RXP_CONTROL_STATUS	CA0h

61.7.1.390.2 Function

The RXP Control Status register establishes the operating mode of Rx Parser and provides some status.

61.7.1.390.3 Diagram



61.7.1.390.4 Fields

Field	Function
31 RXPI	RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. RX Parser in Idle state This status bit is set to 1 when the Rx parser is in Idle State and waiting for a new packet for processing. This bit is used as a handshake with software when parser gets disables. After disabling, when bit is set then software can update the Rx parser instruction table. 0b - RX Parser not in Idle state 1b - RX Parser in Idle state
30-24 —	Reserved
23-16 NPE	Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. Number of parsable entries in the Instruction table This control indicates the number of parsable entries in the Instruction Memory. This is used in Rx parser logic to detect programming Error. In case number of parsed entries for a packet is more than this entry then NPEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set.
15-8 —	Reserved
7-0 NVE	Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. Number of valid entries in the Instruction table This control indicates the number of valid entries in the Instruction Memory. This is used in Rx parser logic to detect any programming Error. In case while parsing Table address (memory address) found to be more than this entry then NVEOVIS bit in the MTL_RXP_INTERRUPT_CONTROL_STATUS register is set. Note: The minimum value of this should be 2.

61.7.1.391 RXP Interrupt Control Status (MTL_RXP_INTERRUPT_CONTROL_STATUS)

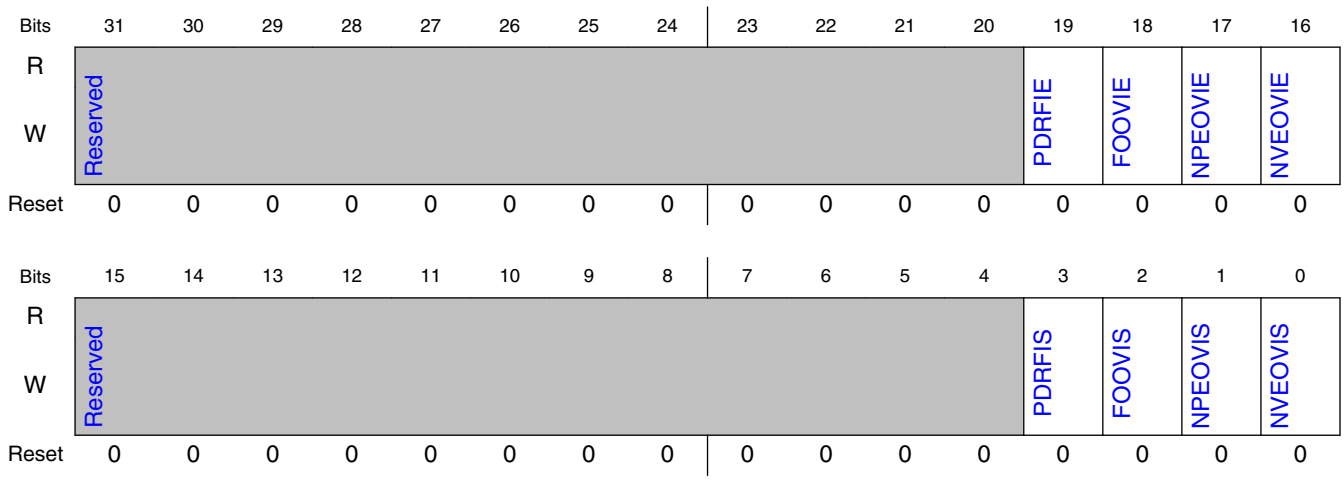
61.7.1.391.1 Offset

Register	Offset
MTL_RXP_INTERRUPT_CONTROL_STATUS	CA4h

61.7.1.391.2 Function

The RXP Interrupt Control Status register provides enable control for the interrupts and provides interrupt status.

61.7.1.391.3 Diagram



61.7.1.391.4 Fields

Field	Function
31-20 —	Reserved
19 PDRFIE	Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. Packet Drop due to RF Interrupt Enable When this bit is set, the PDRFIS interrupt is enabled. When this bit is reset, the PDRFIS interrupt is disabled. 0b - Packet Drop due to RF Interrupt is disabled 1b - Packet Drop due to RF Interrupt is enabled
18 FOOVIE	Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. Frame Offset Overflow Interrupt Enable When this bit is set, the FOOVIS interrupt is enabled. When this bit is reset, the FOOVIS interrupt is disabled. 0b - Frame Offset Overflow Interrupt is disabled 1b - Frame Offset Overflow Interrupt is enabled
17 NPEOVIE	Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. Number of Parsable Entries Overflow Interrupt Enable When this bit is set, the NPEOVIS interrupt is enabled. When this bit is reset, the NPEOVIS interrupt is disabled. 0b - Number of Parsable Entries Overflow Interrupt is disabled

Table continues on the next page...

Field	Function
	1b - Number of Parsable Entries Overflow Interrupt is enabled
16 NVEOVIE	Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. Number of Valid Entries Overflow Interrupt Enable When this bit is set, the NVEOVIS interrupt is enabled. When this bit is reset, the NVEOVIS interrupt is disabled. 0b - Number of Valid Entries Overflow Interrupt is disabled 1b - Number of Valid Entries Overflow Interrupt is enabled
15-4 —	Reserved
3 PDRFIS	Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. Packet Dropped due to RF Interrupt Status If the Rx Parser result says to drop the packet by setting RF=1 in the instruction memory, then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Packet Dropped due to RF Interrupt Status not detected 1b - Packet Dropped due to RF Interrupt Status detected
2 FOOVIS	Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set. Frame Offset Overflow Interrupt Status While parsing if the Instruction table entry's 'Frame Offset' found to be more than EOF offset, then then this bit is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Frame Offset Overflow Interrupt Status not detected 1b - Frame Offset Overflow Interrupt Status detected
1 NPEOVIS	Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. Number of Parsable Entries Overflow Interrupt Status While parsing a packet if the number of parsed entries found to be more than NPE[] (Number of Parseable Entries in MTL_RXP_CONTROL register),then this bit is set to 1. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Number of Parsable Entries Overflow Interrupt Status not detected 1b - Number of Parsable Entries Overflow Interrupt Status detected
0 NVEOVIS	Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1. Number of Valid Entries Overflow Interrupt Status While parsing if the Instruction address found to be more than NVE (Number of Valid Entries in MTL_RXP_CONTROL register), then this bit is set to 1. This bit is cleared when the application writes 1 to this bit.

Field	Function
	Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Number of Valid Entries Overflow Interrupt Status not detected 1b - Number of Valid Entries Overflow Interrupt Status detected

61.7.1.392 RXP Drop Count (MTL_RXP_DROP_CNT)

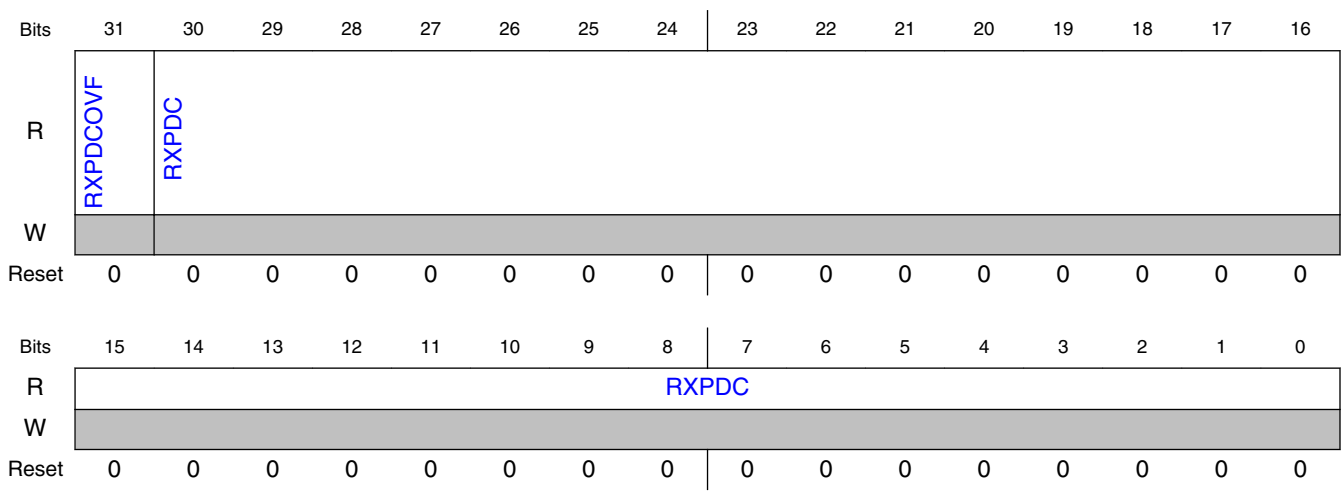
61.7.1.392.1 Offset

Register	Offset
MTL_RXP_DROP_CNT	CA8h

61.7.1.392.2 Function

The MTL_RXP_DROP_CNT register provides the drop count of Rx Parser initiated drops.

61.7.1.392.3 Diagram



61.7.1.392.4 Fields

Field	Function
31 RXPDCOVF	Rx Parser Drop Counter Overflow Bit When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Rx Parser Drop Counter Overflow Bit

Table continues on the next page...

Field	Function
	When set, this bit indicates that the MTL_RXP_DROP_CNT (RXPDC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Drop count overflow not occurred 1b - Rx Parser Drop count overflow occurred
30-0 RXPDC	Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. Rx Parser Drop count This 31-bit counter is implemented whenever a Rx Parser Drops a packet due to RF =1. The counter is cleared when the register is read.

61.7.1.393 RXP Error Count (MTL_RXP_ERROR_CNT)

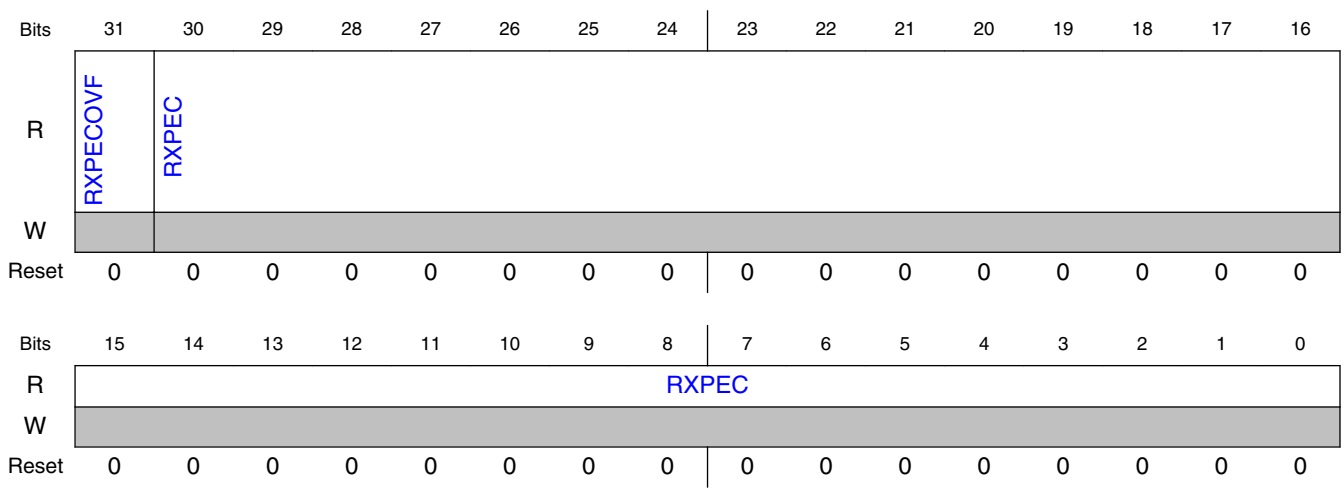
61.7.1.393.1 Offset

Register	Offset
MTL_RXP_ERROR_CNT	CACh

61.7.1.393.2 Function

The RXP_ERROR_CNT register provides the Rx Parser related error occurrence count.

61.7.1.393.3 Diagram



61.7.1.393.4 Fields

Field	Function
31 RXPECOVF	Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Rx Parser Error Counter Overflow Bit When set, this bit indicates that the MTL_RXP_ERROR_CNT (RXPEC) Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Error count overflow not occurred 1b - Rx Parser Error count overflow occurred
30-0 RXPEC	Rx Parser Error count This 31-bit counter is implemented whenever a Rx Parser encounters following Error scenarios - Entry address >= NVE[] - Number Parsed Entries >= NPE[] - Entry address > EOF data entry address The counter is cleared when the register is read.

61.7.1.394 RXP Indirect Access Control and Status (MTL_RXP_INDIRECT_ACC_CONTROL_STATUS)

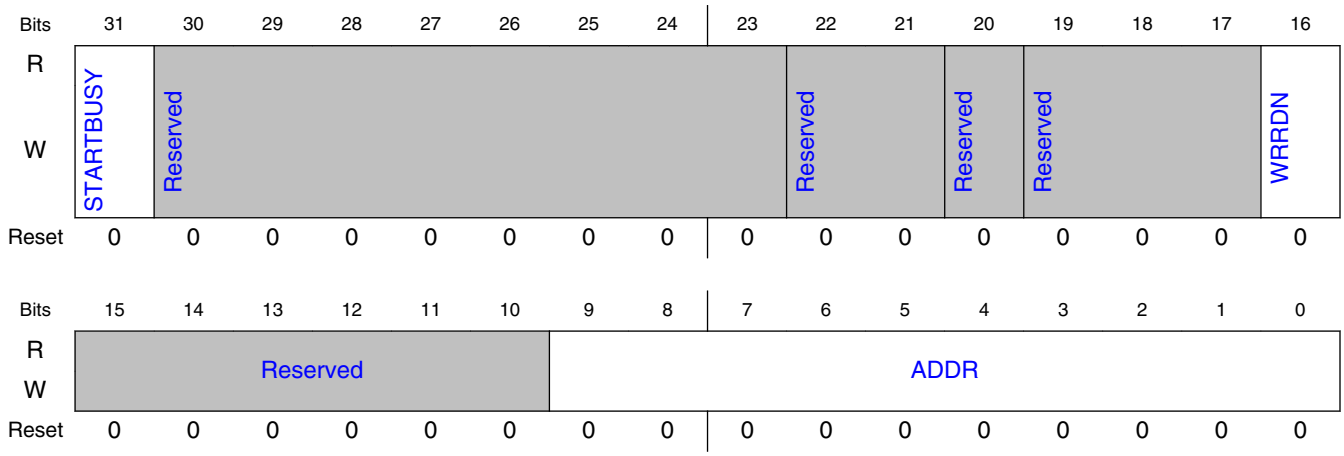
61.7.1.394.1 Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_CONTROL_STATUS	CB0h

61.7.1.394.2 Function

The RXP_INDIRECT_ACC_CONTROL_STATUS register provides the Indirect Access control and status for Rx Parser memory.

61.7.1.394.3 Diagram



61.7.1.394.4 Fields

Field	Function
31 STARTBUSY	FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. FRP Instruction Table Access Busy When this bit is set to 1 by the software then it indicates to start the Read/Write operation from/to the Rx Parser Memory. Software should read this bit as 0 before issuing read or write request to access the Parser Memory Instructions. This bit when set to 1 indicates that hardware is busy until its gets cleared by hardware and software should not issue any read or write operation. 0b - hardware not busy 1b - hardware is busy (Read/Write operation from/to the Rx Parser Memory)
30-23 —	Reserved
22-21 —	Reserved
20 —	Reserved
19-17 —	Reserved
16 WRRDN	Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. Read Write Control When this bit is set to 1 indicates the write operation to the Rx Parser Memory. When this bit is set to 0 indicates the read operation to the Rx Parser Memory. 0b - Read operation to the Rx Parser Memory 1b - Write operation to the Rx Parser Memory

Table continues on the next page...

Field	Function
15-10 —	Reserved
9-0 ADDR	FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. FRP Instruction Table Offset Address This field indicates the ADDR of the 32-bit entry in Rx parser instruction table. Each entry has 128-bit (4x32-bit words). The X depends on the configurable DWC_EQOS_FRP_ENTRIES If DWC_EQOS_FRP_ENTRIES == 256 , then X = 9 If DWC_EQOS_FRP_ENTRIES = 128 , then X = 8 IF DWC_EQOS_FRP_ENTRIES = 64, then X = 7 This is can be written by the software before issuing any Read/Write command. The hardware auto-increments this field after the read-write operation gets completed.

61.7.1.395 RXP Indirect Access Data (MTL_RXP_INDIRECT_ACC_DATA)

61.7.1.395.1 Offset

Register	Offset
MTL_RXP_INDIRECT_ACC_DATA	CB4h

61.7.1.395.2 Function

The RXP_INDIRECT_ACC_DATA registers holds the data associated to Indirect Access to Rx Parser memory.

61.7.1.395.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.395.4 Fields

Field	Function
31-0 DATA	FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. FRP Instruction Table Write/Read Data Software should write this register before issuing any write command. The hardware provides the read data from the Rx Parser Memory for read operation when STARTBUSY =0 after read command.

61.7.1.396 MTL_RXP_BYPASS_CNT (MTL_RXP_BYPASS_CNT)

61.7.1.396.1 Offset

Register	Offset
MTL_RXP_BYPASS_CNT	CB8h

61.7.1.396.2 Function

The MTL_RXP_BYPASS_CNT register provides the bypass count of Rx Parser.

61.7.1.396.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXPBCOF	RXPBC														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXPBC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.396.4 Fields

Field	Function
31 RXPBCOF	Rx Parser bypass Counter Overflow Bit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Indicates that MTL_RXP_BYPASS_CNT[RXPBC] counter field has not crossed the maximum limit 1b - Indicates that MTL_RXP_BYPASS_CNT[RXPBC] counter field has crossed the maximum limit
30-0 RXPBC	Rx Parser Bypass Count This 31-bit counter is implemented when a Rx Parser bypass a packet due to AF=1 and RF=1. The counter is cleared when the register is read.

61.7.1.397 Queue 0 Transmit Operation Mode (MTL_TXQ0_OPERATION_MODE)

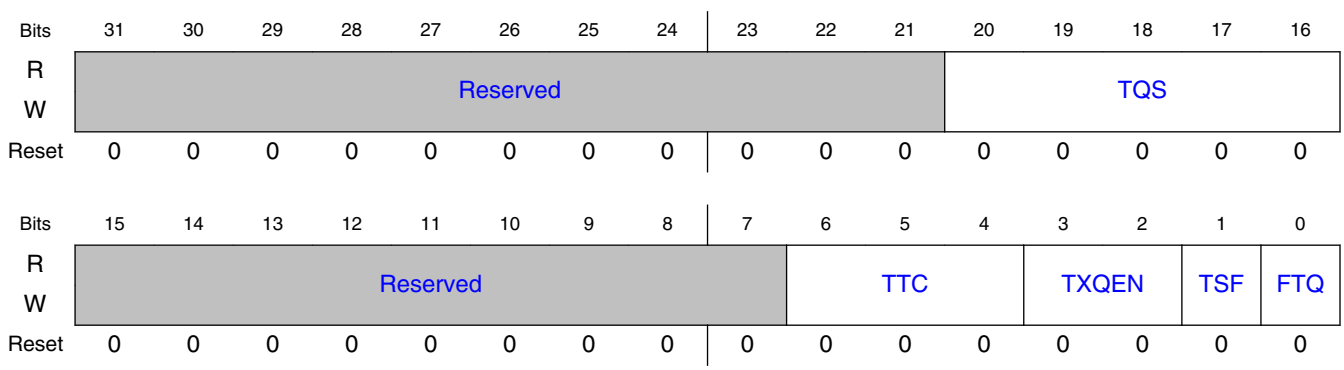
61.7.1.397.1 Offset

Register	Offset
MTL_TXQ0_OPERATION_MODE	D00h

61.7.1.397.2 Function

The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

61.7.1.397.3 Diagram



61.7.1.397.4 Fields

Field	Function
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000b - 32 001b - 64 010b - 96 011b - 128 100b - 192 101b - 256 110b - 384 111b - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Reserved - 2'b10: Enabled - 2'b11: Reserved This field is Read Only in Single Queue configurations and Read Write in Multiple Queue configurations. Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00b - Not enabled 01b - Enable in AV mode (Reserved in non-AV) 10b - Enabled 11b - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue.

Table continues on the next page...

Field	Function
	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0b - Transmit Store and Forward is disabled 1b - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ0_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flush Transmit Queue is disabled 1b - Flush Transmit Queue is enabled

61.7.1.398 Queue 0 Underflow Counter (MTL_TXQ0_UNDERFLOW)

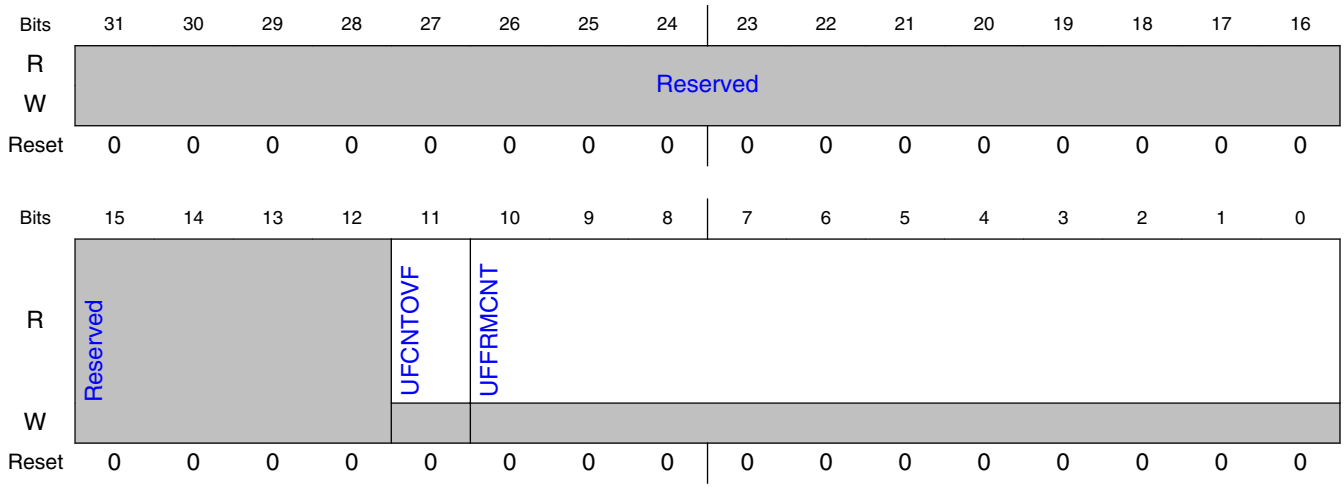
61.7.1.398.1 Offset

Register	Offset
MTL_TXQ0_UNDERFLOW	D04h

61.7.1.398.2 Function

The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

61.7.1.398.3 Diagram



61.7.1.398.4 Fields

Field	Function
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow not detected for Underflow Packet Counter 1b - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.399 Queue 0 Transmit Debug (MTL_TXQ0_DEBUG)

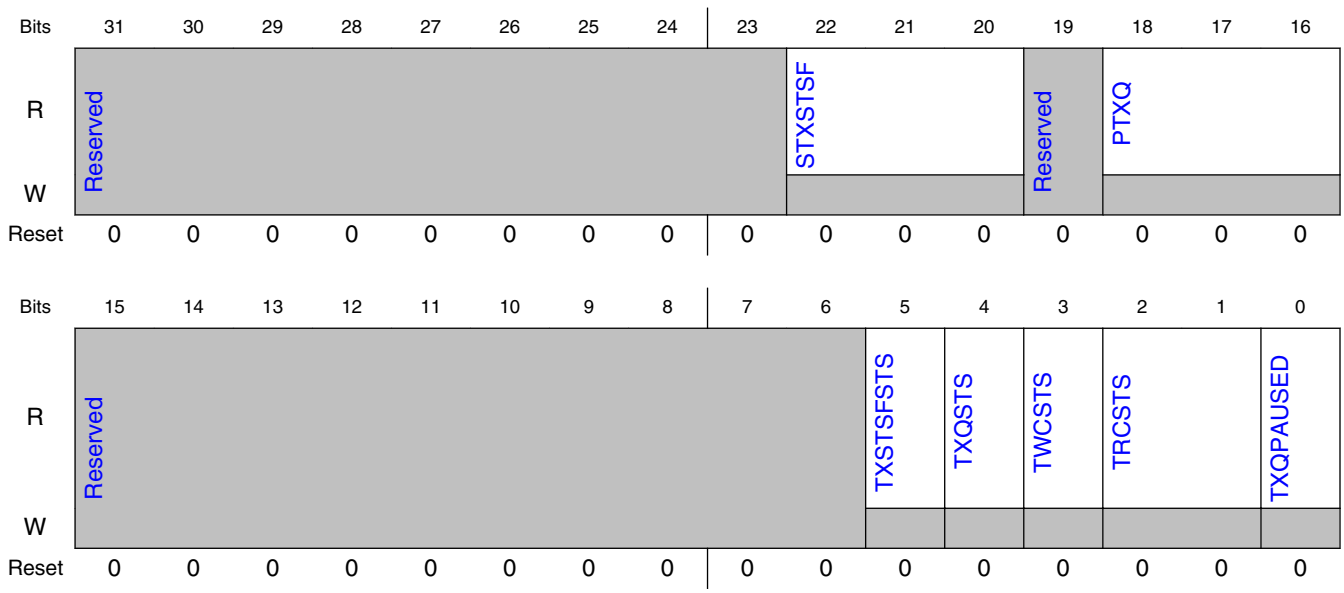
61.7.1.399.1 Offset

Register	Offset
MTL_TXQ0_DEBUG	D08h

61.7.1.399.2 Function

The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

61.7.1.399.3 Diagram



61.7.1.399.4 Fields

Field	Function
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved

Table continues on the next page...

Field	Function
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0b - MTL Tx Status FIFO Full status is not detected 1b - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0b - MTL Tx Queue Not Empty status is not detected 1b - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0b - MTL Tx Queue Write Controller status is not detected 1b - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00b - Idle state 01b - Read state (transferring data to the MAC transmitter) 10b - Waiting for pending Tx Status from the MAC transmitter 11b - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0b - Transmit Queue in Pause status is not detected 1b - Transmit Queue in Pause status is detected

61.7.1.400 Queue 0 ETS Status (MTL_TXQ0_ETS_STATUS)

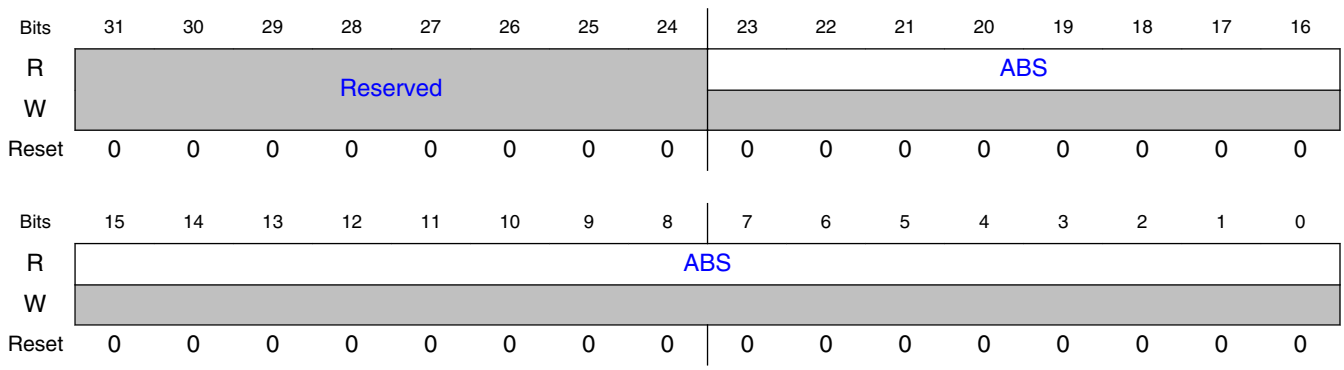
61.7.1.400.1 Offset

Register	Offset
MTL_TXQ0_ETS_STATUS	D14h

61.7.1.400.2 Function

The Queue 0 ETS Status register provides the average traffic transmitted in Queue 0.

61.7.1.400.3 Diagram



61.7.1.400.4 Fields

Field	Function
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. When the DCB operation is enabled for Queue 0, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

61.7.1.401 Queue 0 Quantum or Weights (MTL_TXQ0_QUANTUM_WEIGHT)

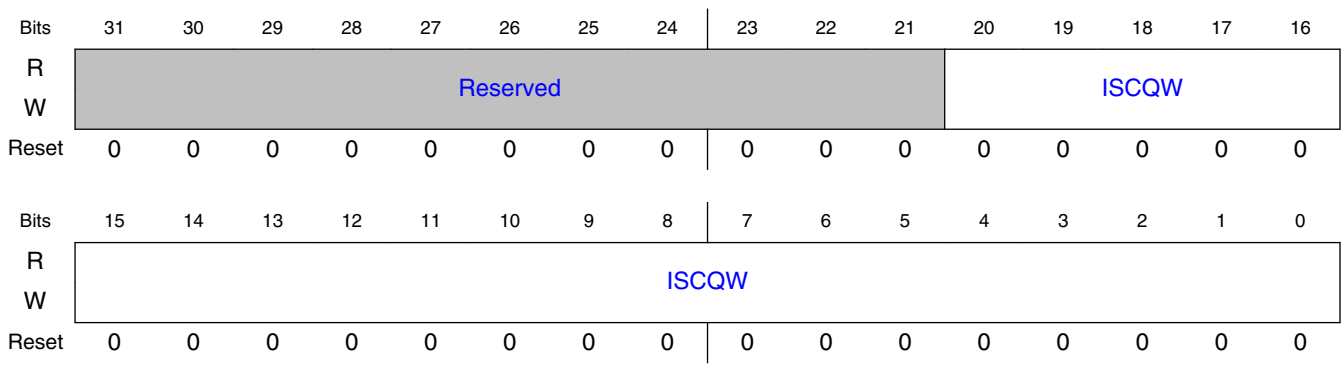
61.7.1.401.1 Offset

Register	Offset
MTL_TXQ0_QUANTUM_WEIGHT	D18h

61.7.1.401.2 Function

The Queue 0 Quantum or Weights register contains the quantum value for Deficit Weighted Round Robin (DWRR), weights for the Weighted Round Robin (WRR), and Weighted Fair Queuing (WFQ) for Queue 0.

61.7.1.401.3 Diagram



61.7.1.401.4 Fields

Field	Function
31-21 —	Reserved
20-0 ISCQW	Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. Quantum or Weights When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. When DCB operation is enabled with WFQ algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. The higher the programmed weights lesser the bandwidth allocated for the particular Transmit Queue. This is because the weights are used to compute the packet finish time (weights*packet_size). Lesser the finish time, higher the probability of the packet getting scheduled first and using more bandwidth.

Field	Function
	When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.

61.7.1.402 Queue 0 Interrupt Control Status (MTL_Q0_INTERRUPT_CONTROL_STATUS)

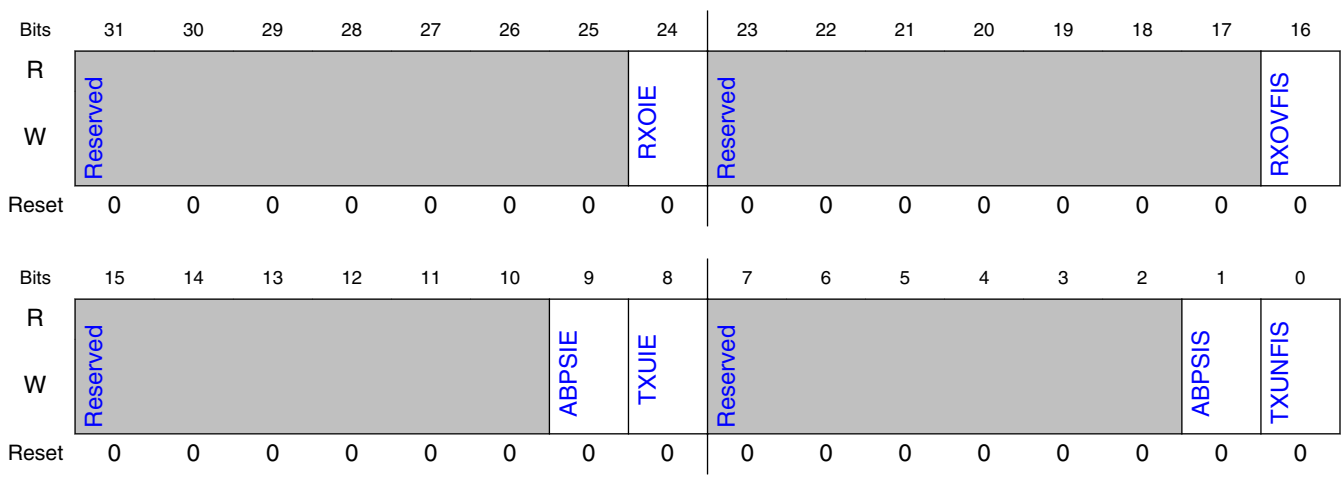
61.7.1.402.1 Offset

Register	Offset
MTL_Q0_INTERRUPT_CONTROL_STATUS	D2Ch

61.7.1.402.2 Function

This register contains the interrupt enable and status bits for the queue 0 interrupts.

61.7.1.402.3 Diagram



61.7.1.402.4 Fields

Field	Function
31-25 —	Reserved

Table continues on the next page...

Field	Function
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0b - Receive Queue Overflow Interrupt is disabled 1b - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Queue Overflow Interrupt Status not detected 1b - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0b - Average Bits Per Slot Interrupt is disabled 1b - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0b - Transmit Queue Underflow Interrupt Status is disabled 1b - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Average Bits Per Slot Interrupt Status not detected

Table continues on the next page...

Field	Function
	1b - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Queue Underflow Interrupt Status not detected 1b - Transmit Queue Underflow Interrupt Status detected

61.7.1.403 Queue 0 Receive Operation Mode (MTL_RXQ0_OPERATION_MODE)

61.7.1.403.1 Offset

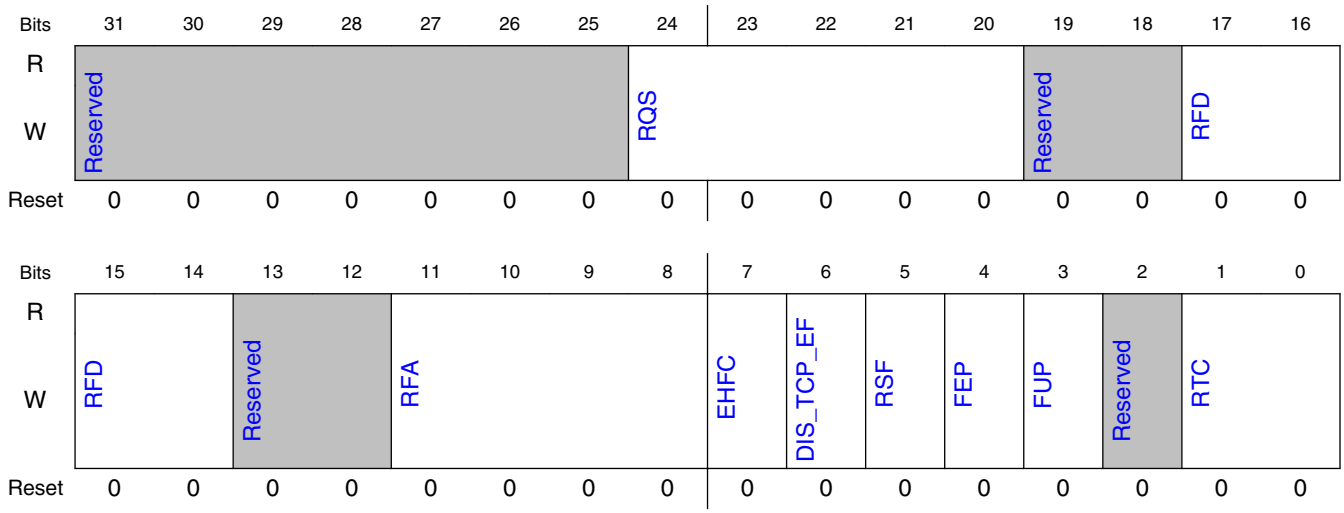
Register	Offset
MTL_RXQ0_OPERATION_MODE	D30h

61.7.1.403.2 Function

The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

61.7.1.403.3 Diagram



61.7.1.403.4 Fields

Field	Function
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB

Table continues on the next page...

Field	Function
	- ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0b - Hardware Flow Control is disabled 1b - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0b - Dropping of TCP/IP Checksum Error Packets is enabled 1b - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0b - Receive Queue Store and Forward is disabled 1b - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets

Table continues on the next page...

Field	Function
	When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0b - Forward Error Packets is disabled 1b - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0b - Forward Undersized Good Packets is disabled 1b - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00b - 64 01b - 32 10b - 96 11b - 128

61.7.1.404 Queue 0 Missed Packet and Overflow Counter (MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT)

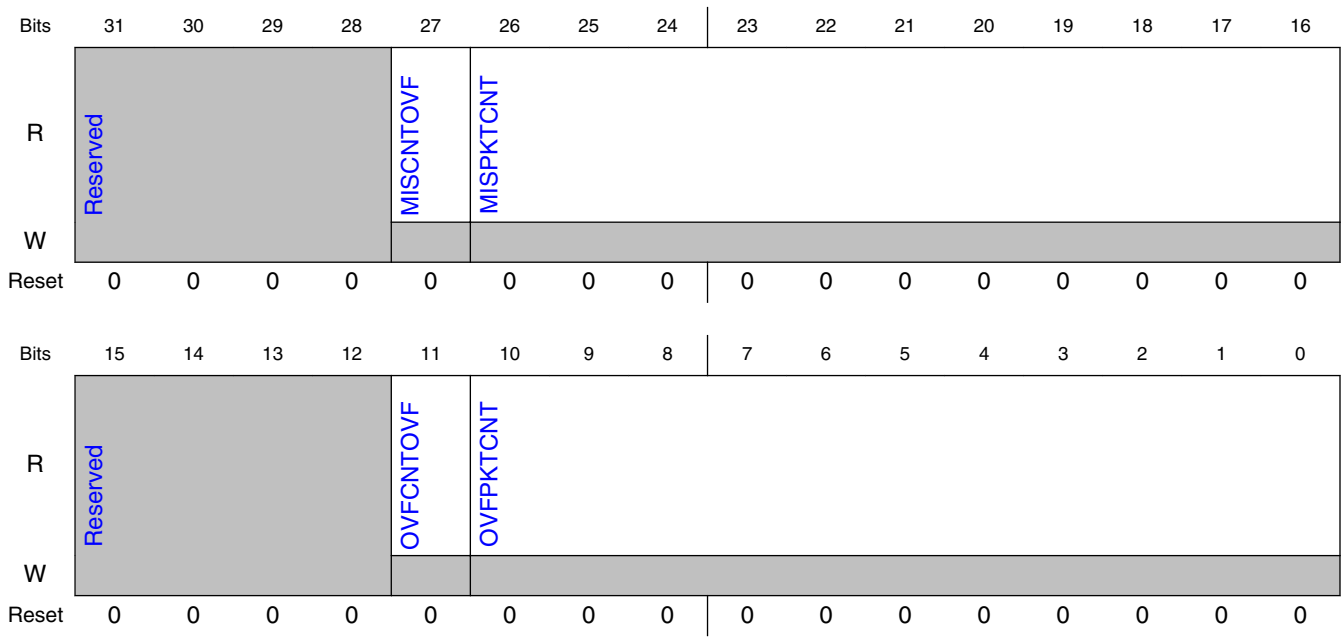
61.7.1.404.1 Offset

Register	Offset
MTL_RXQ0_MISSED_PACKET_OVERFLOW_CNT	D34h

61.7.1.404.2 Function

The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

61.7.1.404.3 Diagram



61.7.1.404.4 Fields

Field	Function
31-28 —	Reserved
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Missed Packet Counter overflow not detected 1b - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter

Table continues on the next page...

Field	Function
	This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow Counter overflow not detected 1b - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.405 Queue 0 Receive Debug (MTL_RXQ0_DEBUG)

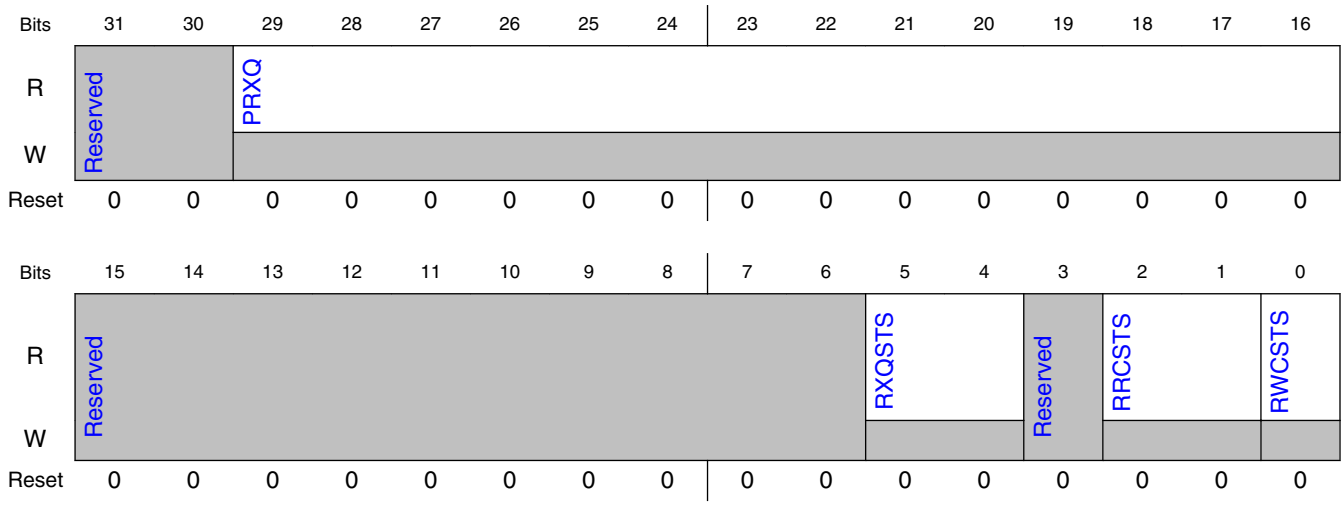
61.7.1.405.1 Offset

Register	Offset
MTL_RXQ0_DEBUG	D38h

61.7.1.405.2 Function

The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue.

61.7.1.405.3 Diagram



61.7.1.405.4 Fields

Field	Function
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00b - Rx Queue empty 01b - Rx Queue fill-level below flow-control deactivate threshold 10b - Rx Queue fill-level above flow-control activate threshold 11b - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00b - Idle state 01b - Reading packet data 10b - Reading packet status (or timestamp) 11b - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0b - MTL Rx Queue Write Controller Active Status not detected 1b - MTL Rx Queue Write Controller Active Status detected

61.7.1.406 Queue 0 Receive Control (MTL_RXQ0_CONTROL)

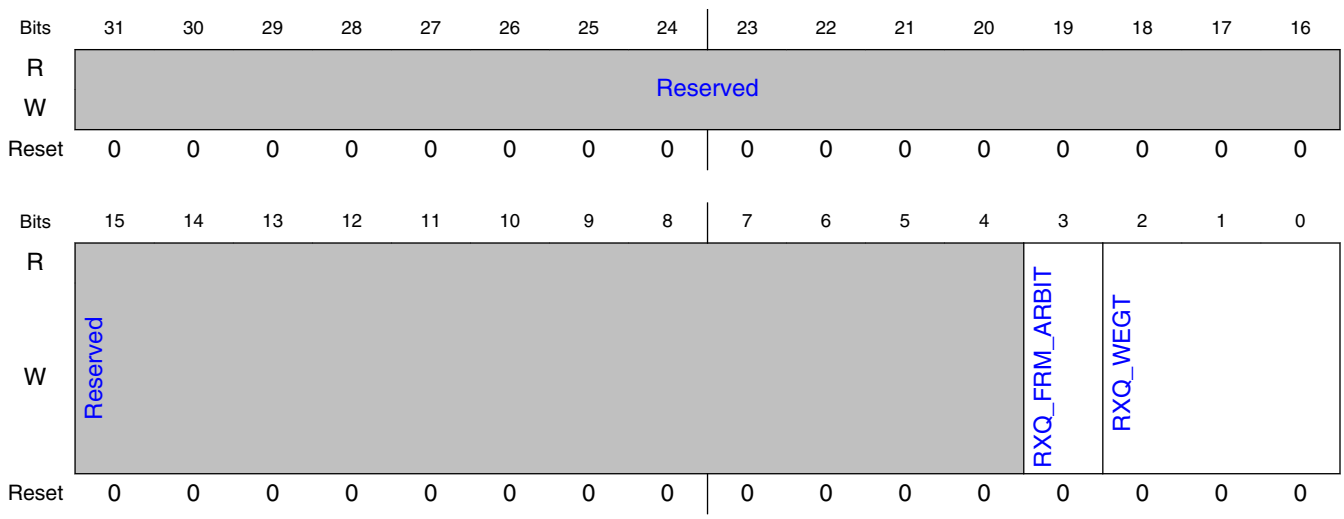
61.7.1.406.1 Offset

Register	Offset
MTL_RXQ0_CONTROL	D3Ch

61.7.1.406.2 Function

The Queue 0 Receive Control register controls the receive arbitration and passing of received packets to the application.

61.7.1.406.3 Diagram



61.7.1.406.4 Fields

Field	Function
31-4 —	Reserved
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue.

Table continues on the next page...

Field	Function
	When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0b - Receive Queue Packet Arbitration is disabled 1b - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 0. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

61.7.1.407 Queue 1 Transmit Operation Mode (MTL_TXQ1_OPERATION_MODE)

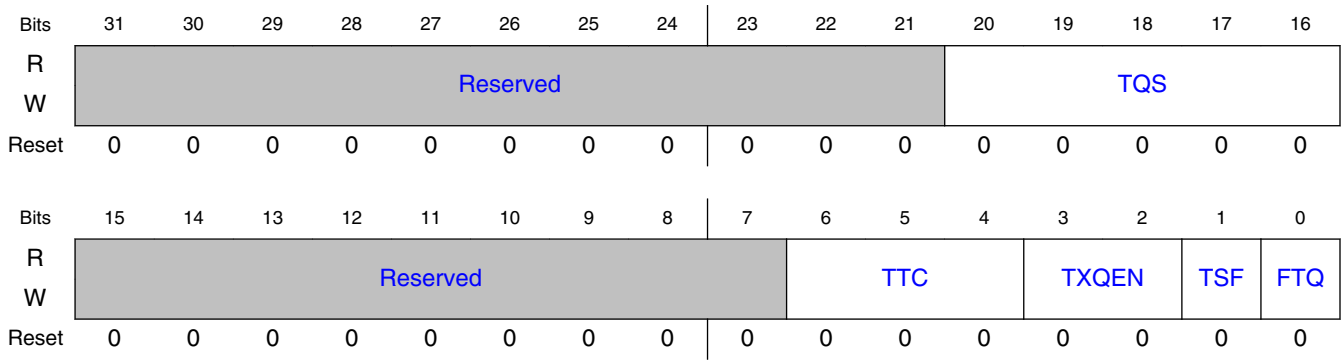
61.7.1.407.1 Offset

Register	Offset
MTL_TXQ1_OPERATION_MODE	D40h

61.7.1.407.2 Function

The Queue 1 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

61.7.1.407.3 Diagram



61.7.1.407.4 Fields

Field	Function
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000b - 32 001b - 64 010b - 96 011b - 128 100b - 192 101b - 256 110b - 384 111b - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable

Table continues on the next page...

Field	Function
	This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00b - Not enabled 01b - Enable in AV mode (Reserved in non-AV) 10b - Enabled 11b - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0b - Transmit Store and Forward is disabled 1b - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ1_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flush Transmit Queue is disabled 1b - Flush Transmit Queue is enabled

61.7.1.408 Queue 1 Underflow Counter (MTL_TXQ1_UNDERFLOW)

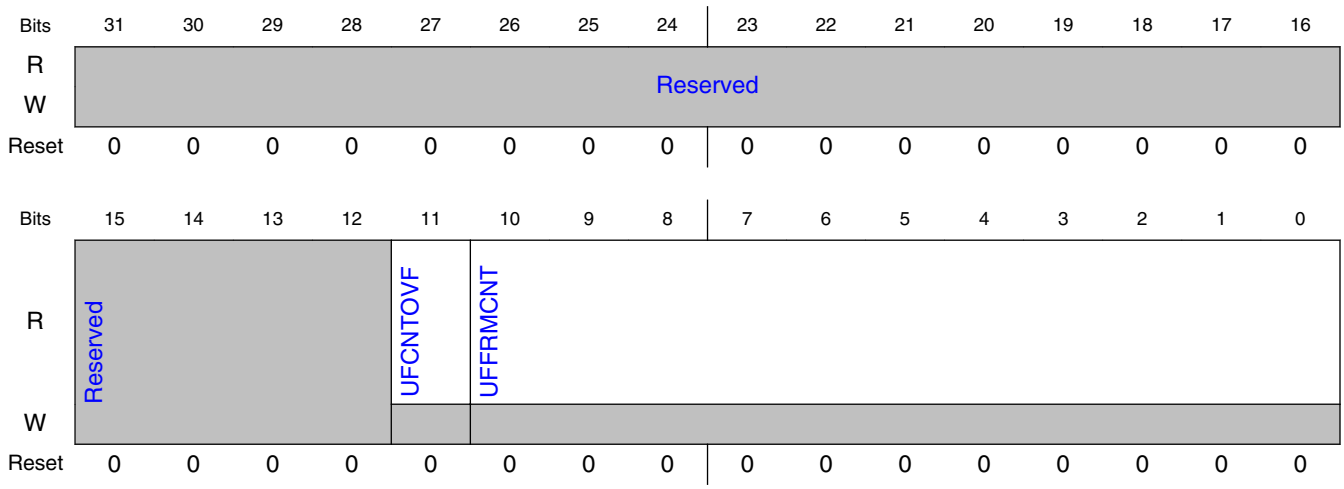
61.7.1.408.1 Offset

Register	Offset
MTL_TXQ1_UNDERFLOW	D44h

61.7.1.408.2 Function

The Queue 1 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

61.7.1.408.3 Diagram



61.7.1.408.4 Fields

Field	Function
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow not detected for Underflow Packet Counter 1b - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.409 Queue 1 Transmit Debug (MTL_TXQ1_DEBUG)

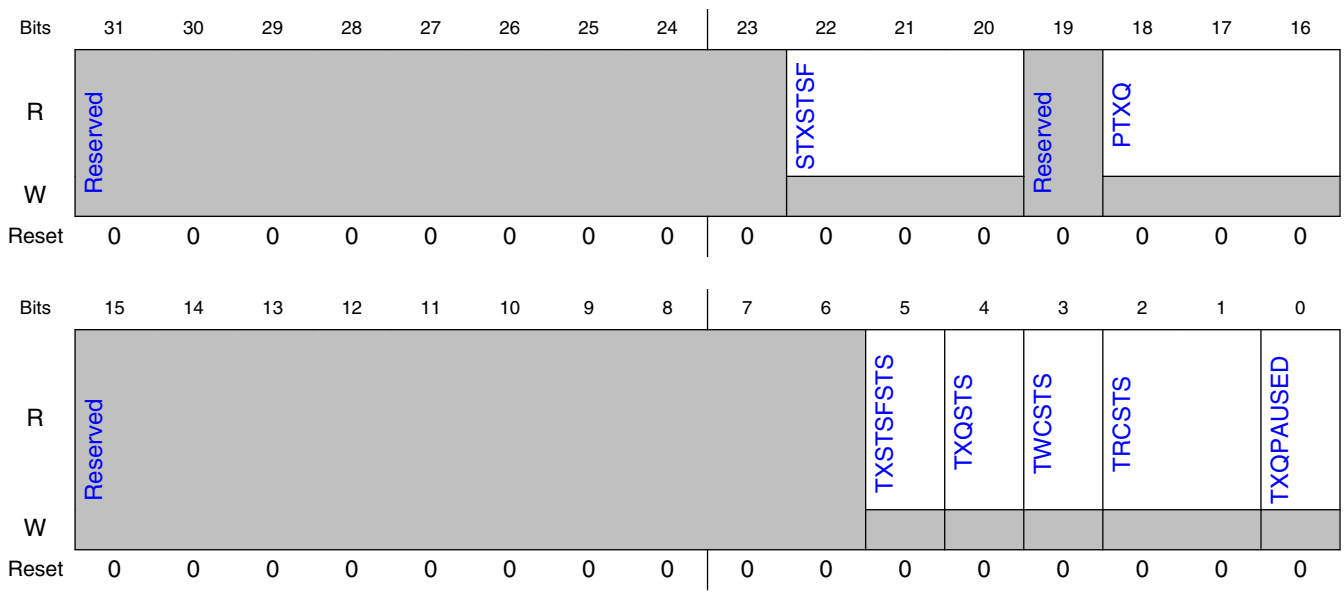
61.7.1.409.1 Offset

Register	Offset
MTL_TXQ1_DEBUG	D48h

61.7.1.409.2 Function

The Queue 1 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

61.7.1.409.3 Diagram



61.7.1.409.4 Fields

Field	Function
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue

Table continues on the next page...

Field	Function
	This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0b - MTL Tx Status FIFO Full status is not detected 1b - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0b - MTL Tx Queue Not Empty status is not detected 1b - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0b - MTL Tx Queue Write Controller status is not detected 1b - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00b - Idle state 01b - Read state (transferring data to the MAC transmitter) 10b - Waiting for pending Tx Status from the MAC transmitter 11b - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0b - Transmit Queue in Pause status is not detected 1b - Transmit Queue in Pause status is detected

61.7.1.410 Queue 1 ETS Control (MTL_TXQ1_ETS_CONTROL)

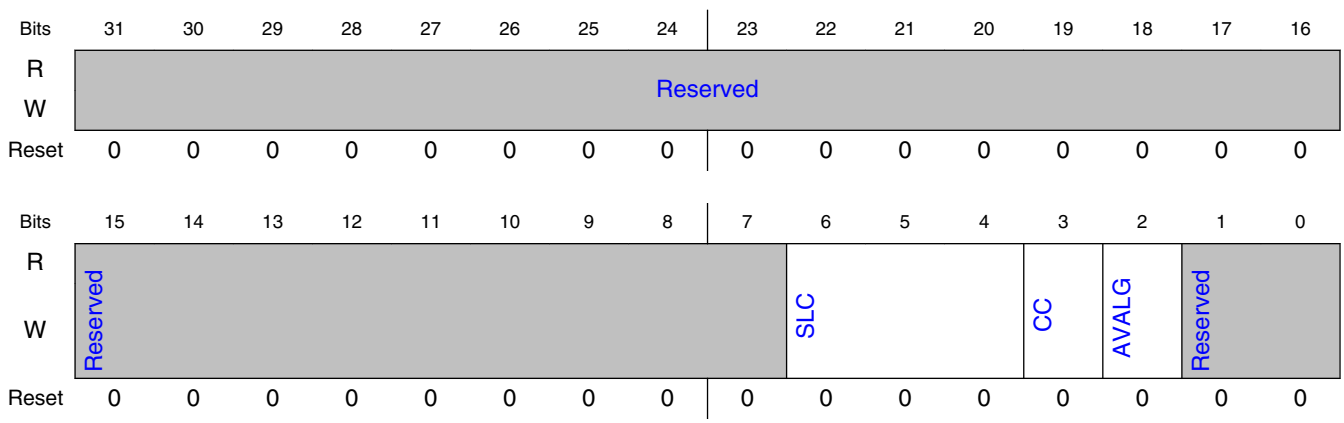
61.7.1.410.1 Offset

Register	Offset
MTL_TXQ1_ETS_CONTROL	D50h

61.7.1.410.2 Function

The Queue 1 ETS Control register controls the enhanced transmission selection operation.

61.7.1.410.3 Diagram



61.7.1.410.4 Fields

Field	Function
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH1_Slot_Interval register) over which the average transmitted bits per slot, provided in the MTL_TXQ1_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000b - 1 slot 001b - 2 slots 010b - 4 slots 011b - 8 slots 100b - 16 slots

Table continues on the next page...

Field	Function
	101b - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 1. The credit accumulates even when there is no packet waiting in Channel 1 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 1. When there is no packet waiting in Channel 1 and other channel is transmitting, no credit is accumulated. 0b - Credit Control is disabled 1b - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. AV Algorithm When Queue 1 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 1 traffic. When reset, strict priority is selected. 0b - CBS Algorithm is disabled 1b - CBS Algorithm is enabled
1-0 —	Reserved

61.7.1.411 Queue 1 ETS Status (MTL_TXQ1_ETS_STATUS)

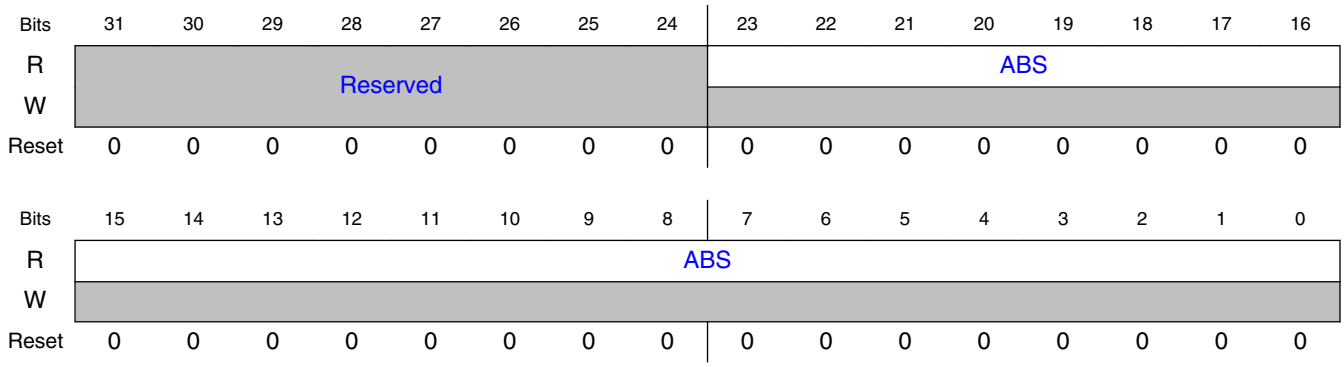
61.7.1.411.1 Offset

Register	Offset
MTL_TXQ1_ETS_STATUS	D54h

61.7.1.411.2 Function

The Queue 1 ETS Status register provides the average traffic transmitted in Queue 1.

61.7.1.411.3 Diagram



61.7.1.411.4 Fields

Field	Function
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ1_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

61.7.1.412 Queue 1 idleSlopeCredit, Quantum or Weights (MTL_TXQ1_QUANTUM_WEIGHT)

61.7.1.412.1 Offset

Register	Offset
MTL_TXQ1_QUANTUM_WEIGHT	D58h

61.7.1.412.2 Function

The Queue 1 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 1.

61.7.1.412.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.412.4 Fields

Field	Function
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register.

Field	Function
	- Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

61.7.1.413 Queue 1 sendSlopeCredit (MTL_TXQ1_SENDSLOPECREDIT)

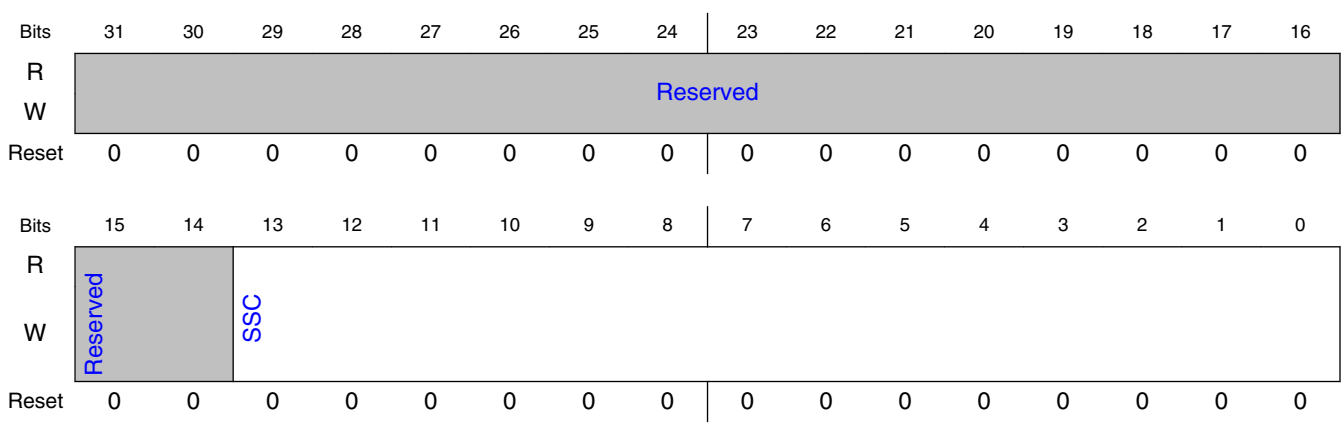
61.7.1.413.1 Offset

Register	Offset
MTL_TXQ1_SENDSLOPECREDIT	D5Ch

61.7.1.413.2 Function

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

61.7.1.413.3 Diagram



61.7.1.413.4 Fields

Field	Function
31-14	Reserved

Table continues on the next page...

Field	Function
—	
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 1 is selected for transmission.

61.7.1.414 Queue 1 hiCredit (MTL_TXQ1_HICREDIT)

61.7.1.414.1 Offset

Register	Offset
MTL_TXQ1_HICREDIT	D60h

61.7.1.414.2 Function

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.414.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			HC												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.414.4 Fields

Field	Function
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

61.7.1.415 Queue 1 loCredit (MTL_TXQ1_LOCREDIT)

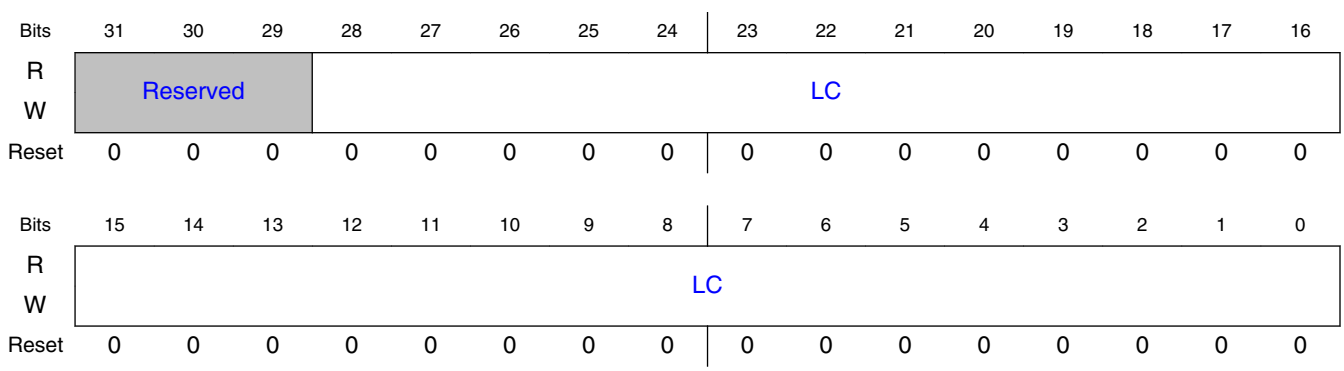
61.7.1.415.1 Offset

Register	Offset
MTL_TXQ1_LOCREDIT	D64h

61.7.1.415.2 Function

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.415.3 Diagram



61.7.1.415.4 Fields

Field	Function
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

61.7.1.416 Queue 1 Interrupt Control Status (MTL_Q1_INTERRUPT_CONTROL_STATUS)

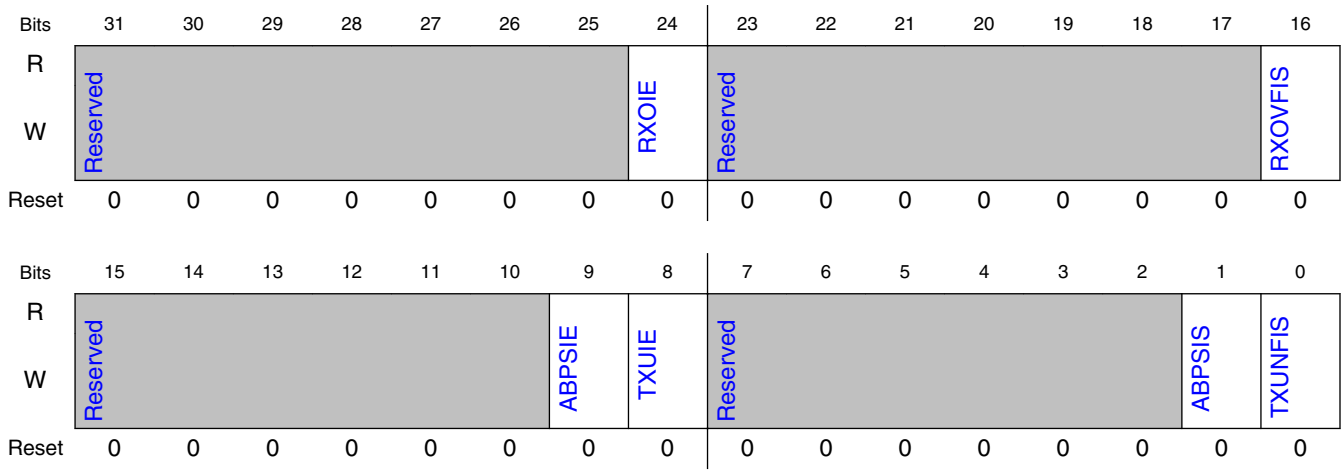
61.7.1.416.1 Offset

Register	Offset
MTL_Q1_INTERRUPT_CONTROL_STATUS	D6Ch

61.7.1.416.2 Function

This register contains the interrupt enable and status bits for the queue 1 interrupts.

61.7.1.416.3 Diagram



61.7.1.416.4 Fields

Field	Function
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0b - Receive Queue Overflow Interrupt is disabled 1b - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Queue Overflow Interrupt Status not detected 1b - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Function
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0b - Average Bits Per Slot Interrupt is disabled 1b - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0b - Transmit Queue Underflow Interrupt Status is disabled 1b - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Average Bits Per Slot Interrupt Status not detected 1b - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Queue Underflow Interrupt Status not detected 1b - Transmit Queue Underflow Interrupt Status detected

61.7.1.417 Queue 1 Receive Operation Mode (MTL_RXQ1_OPERATION_MODE)

61.7.1.417.1 Offset

Register	Offset
MTL_RXQ1_OPERATION_MODE	D70h

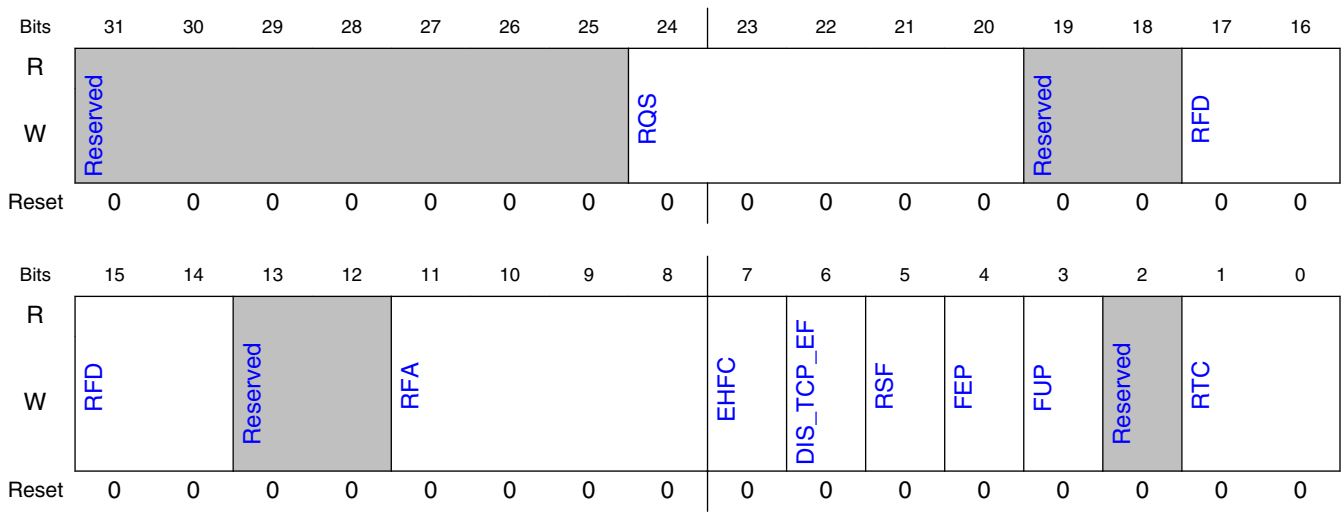
61.7.1.417.2 Function

The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

61.7.1.417.3 Diagram



61.7.1.417.4 Fields

Field	Function
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits

Table continues on the next page...

Field	Function
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0b - Hardware Flow Control is disabled 1b - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0b - Dropping of TCP/IP Checksum Error Packets is enabled

Table continues on the next page...

Field	Function
	1b - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0b - Receive Queue Store and Forward is disabled 1b - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0b - Forward Error Packets is disabled 1b - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0b - Forward Undersized Good Packets is disabled 1b - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00b - 64 01b - 32 10b - 96 11b - 128

61.7.1.418 Queue 1 Missed Packet and Overflow Counter (MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT)

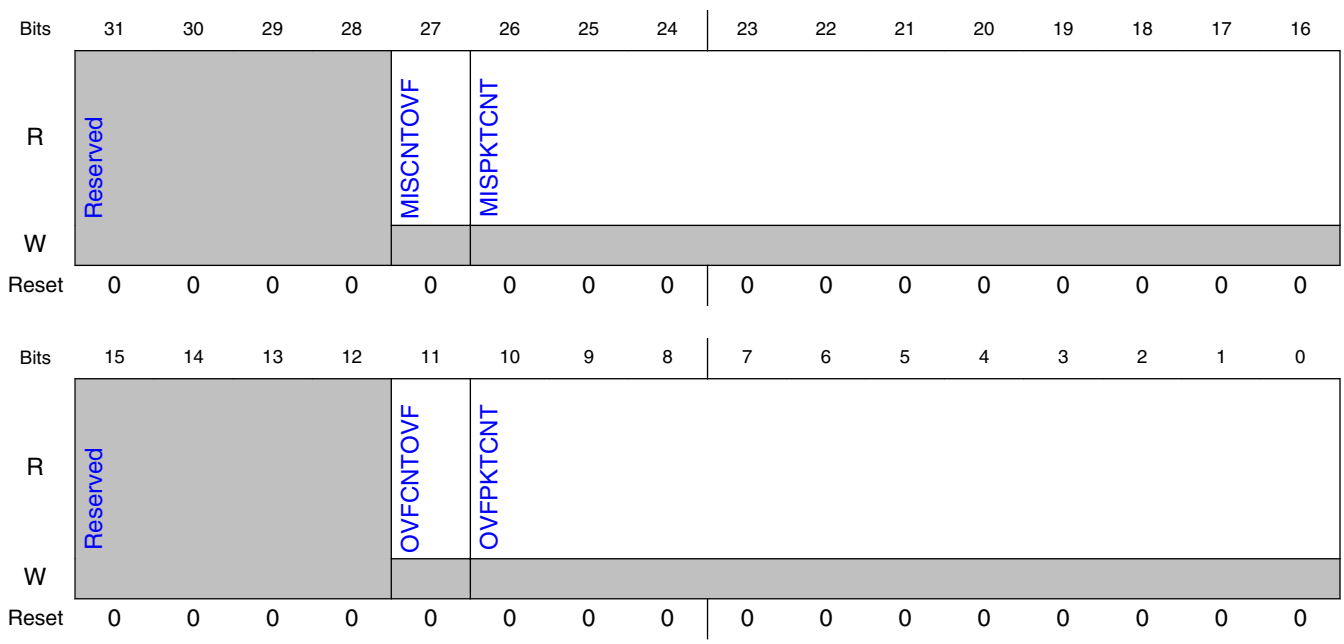
61.7.1.418.1 Offset

Register	Offset
MTL_RXQ1_MISSED_PACKET_OVERFLOW_CNT	D74h

61.7.1.418.2 Function

The Queue 1 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

61.7.1.418.3 Diagram



61.7.1.418.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Missed Packet Counter overflow not detected 1b - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow Counter overflow not detected 1b - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.419 Queue 1 Receive Debug (MTL_RXQ1_DEBUG)

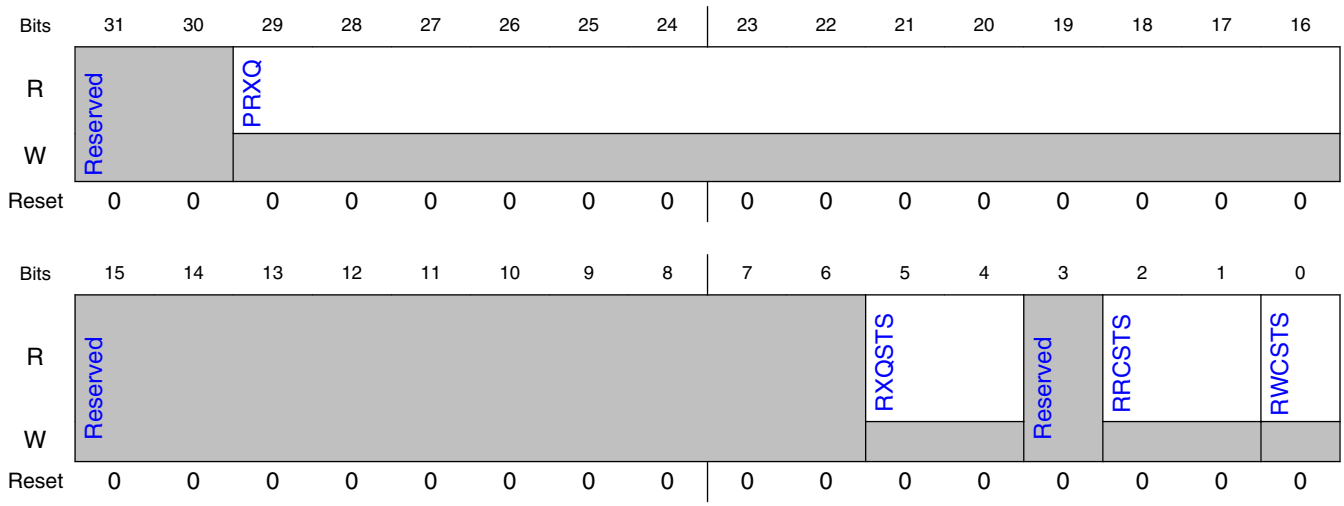
61.7.1.419.1 Offset

Register	Offset
MTL_RXQ1_DEBUG	D78h

61.7.1.419.2 Function

The Queue 1 Receive Debug register gives the debug status of various blocks related to the Receive queue.

61.7.1.419.3 Diagram



61.7.1.419.4 Fields

Field	Function
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00b - Rx Queue empty 01b - Rx Queue fill-level below flow-control deactivate threshold 10b - Rx Queue fill-level above flow-control activate threshold 11b - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00b - Idle state 01b - Reading packet data 10b - Reading packet status (or timestamp)

Table continues on the next page...

Field	Function
	11b - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0b - MTL Rx Queue Write Controller Active Status not detected 1b - MTL Rx Queue Write Controller Active Status detected

61.7.1.420 Queue 1 Receive Control (MTL_RXQ1_CONTROL)

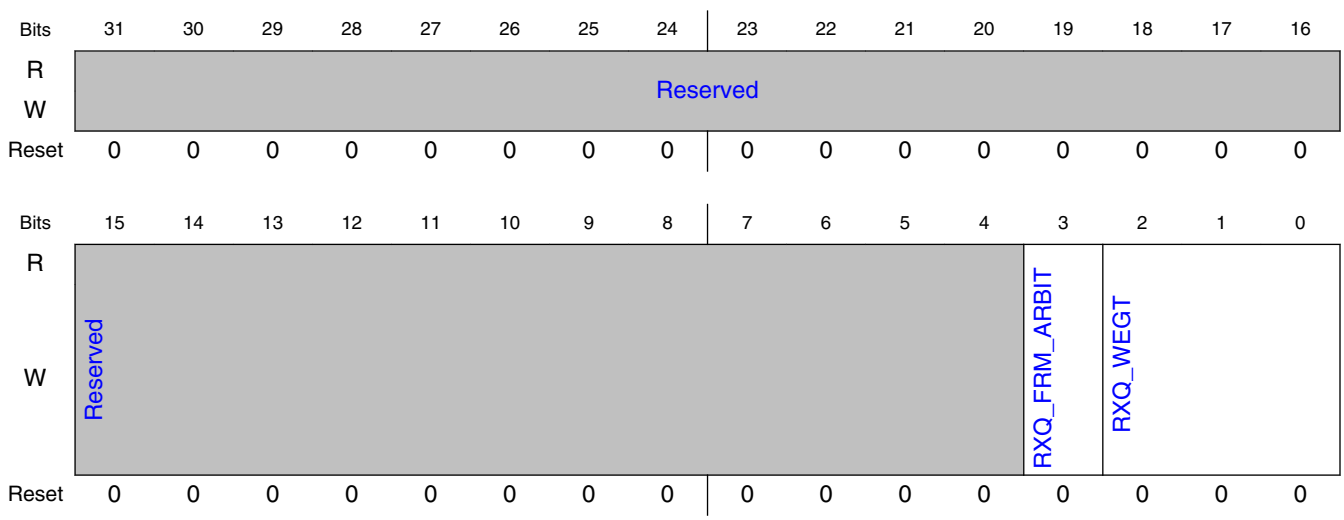
61.7.1.420.1 Offset

Register	Offset
MTL_RXQ1_CONTROL	D7Ch

61.7.1.420.2 Function

The Queue 1 Receive Control register controls the receive arbitration and passing of received packets to the application.

61.7.1.420.3 Diagram



61.7.1.420.4 Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Field	Function
—	
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0b - Receive Queue Packet Arbitration is disabled 1b - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 1. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 1. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

61.7.1.421 Queue 2 Transmit Operation Mode (MTL_TXQ2_OPERATION_MODE)

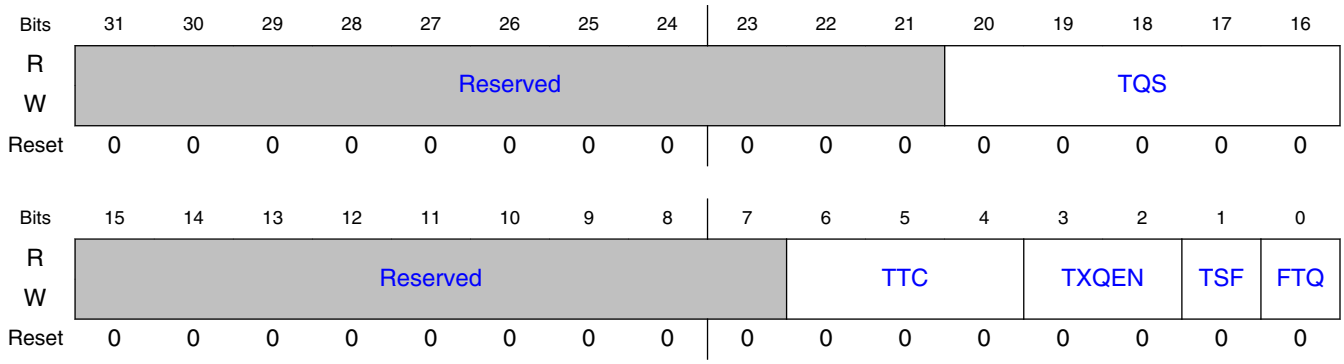
61.7.1.421.1 Offset

Register	Offset
MTL_TXQ2_OPERATION_MODE	D80h

61.7.1.421.2 Function

The Queue 2 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

61.7.1.421.3 Diagram



61.7.1.421.4 Fields

Field	Function
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000b - 32 001b - 64 010b - 96 011b - 128 100b - 192 101b - 256 110b - 384 111b - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable

Table continues on the next page...

Field	Function
	This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00b - Not enabled 01b - Enable in AV mode (Reserved in non-AV) 10b - Enabled 11b - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0b - Transmit Store and Forward is disabled 1b - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ2_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flush Transmit Queue is disabled 1b - Flush Transmit Queue is enabled

61.7.1.422 Queue 2 Underflow Counter (MTL_TXQ2_UNDERFLOW)

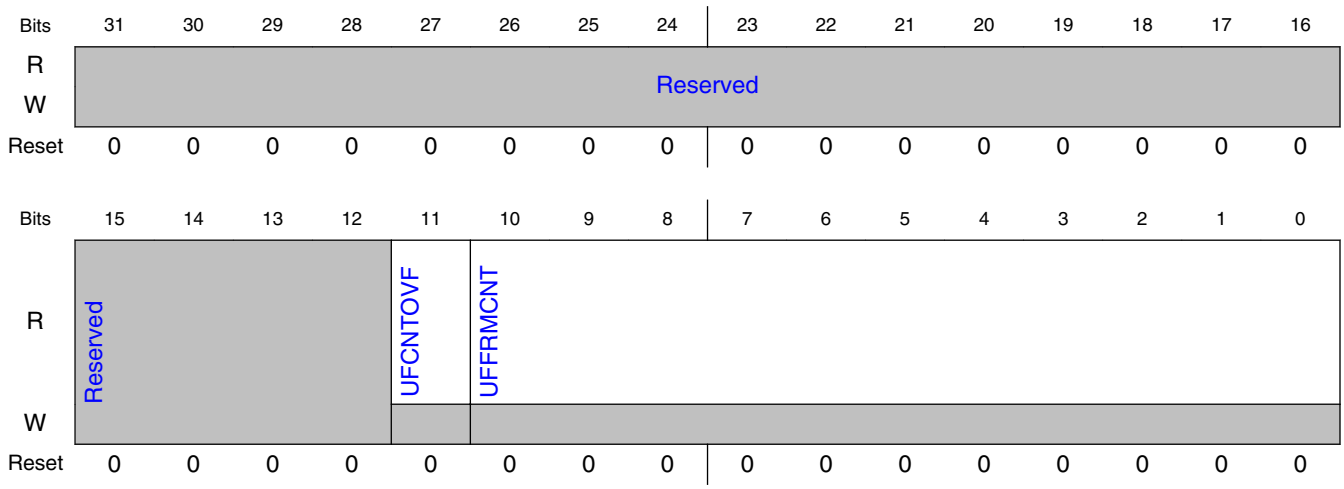
61.7.1.422.1 Offset

Register	Offset
MTL_TXQ2_UNDERFLOW	D84h

61.7.1.422.2 Function

The Queue 2 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

61.7.1.422.3 Diagram



61.7.1.422.4 Fields

Field	Function
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow not detected for Underflow Packet Counter 1b - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.423 Queue 2 Transmit Debug (MTL_TXQ2_DEBUG)

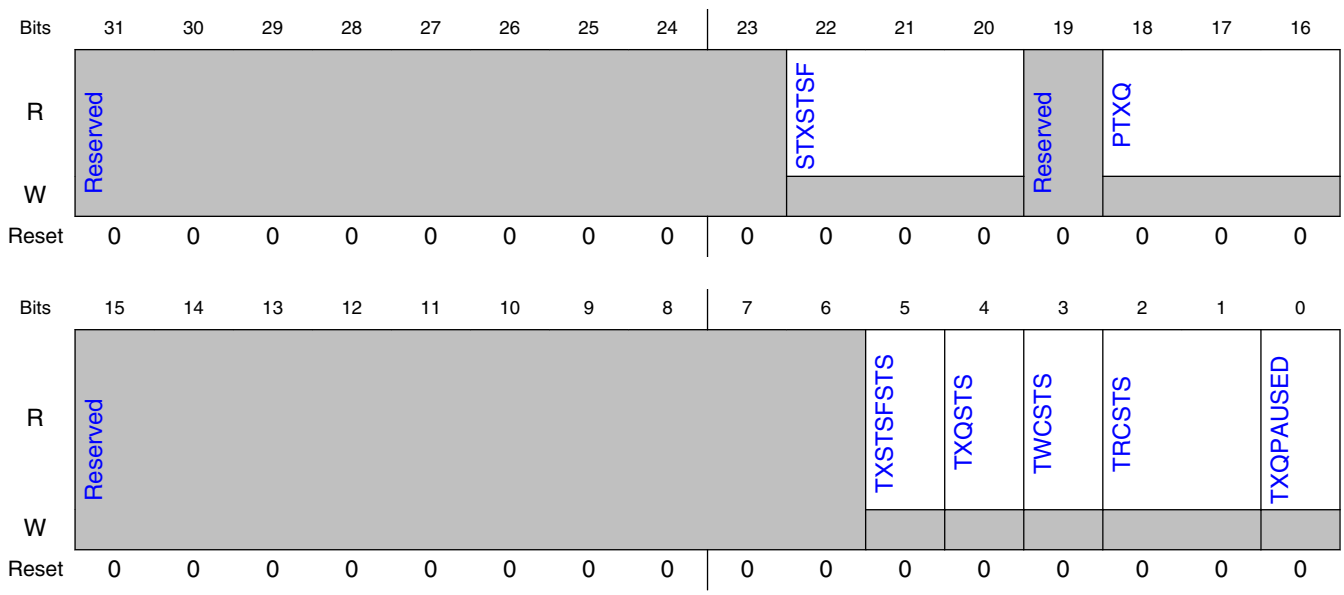
61.7.1.423.1 Offset

Register	Offset
MTL_TXQ2_DEBUG	D88h

61.7.1.423.2 Function

The Queue 2 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

61.7.1.423.3 Diagram



61.7.1.423.4 Fields

Field	Function
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue

Table continues on the next page...

Field	Function
	This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0b - MTL Tx Status FIFO Full status is not detected 1b - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0b - MTL Tx Queue Not Empty status is not detected 1b - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0b - MTL Tx Queue Write Controller status is not detected 1b - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00b - Idle state 01b - Read state (transferring data to the MAC transmitter) 10b - Waiting for pending Tx Status from the MAC transmitter 11b - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0b - Transmit Queue in Pause status is not detected 1b - Transmit Queue in Pause status is detected

61.7.1.424 Queue 2 ETS Control (MTL_TXQ2_ETS_CONTROL)

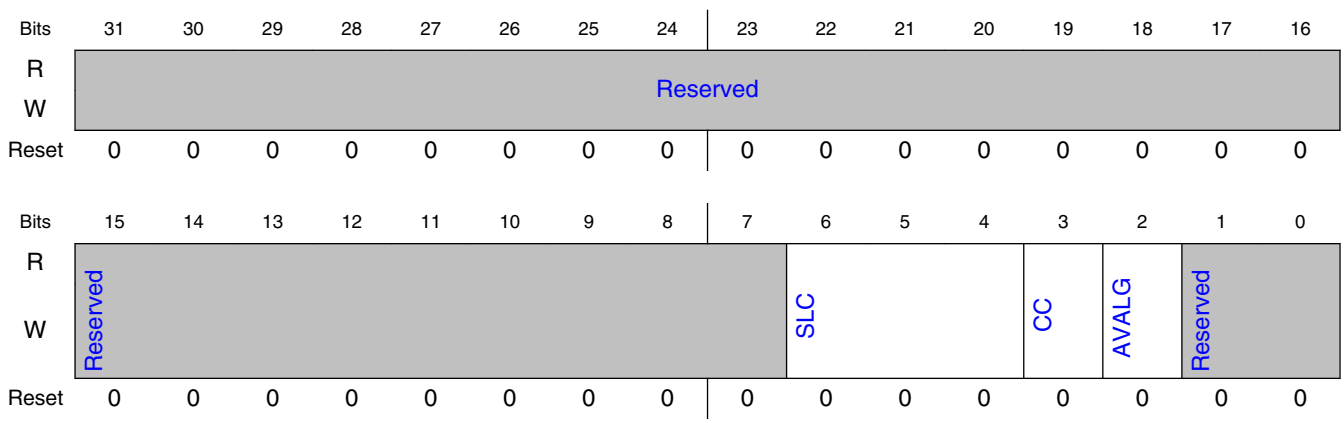
61.7.1.424.1 Offset

Register	Offset
MTL_TXQ2_ETS_CONTROL	D90h

61.7.1.424.2 Function

The Queue 2 ETS Control register controls the enhanced transmission selection operation.

61.7.1.424.3 Diagram



61.7.1.424.4 Fields

Field	Function
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH2_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ2_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000b - 1 slot 001b - 2 slots 010b - 4 slots 011b - 8 slots 100b - 16 slots

Table continues on the next page...

Field	Function
	101b - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 2. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 2. The credit accumulates even when there is no packet waiting in Channel 2 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 2. When there is no packet waiting in Channel 2 and other channel is transmitting, no credit is accumulated. 0b - Credit Control is disabled 1b - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 2 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 2 traffic. AV Algorithm When Queue 2 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 2 traffic. When reset, strict priority is selected. 0b - CBS Algorithm is disabled 1b - CBS Algorithm is enabled
1-0 —	Reserved

61.7.1.425 Queue 2 ETS Status (MTL_TXQ2_ETS_STATUS)

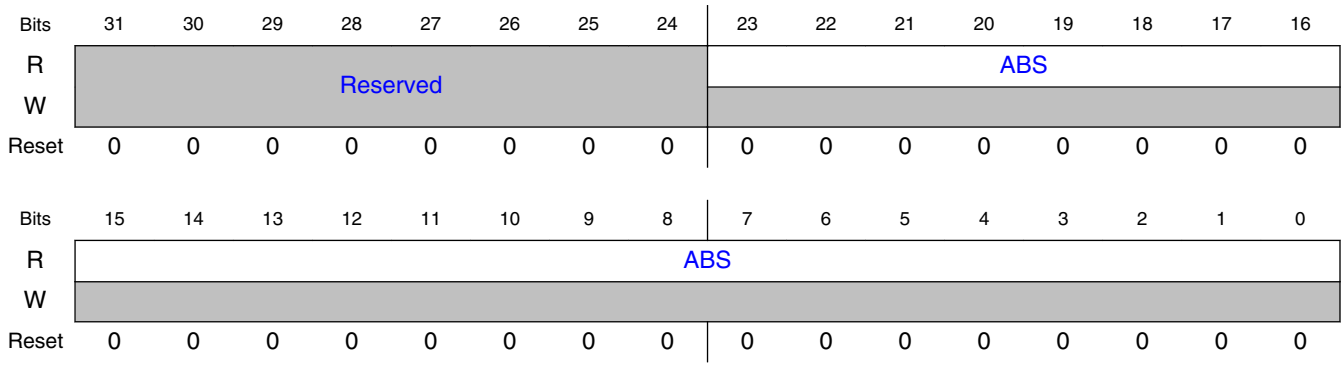
61.7.1.425.1 Offset

Register	Offset
MTL_TXQ2_ETS_STATUS	D94h

61.7.1.425.2 Function

The Queue 2 ETS Status register provides the average traffic transmitted in Queue 2.

61.7.1.425.3 Diagram



61.7.1.425.4 Fields

Field	Function
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ2_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

61.7.1.426 Queue 2 idleSlopeCredit, Quantum or Weights (MTL_TXQ2_QUANTUM_WEIGHT)

61.7.1.426.1 Offset

Register	Offset
MTL_TXQ2_QUANTUM_WEIGHT	D98h

61.7.1.426.2 Function

The Queue 2 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 2.

61.7.1.426.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.426.4 Fields

Field	Function
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 2. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 2. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 2 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 2 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 2 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_OPERATION_MODE register.

Field	Function
	- Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

61.7.1.427 Queue 2 sendSlopeCredit (MTL_TXQ2_SENDSLOPECREDIT)

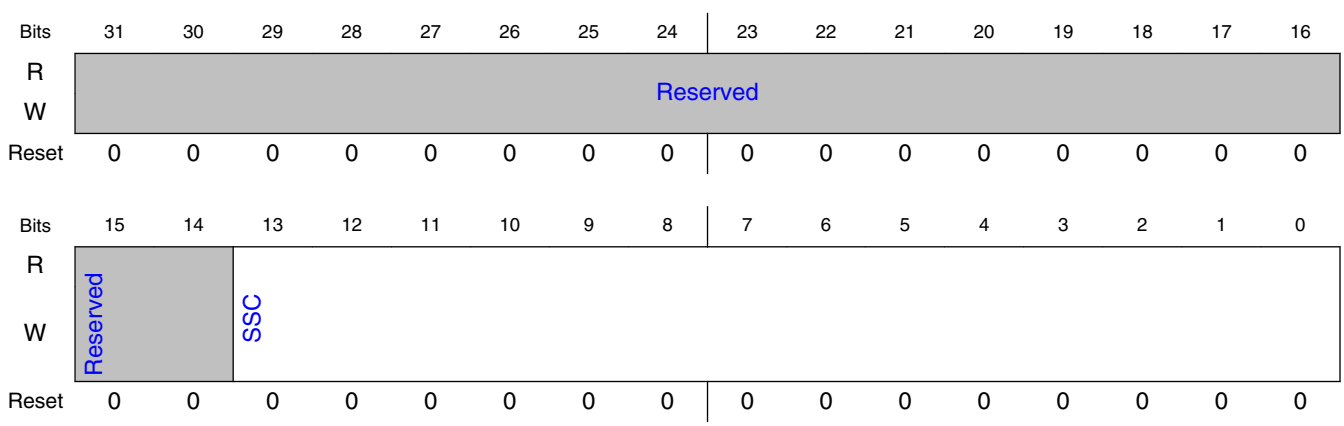
61.7.1.427.1 Offset

Register	Offset
MTL_TXQ2_SENDSLOPECREDIT	D9Ch

61.7.1.427.2 Function

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

61.7.1.427.3 Diagram



61.7.1.427.4 Fields

Field	Function
31-14	Reserved

Table continues on the next page...

Field	Function
—	
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 2. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 2. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 2 is selected for transmission.

61.7.1.428 Queue 2 hiCredit (MTL_TXQ2_HICREDIT)

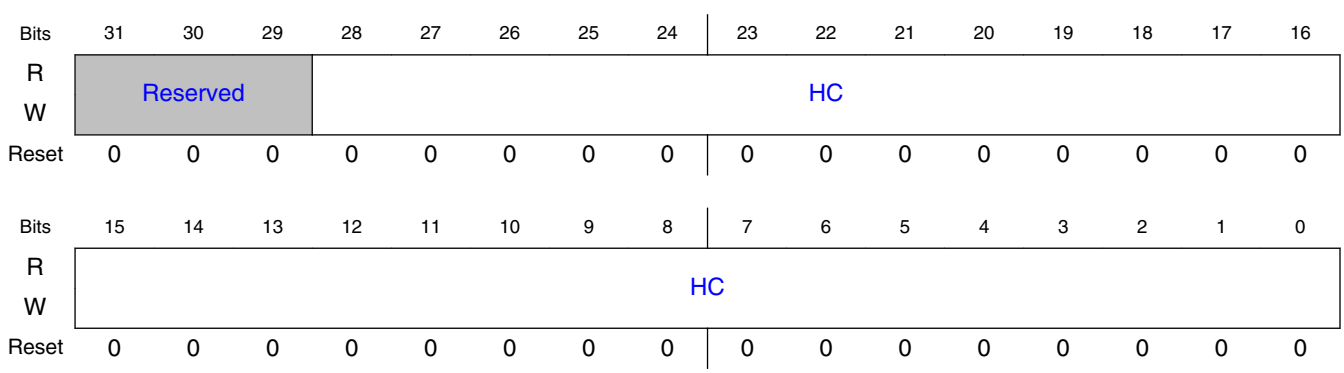
61.7.1.428.1 Offset

Register	Offset
MTL_TXQ2_HICREDIT	DA0h

61.7.1.428.2 Function

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.428.3 Diagram



61.7.1.428.4 Fields

Field	Function
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

61.7.1.429 Queue 2 loCredit (MTL_TXQ2_LOCREDIT)

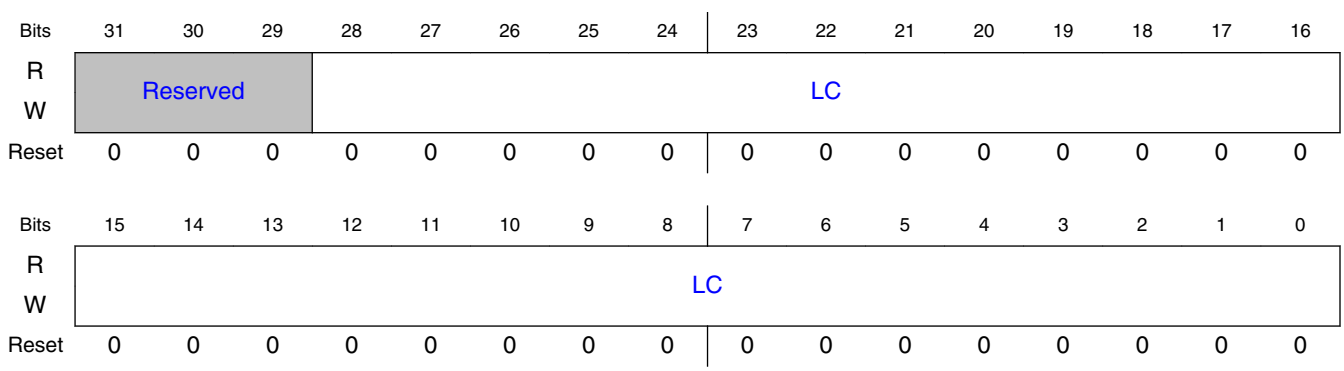
61.7.1.429.1 Offset

Register	Offset
MTL_TXQ2_LOCREDIT	DA4h

61.7.1.429.2 Function

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.429.3 Diagram



61.7.1.429.4 Fields

Field	Function
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

61.7.1.430 Queue 2 Interrupt Control Status (MTL_Q2_INTERRUPT_CONTROL_STATUS)

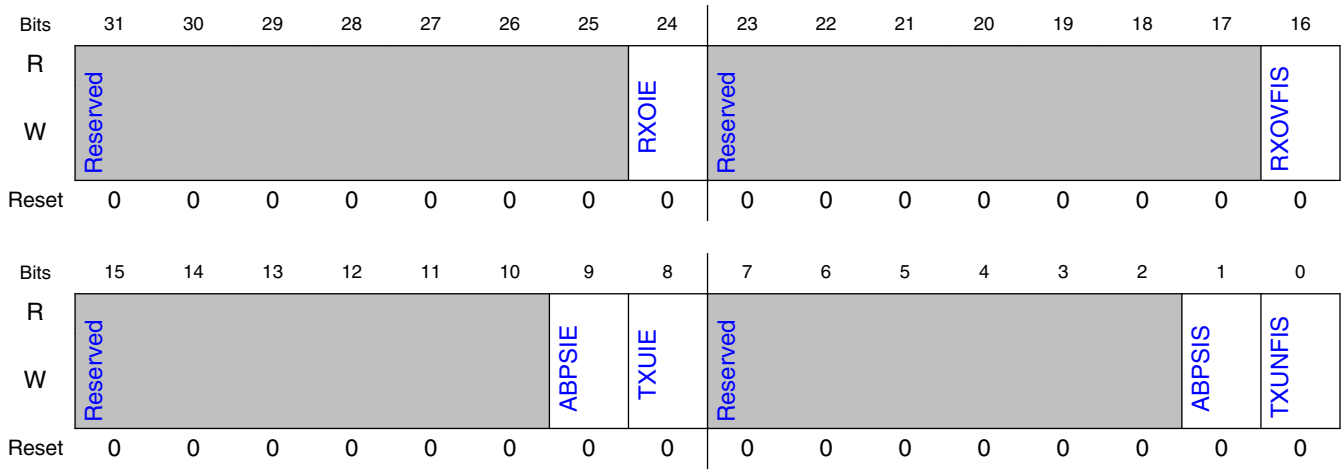
61.7.1.430.1 Offset

Register	Offset
MTL_Q2_INTERRUPT_CONTROL_STATUS	DACh

61.7.1.430.2 Function

This register contains the interrupt enable and status bits for the queue 2 interrupts.

61.7.1.430.3 Diagram



61.7.1.430.4 Fields

Field	Function
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0b - Receive Queue Overflow Interrupt is disabled 1b - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Queue Overflow Interrupt Status not detected 1b - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Function
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0b - Average Bits Per Slot Interrupt is disabled 1b - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0b - Transmit Queue Underflow Interrupt Status is disabled 1b - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Average Bits Per Slot Interrupt Status not detected 1b - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Queue Underflow Interrupt Status not detected 1b - Transmit Queue Underflow Interrupt Status detected

61.7.1.431 Queue 2 Receive Operation Mode (MTL_RXQ2_OPERATION_MODE)

61.7.1.431.1 Offset

Register	Offset
MTL_RXQ2_OPERATION_MODE	DB0h

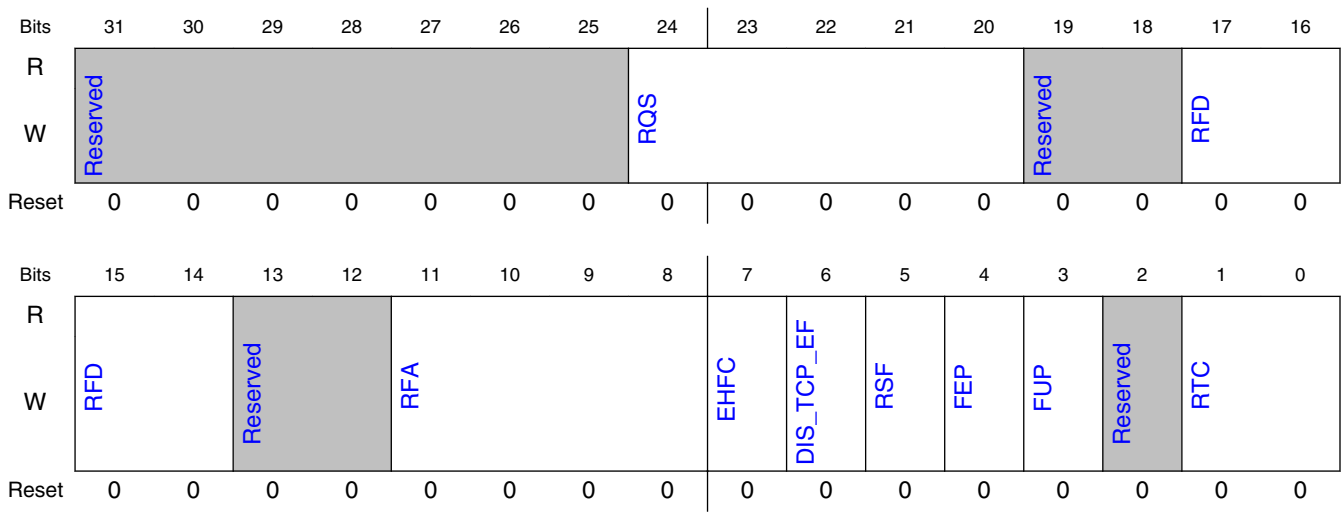
61.7.1.431.2 Function

The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

The Queue 2 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

61.7.1.431.3 Diagram



61.7.1.431.4 Fields

Field	Function
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits

Table continues on the next page...

Field	Function
19-18 —	Reserved
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0b - Hardware Flow Control is disabled 1b - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0b - Dropping of TCP/IP Checksum Error Packets is enabled

Table continues on the next page...

Field	Function
	1b - Dropping of TCP/IP Checksum Error Packets is disabled
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0b - Receive Queue Store and Forward is disabled 1b - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0b - Forward Error Packets is disabled 1b - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0b - Forward Undersized Good Packets is disabled 1b - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00b - 64 01b - 32 10b - 96 11b - 128

61.7.1.432 Queue 2 Missed Packet and Overflow Counter (MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT)

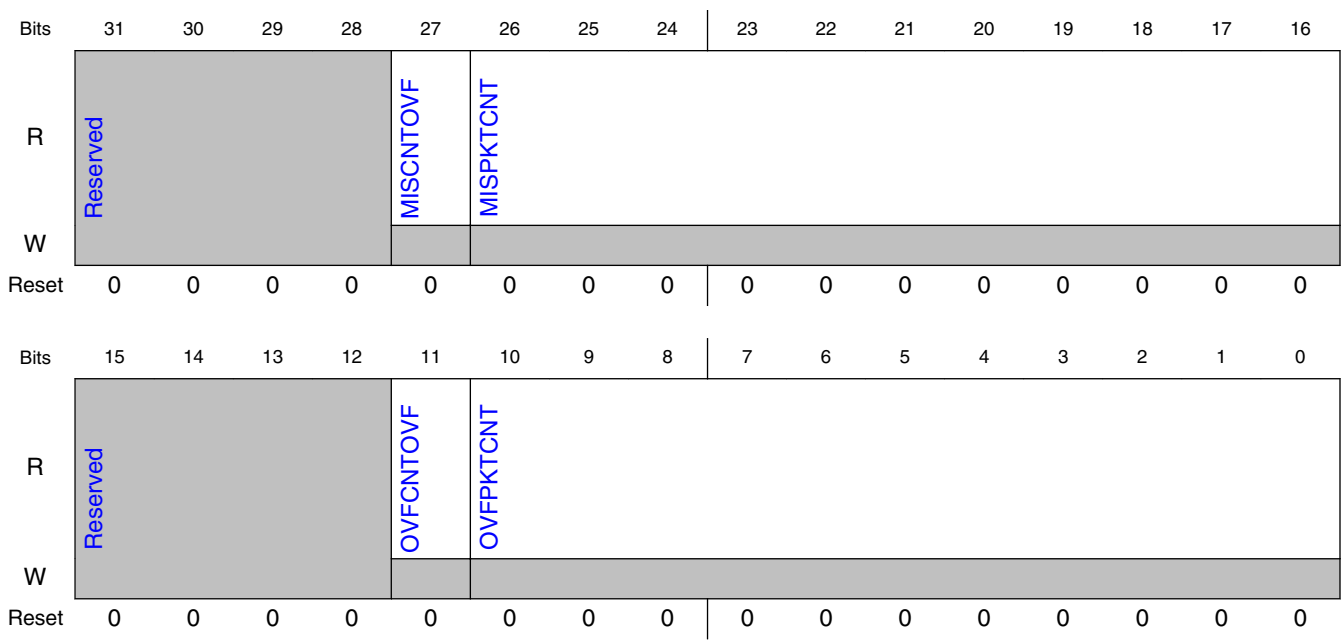
61.7.1.432.1 Offset

Register	Offset
MTL_RXQ2_MISSED_PACKET_OVERFLOW_CNT	DB4h

61.7.1.432.2 Function

The Queue 2 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

61.7.1.432.3 Diagram



61.7.1.432.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Missed Packet Counter overflow not detected 1b - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow Counter overflow not detected 1b - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.433 Queue 2 Receive Debug (MTL_RXQ2_DEBUG)

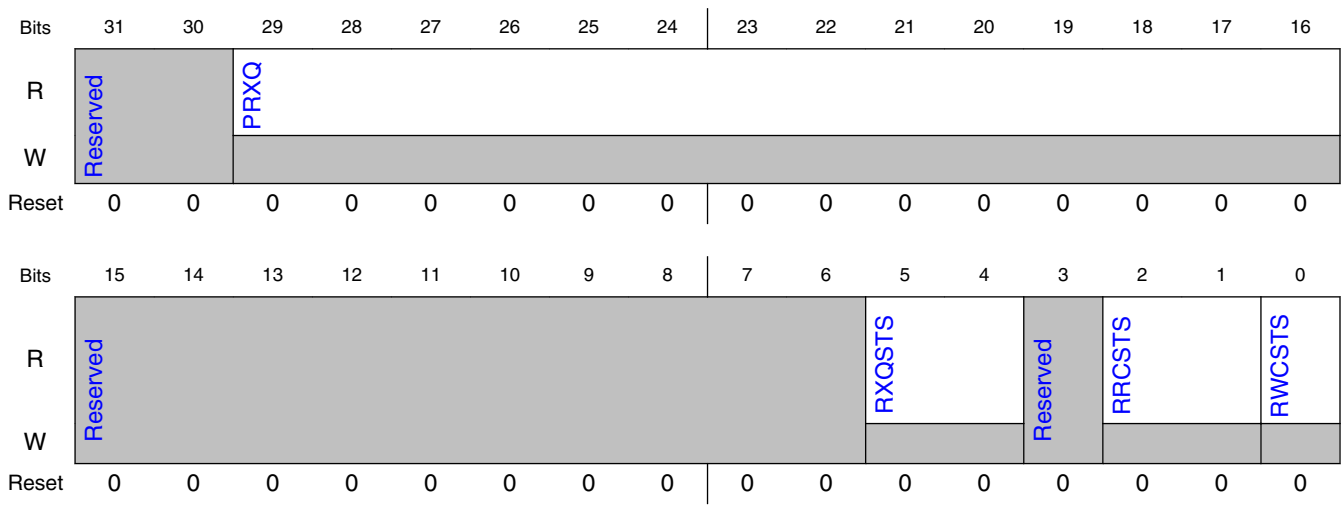
61.7.1.433.1 Offset

Register	Offset
MTL_RXQ2_DEBUG	DB8h

61.7.1.433.2 Function

The Queue 2 Receive Debug register gives the debug status of various blocks related to the Receive queue.

61.7.1.433.3 Diagram



61.7.1.433.4 Fields

Field	Function
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00b - Rx Queue empty 01b - Rx Queue fill-level below flow-control deactivate threshold 10b - Rx Queue fill-level above flow-control activate threshold 11b - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00b - Idle state 01b - Reading packet data 10b - Reading packet status (or timestamp)

Table continues on the next page...

Field	Function
	11b - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0b - MTL Rx Queue Write Controller Active Status not detected 1b - MTL Rx Queue Write Controller Active Status detected

61.7.1.434 Queue 2 Receive Control (MTL_RXQ2_CONTROL)

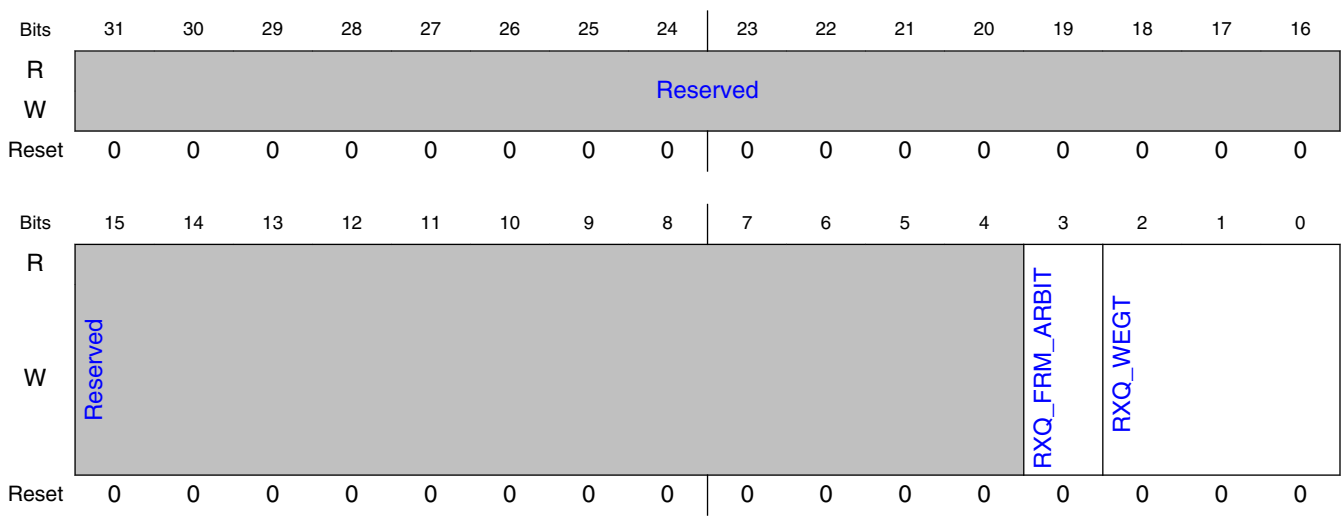
61.7.1.434.1 Offset

Register	Offset
MTL_RXQ2_CONTROL	DBCh

61.7.1.434.2 Function

The Queue 2 Receive Control register controls the receive arbitration and passing of received packets to the application.

61.7.1.434.3 Diagram



61.7.1.434.4 Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Field	Function
—	
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0b - Receive Queue Packet Arbitration is disabled 1b - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 2. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 2. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

61.7.1.435 Queue 3 Transmit Operation Mode (MTL_TXQ3_OPERATION_MODE)

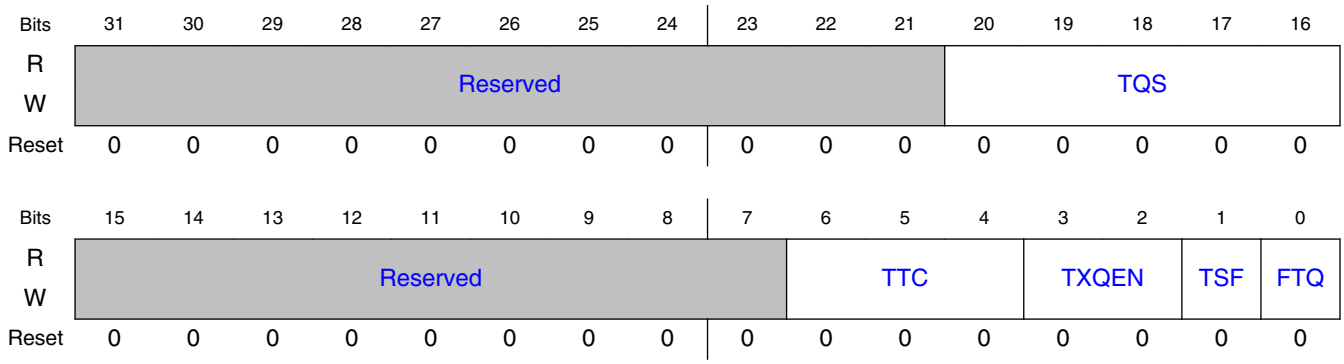
61.7.1.435.1 Offset

Register	Offset
MTL_TXQ3_OPERATION_MODE	DC0h

61.7.1.435.2 Function

The Queue 3 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

61.7.1.435.3 Diagram



61.7.1.435.4 Fields

Field	Function
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000b - 32 001b - 64 010b - 96 011b - 128 100b - 192 101b - 256 110b - 384 111b - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable

Table continues on the next page...

Field	Function
	This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00b - Not enabled 01b - Enable in AV mode (Reserved in non-AV) 10b - Enabled 11b - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0b - Transmit Store and Forward is disabled 1b - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ3_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flush Transmit Queue is disabled 1b - Flush Transmit Queue is enabled

61.7.1.436 Queue 3 Underflow Counter (MTL_TXQ3_UNDERFLOW)

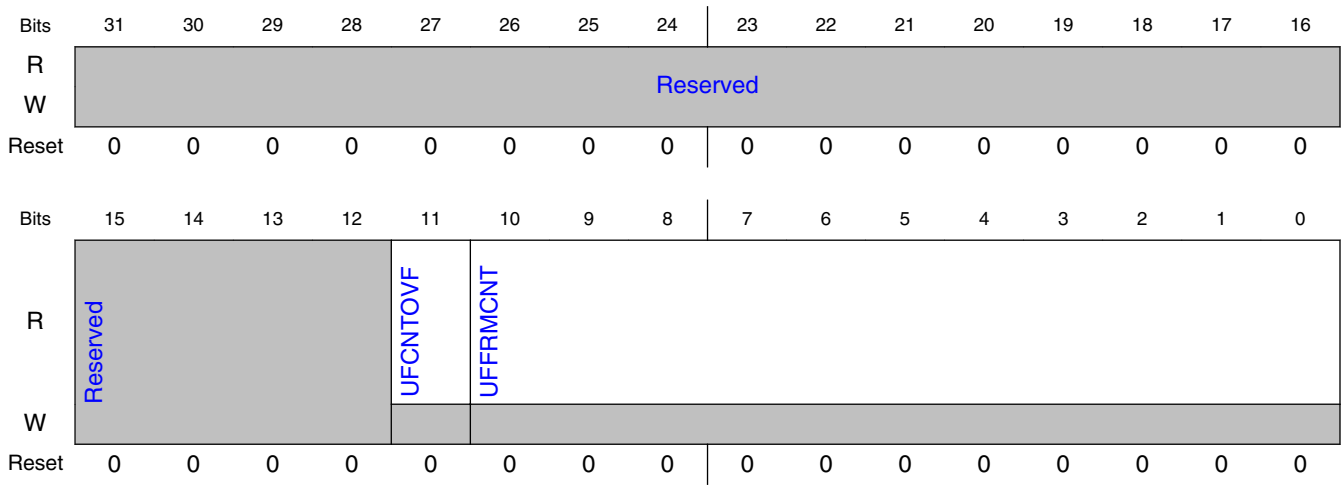
61.7.1.436.1 Offset

Register	Offset
MTL_TXQ3_UNDERFLOW	DC4h

61.7.1.436.2 Function

The Queue 3 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

61.7.1.436.3 Diagram



61.7.1.436.4 Fields

Field	Function
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow not detected for Underflow Packet Counter 1b - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.437 Queue 3 Transmit Debug (MTL_TXQ3_DEBUG)

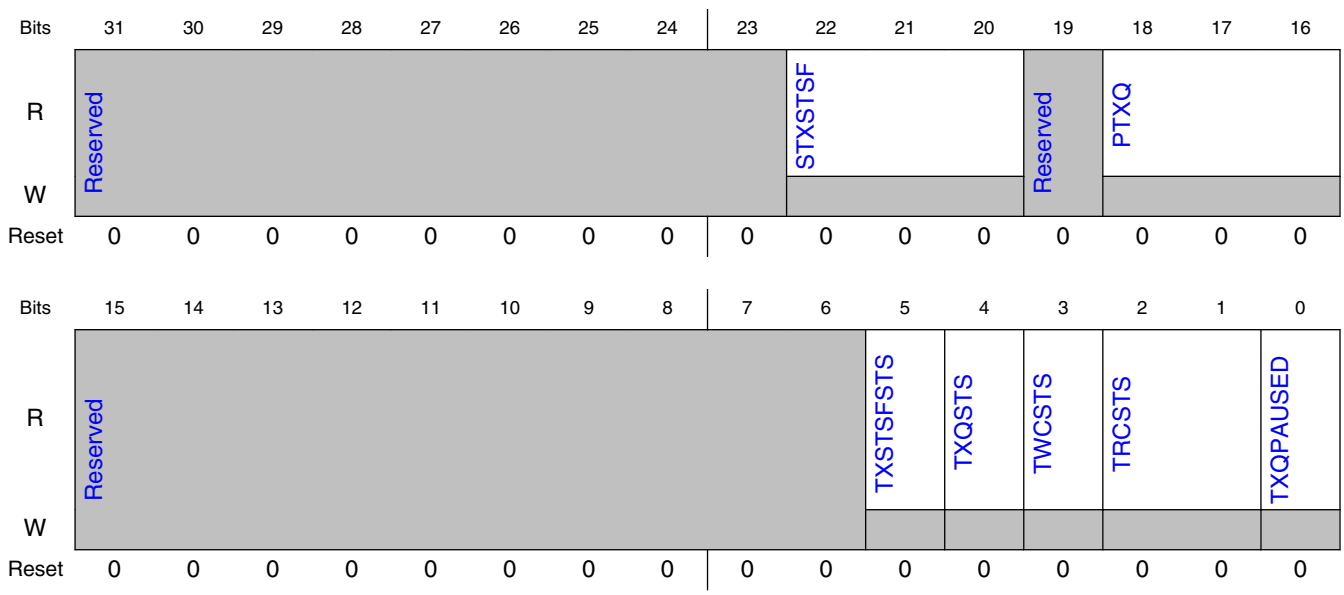
61.7.1.437.1 Offset

Register	Offset
MTL_TXQ3_DEBUG	DC8h

61.7.1.437.2 Function

The Queue 3 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

61.7.1.437.3 Diagram



61.7.1.437.4 Fields

Field	Function
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue

Table continues on the next page...

Field	Function
	This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0b - MTL Tx Status FIFO Full status is not detected 1b - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0b - MTL Tx Queue Not Empty status is not detected 1b - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0b - MTL Tx Queue Write Controller status is not detected 1b - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00b - Idle state 01b - Read state (transferring data to the MAC transmitter) 10b - Waiting for pending Tx Status from the MAC transmitter 11b - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0b - Transmit Queue in Pause status is not detected 1b - Transmit Queue in Pause status is detected

61.7.1.438 Queue 3 ETS Control (MTL_TXQ3_ETS_CONTROL)

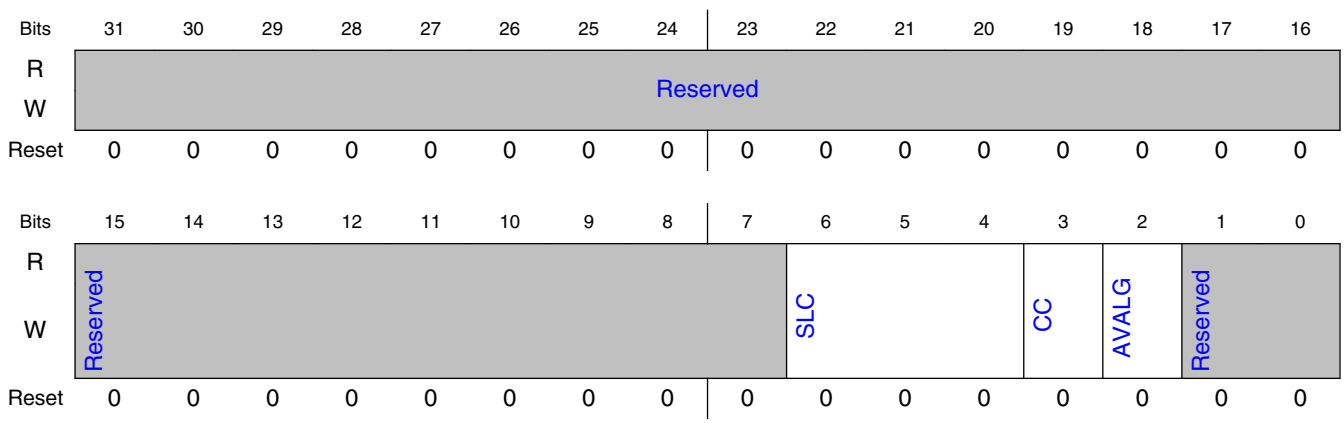
61.7.1.438.1 Offset

Register	Offset
MTL_TXQ3_ETS_CONTROL	DD0h

61.7.1.438.2 Function

The Queue 3 ETS Control register controls the enhanced transmission selection operation.

61.7.1.438.3 Diagram



61.7.1.438.4 Fields

Field	Function
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH3_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ3_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000b - 1 slot 001b - 2 slots 010b - 4 slots 011b - 8 slots 100b - 16 slots

Table continues on the next page...

Field	Function
	101b - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 3. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 3. The credit accumulates even when there is no packet waiting in Channel 3 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 3. When there is no packet waiting in Channel 3 and other channel is transmitting, no credit is accumulated. 0b - Credit Control is disabled 1b - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 3 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 3 traffic. AV Algorithm When Queue 3 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 3 traffic. When reset, strict priority is selected. 0b - CBS Algorithm is disabled 1b - CBS Algorithm is enabled
1-0 —	Reserved

61.7.1.439 Queue 3 ETS Status (MTL_TXQ3_ETS_STATUS)

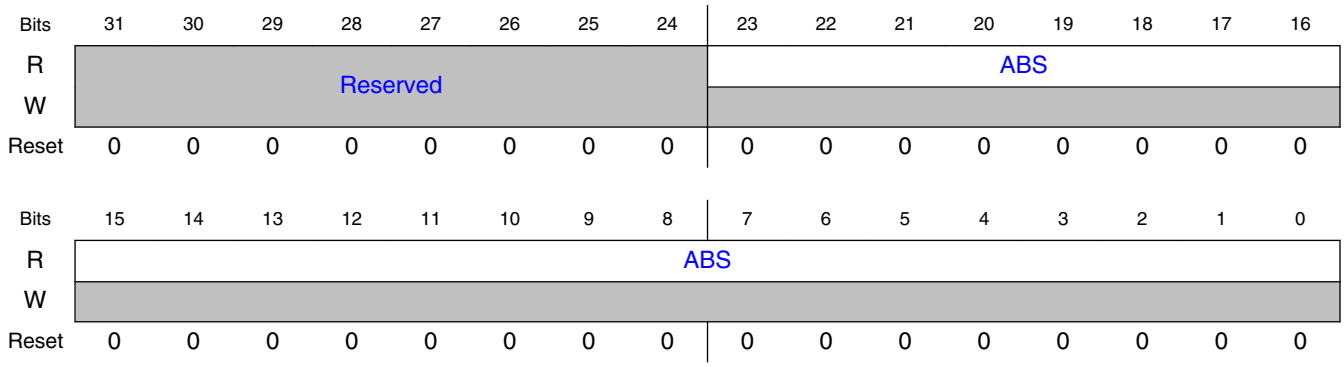
61.7.1.439.1 Offset

Register	Offset
MTL_TXQ3_ETS_STATUS	DD4h

61.7.1.439.2 Function

The Queue 3 ETS Status register provides the average traffic transmitted in Queue 3.

61.7.1.439.3 Diagram



61.7.1.439.4 Fields

Field	Function
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ3_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

61.7.1.440 Queue 3 idleSlopeCredit, Quantum or Weights (MTL_TXQ3_QUANTUM_WEIGHT)

61.7.1.440.1 Offset

Register	Offset
MTL_TXQ3_QUANTUM_WEIGHT	DD8h

61.7.1.440.2 Function

The Queue 3 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 3.

61.7.1.440.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.440.4 Fields

Field	Function
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 3. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 3. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 3 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 3 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 3 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register.

Field	Function
	- Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

61.7.1.441 Queue 3 sendSlopeCredit (MTL_TXQ3_SENDSLOPECREDIT)

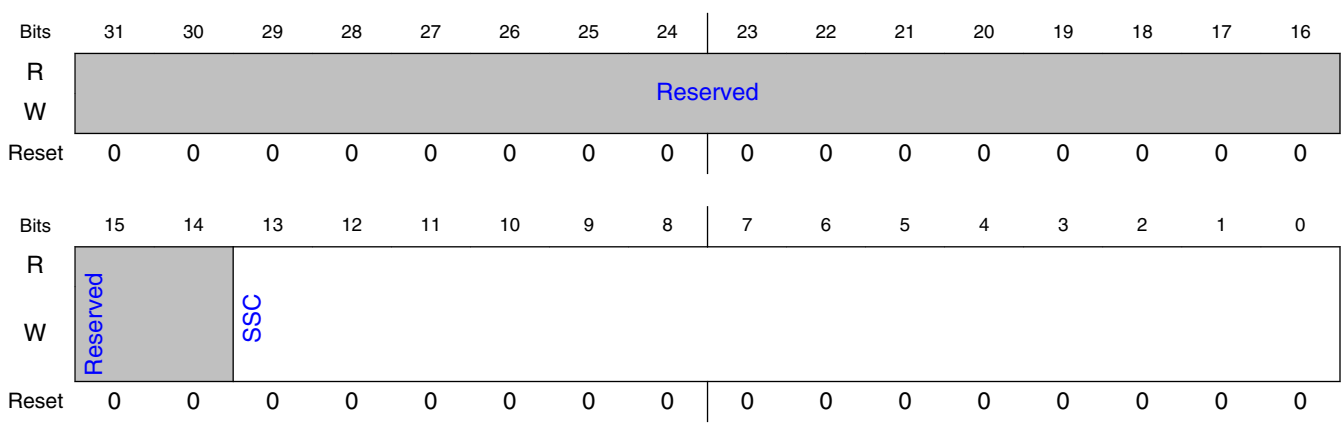
61.7.1.441.1 Offset

Register	Offset
MTL_TXQ3_SENDSLOPECREDIT	DDCh

61.7.1.441.2 Function

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

61.7.1.441.3 Diagram



61.7.1.441.4 Fields

Field	Function
31-14	Reserved

Table continues on the next page...

Field	Function
—	
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 3. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 3. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 3 is selected for transmission.

61.7.1.442 Queue 3 hiCredit (MTL_TXQ3_HICREDIT)

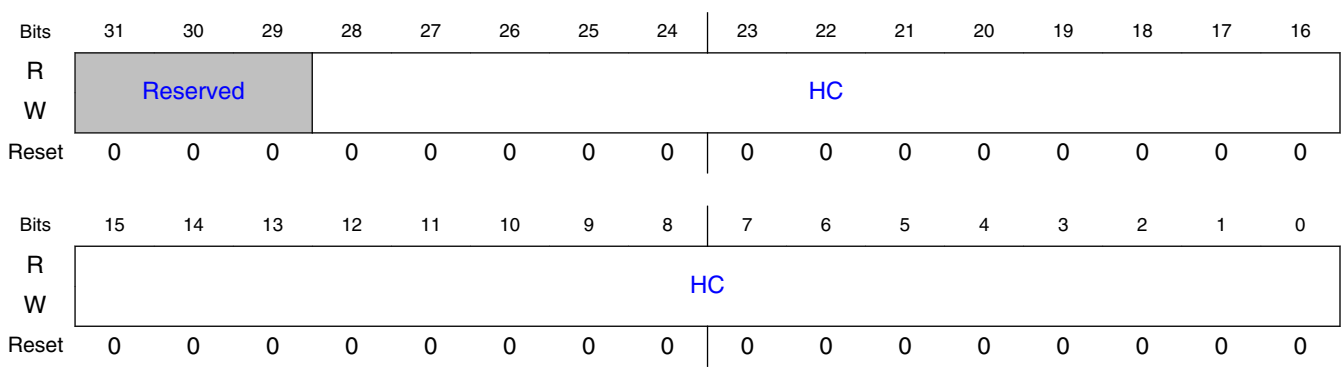
61.7.1.442.1 Offset

Register	Offset
MTL_TXQ3_HICREDIT	DE0h

61.7.1.442.2 Function

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.442.3 Diagram



61.7.1.442.4 Fields

Field	Function
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

61.7.1.443 Queue 3 loCredit (MTL_TXQ3_LOCREDIT)

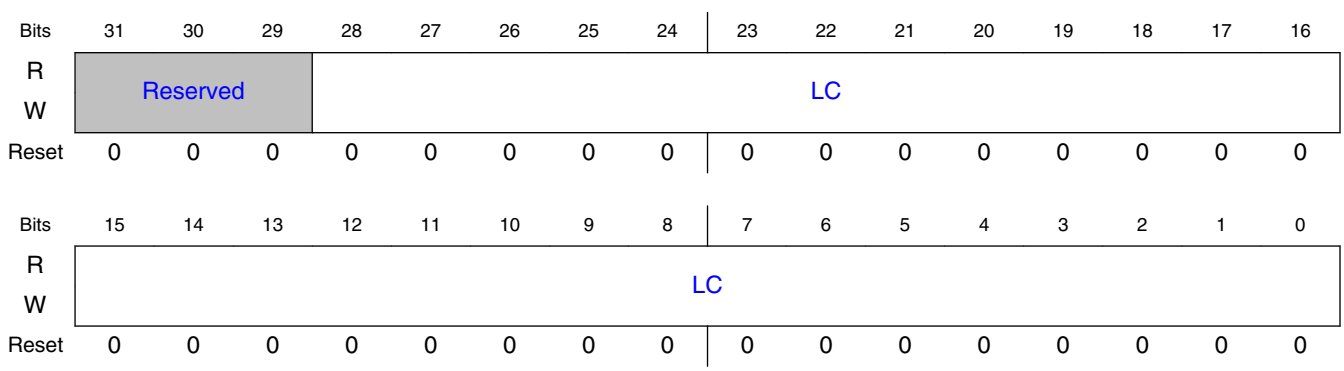
61.7.1.443.1 Offset

Register	Offset
MTL_TXQ3_LOCREDIT	DE4h

61.7.1.443.2 Function

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.443.3 Diagram



61.7.1.443.4 Fields

Field	Function
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

61.7.1.444 Queue 3 Interrupt Control Status (MTL_Q3_INTERRUPT_CONTROL_STATUS)

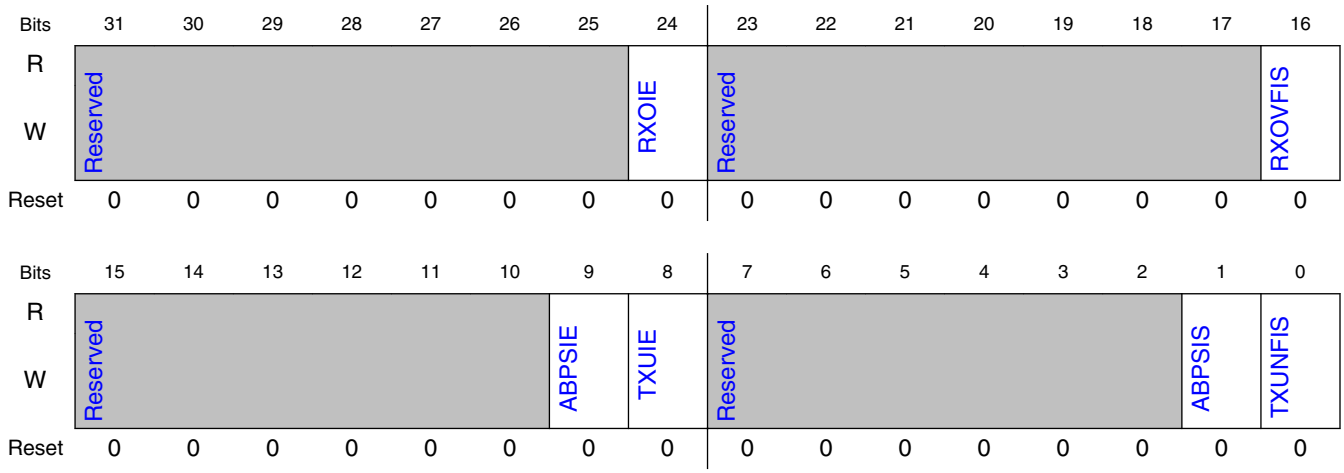
61.7.1.444.1 Offset

Register	Offset
MTL_Q3_INTERRUPT_CONTROL_STATUS	DECh

61.7.1.444.2 Function

This register contains the interrupt enable and status bits for the queue 3 interrupts.

61.7.1.444.3 Diagram



61.7.1.444.4 Fields

Field	Function
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0b - Receive Queue Overflow Interrupt is disabled 1b - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Queue Overflow Interrupt Status not detected 1b - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Function
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0b - Average Bits Per Slot Interrupt is disabled 1b - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0b - Transmit Queue Underflow Interrupt Status is disabled 1b - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Average Bits Per Slot Interrupt Status not detected 1b - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Queue Underflow Interrupt Status not detected 1b - Transmit Queue Underflow Interrupt Status detected

61.7.1.445 Queue 3 Receive Operation Mode (MTL_RXQ3_OPERATION_MODE)

61.7.1.445.1 Offset

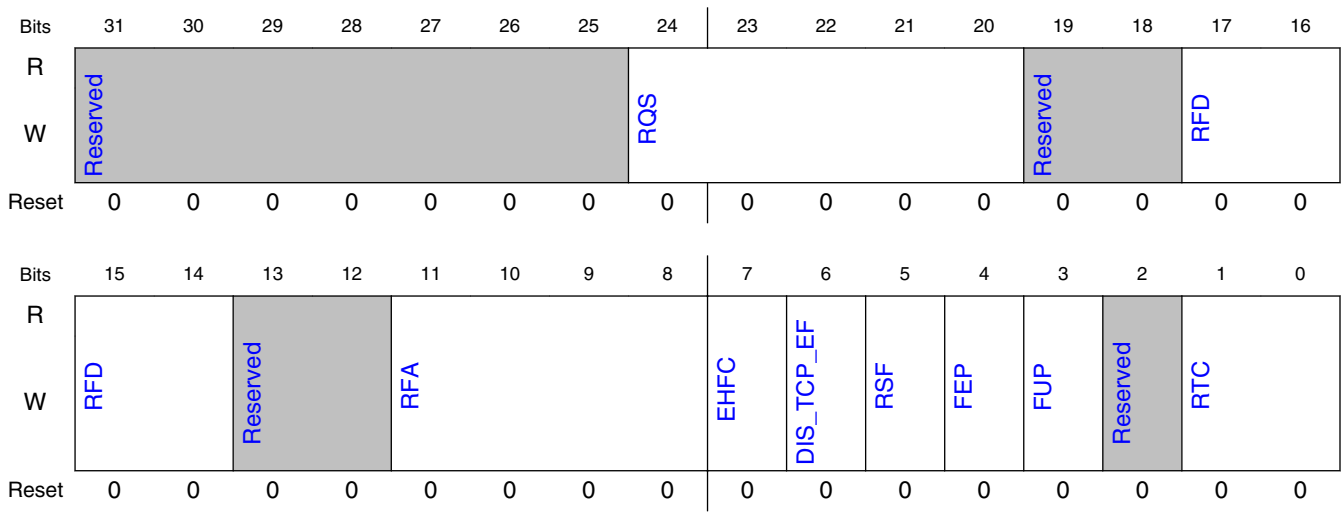
Register	Offset
MTL_RXQ3_OPERATION_MODE	DF0h

61.7.1.445.2 Function

The Queue 3 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

61.7.1.445.3 Diagram



61.7.1.445.4 Fields

Field	Function
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved

Table continues on the next page...

Field	Function
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0b - Hardware Flow Control is disabled 1b - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0b - Dropping of TCP/IP Checksum Error Packets is enabled 1b - Dropping of TCP/IP Checksum Error Packets is disabled

Table continues on the next page...

Field	Function
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0b - Receive Queue Store and Forward is disabled 1b - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0b - Forward Error Packets is disabled 1b - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0b - Forward Undersized Good Packets is disabled 1b - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00b - 64 01b - 32 10b - 96 11b - 128

61.7.1.446 Queue 3 Missed Packet and Overflow Counter (MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT)

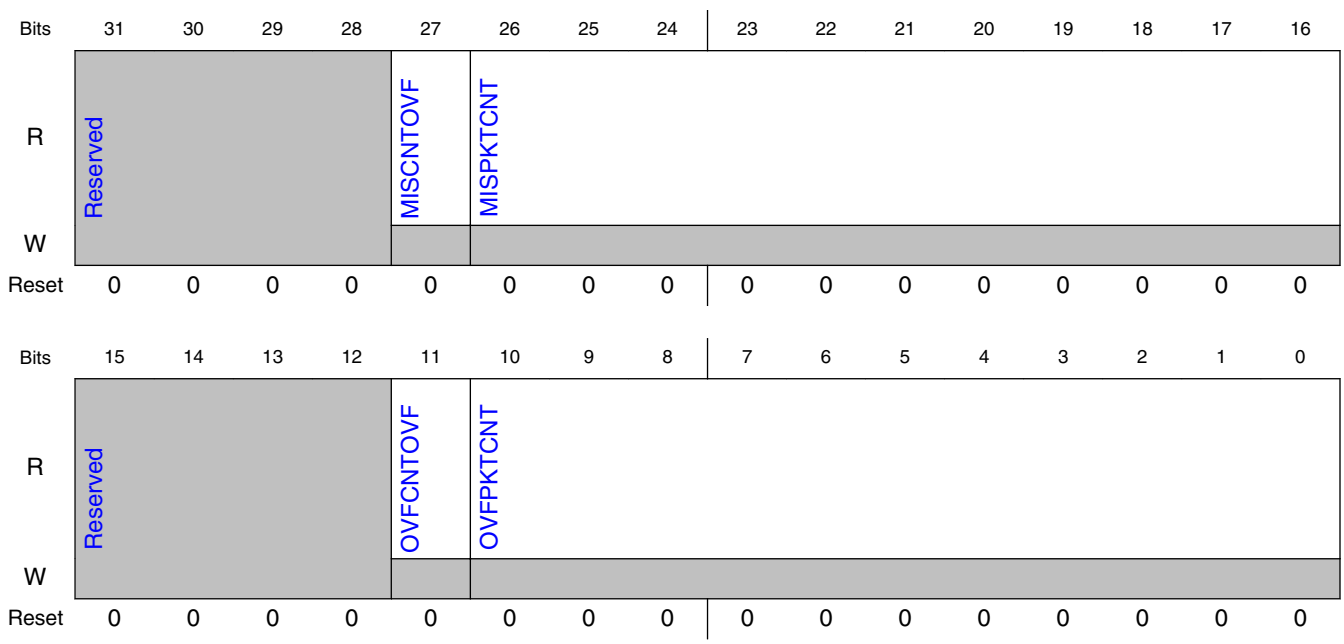
61.7.1.446.1 Offset

Register	Offset
MTL_RXQ3_MISSED_PACKET_OVERFLOW_CNT	DF4h

61.7.1.446.2 Function

The Queue 3 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

61.7.1.446.3 Diagram



61.7.1.446.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Missed Packet Counter overflow not detected 1b - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow Counter overflow not detected 1b - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.447 Queue 3 Receive Debug (MTL_RXQ3_DEBUG)

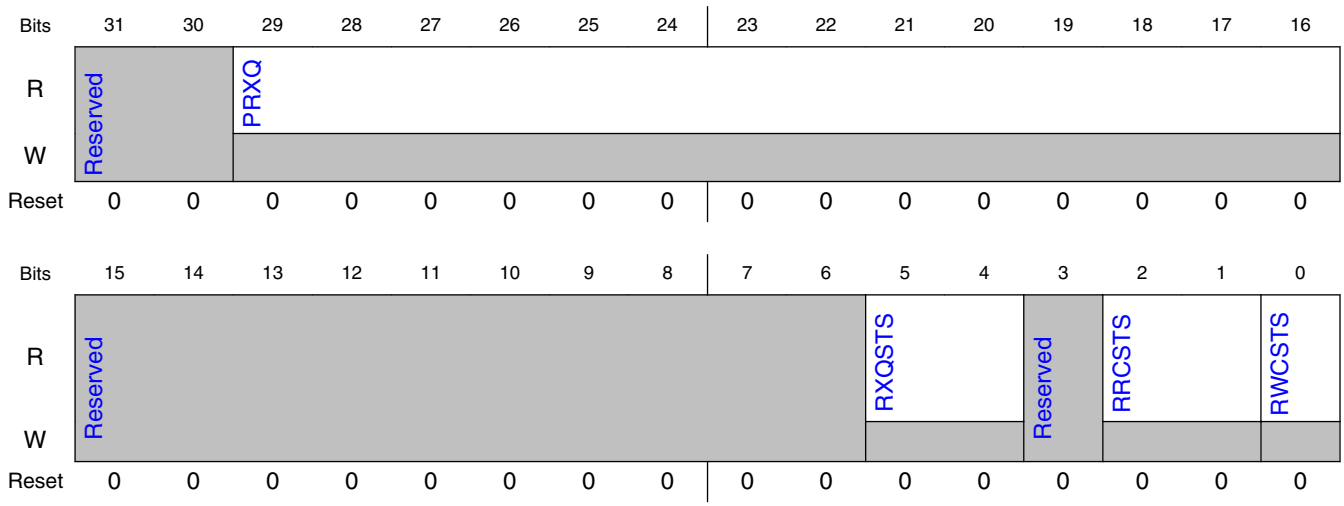
61.7.1.447.1 Offset

Register	Offset
MTL_RXQ3_DEBUG	DF8h

61.7.1.447.2 Function

The Queue 3 Receive Debug register gives the debug status of various blocks related to the Receive queue.

61.7.1.447.3 Diagram



61.7.1.447.4 Fields

Field	Function
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00b - Rx Queue empty 01b - Rx Queue fill-level below flow-control deactivate threshold 10b - Rx Queue fill-level above flow-control activate threshold 11b - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00b - Idle state 01b - Reading packet data 10b - Reading packet status (or timestamp)

Table continues on the next page...

Field	Function
	11b - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0b - MTL Rx Queue Write Controller Active Status not detected 1b - MTL Rx Queue Write Controller Active Status detected

61.7.1.448 Queue 3 Receive Control (MTL_RXQ3_CONTROL)

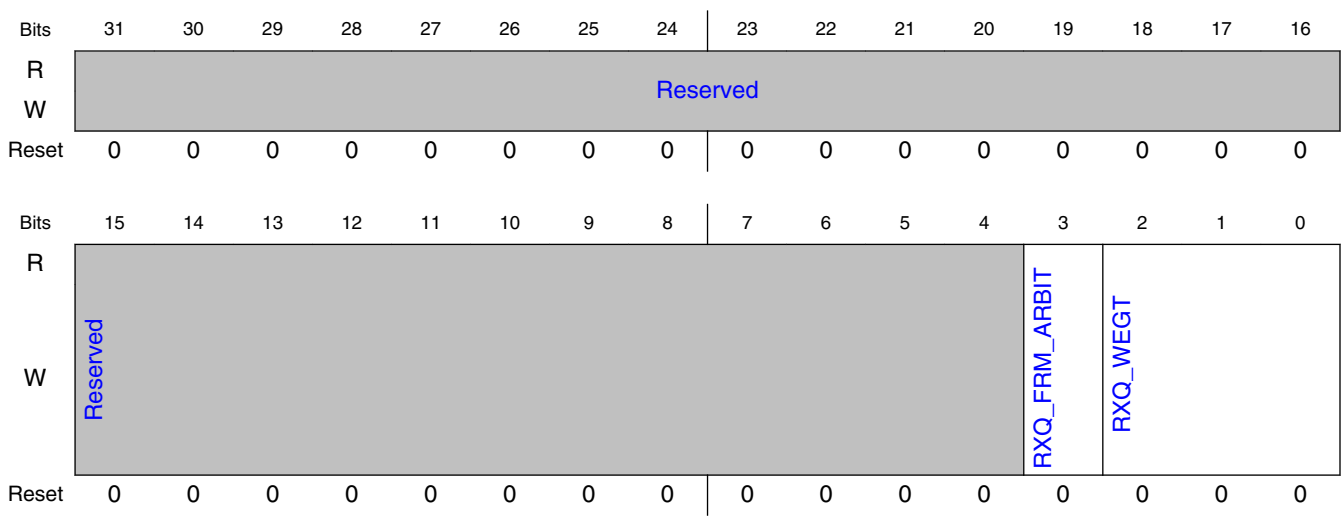
61.7.1.448.1 Offset

Register	Offset
MTL_RXQ3_CONTROL	DFCh

61.7.1.448.2 Function

The Queue 3 Receive Control register controls the receive arbitration and passing of received packets to the application.

61.7.1.448.3 Diagram



61.7.1.448.4 Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Field	Function
—	
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0b - Receive Queue Packet Arbitration is disabled 1b - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 3. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 3. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

61.7.1.449 Queue 4 Transmit Operation Mode (MTL_TXQ4_OPERATION_MODE)

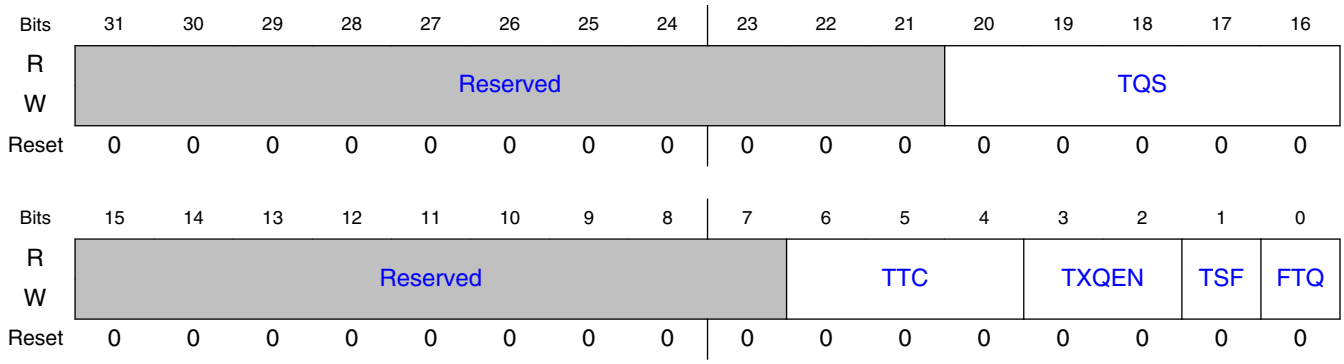
61.7.1.449.1 Offset

Register	Offset
MTL_TXQ4_OPERATION_MODE	E00h

61.7.1.449.2 Function

The Queue 4 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

61.7.1.449.3 Diagram



61.7.1.449.4 Fields

Field	Function
31-21 —	Reserved
20-16 TQS	Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. Transmit Queue Size This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
15-7 —	Reserved
6-4 TTC	Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. 000b - 32 001b - 64 010b - 96 011b - 128 100b - 192 101b - 256 110b - 384 111b - 512
3-2 TXQEN	Transmit Queue Enable This field is used to enable/disable the transmit queue 0. Transmit Queue Enable

Table continues on the next page...

Field	Function
	This field is used to enable/disable the transmit queue 0. - 2'b00: Not enabled - 2'b01: Enable in AV mode - 2'b10: Enabled - 2'b11: Reserved Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field. 00b - Not enabled 01b - Enable in AV mode (Reserved in non-AV) 10b - Enabled 11b - Reserved
1 TSF	Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. 0b - Transmit Store and Forward is disabled 1b - Transmit Store and Forward is enabled
0 FTQ	Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TXQ4_OPERATION_MODE register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Flush Transmit Queue is disabled 1b - Flush Transmit Queue is enabled

61.7.1.450 Queue 4 Underflow Counter (MTL_TXQ4_UNDERFLOW)

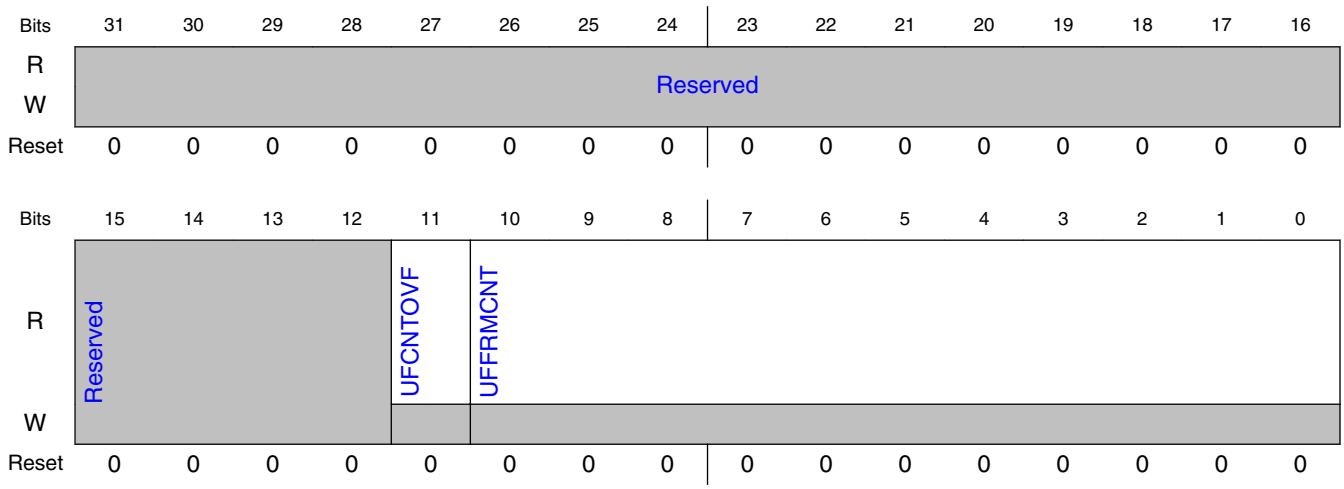
61.7.1.450.1 Offset

Register	Offset
MTL_TXQ4_UNDERFLOW	E04h

61.7.1.450.2 Function

The Queue 4 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

61.7.1.450.3 Diagram



61.7.1.450.4 Fields

Field	Function
31-12 —	Reserved
11 UFCNTOVF	Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow not detected for Underflow Packet Counter 1b - Overflow detected for Underflow Packet Counter
10-0 UFFRMCNT	Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.451 Queue 4 Transmit Debug (MTL_TXQ4_DEBUG)

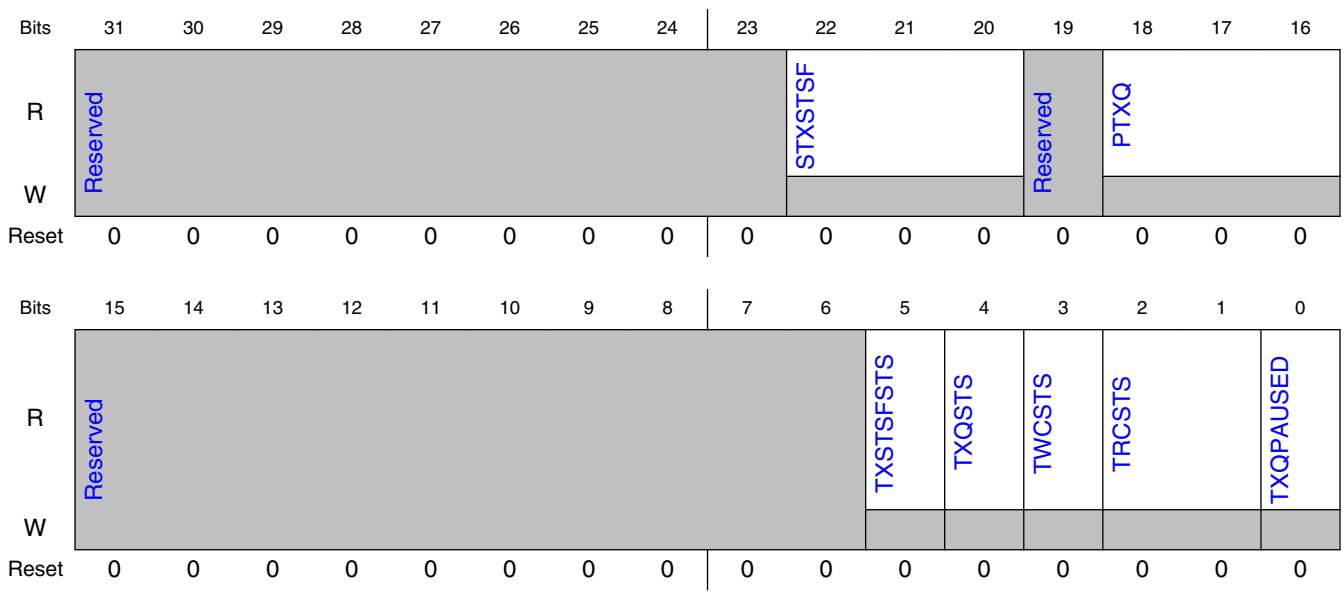
61.7.1.451.1 Offset

Register	Offset
MTL_TXQ4_DEBUG	E08h

61.7.1.451.2 Function

The Queue 4 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

61.7.1.451.3 Diagram



61.7.1.451.4 Fields

Field	Function
31-23 —	Reserved
22-20 STXSTSF	Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue. Number of Status Words in Tx Status FIFO of Queue

Table continues on the next page...

Field	Function
	This field indicates the current number of status in the Tx Status FIFO of this queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19 —	Reserved
18-16 PTXQ	Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_OPERATION_MODE register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15-6 —	Reserved
5 TXSTSFSTS	MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. 0b - MTL Tx Status FIFO Full status is not detected 1b - MTL Tx Status FIFO Full status is detected
4 TXQSTS	MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. 0b - MTL Tx Queue Not Empty status is not detected 1b - MTL Tx Queue Not Empty status is detected
3 TWCSTS	MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. 0b - MTL Tx Queue Write Controller status is not detected 1b - MTL Tx Queue Write Controller status is detected
2-1 TRCSTS	MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: 00b - Idle state 01b - Read state (transferring data to the MAC transmitter) 10b - Waiting for pending Tx Status from the MAC transmitter 11b - Flushing the Tx queue because of the Packet Abort request from the MAC
0 TXQPAUSED	Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802. Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: - Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled - Reception of 802.3x Pause packet when PFC is disabled 0b - Transmit Queue in Pause status is not detected 1b - Transmit Queue in Pause status is detected

61.7.1.452 Queue 4 ETS Control (MTL_TXQ4_ETS_CONTROL)

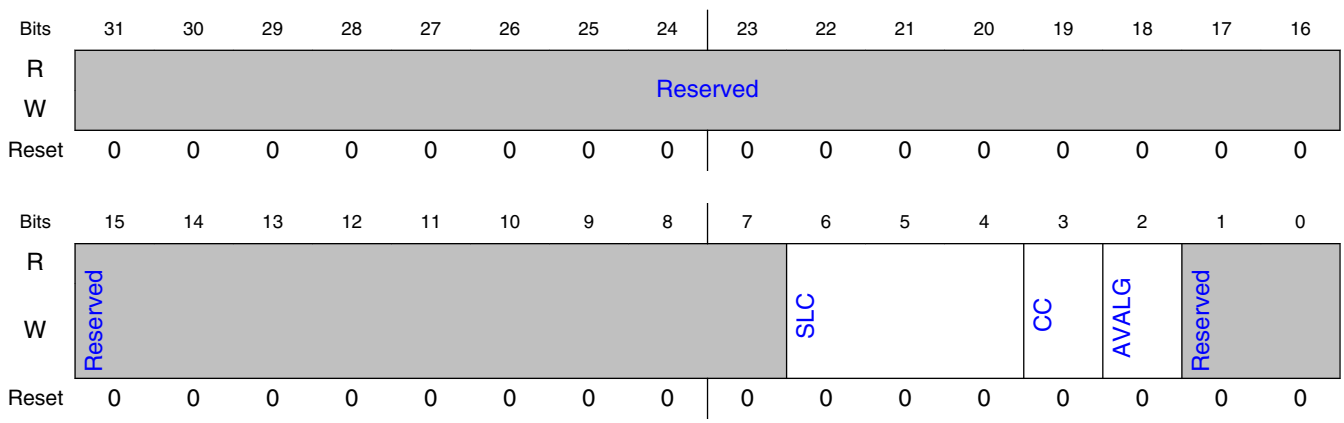
61.7.1.452.1 Offset

Register	Offset
MTL_TXQ4_ETS_CONTROL	E10h

61.7.1.452.2 Function

The Queue 4 ETS Control register controls the enhanced transmission selection operation.

61.7.1.452.3 Diagram



61.7.1.452.4 Fields

Field	Function
31-7 —	Reserved
6-4 SLC	Slot Count If the credit-based shaper algorithm is enabled, the software can program the number of slots (of duration programmed in DMA_CH4_SLOT_INTERVAL register) over which the average transmitted bits per slot, provided in the MTL_TXQ4_ETS_STATUS register, need to be computed for Queue. The encoding is as follows: 000b - 1 slot 001b - 2 slots 010b - 4 slots 011b - 8 slots 100b - 16 slots

Table continues on the next page...

Field	Function
	101b - Reserved
3 CC	Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 4. Credit Control When this bit is set, the accumulated credit parameter in the credit-based shaper algorithm logic is not reset to zero when there is positive credit and no packet to transmit in Channel 4. The credit accumulates even when there is no packet waiting in Channel 4 and another channel is transmitting. When this bit is reset, the accumulated credit parameter in the credit-based shaper algorithm logic is set to zero when there is positive credit and no packet to transmit in Channel 4. When there is no packet waiting in Channel 4 and other channel is transmitting, no credit is accumulated. 0b - Credit Control is disabled 1b - Credit Control is enabled
2 AVALG	AV Algorithm When Queue 4 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 4 traffic. AV Algorithm When Queue 4 is programmed for AV, this field configures the scheduling algorithm for this queue: This bit when set, indicates credit based shaper algorithm (CBS) is selected for Queue 4 traffic. When reset, strict priority is selected. 0b - CBS Algorithm is disabled 1b - CBS Algorithm is enabled
1-0 —	Reserved

61.7.1.453 Queue 4 ETS Status (MTL_TXQ4_ETS_STATUS)

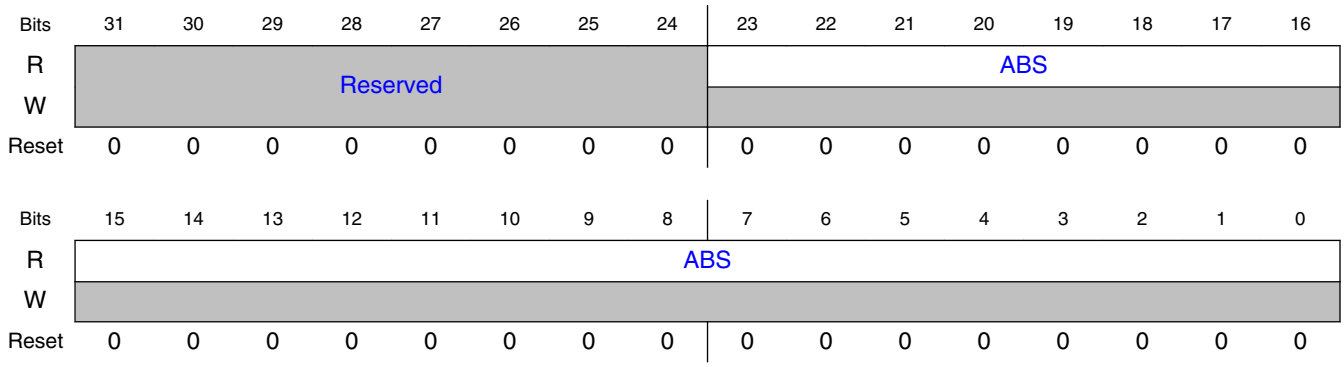
61.7.1.453.1 Offset

Register	Offset
MTL_TXQ4_ETS_STATUS	E14h

61.7.1.453.2 Function

The Queue 4 ETS Status register provides the average traffic transmitted in Queue 4.

61.7.1.453.3 Diagram



61.7.1.453.4 Fields

Field	Function
31-24 —	Reserved
23-0 ABS	Average Bits per Slot This field contains the average transmitted bits per slot. Average Bits per Slot This field contains the average transmitted bits per slot. If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TXQ4_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively. When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.

61.7.1.454 Queue 4 idleSlopeCredit, Quantum or Weights (MTL_TXQ4_QUANTUM_WEIGHT)

61.7.1.454.1 Offset

Register	Offset
MTL_TXQ4_QUANTUM_WEIGHT	E18h

61.7.1.454.2 Function

The Queue 4 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 4.

61.7.1.454.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											ISCQW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ISCQW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.454.4 Fields

Field	Function
31-21 —	Reserved
20-0 ISCQW	idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 4. idleSlopeCredit, Quantum or Weights - idleSlopeCredit When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 4. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero. - Quantum When the DCB operation is enabled with DWRR algorithm for Queue 4 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes. - Weights When DCB operation is enabled with WFQ algorithm for Queue 4 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 4 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero. - Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is OPERATION_MODE register.

Field	Function
	- Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size)) - Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.

61.7.1.455 Queue 4 sendSlopeCredit (MTL_TXQ4_SENDSLOPECREDIT)

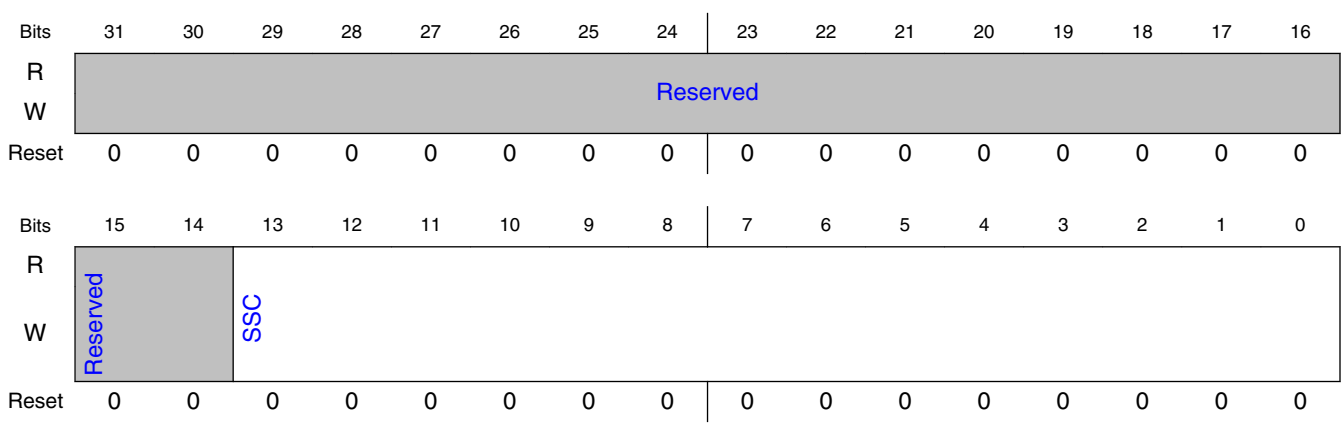
61.7.1.455.1 Offset

Register	Offset
MTL_TXQ4_SENDSLOPECREDIT	E1Ch

61.7.1.455.2 Function

The sendSlopeCredit register contains the sendSlope credit value required for the credit-based shaper algorithm for the Queue.

61.7.1.455.3 Diagram



61.7.1.455.4 Fields

Field	Function
31-14	Reserved

Table continues on the next page...

Field	Function
—	
13-0 SSC	sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 4. sendSlopeCredit Value When AV operation is enabled, this field contains the sendSlopeCredit value required for credit-based shaper algorithm for Queue 4. This is the rate of change of credit in bits per cycle (40 ns, 8 ns and 3.2 ns for 100 Mbps, 1000 Mbps and 2500 Mbps respectively) when the credit is decreasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. This field should be programmed with absolute sendSlopeCredit value. The credit-based shaper logic subtracts it from the accumulated credit when Channel 4 is selected for transmission.

61.7.1.456 Queue 4 hiCredit (MTL_TXQ4_HICREDIT)

61.7.1.456.1 Offset

Register	Offset
MTL_TXQ4_HICREDIT	E20h

61.7.1.456.2 Function

The hiCredit register contains the hiCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.456.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			HC												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.456.4 Fields

Field	Function
31-29 —	Reserved
28-0 HC	hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. hiCredit Value When the AV feature is enabled, this field contains the hiCredit value required for the credit-based shaper algorithm. This is the maximum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value is maxInterferenceSize, that is, best-effort maximum packet size (16,384 bytes or 131,072 bits). The value to be specified is 131,072 * 1,024 = 134,217,728 or 0x0800_0000.

61.7.1.457 Queue 4 loCredit (MTL_TXQ4_LOCREDIT)

61.7.1.457.1 Offset

Register	Offset
MTL_TXQ4_LOCREDIT	E24h

61.7.1.457.2 Function

The loCredit register contains the loCredit value required for the credit-based shaper algorithm for the Queue.

61.7.1.457.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LC												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.457.4 Fields

Field	Function
31-29 —	Reserved
28-0 LC	loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. loCredit Value When AV operation is enabled, this field contains the loCredit value required for the credit-based shaper algorithm. This is the minimum value that can be accumulated in the credit parameter. This is specified in bits scaled by 1,024. The maximum value to be programmed is corresponds to twice the maxFrameSize transmitted from this queue. If the maxFrameSize is 8192 bytes, then (8192*2) * 8 * 1024 = 134,217,728 or 0x0800_0000. Because it is a negative value, the programmed value is 2's complement of the value, that is, 0x1800_0000.

61.7.1.458 Queue 4 Interrupt Control Status (MTL_Q4_INTERRUPT_CONTROL_STATUS)

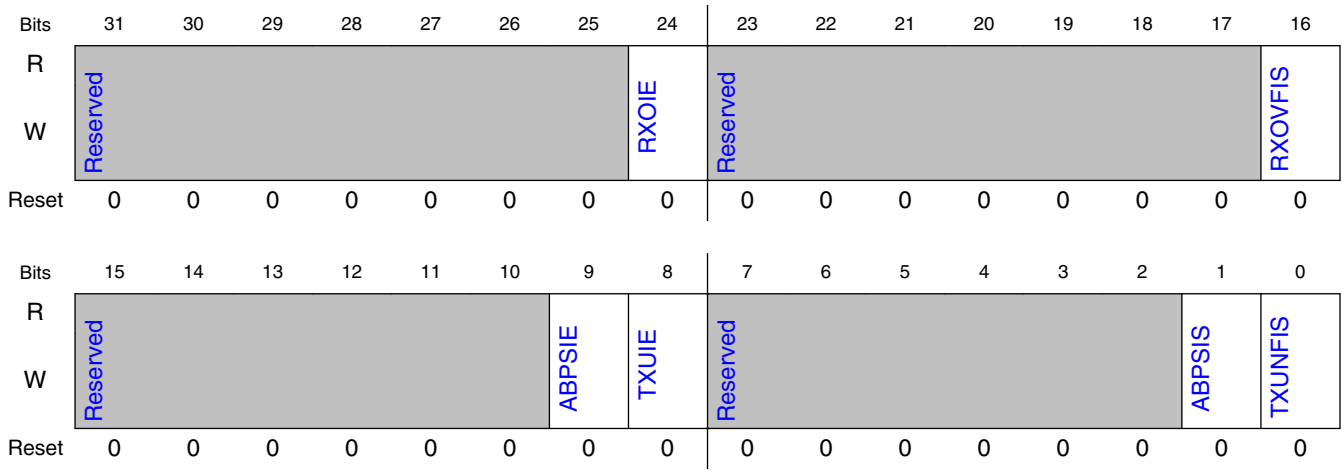
61.7.1.458.1 Offset

Register	Offset
MTL_Q4_INTERRUPT_CONTROL_STATUS	E2Ch

61.7.1.458.2 Function

This register contains the interrupt enable and status bits for the queue 4 interrupts.

61.7.1.458.3 Diagram



61.7.1.458.4 Fields

Field	Function
31-25 —	Reserved
24 RXOIE	Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. 0b - Receive Queue Overflow Interrupt is disabled 1b - Receive Queue Overflow Interrupt is enabled
23-17 —	Reserved
16 RXOVFIS	Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Queue Overflow Interrupt Status not detected 1b - Receive Queue Overflow Interrupt Status detected
15-10 —	Reserved
9 ABPSIE	Average Bits Per Slot Interrupt Enable When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. Average Bits Per Slot Interrupt Enable

Table continues on the next page...

Field	Function
	When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated. When this bit is cleared, the interrupt is not asserted for such an event. 0b - Average Bits Per Slot Interrupt is disabled 1b - Average Bits Per Slot Interrupt is enabled
8 TXUIE	Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. 0b - Transmit Queue Underflow Interrupt Status is disabled 1b - Transmit Queue Underflow Interrupt Status is enabled
7-2 —	Reserved
1 ABPSIS	Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. Average Bits Per Slot Interrupt Status When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Average Bits Per Slot Interrupt Status not detected 1b - Average Bits Per Slot Interrupt Status detected
0 TXUNFIS	Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Queue Underflow Interrupt Status not detected 1b - Transmit Queue Underflow Interrupt Status detected

61.7.1.459 Queue 4 Receive Operation Mode (MTL_RXQ4_OPERATION_MODE)

61.7.1.459.1 Offset

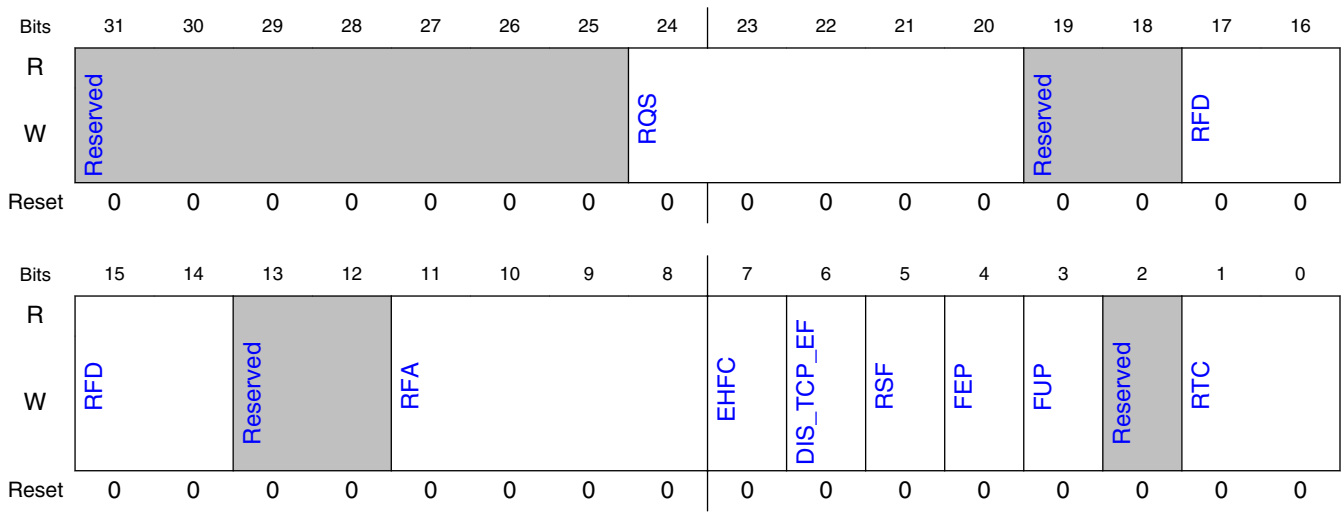
Register	Offset
MTL_RXQ4_OPERATION_MODE	E30h

61.7.1.459.2 Function

The Queue 4 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

61.7.1.459.3 Diagram



61.7.1.459.4 Fields

Field	Function
31-25 —	Reserved
24-20 RQS	Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. Receive Queue Size This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value. The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits: LOG2(2048/256) = LOG2(8) = 3 bits
19-18 —	Reserved

Table continues on the next page...

Field	Function
17-14 RFD	Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1. Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: - 0: Full minus 1 KB, that is, FULL 1 KB - 1: Full minus 1.5 KB, that is, FULL 1.5 KB - 2: Full minus 2 KB, that is, FULL 2 KB - 3: Full minus 2.5 KB, that is, FULL 2.5 KB - ... - 14: Full minus 8 KB, that is, FULL 8 KB - 15: Full minus 8.5 KB, that is, FULL 8.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13-12 —	Reserved
11-8 RFA	Threshold for Activating Flow Control (in half-duplex and full-duplex These bits control the threshold (fill-level of Rx queue) at which the flow control is activated: For more information on encoding for this field, see RFD.
7 EHFC	Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. 0b - Hardware Flow Control is disabled 1b - Hardware Flow Control is enabled
6 DIS_TCP_EF	Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. 0b - Dropping of TCP/IP Checksum Error Packets is enabled 1b - Dropping of TCP/IP Checksum Error Packets is disabled

Table continues on the next page...

Field	Function
5 RSF	Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. Receive Queue Store and Forward When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. 0b - Receive Queue Store and Forward is disabled 1b - Receive Queue Store and Forward is enabled
4 FEP	Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. 0b - Forward Error Packets is disabled 1b - Forward Error Packets is enabled
3 FUP	Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. 0b - Forward Undersized Good Packets is disabled 1b - Forward Undersized Good Packets is enabled
2 —	Reserved
1-0 RTC	Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1. 00b - 64 01b - 32 10b - 96 11b - 128

61.7.1.460 Queue 4 Missed Packet and Overflow Counter (MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT)

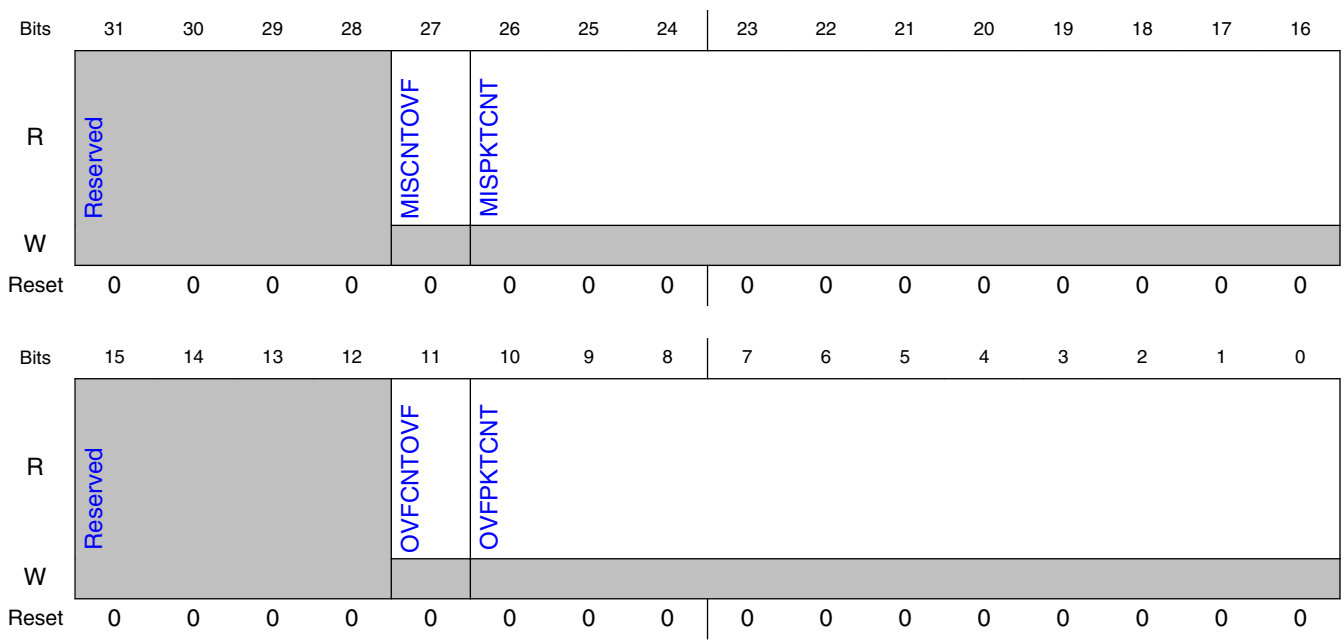
61.7.1.460.1 Offset

Register	Offset
MTL_RXQ4_MISSED_PACKET_OVERFLOW_CNT	E34h

61.7.1.460.2 Function

The Queue 4 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

61.7.1.460.3 Diagram



61.7.1.460.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27 MISCNTOVF	Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Missed Packet Counter overflow not detected 1b - Missed Packet Counter overflow detected
26-16 MISPKTCNT	Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. Missed Packet Counter This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15-12 —	Reserved
11 OVFCNTOVF	Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Overflow Counter overflow not detected 1b - Overflow Counter overflow detected
10-0 OVFPKTCNT	Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. Overflow Packet Counter This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.461 Queue 4 Receive Debug (MTL_RXQ4_DEBUG)

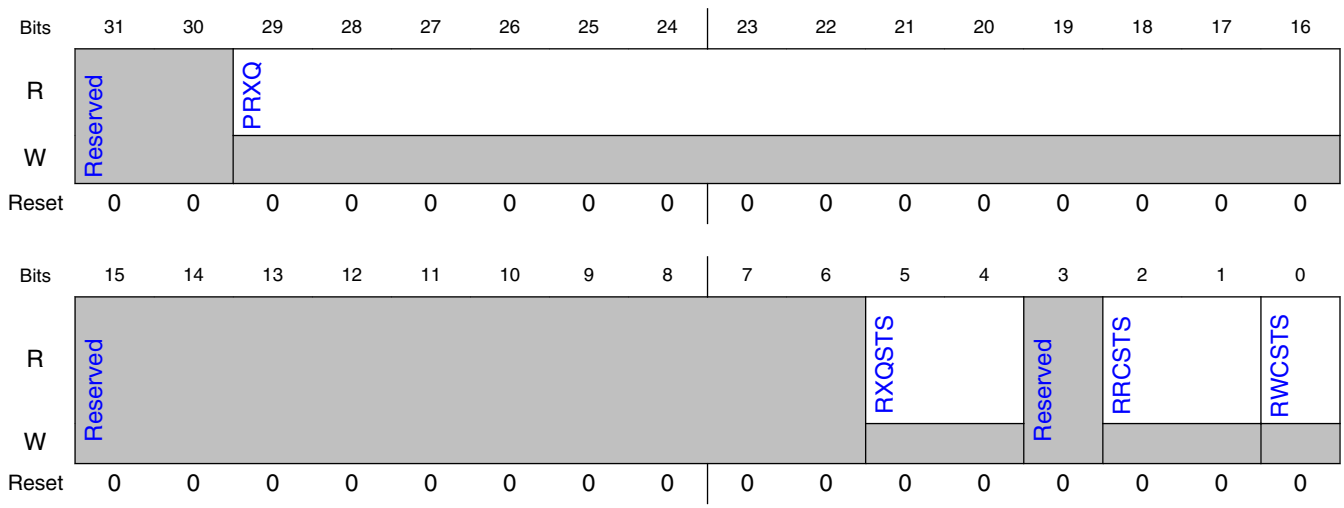
61.7.1.461.1 Offset

Register	Offset
MTL_RXQ4_DEBUG	E38h

61.7.1.461.2 Function

The Queue 4 Receive Debug register gives the debug status of various blocks related to the Receive queue.

61.7.1.461.3 Diagram



61.7.1.461.4 Fields

Field	Function
31-30 —	Reserved
29-16 PRXQ	Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15-6 —	Reserved
5-4 RXQSTS	MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: 00b - Rx Queue empty 01b - Rx Queue fill-level below flow-control deactivate threshold 10b - Rx Queue fill-level above flow-control activate threshold 11b - Rx Queue full
3 —	Reserved
2-1 RRCSTS	MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: 00b - Idle state 01b - Reading packet data 10b - Reading packet status (or timestamp)

Table continues on the next page...

Field	Function
	11b - Flushing the packet data and status
0 RWCSTS	MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. 0b - MTL Rx Queue Write Controller Active Status not detected 1b - MTL Rx Queue Write Controller Active Status detected

61.7.1.462 Queue 4 Receive Control (MTL_RXQ4_CONTROL)

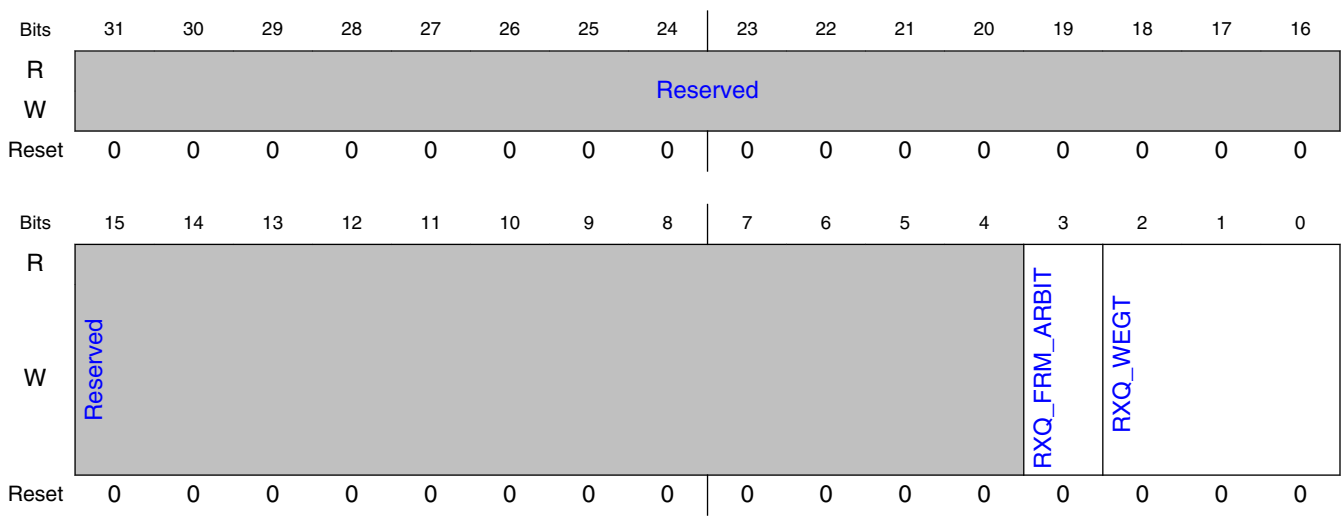
61.7.1.462.1 Offset

Register	Offset
MTL_RXQ4_CONTROL	E3Ch

61.7.1.462.2 Function

The Queue 4 Receive Control register controls the receive arbitration and passing of received packets to the application.

61.7.1.462.3 Diagram



61.7.1.462.4 Fields

Field	Function
31-4	Reserved

Table continues on the next page...

Field	Function
—	
3 RXQ_FRM_ARBIT	Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. Receive Queue Packet Arbitration When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue. When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue: - PBL amount of data (indicated by ari_qN_pbl_i[]) or - Complete data of a packet The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode). 0b - Receive Queue Packet Arbitration is disabled 1b - Receive Queue Packet Arbitration is enabled
2-0 RXQ_WEGT	Receive Queue Weight This field indicates the weight assigned to the Rx Queue 4. Receive Queue Weight This field indicates the weight assigned to the Rx Queue 4. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.

61.7.1.463 DMA Bus Mode (DMA_MODE)

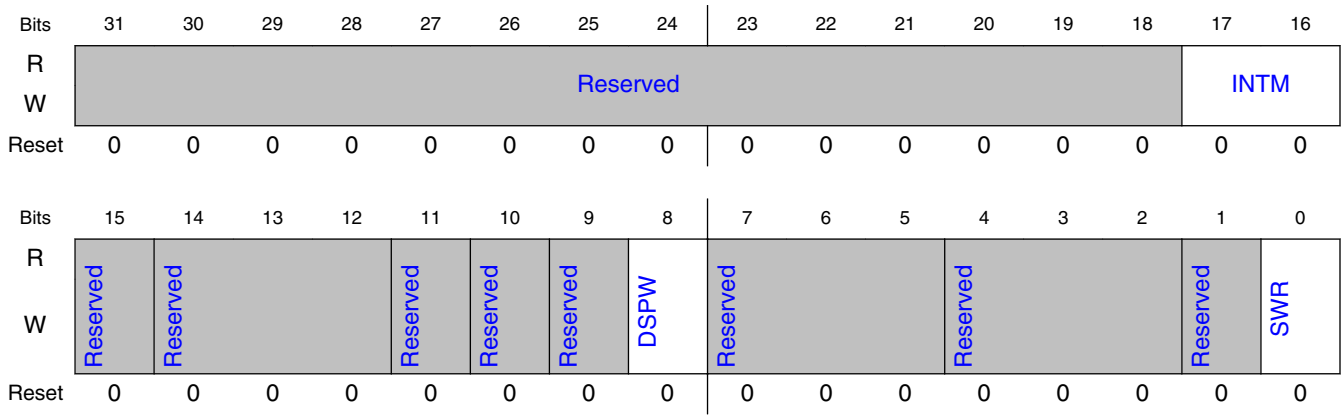
61.7.1.463.1 Offset

Register	Offset
DMA_MODE	1000h

61.7.1.463.2 Function

The Bus Mode register establishes the bus operating modes for the DMA.

61.7.1.463.3 Diagram



61.7.1.463.4 Fields

Field	Function
31-18 —	Reserved
17-16 INTM	Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. Interrupt Mode This field defines the interrupt mode of DWC_ether_qos. The behavior of the following outputs changes depending on the following settings: - sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) - sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) - sbd_intr_o (Common Interrupt) It also changes the behavior of the RI/TI bits in the STATUS. - 00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits. - 01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. - 11: Reserved For more details please refer Table "DWC_ether_qos Transfer Complete Interrupt Behavior". 00b - See above description 01b - See above description 10b - See above description 11b - Reserved

Table continues on the next page...

Field	Function
15 —	Reserved
14-12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 DSPW	Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. When this bit is set to 1, the descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor, otherwise the descriptor writes are always posted. 0b - Descriptor Posted Write is disabled 1b - Descriptor Posted Write is enabled
7-5 —	Reserved
4-2 —	Reserved
1 —	Reserved
0 SWR	Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all DWC_ether_qos clock domains. Before reprogramming any DWC_ether_qos register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. 0b - Software Reset is disabled 1b - Software Reset is enabled

61.7.1.464 DMA System Bus Mode (DMA_SYSBUS_MODE)

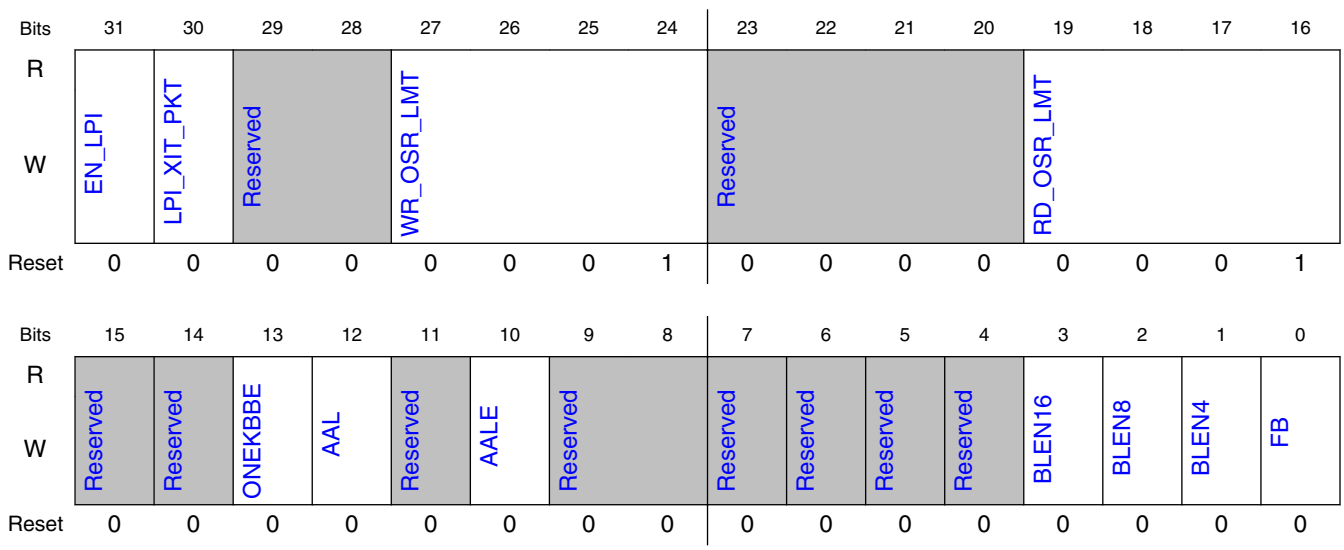
61.7.1.464.1 Offset

Register	Offset
DMA_SYSBUS_MODE	1004h

61.7.1.464.2 Function

The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and number of outstanding requests.

61.7.1.464.3 Diagram



61.7.1.464.4 Fields

Field	Function
31 EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller. 0b - Low Power Interface (LPI) is disabled 1b - Low Power Interface (LPI) is enabled

Table continues on the next page...

Field	Function
30 LPI_XIT_PKT	Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. When set to 0, this bit enables the AXI master to come out of the LPI mode when any packet is received. 0b - Unlock on Magic Packet or Remote Wake-Up Packet is disabled 1b - Unlock on Magic Packet or Remote Wake-Up Packet is enabled
29-28 —	Reserved
27-24 WR_OSR_LMT	AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1 Note: - Bit 26 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ = 4 - Bit 27 is reserved if DWC_ETHER_QOS_AXI_MAX_WR_REQ!= 16
23-20 —	Reserved
19-16 RD_OSR_LMT	AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1 Note: - Bit 18 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ = 4 - Bit 19 is reserved if parameter DWC_ETHER_QOS_AXI_MAX_RD_REQ!= 16
15 —	Reserved
14 —	Reserved
13 ONEKBBE	1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. 1 KB Boundary Crossing Enable for the EQOS-AXI Master When set, the burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary. When reset, the burst transfers performed by the EQOS-AXI master do not cross 4 KB boundary. 0b - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is disabled 1b - 1 KB Boundary Crossing for the EQOS-AXI Master Beats is enabled
12 AAL	Address-Aligned Beats When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels. 0b - Address-Aligned Beats is disabled 1b - Address-Aligned Beats is enabled
11	Reserved

Table continues on the next page...

Field	Function
—	
10 AALE	Automatic AXI LPI enable When set to 1, enables the AXI master to enter into LPI state when there is no activity in the DWC_ether_qos for number of system clock cycles programmed in the LPIEI field of DMA_AXI_LPI_ENTRY_INTERVAL register. 0b - Automatic AXI LPI is disabled 1b - Automatic AXI LPI is enabled
9-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 BLEN16	AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0b - No effect 1b - AXI Burst Length 16
2 BLEN8	AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0b - No effect 1b - AXI Burst Length 8
1 BLEN4	AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. 0b - No effect 1b - AXI Burst Length 4
0 FB	Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. Fixed Burst Length

Field	Function
	When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. - Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field - Burst transfers of length 1 When this bit is set to 0, the EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:1]. 0b - Fixed Burst Length is disabled 1b - Fixed Burst Length is enabled

61.7.1.465 DMA Interrupt Status (DMA_INTERRUPT_STATUS)

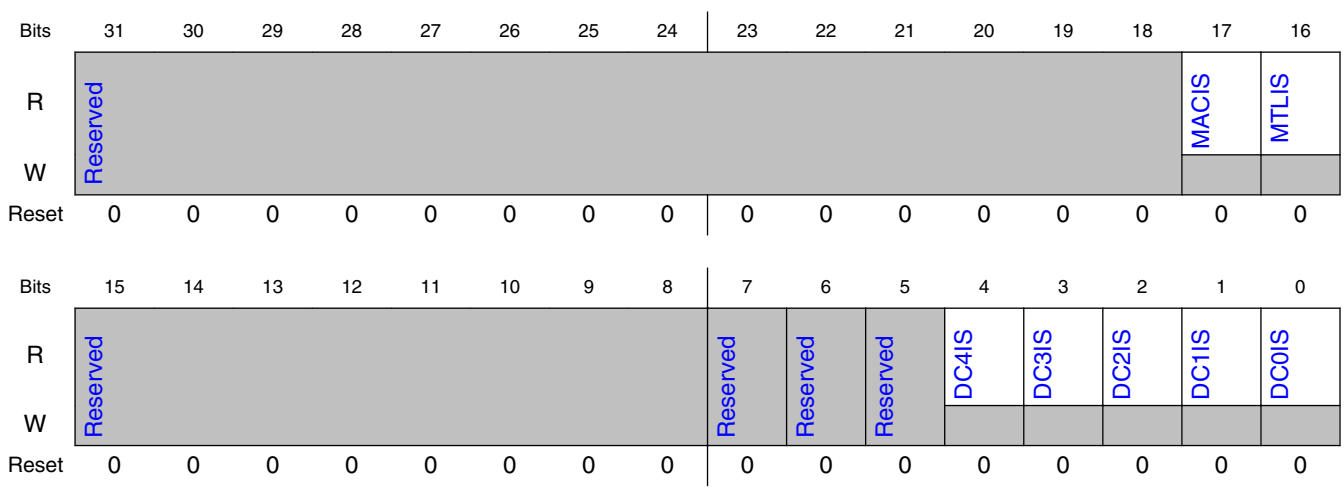
61.7.1.465.1 Offset

Register	Offset
DMA_INTERRUPT_STATUS	1008h

61.7.1.465.2 Function

The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

61.7.1.465.3 Diagram



61.7.1.465.4 Fields

Field	Function
31-18 —	Reserved
17 MACIS	MAC Interrupt Status This bit indicates an interrupt event in the MAC. MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source. 0b - MAC Interrupt Status not detected 1b - MAC Interrupt Status detected
16 MTLIS	MTL Interrupt Status This bit indicates an interrupt event in the MTL. MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source. 0b - MTL Interrupt Status not detected 1b - MTL Interrupt Status detected
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 DC4IS	DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. DMA Channel 4 Interrupt Status This bit indicates an interrupt event in DMA Channel 4. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 4 to get the exact cause of the interrupt and clear its source. 0b - DMA Channel 4 Interrupt Status not detected 1b - DMA Channel 4 Interrupt Status detected
3 DC3IS	DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. DMA Channel 3 Interrupt Status This bit indicates an interrupt event in DMA Channel 3. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 3 to get the exact cause of the interrupt and clear its source. 0b - DMA Channel 3 Interrupt Status not detected 1b - DMA Channel 3 Interrupt Status detected
2 DC2IS	DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. DMA Channel 2 Interrupt Status This bit indicates an interrupt event in DMA Channel 2. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 2 to get the exact cause of the interrupt and clear its source. 0b - DMA Channel 2 Interrupt Status not detected 1b - DMA Channel 2 Interrupt Status detected

Table continues on the next page...

Field	Function
1 DC1IS	DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. DMA Channel 1 Interrupt Status This bit indicates an interrupt event in DMA Channel 1. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 1 to get the exact cause of the interrupt and clear its source. 0b - DMA Channel 1 Interrupt Status not detected 1b - DMA Channel 1 Interrupt Status detected
0 DC0IS	DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. 0b - DMA Channel 0 Interrupt Status not detected 1b - DMA Channel 0 Interrupt Status detected

61.7.1.466 DMA Debug Status 0 (DMA_DEBUG_STATUS0)

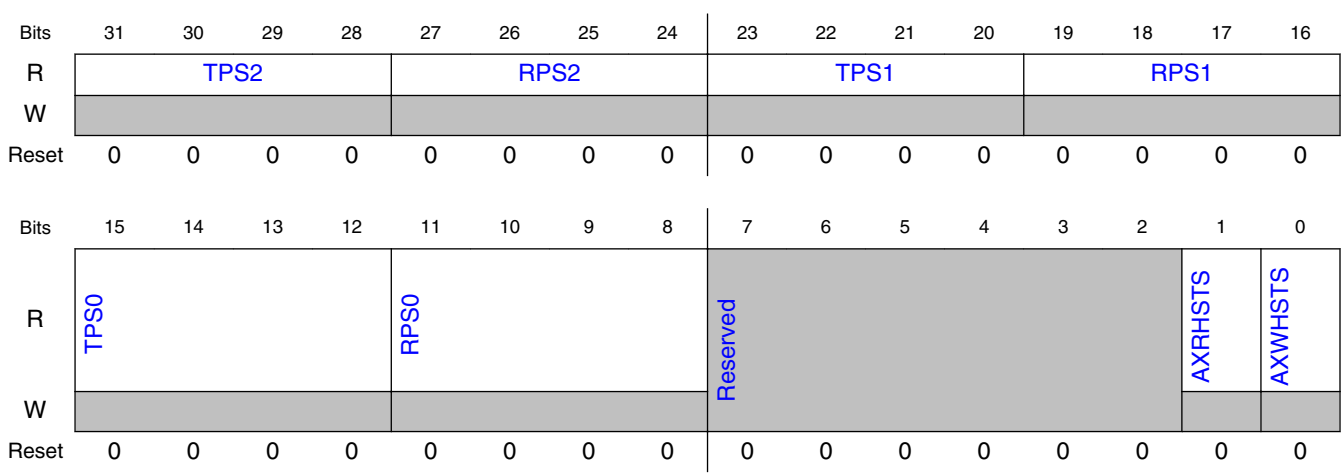
61.7.1.466.1 Offset

Register	Offset
DMA_DEBUG_STATUS0	100Ch

61.7.1.466.2 Function

The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose.

61.7.1.466.3 Diagram



61.7.1.466.4 Fields

Field	Function
31-28 TPS2	DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. DMA Channel 2 Transmit Process State This field indicates the Tx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Transmit Command issued) 0001b - Running (Fetching Tx Transfer Descriptor) 0010b - Running (Waiting for status) 0011b - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100b - Timestamp write state 0101b - Reserved for future use 0110b - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111b - Running (Closing Tx Descriptor)
27-24 RPS2	DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. DMA Channel 2 Receive Process State This field indicates the Rx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Receive Command issued) 0001b - Running (Fetching Rx Transfer Descriptor) 0010b - Reserved for future use 0011b - Running (Waiting for Rx packet) 0100b - Suspended (Rx Descriptor Unavailable) 0101b - Running (Closing the Rx Descriptor) 0110b - Timestamp write state 0111b - Running (Transferring the received packet data from the Rx buffer to the system memory)
23-20 TPS1	DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1. DMA Channel 1 Transmit Process State This field indicates the Tx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Transmit Command issued) 0001b - Running (Fetching Tx Transfer Descriptor) 0010b - Running (Waiting for status) 0011b - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100b - Timestamp write state 0101b - Reserved for future use 0110b - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111b - Running (Closing Tx Descriptor)
19-16 RPS1	DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1. DMA Channel 1 Receive Process State This field indicates the Rx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Receive Command issued)

Table continues on the next page...

Field	Function
	0001b - Running (Fetching Rx Transfer Descriptor) 0010b - Reserved for future use 0011b - Running (Waiting for Rx packet) 0100b - Suspended (Rx Descriptor Unavailable) 0101b - Running (Closing the Rx Descriptor) 0110b - Timestamp write state 0111b - Running (Transferring the received packet data from the Rx buffer to the system memory)
15-12 TPS0	DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Transmit Command issued) 0001b - Running (Fetching Tx Transfer Descriptor) 0010b - Running (Waiting for status) 0011b - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100b - Timestamp write state 0101b - Reserved for future use 0110b - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111b - Running (Closing Tx Descriptor)
11-8 RPS0	DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Receive Command issued) 0001b - Running (Fetching Rx Transfer Descriptor) 0010b - Reserved for future use 0011b - Running (Waiting for Rx packet) 0100b - Suspended (Rx Descriptor Unavailable) 0101b - Running (Closing the Rx Descriptor) 0110b - Timestamp write state 0111b - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-2 —	Reserved
1 AXRHSTS	AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data. 0b - AXI Master Read Channel Status not detected 1b - AXI Master Read Channel Status detected
0 AXWHSTS	AXI Master Write Channel When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. 0b - AXI Master Write Channel or AHB Master Status not detected 1b - AXI Master Write Channel or AHB Master Status detected

61.7.1.467 DMA Debug Status 1 (DMA_DEBUG_STATUS1)

61.7.1.467.1 Offset

Register	Offset
DMA_DEBUG_STATUS1	1010h

61.7.1.467.2 Function

The Debug Status1 register gives the Receive and Transmit process status for DMA Channel 3-Channel 6.

61.7.1.467.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				Reserved				Reserved				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TPS4				RPS4				TPS3				RPS3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.467.4 Fields

Field	Function
31-28 —	Reserved
27-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 TPS4	DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. DMA Channel 4 Transmit Process State This field indicates the Tx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Transmit Command issued) 0001b - Running (Fetching Tx Transfer Descriptor) 0010b - Running (Waiting for status)

Table continues on the next page...

Field	Function
	0011b - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100b - Timestamp write state 0101b - Reserved for future use 0110b - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111b - Running (Closing Tx Descriptor)
11-8 RPS4	DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. DMA Channel 4 Receive Process State This field indicates the Rx DMA FSM state for Channel 4. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Receive Command issued) 0001b - Running (Fetching Rx Transfer Descriptor) 0010b - Reserved for future use 0011b - Running (Waiting for Rx packet) 0100b - Suspended (Rx Descriptor Unavailable) 0101b - Running (Closing the Rx Descriptor) 0110b - Timestamp write state 0111b - Running (Transferring the received packet data from the Rx buffer to the system memory)
7-4 TPS3	DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. DMA Channel 3 Transmit Process State This field indicates the Tx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Transmit Command issued) 0001b - Running (Fetching Tx Transfer Descriptor) 0010b - Running (Waiting for status) 0011b - Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0100b - Timestamp write state 0101b - Reserved for future use 0110b - Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0111b - Running (Closing Tx Descriptor)
3-0 RPS3	DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. DMA Channel 3 Receive Process State This field indicates the Rx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt. 0000b - Stopped (Reset or Stop Receive Command issued) 0001b - Running (Fetching Rx Transfer Descriptor) 0010b - Reserved for future use 0011b - Running (Waiting for Rx packet) 0100b - Suspended (Rx Descriptor Unavailable) 0101b - Running (Closing the Rx Descriptor) 0110b - Timestamp write state 0111b - Running (Transferring the received packet data from the Rx buffer to the system memory)

61.7.1.468 AXI LPI Entry Interval Control (DMA_AXI_LPI_ENTRY_INTERVAL)

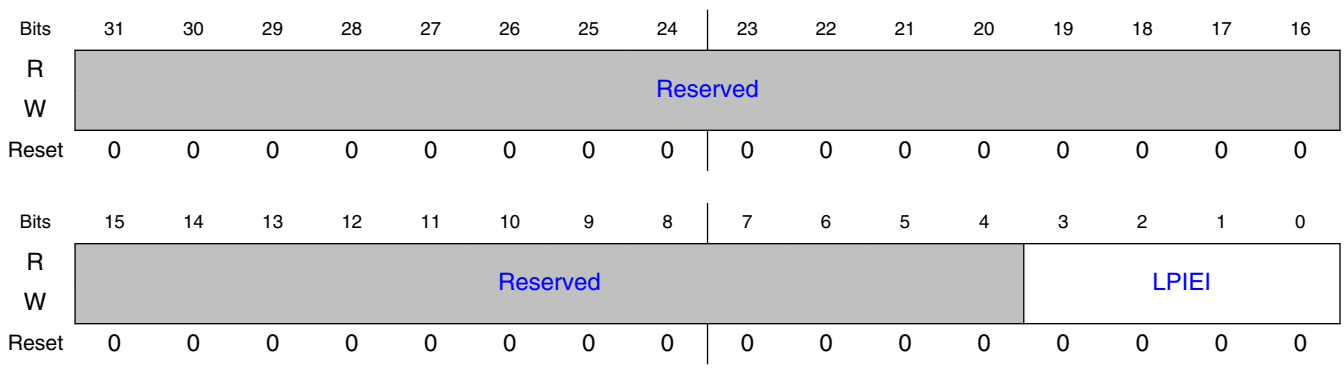
61.7.1.468.1 Offset

Register	Offset
DMA_AXI_LPI_ENTRY_INTERVAL	1040h

61.7.1.468.2 Function

This register is used to control the AXI LPI entry interval.

61.7.1.468.3 Diagram



61.7.1.468.4 Fields

Field	Function
31-4 —	Reserved
3-0 LPIEI	LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the DWC_ether_qos to enter into the AXI low power state 0 indicates 64 clock cycles

61.7.1.469 DMA_TBS_CTRL0 (DMA_TBS_CTRL0)

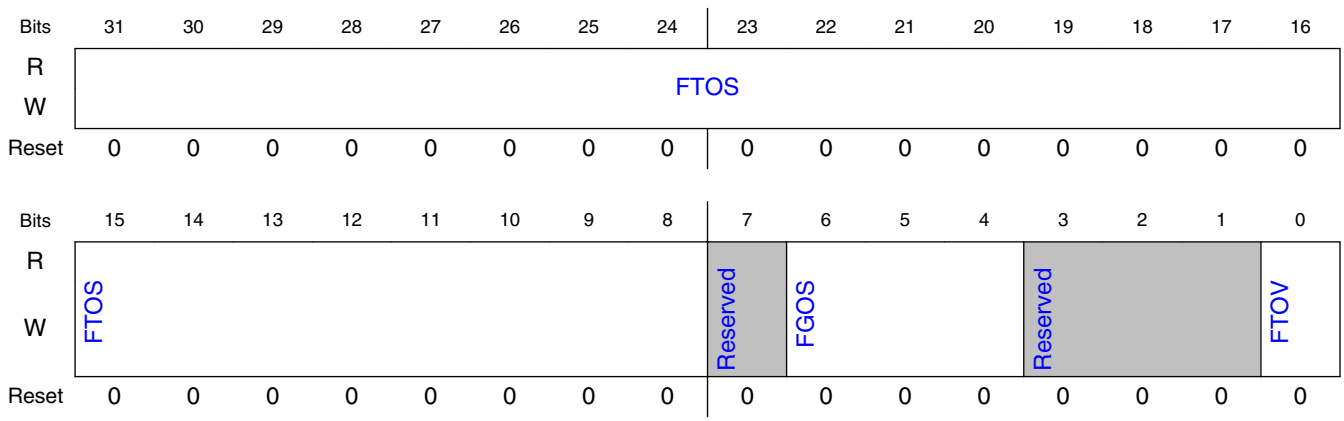
61.7.1.469.1 Offset

Register	Offset
DMA_TBS_CTRL0	1050h

61.7.1.469.2 Function

This register is used to control the TBS attributes.

61.7.1.469.3 Diagram



61.7.1.469.4 Fields

Field	Function
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0b - Fetch Time Offset is invalid 1b - Fetch Time Offset is valid

61.7.1.470 DMA_TBS_CTRL1 (DMA_TBS_CTRL1)

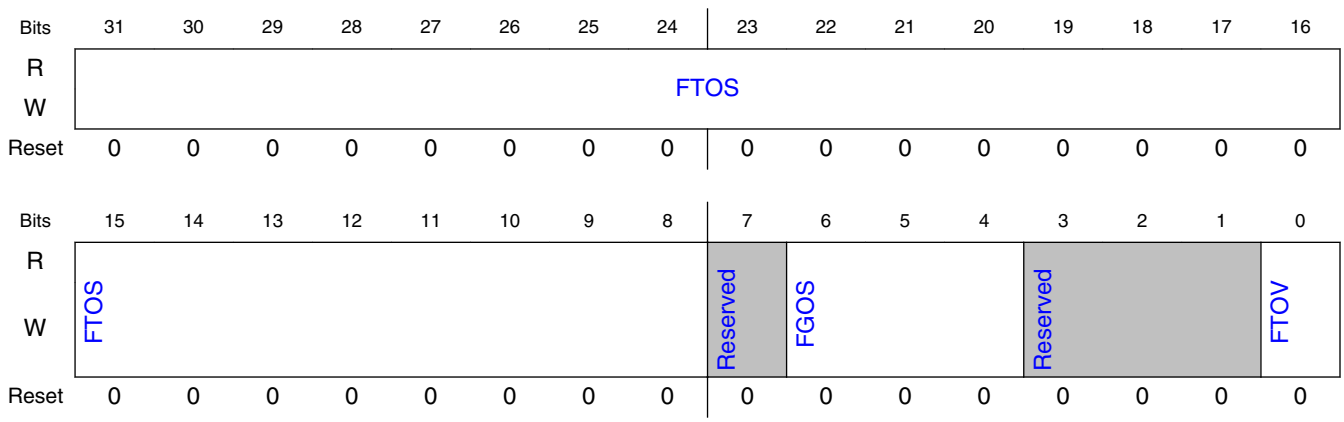
61.7.1.470.1 Offset

Register	Offset
DMA_TBS_CTRL1	1054h

61.7.1.470.2 Function

This register is used to control the TBS attributes.

61.7.1.470.3 Diagram



61.7.1.470.4 Fields

Field	Function
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved

Table continues on the next page...

Field	Function
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0b - Fetch Time Offset is invalid 1b - Fetch Time Offset is valid

61.7.1.471 DMA_TBS_CTRL2 (DMA_TBS_CTRL2)

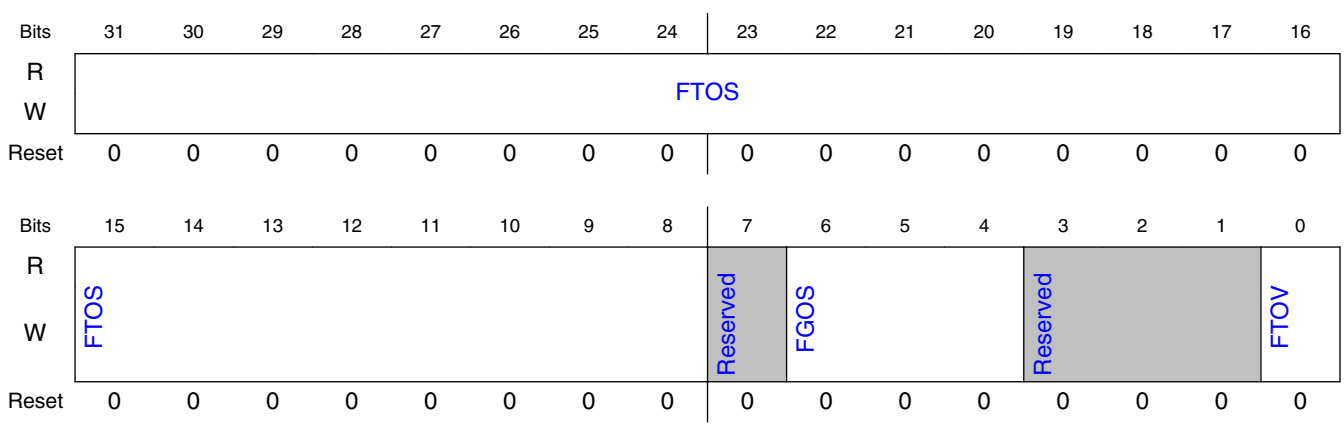
61.7.1.471.1 Offset

Register	Offset
DMA_TBS_CTRL2	1058h

61.7.1.471.2 Function

This register is used to control the TBS attributes.

61.7.1.471.3 Diagram



61.7.1.471.4 Fields

Field	Function
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.

Table continues on the next page...

Field	Function
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0b - Fetch Time Offset is invalid 1b - Fetch Time Offset is valid

61.7.1.472 DMA_TBS_CTRL3 (DMA_TBS_CTRL3)

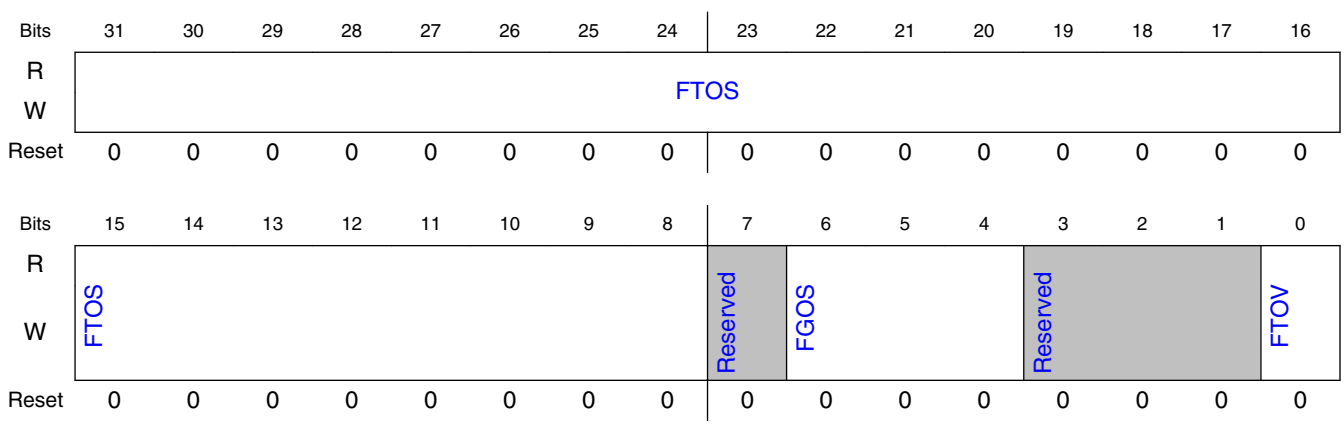
61.7.1.472.1 Offset

Register	Offset
DMA_TBS_CTRL3	105Ch

61.7.1.472.2 Function

This register is used to control the TBS attributes.

61.7.1.472.3 Diagram



61.7.1.472.4 Fields

Field	Function
31-8 FTOS	Fetch Time Offset The value in units of 256 nanoseconds, that has to be deducted from the Launch time to compute the Fetch Time. Max value: 999,999,999 ns, additionally should be smaller than CTR-1 value when ESTM mode is set since this value is a modulo CTR value.
7 —	Reserved
6-4 FGOS	Fetch GSN Offset The number GSN slots that must be deducted from the Launch GSN to compute the Fetch GSN. Value valid only when FTOV is set.
3-1 —	Reserved
0 FTOV	Fetch Time Offset Valid When set indicates the FTOS field is valid. When not set, indicates the Fetch Offset is not valid and the DMA engine can fetch the frames from host memory without any time restrictions. 0b - Fetch Time Offset is invalid 1b - Fetch Time Offset is valid

61.7.1.473 DMA Channel 0 Control (DMA_CH0_CONTROL)

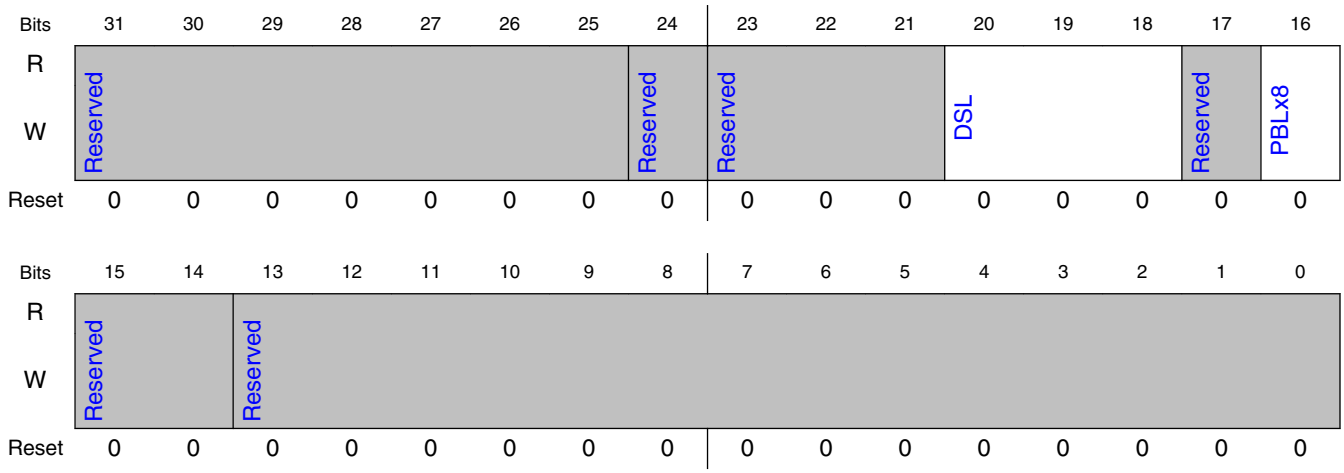
61.7.1.473.1 Offset

Register	Offset
DMA_CH0_CONTROL	1100h

61.7.1.473.2 Function

The DMA Channel 0 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

61.7.1.473.3 Diagram



61.7.1.473.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_TX_CONTROL and Bits[21:16] in DMA_CH0_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0b - 8xPBL mode is disabled 1b - 8xPBL mode is enabled
15-14 —	Reserved

Table continues on the next page...

Field	Function
13-0 —	Reserved

61.7.1.474 DMA Channel 0 Transmit Control (DMA_CH0_TX_CONTROL)

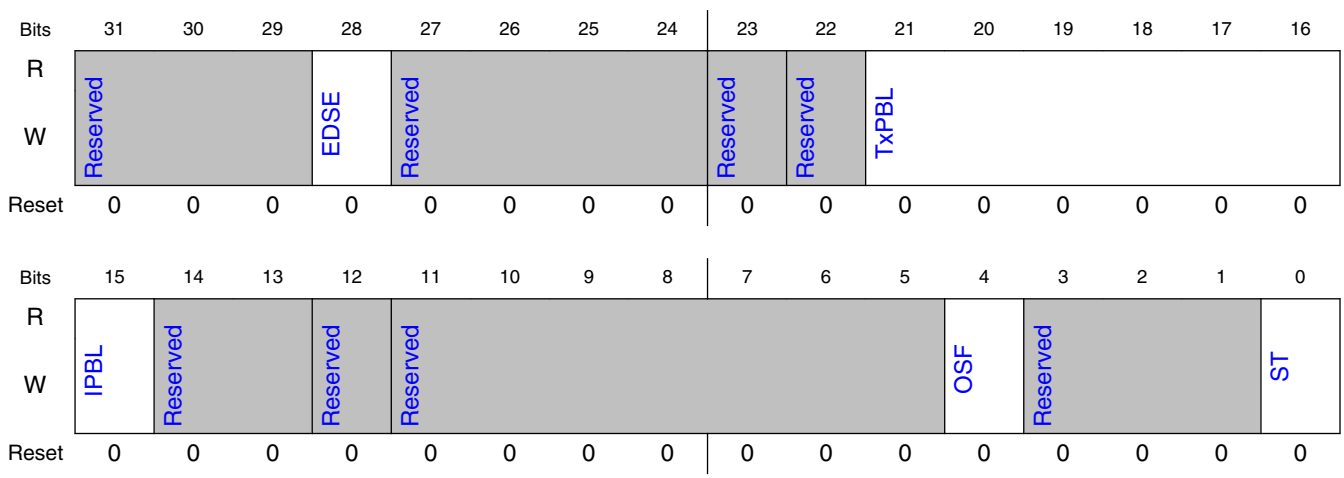
61.7.1.474.1 Offset

Register	Offset
DMA_CH0_TX_CONTROL	1104h

61.7.1.474.2 Function

The DMA Channel 0 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

61.7.1.474.3 Diagram



61.7.1.474.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Field	Function
—	
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0b - Enhanced Descriptor is disabled 1b - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ0_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0b - Ignore PBL Requirement is disabled 1b - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved

Table continues on the next page...

Field	Function
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0b - Operate on Second Packet disabled 1b - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH0_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0b - Stop Transmission Command 1b - Start Transmission Command

61.7.1.475 DMA Channel 0 Receive Control (DMA_CH0_RX_CONTROL)

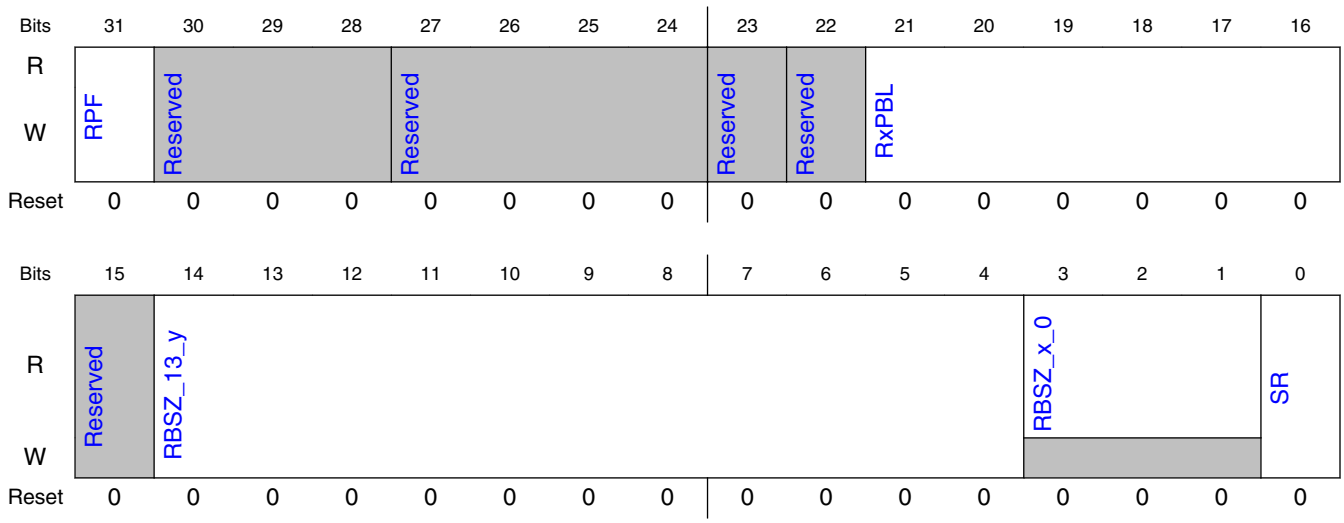
61.7.1.475.1 Offset

Register	Offset
DMA_CH0_RX_CONTROL	1108h

61.7.1.475.2 Function

The DMA Channel 0 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

61.7.1.475.3 Diagram



61.7.1.475.4 Fields

Field	Function
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0b - Rx Packet Flush is disabled 1b - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22	Reserved

Table continues on the next page...

Field	Function
—	
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH0_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ0_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH0_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Field	Function
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0b - Stop Receive 1b - Start Receive

61.7.1.476 Channel 0 Tx Descriptor List Address register (DMA_CH0_TXDESC_LIST_ADDRESS)

61.7.1.476.1 Offset

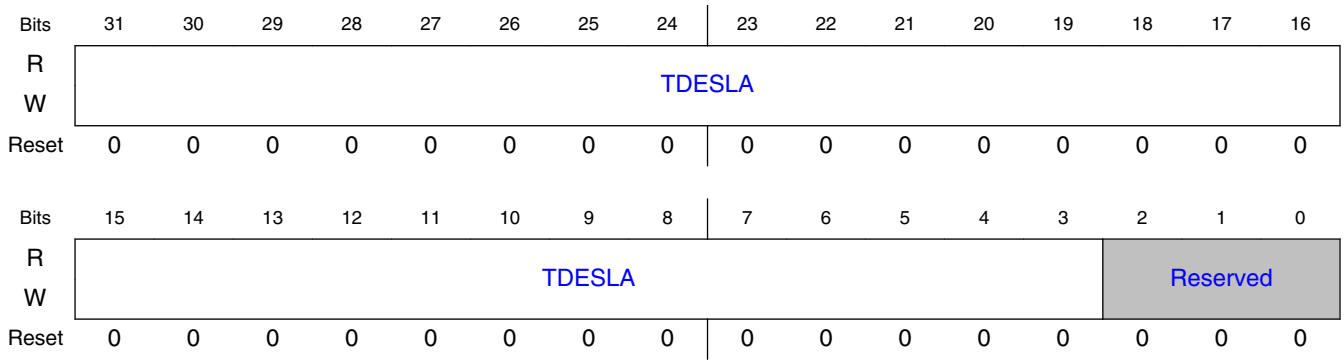
Register	Offset
DMA_CH0_TXDESC_LIST_ADDRESS	1114h

61.7.1.476.2 Function

The Channel 0 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

61.7.1.476.3 Diagram



61.7.1.476.4 Fields

Field	Function
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.477 Channel 0 Rx Descriptor List Address register (DMA_CH0_RXDESC_LIST_ADDRESS)

61.7.1.477.1 Offset

Register	Offset
DMA_CH0_RXDESC_LIST_ADDRESS	111Ch

61.7.1.477.2 Function

The Channel 0 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

61.7.1.477.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.477.4 Fields

Field	Function
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0	Reserved

Field	Function
—	

61.7.1.478 Channel 0 Tx Descriptor Tail Pointer (DMA_CH0_TXDESC_TAIL_POINTER)

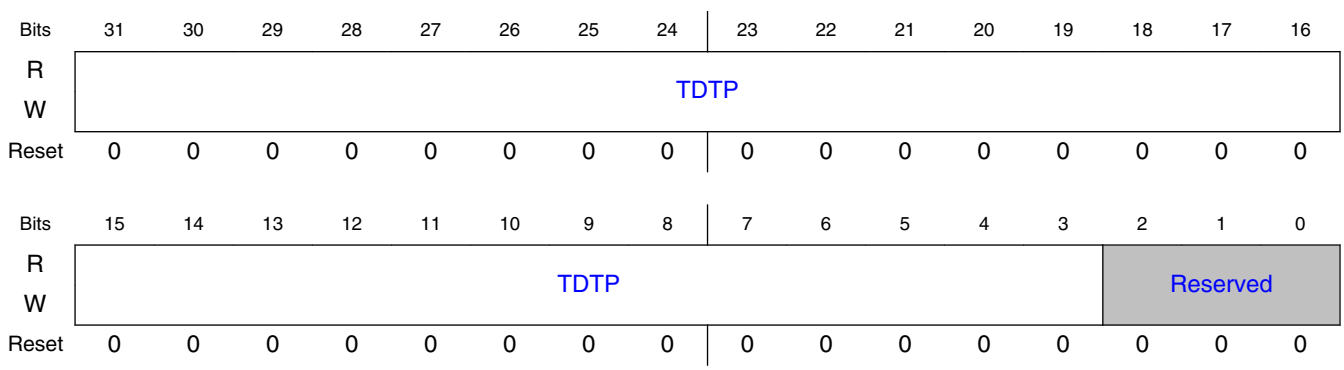
61.7.1.478.1 Offset

Register	Offset
DMA_CH0_TXDESC_TAIL_POINTER	1120h

61.7.1.478.2 Function

The Channel 0 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.478.3 Diagram



61.7.1.478.4 Fields

Field	Function
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Function
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.479 Channel 0 Rx Descriptor Tail Pointer (DMA_CH0_RXDESC_TAIL_POINTER)

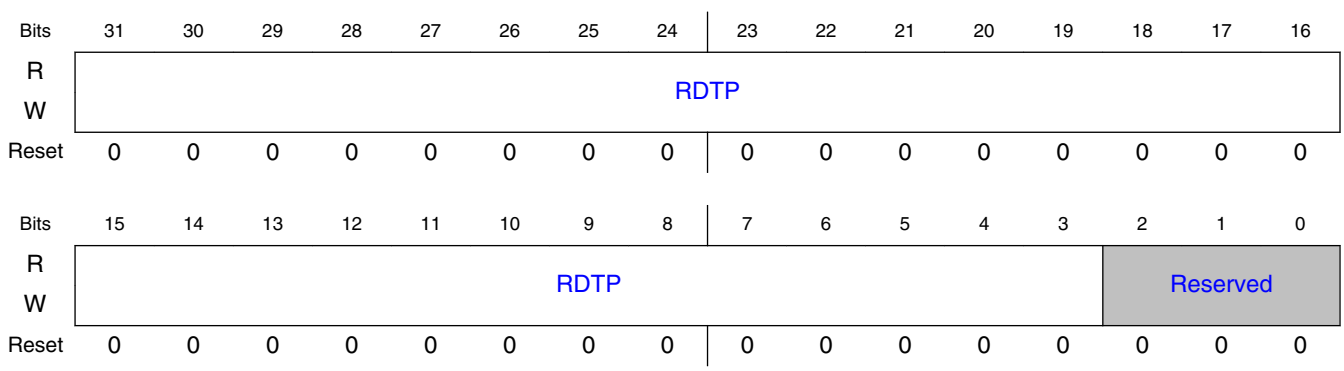
61.7.1.479.1 Offset

Register	Offset
DMA_CH0_RXDESC_TAIL_POINTER	1128h

61.7.1.479.2 Function

The Channel 0 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.479.3 Diagram



61.7.1.479.4 Fields

Field	Function
31-3	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring.
RDTP	Receive Descriptor Tail Pointer

Table continues on the next page...

Field	Function
	This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.480 Channel 0 Tx Descriptor Ring Length (DMA_CH0_TXDESC_RING_LENGTH)

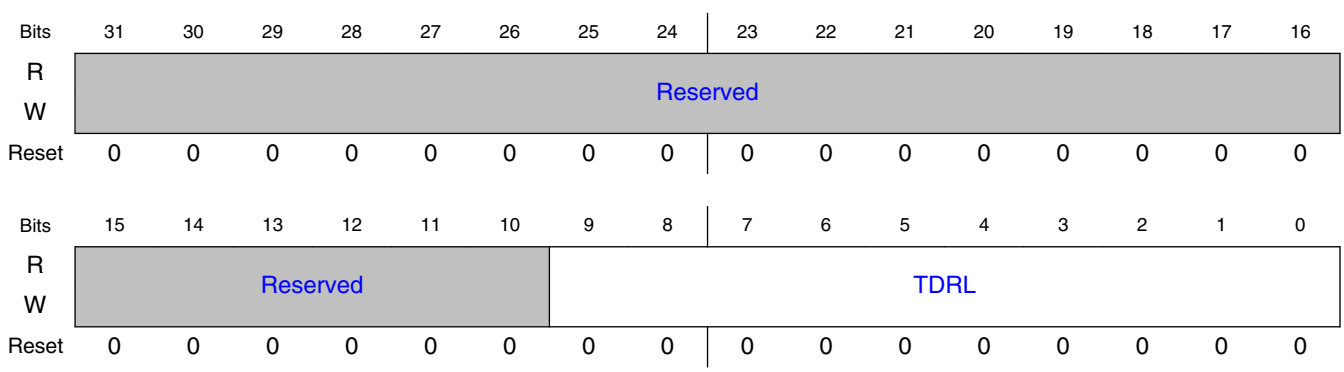
61.7.1.480.1 Offset

Register	Offset
DMA_CH0_TXDESC_RING_LENGTH	112Ch

61.7.1.480.2 Function

The Channel 0 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

61.7.1.480.3 Diagram



61.7.1.480.4 Fields

Field	Function
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.481 Channel 0 Receive Control 2 register (DMA_CH0_RX_CONTROL2)

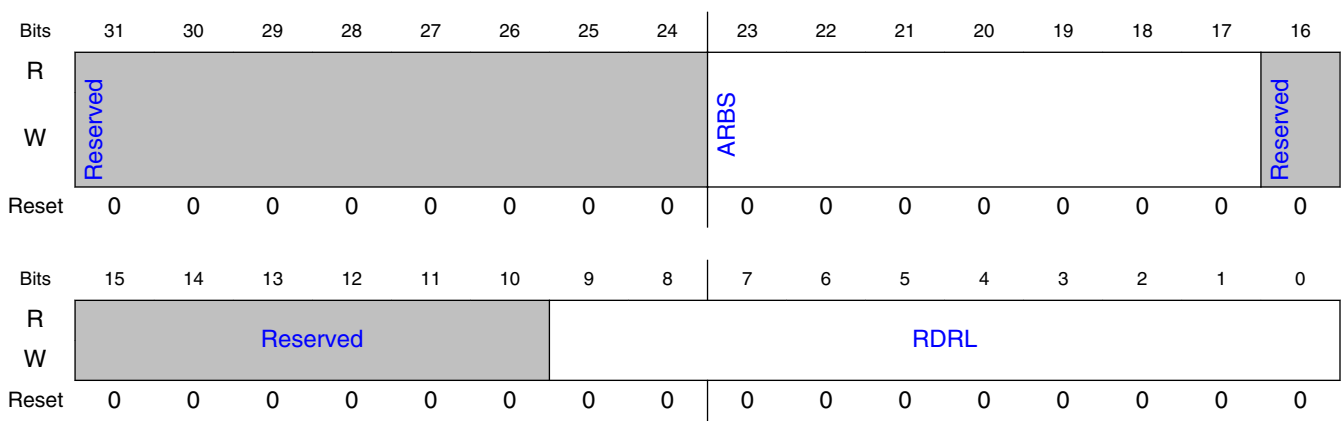
61.7.1.481.1 Offset

Register	Offset
DMA_CH0_RX_CONTROL2	1130h

61.7.1.481.2 Function

The Channel 0 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

61.7.1.481.3 Diagram



61.7.1.481.4 Fields

Field	Function
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH0_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.482 Channel 0 Interrupt Enable (DMA_CH0_INTERRUPT_ENABLE)

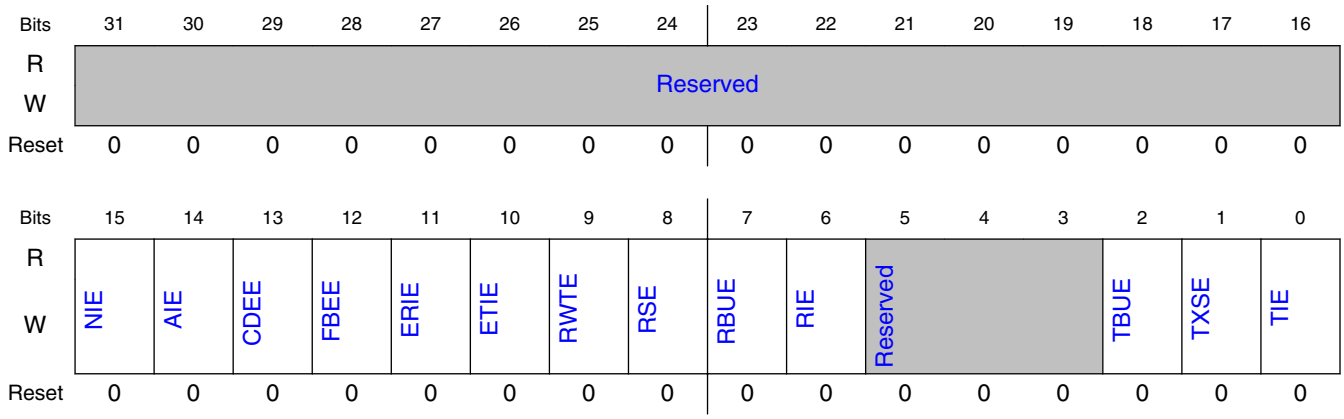
61.7.1.482.1 Offset

Register	Offset
DMA_CH0_INTERRUPT_ENABLE	1134h

61.7.1.482.2 Function

The Channel 0 Interrupt Enable register enables the interrupts reported by the Status register.

61.7.1.482.3 Diagram



61.7.1.482.4 Fields

Field	Function
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0b - Normal Interrupt Summary is disabled 1b - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled.

Table continues on the next page...

Field	Function
	0b - Abnormal Interrupt Summary is disabled 1b - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0b - Context Descriptor Error is disabled 1b - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0b - Fatal Bus Error is disabled 1b - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0b - Early Receive Interrupt is disabled 1b - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0b - Early Transmit Interrupt is disabled 1b - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0b - Receive Watchdog Timeout is disabled 1b - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0b - Receive Stopped is disabled 1b - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled.

Table continues on the next page...

Field	Function
	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0b - Receive Buffer Unavailable is disabled 1b - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0b - Receive Interrupt is disabled 1b - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0b - Transmit Buffer Unavailable is disabled 1b - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0b - Transmit Stopped is disabled 1b - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0b - Transmit Interrupt is disabled 1b - Transmit Interrupt is enabled

61.7.1.483 Channel 0 Receive Interrupt Watchdog Timer (DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER)

61.7.1.483.1 Offset

Register	Offset
DMA_CH0_RX_INTERRUPT_WATCHDOG_TIMER	1138h

61.7.1.483.2 Function

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA.

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH0_Status register.

61.7.1.483.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.483.4 Fields

Field	Function
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048

Table continues on the next page...

Field	Function
	For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

61.7.1.484 Channel 0 Slot Function Control and Status (DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS)

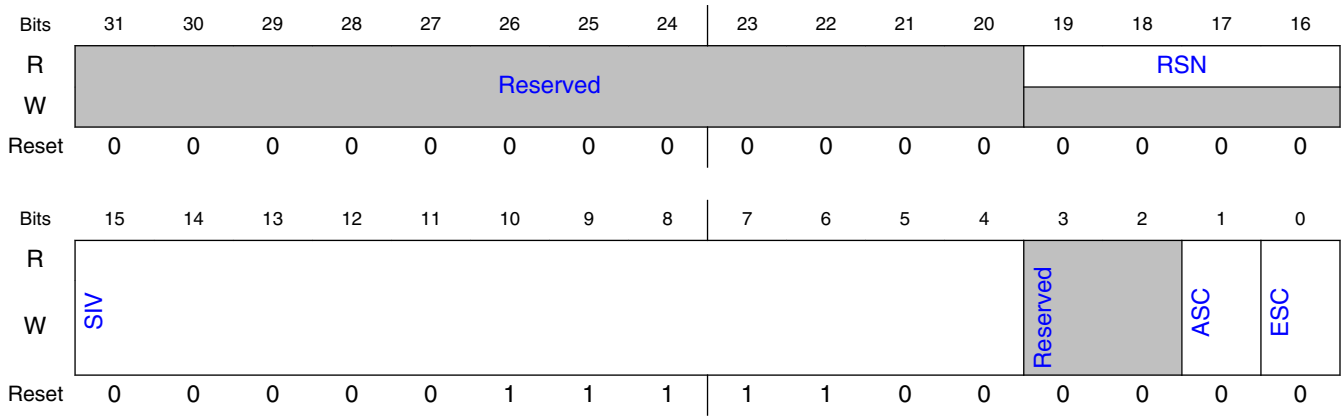
61.7.1.484.1 Offset

Register	Offset
DMA_CH0_SLOT_FUNCTION_CONTROL_STATUS	113Ch

61.7.1.484.2 Function

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

61.7.1.484.3 Diagram



61.7.1.484.4 Fields

Field	Function
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0b - Advance Slot Check is disabled 1b - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or

Field	Function
	- ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0b - Slot Comparison is disabled 1b - Slot Comparison is enabled

61.7.1.485 Channel 0 Current Application Transmit Descriptor (DMA_CH0_CURRENT_APP_TXDESC)

61.7.1.485.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXDESC	1144h

61.7.1.485.2 Function

The Channel 0 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

61.7.1.485.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.485.4 Fields

Field	Function
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.486 Channel 0 Current Application Receive Descriptor (DMA_CH0_CURRENT_APP_RXDESC)

61.7.1.486.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXDESC	114Ch

61.7.1.486.2 Function

The Channel 0 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

61.7.1.486.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.486.4 Fields

Field	Function
31-0	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer

Field	Function
CURRDESAPTR	The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.487 Channel 0 Current Application Transmit Buffer Address (DMA_CH0_CURRENT_APP_TXBUFFER)

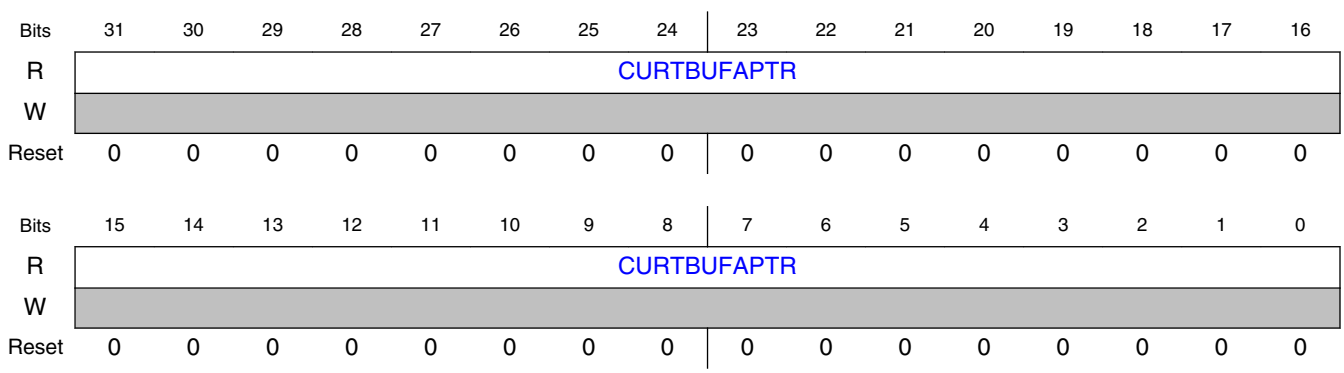
61.7.1.487.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_TXBUFFER	1154h

61.7.1.487.2 Function

The Channel 0 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

61.7.1.487.3 Diagram



61.7.1.487.4 Fields

Field	Function
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.488 Channel 0 Current Application Receive Buffer Address (DMA_CH0_CURRENT_APP_RXBUFFER)

61.7.1.488.1 Offset

Register	Offset
DMA_CH0_CURRENT_APP_RXBUFFER	115Ch

61.7.1.488.2 Function

The Channel 0 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

61.7.1.488.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.488.4 Fields

Field	Function
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.489 DMA Channel 0 Status (DMA_CH0_STATUS)

61.7.1.489.1 Offset

Register	Offset
DMA_CH0_STATUS	1160h

61.7.1.489.2 Function

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

61.7.1.489.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.489.4 Fields

Field	Function
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access

Table continues on the next page...

Field	Function
	-- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Normal Interrupt Summary status not detected 1b - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit.

Table continues on the next page...

Field	Function
	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH0_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Abnormal Interrupt Summary status not detected 1b - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all ones descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Context Descriptor Error status not detected 1b - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Fatal Bus Error status not detected 1b - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Receive Interrupt status not detected

Table continues on the next page...

Field	Function
	1b - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Transmit Interrupt status not detected 1b - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0b - Receive Watchdog Timeout status not detected 1b - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Process Stopped status not detected 1b - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Buffer Unavailable status not detected 1b - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Interrupt status not detected 1b - Receive Interrupt status detected
5-3 —	Reserved

Table continues on the next page...

Field	Function
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Buffer Unavailable status not detected 1b - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Process Stopped status not detected 1b - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Interrupt status not detected 1b - Transmit Interrupt status detected

61.7.1.490 Channel 0 Missed Frame Counter (DMA_CH0_MISS_FRAME_CNT)

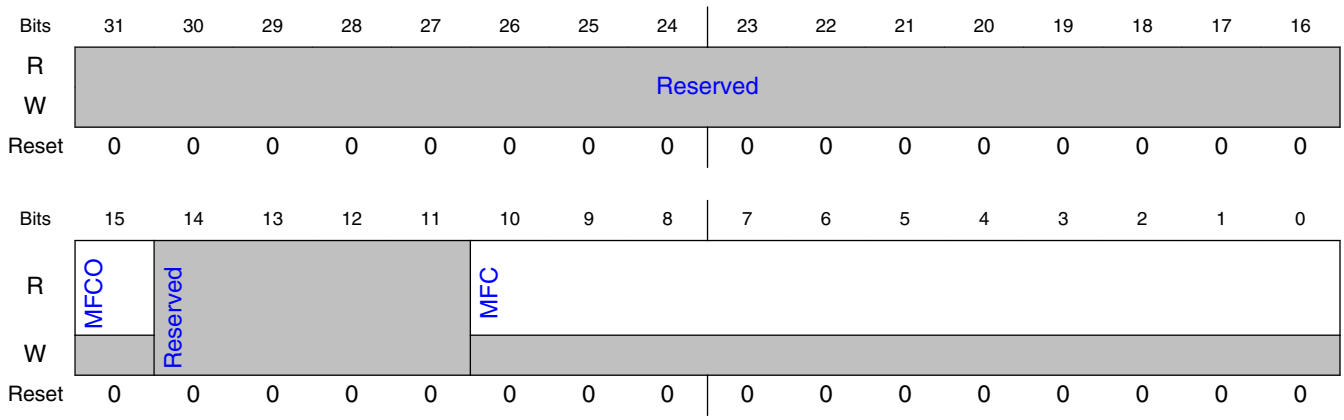
61.7.1.490.1 Offset

Register	Offset
DMA_CH0_MISS_FRAME_CNT	1164h

61.7.1.490.2 Function

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

61.7.1.490.3 Diagram



61.7.1.490.4 Fields

Field	Function
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Miss Frame Counter overflow not occurred 1b - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.491 Channel 0 RXP Frames Accepted Counter (DMA_CH0_RXP_ACCEPT_CNT)

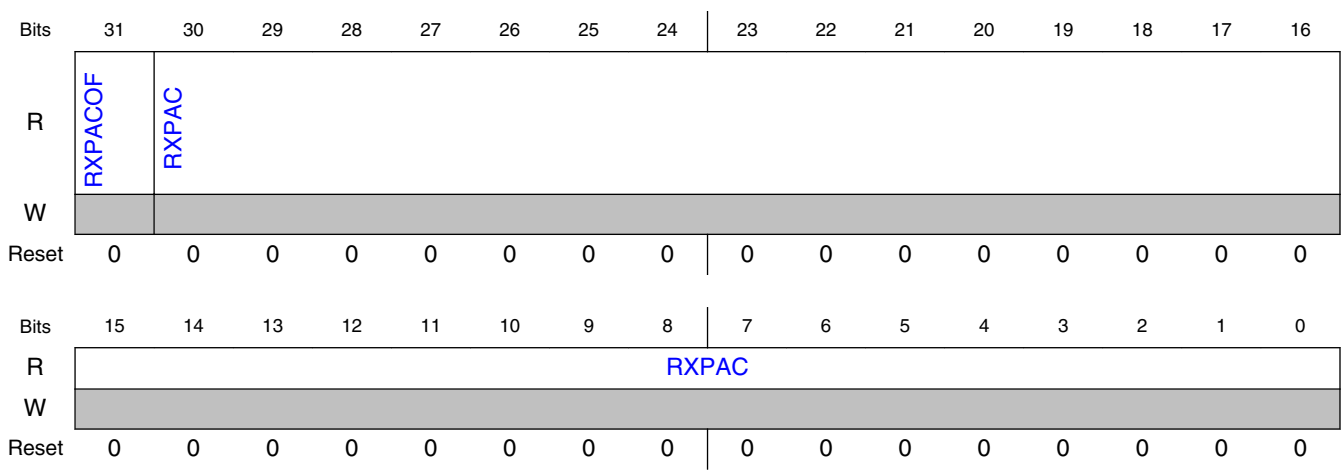
61.7.1.491.1 Offset

Register	Offset
DMA_CH0_RXP_ACCEPT_CNT	1168h

61.7.1.491.2 Function

The RXP_ACCEPT_CNT register provides the count of the number of frames accepted by Rx Parser.

61.7.1.491.3 Diagram



61.7.1.491.4 Fields

Field	Function
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Accept Counter overflow not occurred 1b - Rx Parser Accept Counter overflow occurred

Table continues on the next page...

Field	Function
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

61.7.1.492 DMA Channel 1 Control (DMA_CH1_CONTROL)

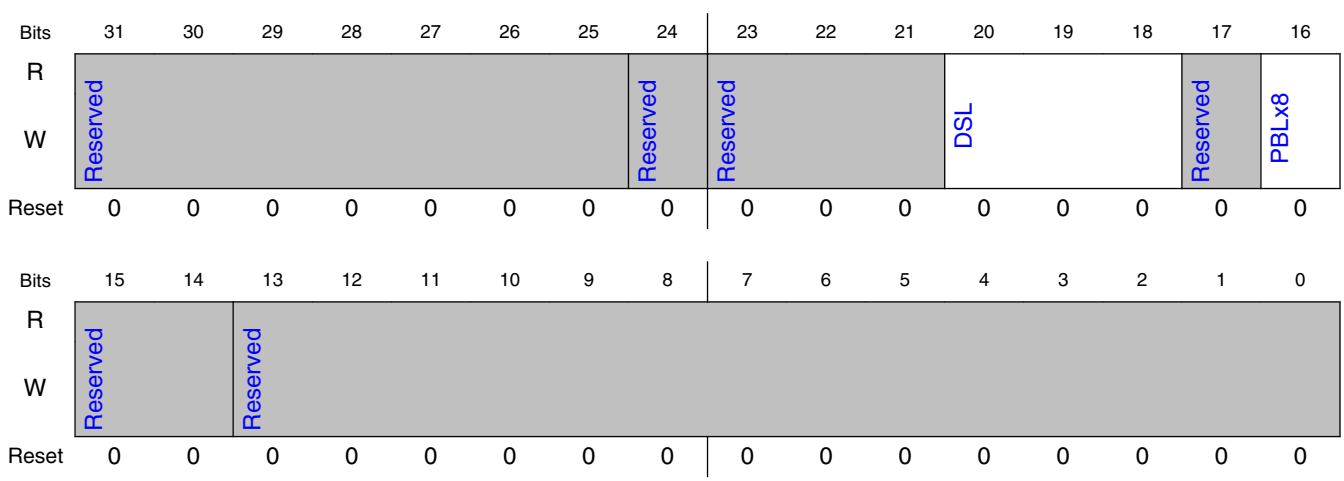
61.7.1.492.1 Offset

Register	Offset
DMA_CH1_CONTROL	1180h

61.7.1.492.2 Function

The DMA Channel 1 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

61.7.1.492.3 Diagram



61.7.1.492.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in TX_CONTROL and Bits[21:16] in DMA_CH1_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0b - 8xPBL mode is disabled 1b - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

61.7.1.493 DMA Channel 1 Transmit Control (DMA_CH1_TX_CONTROL)

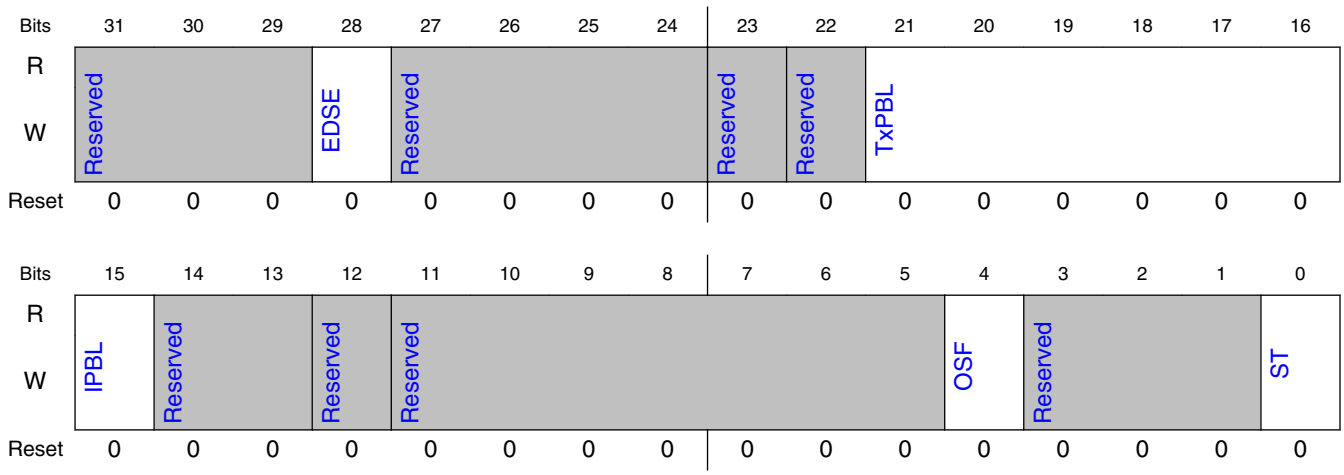
61.7.1.493.1 Offset

Register	Offset
DMA_CH1_TX_CONTROL	1184h

61.7.1.493.2 Function

The DMA Channel 1 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

61.7.1.493.3 Diagram



61.7.1.493.4 Fields

Field	Function
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0b - Enhanced Descriptor is disabled 1b - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length

Table continues on the next page...

Field	Function
	These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ1_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0b - Ignore PBL Requirement is disabled 1b - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0b - Operate on Second Packet disabled 1b - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH1_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped

Field	Function
	If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH1_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH1_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0b - Stop Transmission Command 1b - Start Transmission Command

61.7.1.494 DMA Channel 1 Receive Control (DMA_CH1_RX_CONTROL)

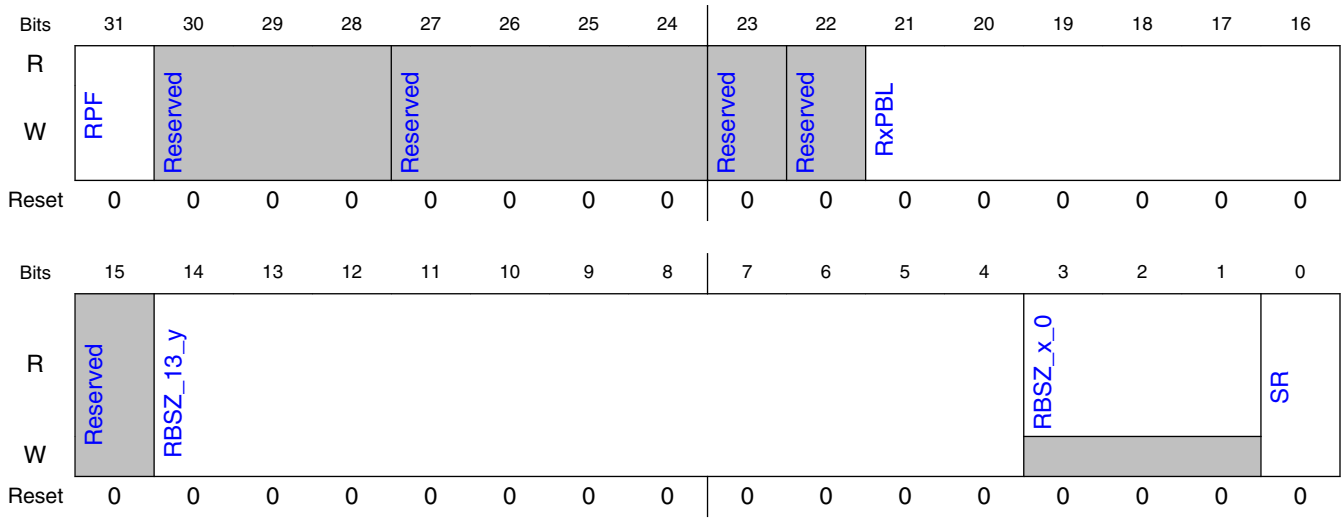
61.7.1.494.1 Offset

Register	Offset
DMA_CH1_RX_CONTROL	1188h

61.7.1.494.2 Function

The DMA Channel 1 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

61.7.1.494.3 Diagram



61.7.1.494.4 Fields

Field	Function
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0b - Rx Packet Flush is disabled 1b - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length

Table continues on the next page...

Field	Function
	These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ1_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH1_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH1_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH1_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state.

Field	Function
	0b - Stop Receive 1b - Start Receive

61.7.1.495 Channel 1 Tx Descriptor List Address (DMA_CH1_TXDESC_LIST_ADDRESS)

61.7.1.495.1 Offset

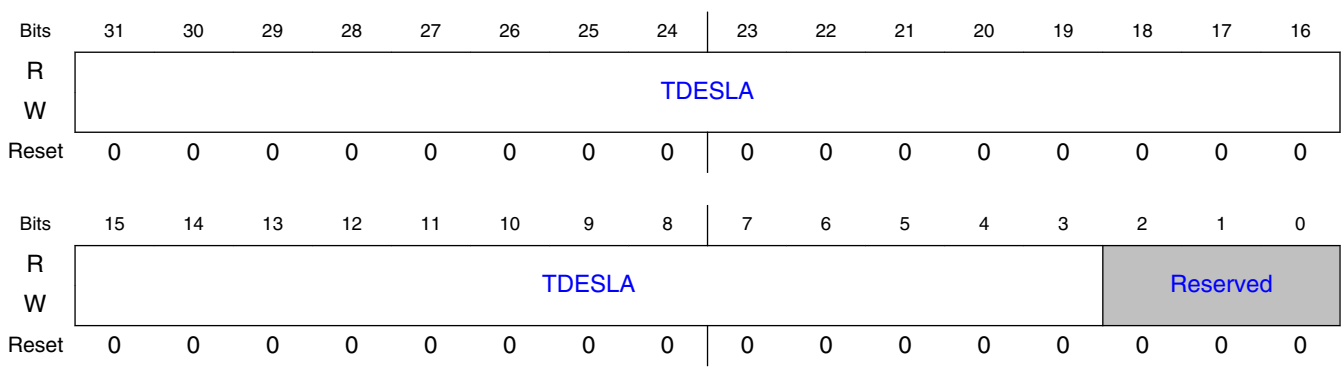
Register	Offset
DMA_CH1_TXDESC_LIST_ADDRESS	1194h

61.7.1.495.2 Function

The Channel 1 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH1_TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

61.7.1.495.3 Diagram



61.7.1.495.4 Fields

Field	Function
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.496 Channel 1 Rx Descriptor List Address (DMA_CH1_RXDESC_LIST_ADDRESS)

61.7.1.496.1 Offset

Register	Offset
DMA_CH1_RXDESC_LIST_ADDRESS	119Ch

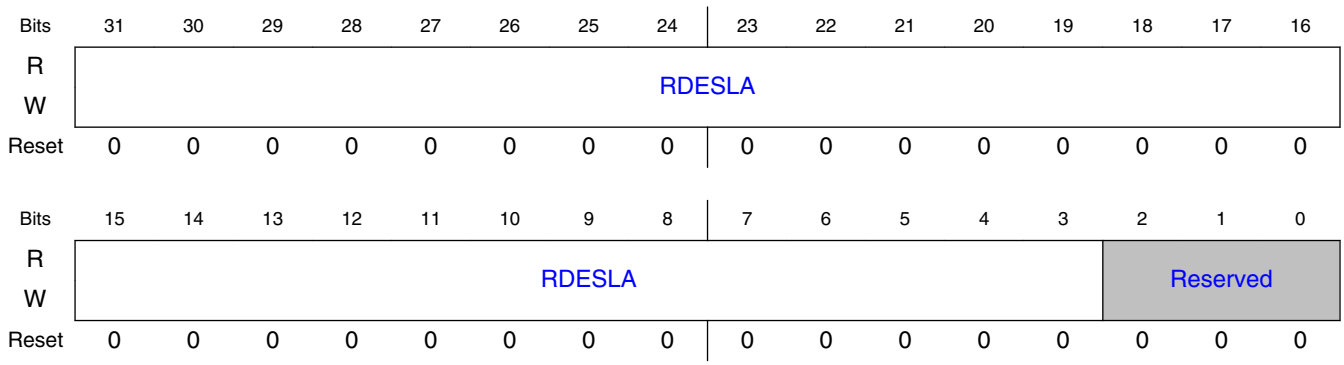
61.7.1.496.2 Function

The Channel 1 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA_CH1_RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

61.7.1.496.3 Diagram



61.7.1.496.4 Fields

Field	Function
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.497 Channel 1 Tx Descriptor Tail Pointer (DMA_CH1_TXDESC_TAIL_POINTER)

61.7.1.497.1 Offset

Register	Offset
DMA_CH1_TXDESC_TAIL_POINTER	11A0h

61.7.1.497.2 Function

The Channel 1 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.497.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.497.4 Fields

Field	Function
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.498 Channel 1 Rx Descriptor Tail Pointer (DMA_CH1_RXDESC_TAIL_POINTER)

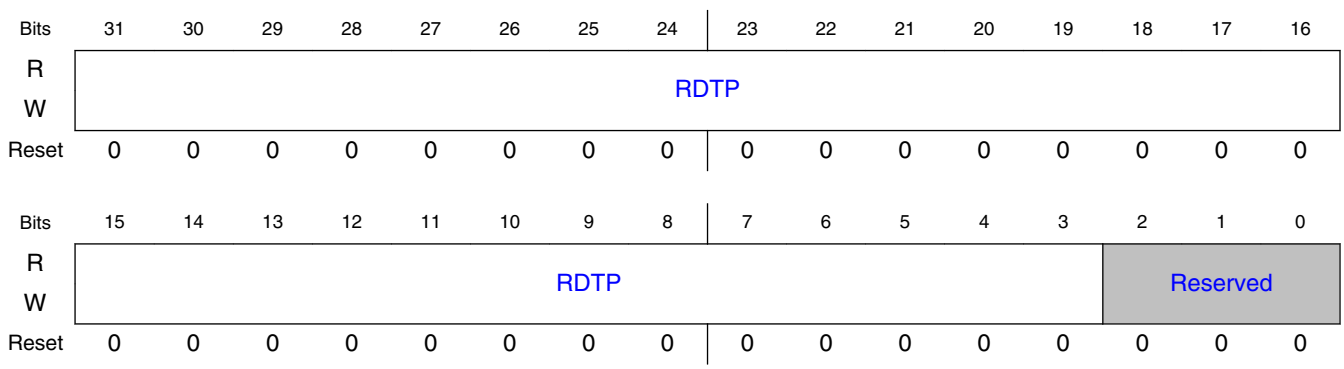
61.7.1.498.1 Offset

Register	Offset
DMA_CH1_RXDESC_TAIL_POINTER	11A8h

61.7.1.498.2 Function

The Channel 1 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.498.3 Diagram



61.7.1.498.4 Fields

Field	Function
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.499 Channel 1 Tx Descriptor Ring Length (DMA_CH1_TXDESC_RING_LENGTH)

61.7.1.499.1 Offset

Register	Offset
DMA_CH1_TXDESC_RING_LENGTH	11ACh

61.7.1.499.2 Function

The Channel 1 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

61.7.1.499.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TDRL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.499.4 Fields

Field	Function
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.500 Channel 1 Receive Control 2 register (DMA_CH1_RX_CONTROL2)

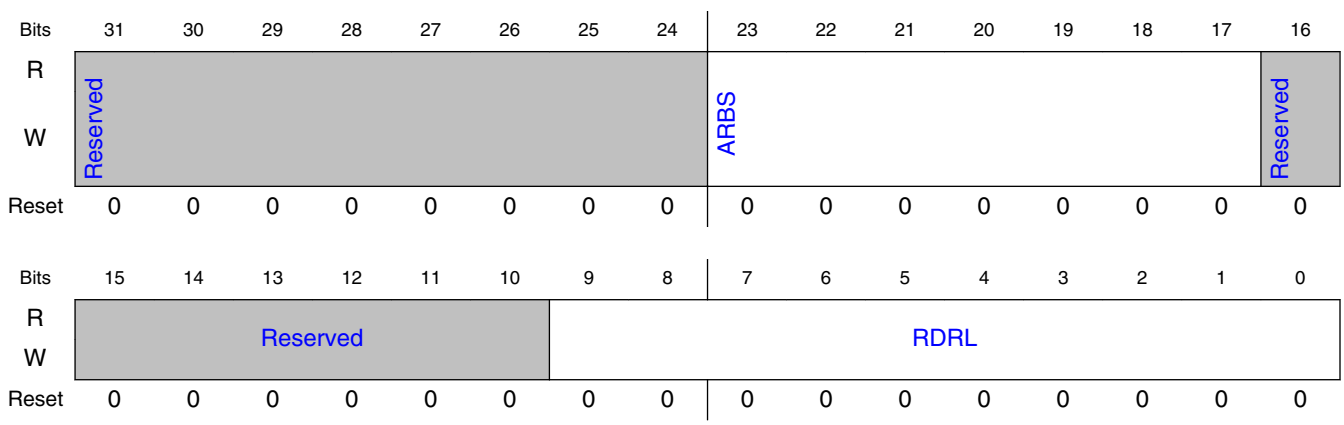
61.7.1.500.1 Offset

Register	Offset
DMA_CH1_RX_CONTROL2	11B0h

61.7.1.500.2 Function

The Channel 1 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

61.7.1.500.3 Diagram



61.7.1.500.4 Fields

Field	Function
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH1_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).

Table continues on the next page...

Field	Function
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.501 Channel 1 Interrupt Enable (DMA_CH1_INTERRUPT_ENABLE)

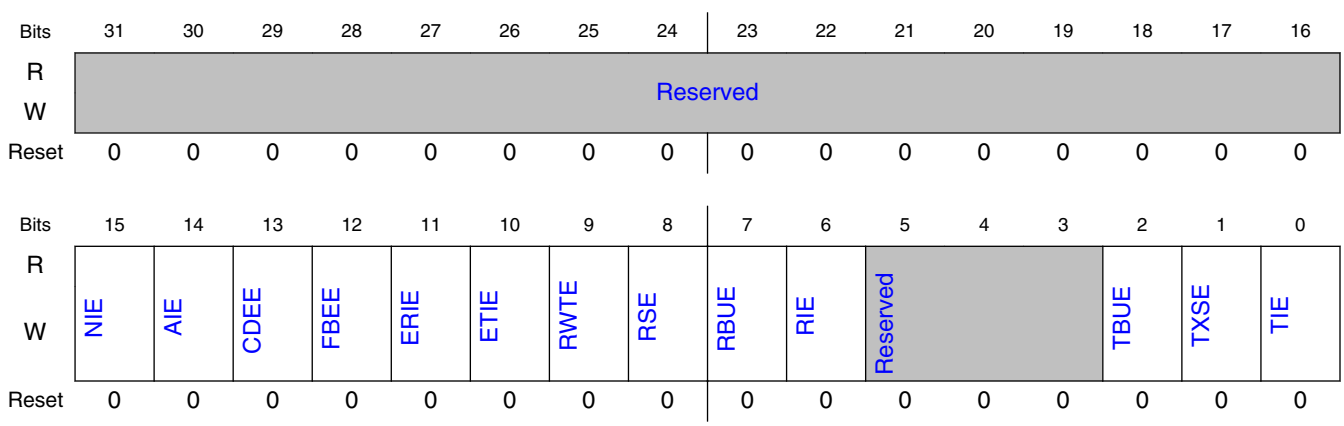
61.7.1.501.1 Offset

Register	Offset
DMA_CH1_INTERRUPT_ENABLE	11B4h

61.7.1.501.2 Function

The Channel 1 Interrupt Enable register enables the interrupts reported by the Status register.

61.7.1.501.3 Diagram



61.7.1.501.4 Fields

Field	Function
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0b - Normal Interrupt Summary is disabled 1b - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0b - Abnormal Interrupt Summary is disabled 1b - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0b - Context Descriptor Error is disabled 1b - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0b - Fatal Bus Error is disabled 1b - Fatal Bus Error is enabled

Table continues on the next page...

Field	Function
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0b - Early Receive Interrupt is disabled 1b - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0b - Early Transmit Interrupt is disabled 1b - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0b - Receive Watchdog Timeout is disabled 1b - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0b - Receive Stopped is disabled 1b - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0b - Receive Buffer Unavailable is disabled 1b - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0b - Receive Interrupt is disabled 1b - Receive Interrupt is enabled
5-3 —	Reserved
2	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled.

Table continues on the next page...

Field	Function
TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0b - Transmit Buffer Unavailable is disabled 1b - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0b - Transmit Stopped is disabled 1b - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0b - Transmit Interrupt is disabled 1b - Transmit Interrupt is enabled

61.7.1.502 Channel 1 Receive Interrupt Watchdog Timer (DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER)

61.7.1.502.1 Offset

Register	Offset
DMA_CH1_RX_INTERRUPT_WATCHDOG_TIMER	11B8h

61.7.1.502.2 Function

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH1_Status register.

61.7.1.502.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.502.4 Fields

Field	Function
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

61.7.1.503 Channel 1 Slot Function Control and Status (DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS)

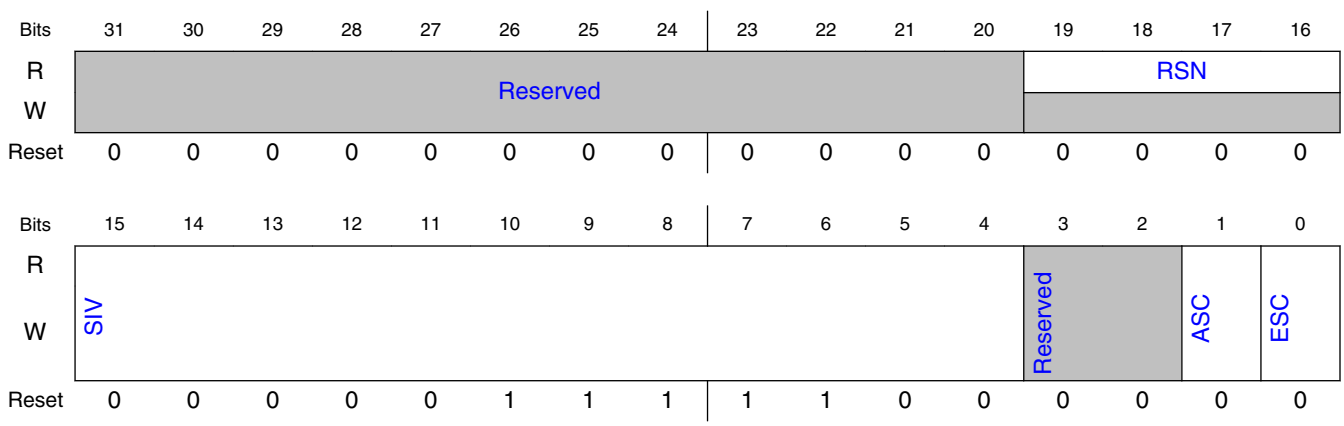
61.7.1.503.1 Offset

Register	Offset
DMA_CH1_SLOT_FUNCTION_CONTROL_STATUS	11BCh

61.7.1.503.2 Function

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

61.7.1.503.3 Diagram



61.7.1.503.4 Fields

Field	Function
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value

Table continues on the next page...

Field	Function
	This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0b - Advance Slot Check is disabled 1b - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0b - Slot Comparison is disabled 1b - Slot Comparison is enabled

61.7.1.504 Channel 1 Current Application Transmit Descriptor (DMA_CH1_CURRENT_APP_TXDESC)

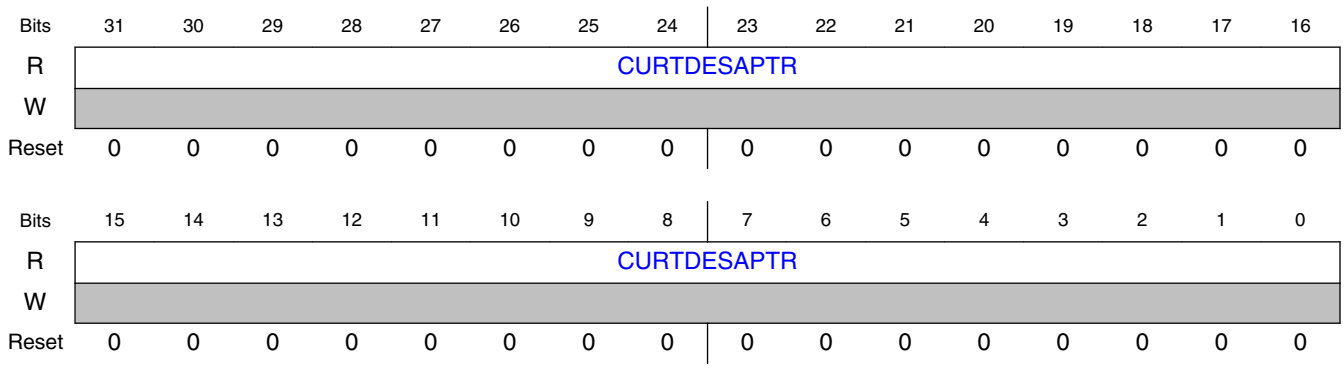
61.7.1.504.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXDESC	11C4h

61.7.1.504.2 Function

The Channel 1 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

61.7.1.504.3 Diagram



61.7.1.504.4 Fields

Field	Function
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.505 Channel 1 Current Application Receive Descriptor (DMA_CH1_CURRENT_APP_RXDESC)

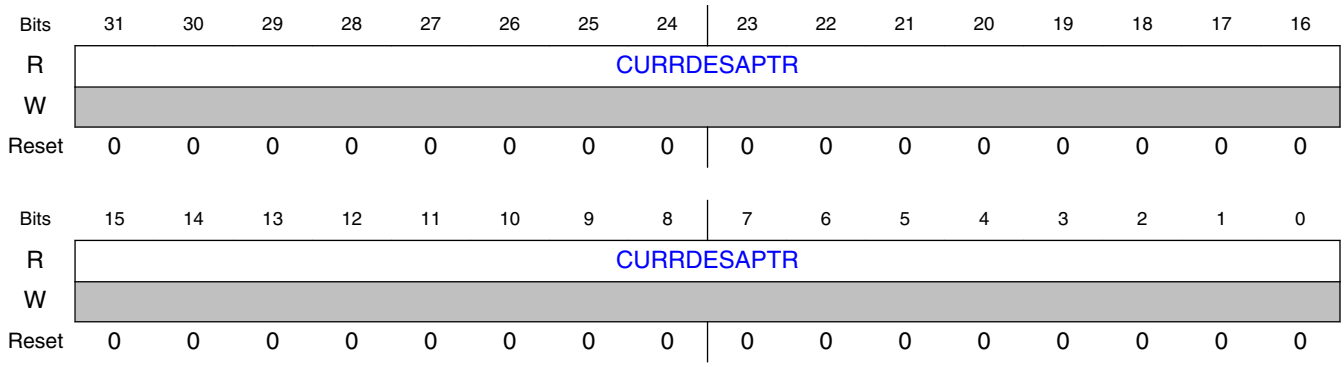
61.7.1.505.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXDESC	11CCh

61.7.1.505.2 Function

The Channel 1 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

61.7.1.505.3 Diagram



61.7.1.505.4 Fields

Field	Function
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.506 Channel 1 Current Application Transmit Buffer Address (DMA_CH1_CURRENT_APP_TXBUFFER)

61.7.1.506.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_TXBUFFER	11D4h

61.7.1.506.2 Function

The Channel 1 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

61.7.1.506.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.506.4 Fields

Field	Function
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.507 Channel 1 Current Application Receive Buffer Address (DMA_CH1_CURRENT_APP_RXBUFFER)

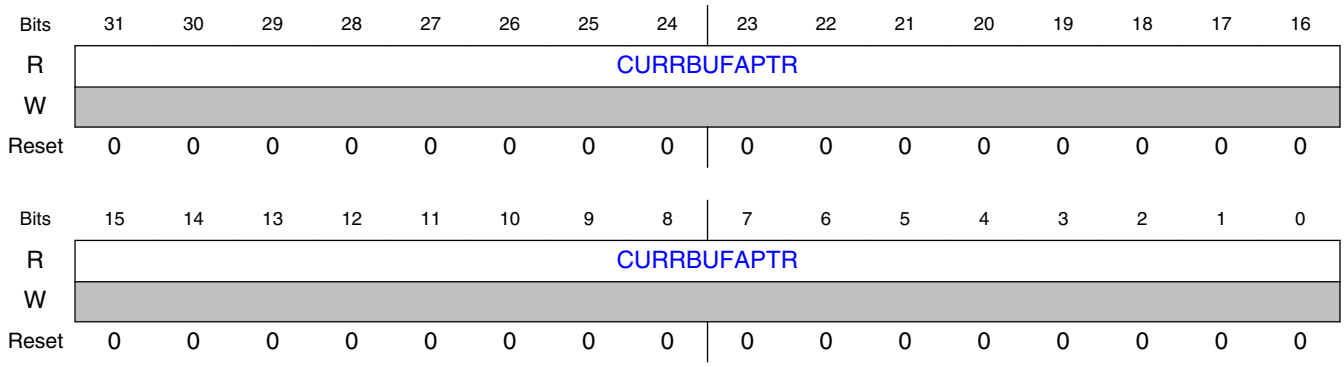
61.7.1.507.1 Offset

Register	Offset
DMA_CH1_CURRENT_APP_RXBUFFER	11DCh

61.7.1.507.2 Function

The Channel 1 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

61.7.1.507.3 Diagram



61.7.1.507.4 Fields

Field	Function
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.508 DMA Channel 1 Status (DMA_CH1_STATUS)

61.7.1.508.1 Offset

Register	Offset
DMA_CH1_STATUS	11E0h

61.7.1.508.2 Function

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

61.7.1.508.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.508.4 Fields

Field	Function
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access

Table continues on the next page...

Field	Function
	-- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH1_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH1_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Normal Interrupt Summary status not detected 1b - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Abnormal Interrupt Summary status not detected 1b - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Function
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Context Descriptor Error status not detected 1b - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Fatal Bus Error status not detected 1b - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Receive Interrupt status not detected 1b - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Transmit Interrupt status not detected 1b - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0b - Receive Watchdog Timeout status not detected 1b - Receive Watchdog Timeout status detected
8	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state.

Table continues on the next page...

Field	Function
RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Process Stopped status not detected 1b - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Buffer Unavailable status not detected 1b - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Interrupt status not detected 1b - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Buffer Unavailable status not detected 1b - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped.

Table continues on the next page...

Field	Function
	Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Process Stopped status not detected 1b - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Interrupt status not detected 1b - Transmit Interrupt status detected

61.7.1.509 Channel 1 Missed Frame Counter (DMA_CH1_MISS_FRAME_CNT)

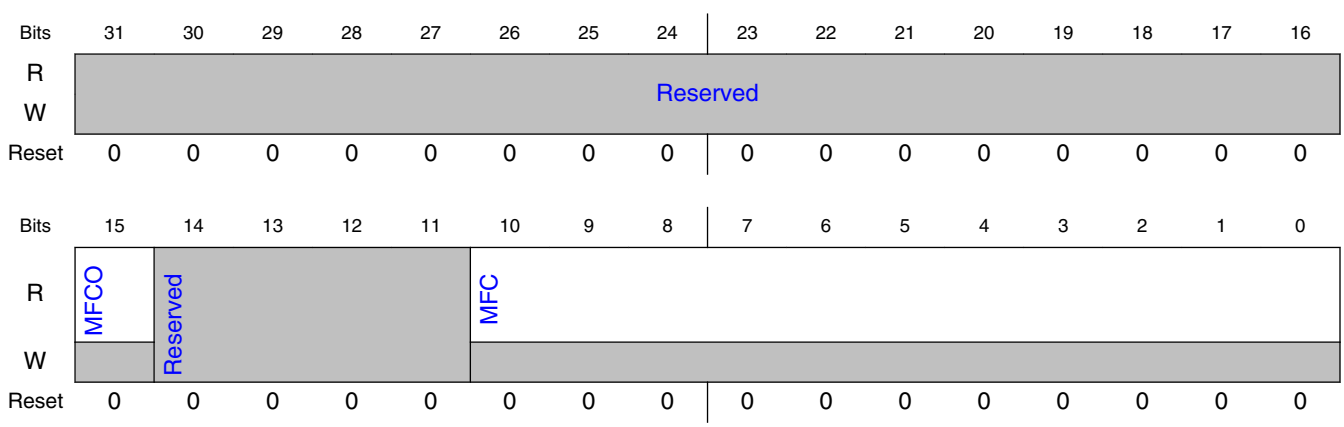
61.7.1.509.1 Offset

Register	Offset
DMA_CH1_MISS_FRAME_CNT	11E4h

61.7.1.509.2 Function

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in RX_CONTROL register.

61.7.1.509.3 Diagram



61.7.1.509.4 Fields

Field	Function
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Miss Frame Counter overflow not occurred 1b - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH1_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.510 Channel 1 RXP Frames Accepted Counter (DMA_CH1_RXP_ACCEPT_CNT)

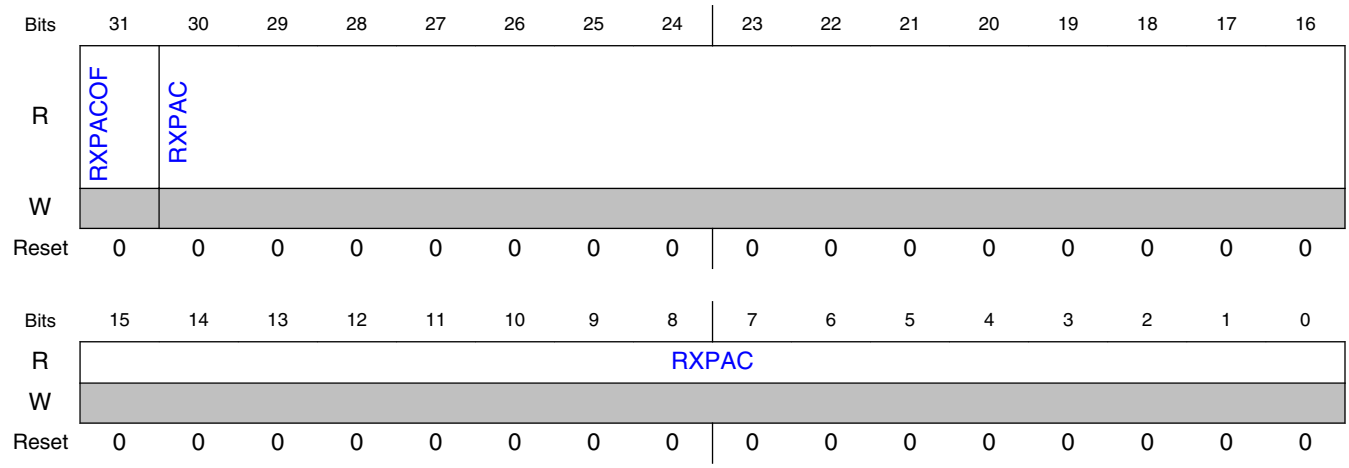
61.7.1.510.1 Offset

Register	Offset
DMA_CH1_RXP_ACCEPT_CNT	11E8h

61.7.1.510.2 Function

The DMA_CH1_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

61.7.1.510.3 Diagram



61.7.1.510.4 Fields

Field	Function
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Accept Counter overflow not occurred 1b - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

61.7.1.511 DMA Channel 2 Control (DMA_CH2_CONTROL)

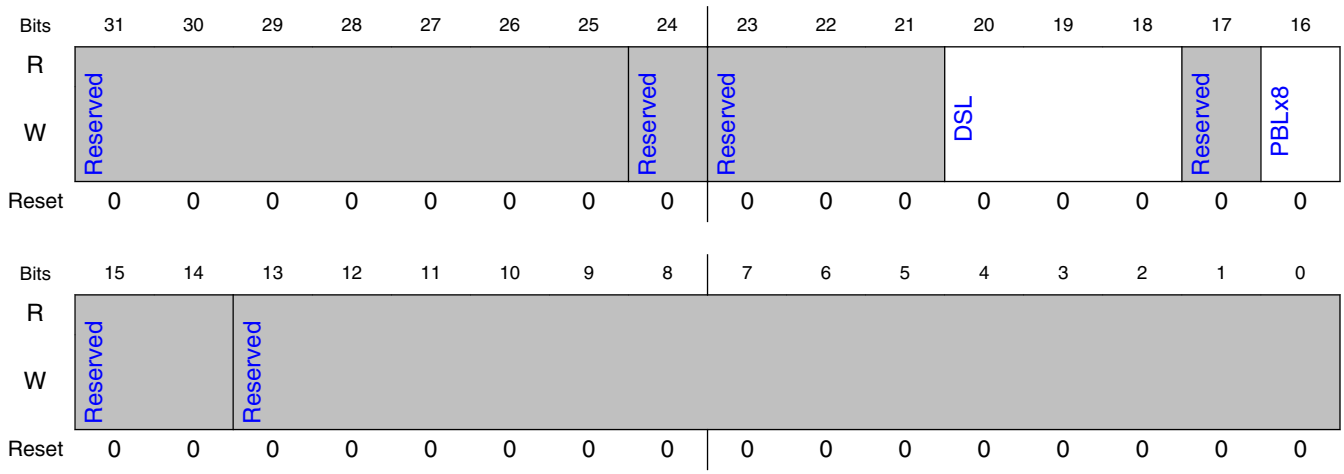
61.7.1.511.1 Offset

Register	Offset
DMA_CH2_CONTROL	1200h

61.7.1.511.2 Function

The DMA Channel 2 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

61.7.1.511.3 Diagram



61.7.1.511.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times. 8xPBL mode

Table continues on the next page...

Field	Function
	When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH2_TX_CONTROL and Bits[21:16] in DMA_CH2_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0b - 8xPBL mode is disabled 1b - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

61.7.1.512 DMA Channel 2 Transmit Control (DMA_CH2_TX_CONTROL)

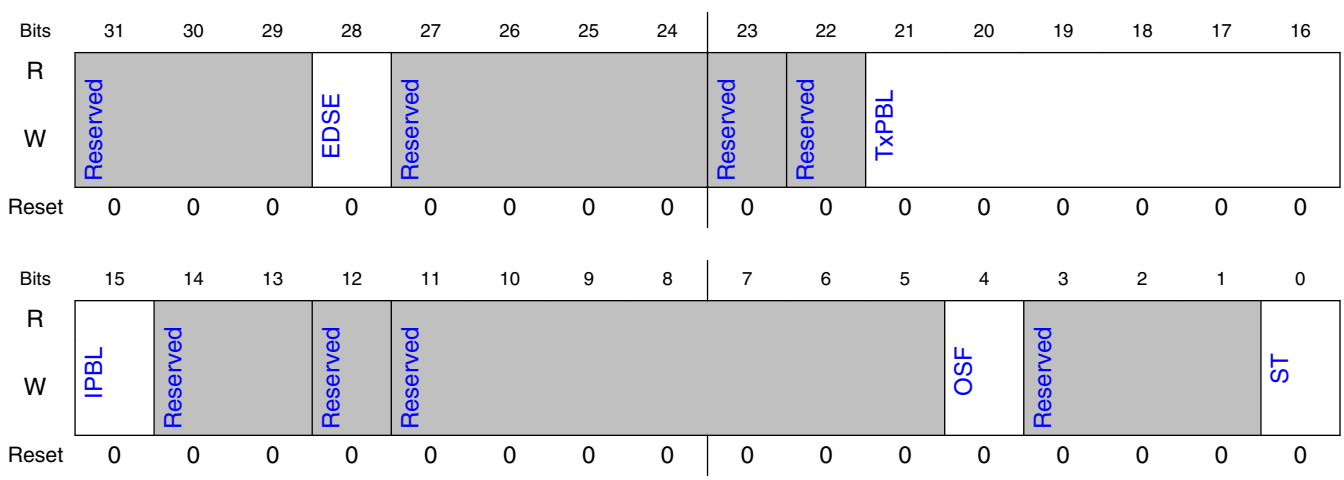
61.7.1.512.1 Offset

Register	Offset
DMA_CH2_TX_CONTROL	1204h

61.7.1.512.2 Function

The DMA Channel 2 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

61.7.1.512.3 Diagram



61.7.1.512.4 Fields

Field	Function
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0b - Enhanced Descriptor is disabled 1b - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH2_CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ2_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0b - Ignore PBL Requirement is disabled 1b - Ignore PBL Requirement is enabled

Table continues on the next page...

Field	Function
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0b - Operate on Second Packet disabled 1b - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH2_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH2_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH2_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH2_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0b - Stop Transmission Command 1b - Start Transmission Command

61.7.1.513 DMA Channel 2 Receive Control (DMA_CH2_RX_CONTROL)

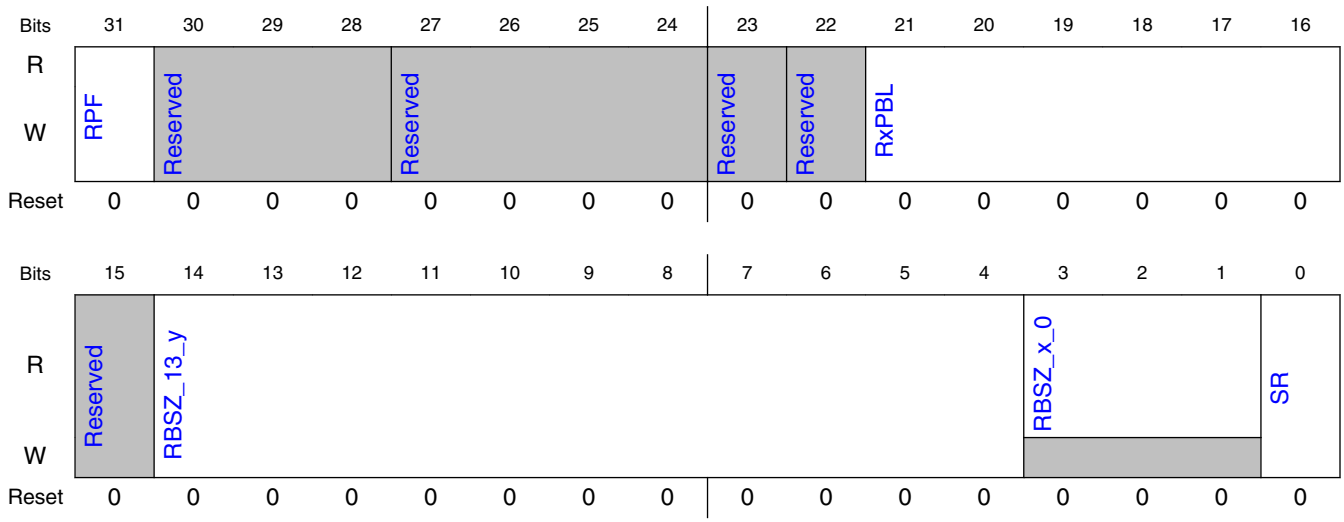
61.7.1.513.1 Offset

Register	Offset
DMA_CH2_RX_CONTROL	1208h

61.7.1.513.2 Function

The DMA Channel 2 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

61.7.1.513.3 Diagram



61.7.1.513.4 Fields

Field	Function
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0b - Rx Packet Flush is disabled 1b - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved

Table continues on the next page...

Field	Function
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH2_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ2_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH2_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Field	Function
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH2_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH2_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0b - Stop Receive 1b - Start Receive

61.7.1.514 Channel 2 Tx Descriptor List Address (DMA_CH2_TXDESC_LIST_ADDRESS)

61.7.1.514.1 Offset

Register	Offset
DMA_CH2_TXDESC_LIST_ADDRESS	1214h

61.7.1.514.2 Function

The Channel 2 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

61.7.1.514.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.514.4 Fields

Field	Function
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.515 Channel 2 Rx Descriptor List Address (DMA_CH2_RXDESC_LIST_ADDRESS)

61.7.1.515.1 Offset

Register	Offset
DMA_CH2_RXDESC_LIST_ADDRESS	121Ch

61.7.1.515.2 Function

The Channel 2 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

61.7.1.515.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.515.4 Fields

Field	Function
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0	Reserved

Field	Function
—	

61.7.1.516 Channel 2 Tx Descriptor Tail Pointer (DMA_CH2_TXDESC_TAIL_POINTER)

61.7.1.516.1 Offset

Register	Offset
DMA_CH2_TXDESC_TAIL_POINTER	1220h

61.7.1.516.2 Function

The Channel 2 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.516.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.516.4 Fields

Field	Function
31-3	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring.
TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Function
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.517 Channel 2 Rx Descriptor Tail Pointer (DMA_CH2_RXDESC_TAIL_POINTER)

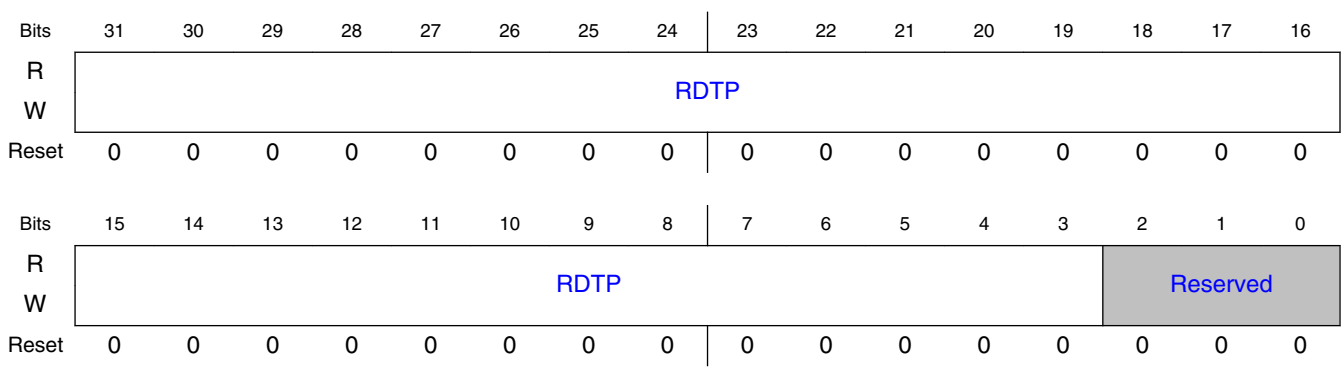
61.7.1.517.1 Offset

Register	Offset
DMA_CH2_RXDESC_TAIL_POINTER	1228h

61.7.1.517.2 Function

The Channel 2 Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.517.3 Diagram



61.7.1.517.4 Fields

Field	Function
31-3	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring.
RDTP	Receive Descriptor Tail Pointer

Table continues on the next page...

Field	Function
	This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.518 Channel 2 Tx Descriptor Ring Length (DMA_CH2_TXDESC_RING_LENGTH)

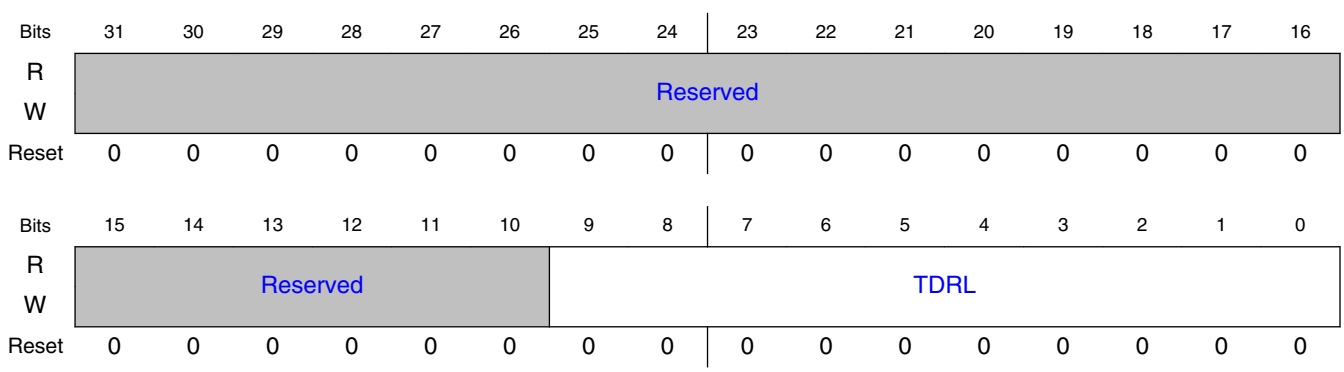
61.7.1.518.1 Offset

Register	Offset
DMA_CH2_TXDESC_RING_LENGTH	122Ch

61.7.1.518.2 Function

The Channel 2 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

61.7.1.518.3 Diagram



61.7.1.518.4 Fields

Field	Function
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.519 Channel 2 Receive Control 2 register (DMA_CH2_RX_CONTROL2)

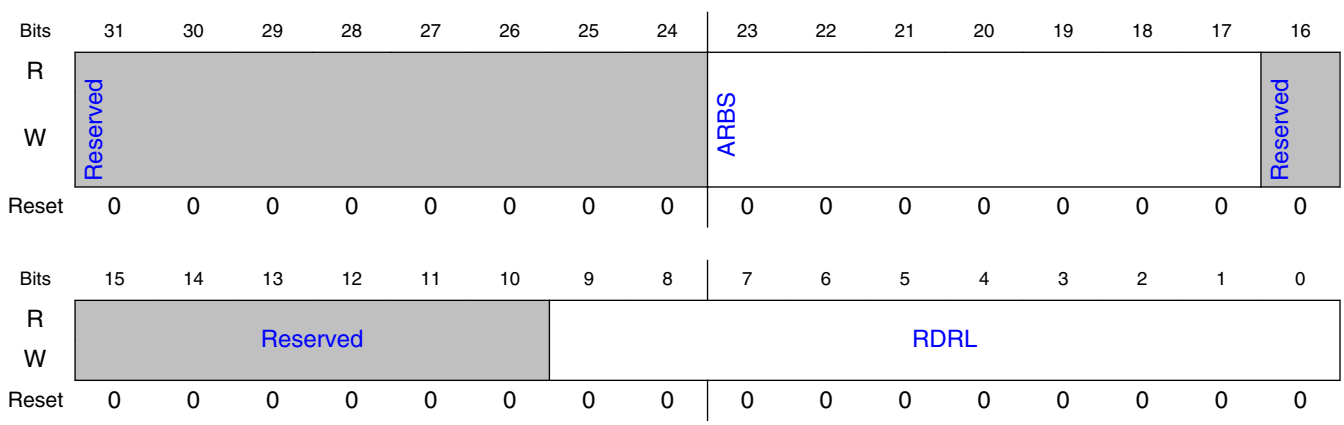
61.7.1.519.1 Offset

Register	Offset
DMA_CH2_RX_CONTROL2	1230h

61.7.1.519.2 Function

The Channel 2 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

61.7.1.519.3 Diagram



61.7.1.519.4 Fields

Field	Function
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.520 Channel 2 Interrupt Enable (DMA_CH2_INTERRUPT_ENABLE)

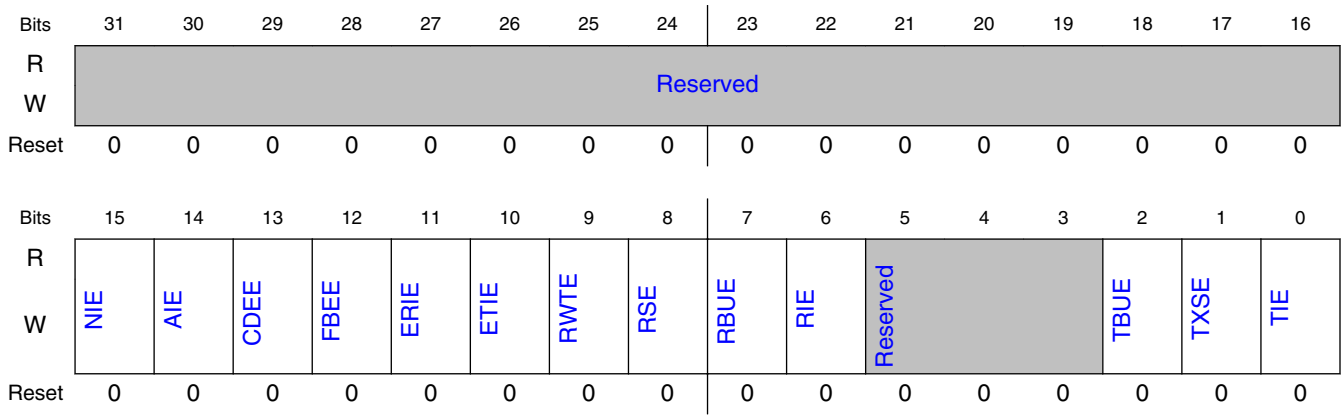
61.7.1.520.1 Offset

Register	Offset
DMA_CH2_INTERRUPT_ENABLE	1234h

61.7.1.520.2 Function

The Channel 2 Interrupt Enable register enables the interrupts reported by the Status register.

61.7.1.520.3 Diagram



61.7.1.520.4 Fields

Field	Function
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0b - Normal Interrupt Summary is disabled 1b - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled.

Table continues on the next page...

Field	Function
	0b - Abnormal Interrupt Summary is disabled 1b - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0b - Context Descriptor Error is disabled 1b - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0b - Fatal Bus Error is disabled 1b - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0b - Early Receive Interrupt is disabled 1b - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0b - Early Transmit Interrupt is disabled 1b - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0b - Receive Watchdog Timeout is disabled 1b - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0b - Receive Stopped is disabled 1b - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled.

Table continues on the next page...

Field	Function
	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0b - Receive Buffer Unavailable is disabled 1b - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0b - Receive Interrupt is disabled 1b - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0b - Transmit Buffer Unavailable is disabled 1b - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0b - Transmit Stopped is disabled 1b - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0b - Transmit Interrupt is disabled 1b - Transmit Interrupt is enabled

61.7.1.521 Channel 2 Receive Interrupt Watchdog Timer (DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER)

61.7.1.521.1 Offset

Register	Offset
DMA_CH2_RX_INTERRUPT_WATCHDOG_TIMER	1238h

61.7.1.521.2 Function

The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH2_Status register.

61.7.1.521.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.521.4 Fields

Field	Function
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

61.7.1.522 Channel 2 Slot Function Control and Status (DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS)

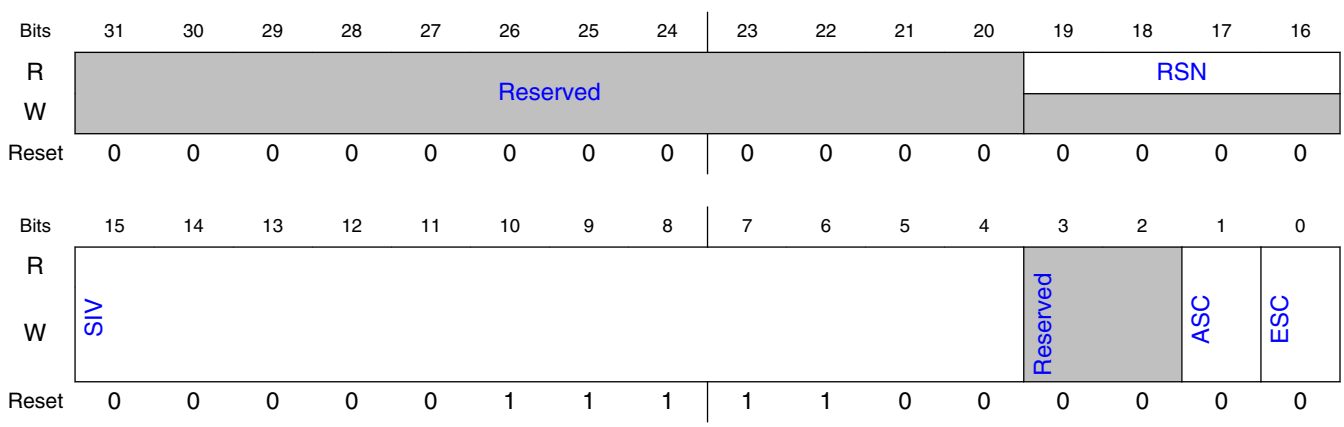
61.7.1.522.1 Offset

Register	Offset
DMA_CH2_SLOT_FUNCTION_CONTROL_STATUS	123Ch

61.7.1.522.2 Function

The Channel 2 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

61.7.1.522.3 Diagram



61.7.1.522.4 Fields

Field	Function
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0b - Advance Slot Check is disabled 1b - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or - ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0b - Slot Comparison is disabled 1b - Slot Comparison is enabled

61.7.1.523 Channel 2 Current Application Transmit Descriptor (DMA_CH2_CURRENT_APP_TXDESC)

61.7.1.523.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXDESC	1244h

61.7.1.523.2 Function

The Channel 2 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

61.7.1.523.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.523.4 Fields

Field	Function
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.524 Channel 2 Current Application Receive Descriptor (DMA_CH2_CURRENT_APP_RXDESC)

61.7.1.524.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXDESC	124Ch

61.7.1.524.2 Function

The Channel 2 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

61.7.1.524.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.524.4 Fields

Field	Function
31-0 CURRDESAPTR	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.525 Channel 2 Current Application Transmit Buffer Address (DMA_CH2_CURRENT_APP_TXBUFFER)

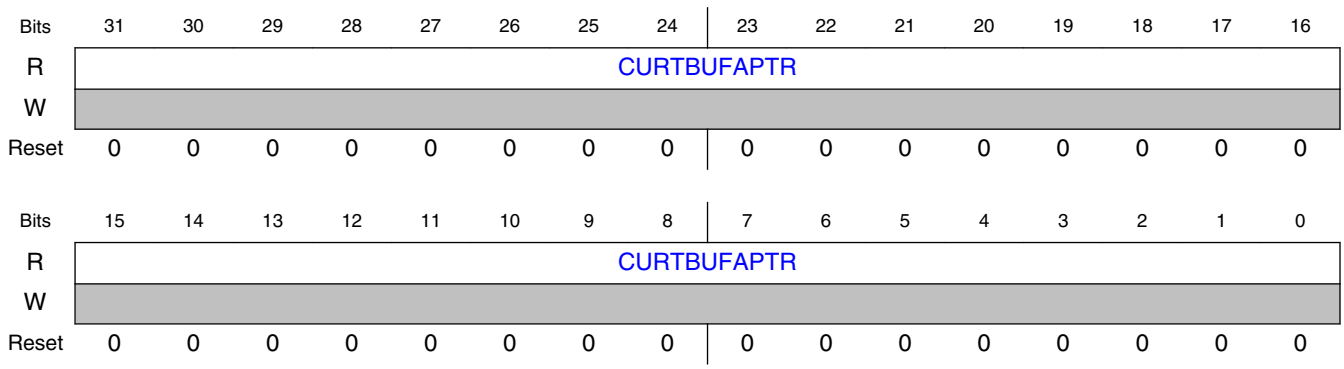
61.7.1.525.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_TXBUFFER	1254h

61.7.1.525.2 Function

The Channel 2 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

61.7.1.525.3 Diagram



61.7.1.525.4 Fields

Field	Function
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.526 Channel 2 Current Application Receive Buffer Address (DMA_CH2_CURRENT_APP_RXBUFFER)

61.7.1.526.1 Offset

Register	Offset
DMA_CH2_CURRENT_APP_RXBUFFER	125Ch

61.7.1.526.2 Function

The Channel 2 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

61.7.1.526.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.526.4 Fields

Field	Function
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.527 DMA Channel 2 Status (DMA_CH2_STATUS)

61.7.1.527.1 Offset

Register	Offset
DMA_CH2_STATUS	1260h

61.7.1.527.2 Function

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

61.7.1.527.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.527.4 Fields

Field	Function
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access

Table continues on the next page...

Field	Function
	-- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH2_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Normal Interrupt Summary status not detected 1b - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH2_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Abnormal Interrupt Summary status not detected 1b - Abnormal Interrupt Summary status detected

Table continues on the next page...

Field	Function
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Context Descriptor Error status not detected 1b - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Fatal Bus Error status not detected 1b - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Receive Interrupt status not detected 1b - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Transmit Interrupt status not detected 1b - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0b - Receive Watchdog Timeout status not detected 1b - Receive Watchdog Timeout status detected
8	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state.

Table continues on the next page...

Field	Function
RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Process Stopped status not detected 1b - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Buffer Unavailable status not detected 1b - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Interrupt status not detected 1b - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Buffer Unavailable status not detected 1b - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped.

Table continues on the next page...

Field	Function
	Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Process Stopped status not detected 1b - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Interrupt status not detected 1b - Transmit Interrupt status detected

61.7.1.528 Channel 2 Missed Frame Counter (DMA_CH2_MISS_FRAME_CNT)

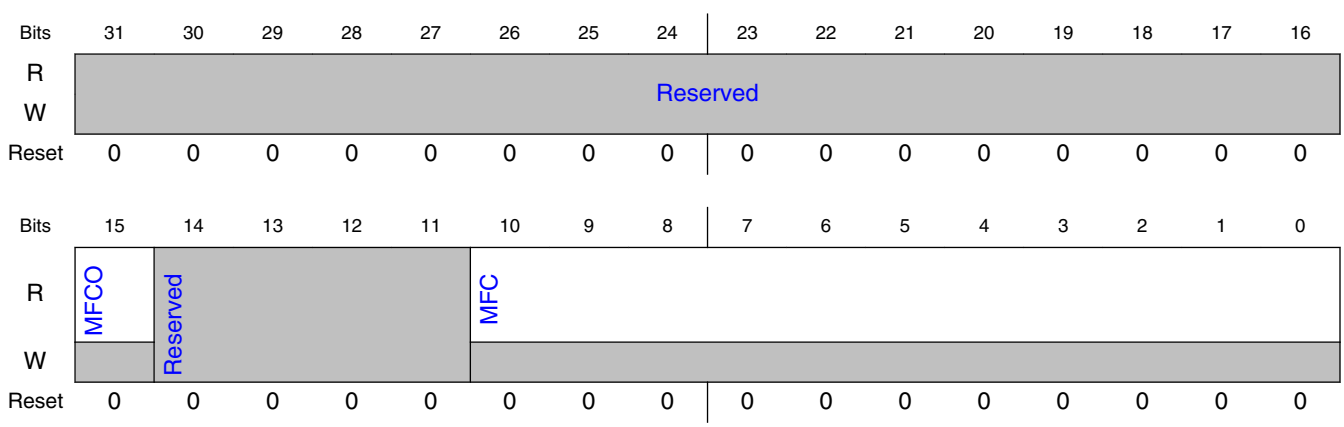
61.7.1.528.1 Offset

Register	Offset
DMA_CH2_MISS_FRAME_CNT	1264h

61.7.1.528.2 Function

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH2_RX_CONTROL register.

61.7.1.528.3 Diagram



61.7.1.528.4 Fields

Field	Function
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Miss Frame Counter overflow not occurred 1b - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH2_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.529 Channel 2 RXP Frames Accepted Counter (DMA_CH2_RXP_ACCEPT_CNT)

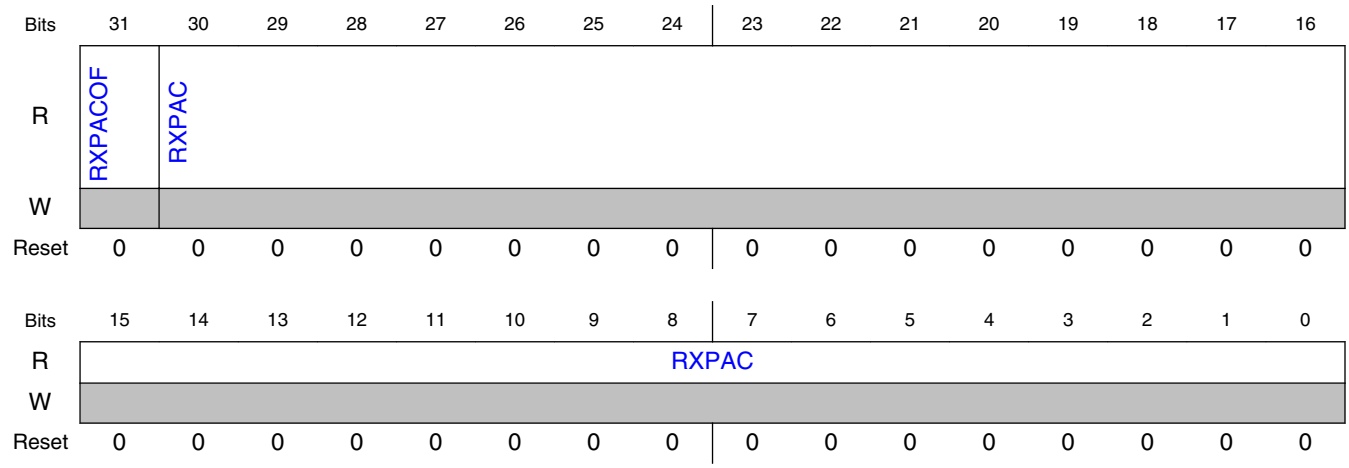
61.7.1.529.1 Offset

Register	Offset
DMA_CH2_RXP_ACCEPT_CNT	1268h

61.7.1.529.2 Function

The DMA_CH2_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

61.7.1.529.3 Diagram



61.7.1.529.4 Fields

Field	Function
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Accept Counter overflow not occurred 1b - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

61.7.1.530 DMA Channel 3 Control (DMA_CH3_CONTROL)

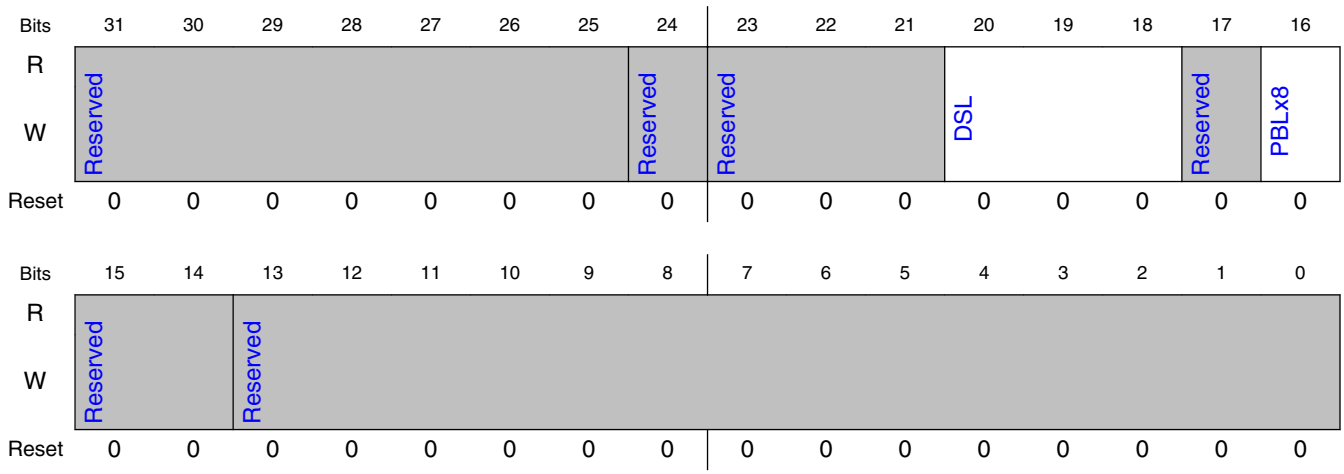
61.7.1.530.1 Offset

Register	Offset
DMA_CH3_CONTROL	1280h

61.7.1.530.2 Function

The DMA Channel 3 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

61.7.1.530.3 Diagram



61.7.1.530.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. 8xPBL mode

Table continues on the next page...

Field	Function
	When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH3_TX_CONTROL and Bits[21:16] in DMA_CH3_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0b - 8xPBL mode is disabled 1b - 8xPBL mode is enabled
15-14 —	Reserved
13-0 —	Reserved

61.7.1.531 DMA Channel 3 Transmit Control (DMA_CH3_TX_CONTROL)

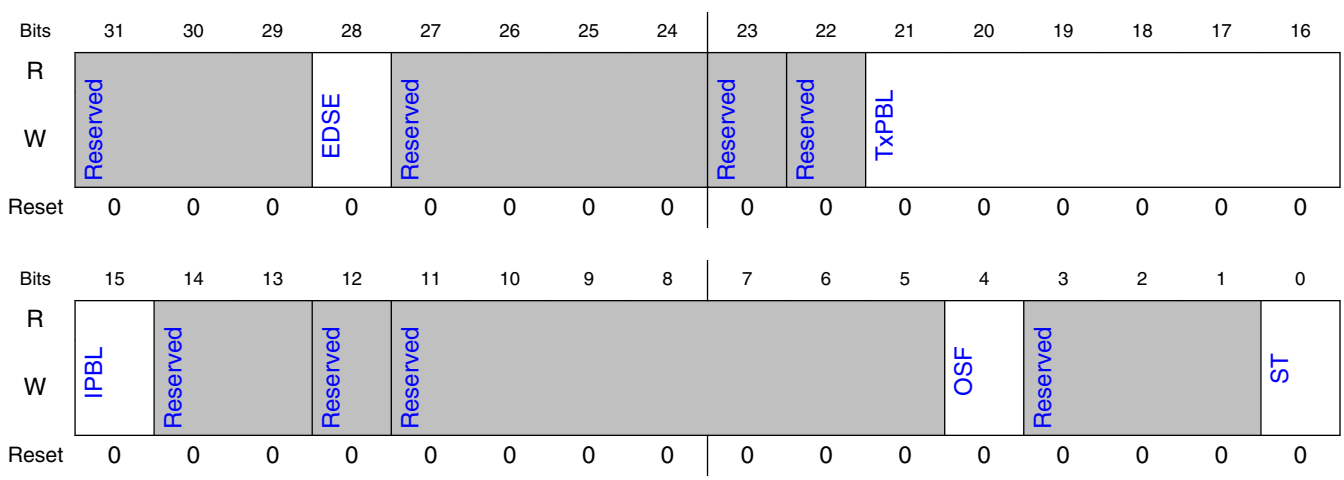
61.7.1.531.1 Offset

Register	Offset
DMA_CH3_TX_CONTROL	1284h

61.7.1.531.2 Function

The DMA Channel 3 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

61.7.1.531.3 Diagram



61.7.1.531.4 Fields

Field	Function
31-29 —	Reserved
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0b - Enhanced Descriptor is disabled 1b - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH3_CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ3_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0b - Ignore PBL Requirement is disabled 1b - Ignore PBL Requirement is enabled

Table continues on the next page...

Field	Function
14-13 —	Reserved
12 —	Reserved
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0b - Operate on Second Packet disabled 1b - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH3_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH3_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH3_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH3_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0b - Stop Transmission Command 1b - Start Transmission Command

61.7.1.532 DMA Channel 3 Receive Control (DMA_CH3_RX_CONTROL)

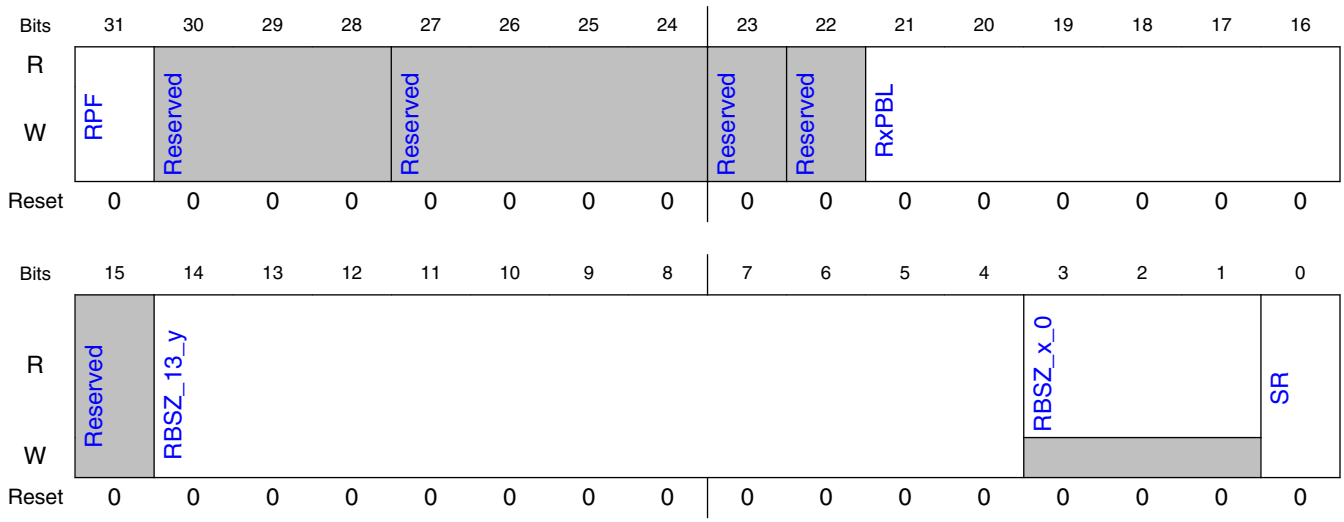
61.7.1.532.1 Offset

Register	Offset
DMA_CH3_RX_CONTROL	1288h

61.7.1.532.2 Function

The DMA Channel 3 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

61.7.1.532.3 Diagram



61.7.1.532.4 Fields

Field	Function
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0b - Rx Packet Flush is disabled 1b - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved

Table continues on the next page...

Field	Function
22 —	Reserved
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ3_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH3_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Field	Function
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH3_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH3_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0b - Stop Receive 1b - Start Receive

61.7.1.533 Channel 3 Tx Descriptor List Address (DMA_CH3_TXDESC_LIST_ADDRESS)

61.7.1.533.1 Offset

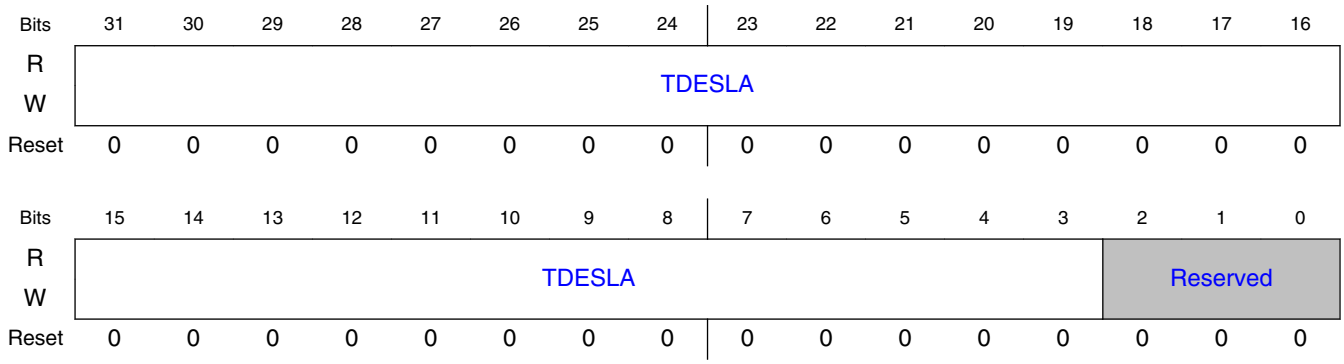
Register	Offset
DMA_CH3_TXDESC_LIST_ADDRESS	1294h

61.7.1.533.2 Function

The Channel 3 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

61.7.1.533.3 Diagram



61.7.1.533.4 Fields

Field	Function
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.534 Channel 3 Rx Descriptor List Address (DMA_CH3_RXDESC_LIST_ADDRESS)

61.7.1.534.1 Offset

Register	Offset
DMA_CH3_RXDESC_LIST_ADDRESS	129Ch

61.7.1.534.2 Function

The Channel 3 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

61.7.1.534.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.534.4 Fields

Field	Function
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0	Reserved

Field	Function
—	

61.7.1.535 Channel 3 Tx Descriptor Tail Pointer (DMA_CH3_TXDESC_TAIL_POINTER)

61.7.1.535.1 Offset

Register	Offset
DMA_CH3_TXDESC_TAIL_POINTER	12A0h

61.7.1.535.2 Function

The Channel 3 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.535.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.535.4 Fields

Field	Function
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Function
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.536 Channel 3 Rx Descriptor Tail Pointer (DMA_CH3_RXDESC_TAIL_POINTER)

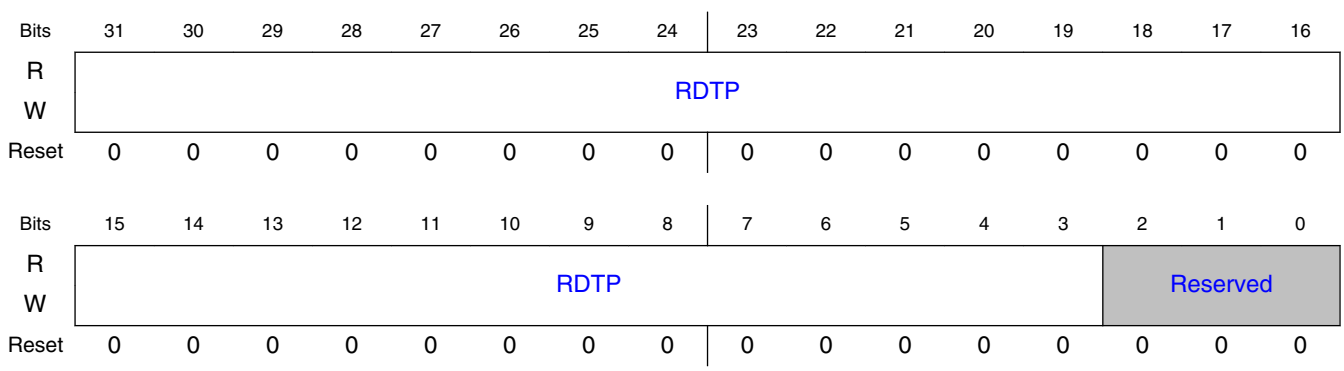
61.7.1.536.1 Offset

Register	Offset
DMA_CH3_RXDESC_TAIL_POINTER	12A8h

61.7.1.536.2 Function

The Channel 3 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.536.3 Diagram



61.7.1.536.4 Fields

Field	Function
31-3	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring.
RDTP	Receive Descriptor Tail Pointer

Table continues on the next page...

Field	Function
	This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.537 Channel 3 Tx Descriptor Ring Length (DMA_CH3_TXDESC_RING_LENGTH)

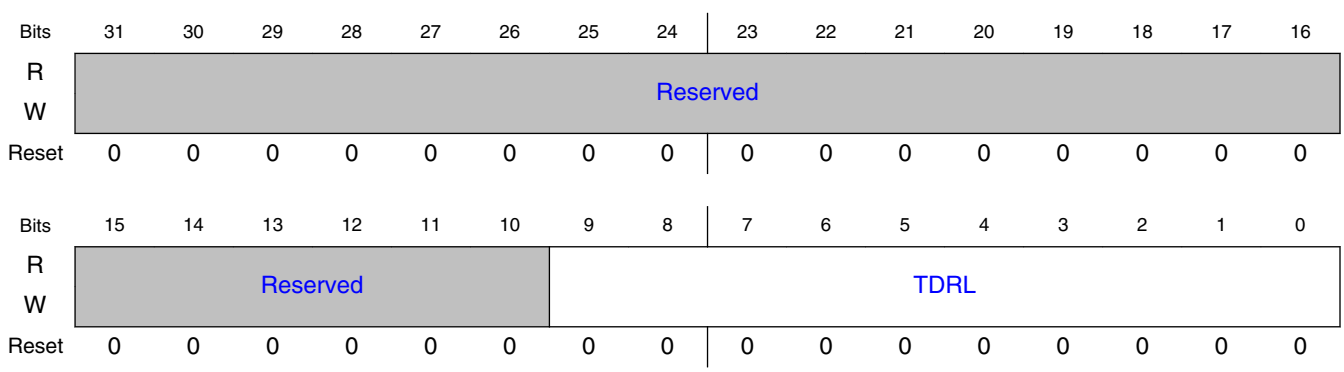
61.7.1.537.1 Offset

Register	Offset
DMA_CH3_TXDESC_RING_LENGTH	12ACh

61.7.1.537.2 Function

The Channel 3 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

61.7.1.537.3 Diagram



61.7.1.537.4 Fields

Field	Function
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.538 Channel 3 Receive Control 2 register (DMA_CH3_RX_CONTROL2)

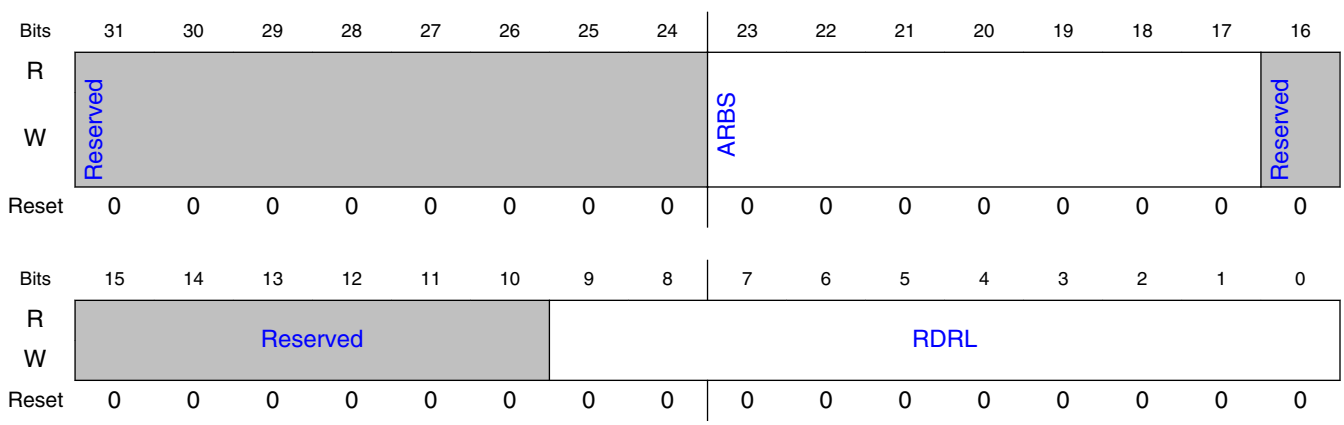
61.7.1.538.1 Offset

Register	Offset
DMA_CH3_RX_CONTROL2	12B0h

61.7.1.538.2 Function

The Channel 3 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

61.7.1.538.3 Diagram



61.7.1.538.4 Fields

Field	Function
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.539 Channel 3 Interrupt Enable (DMA_CH3_INTERRUPT_ENABLE)

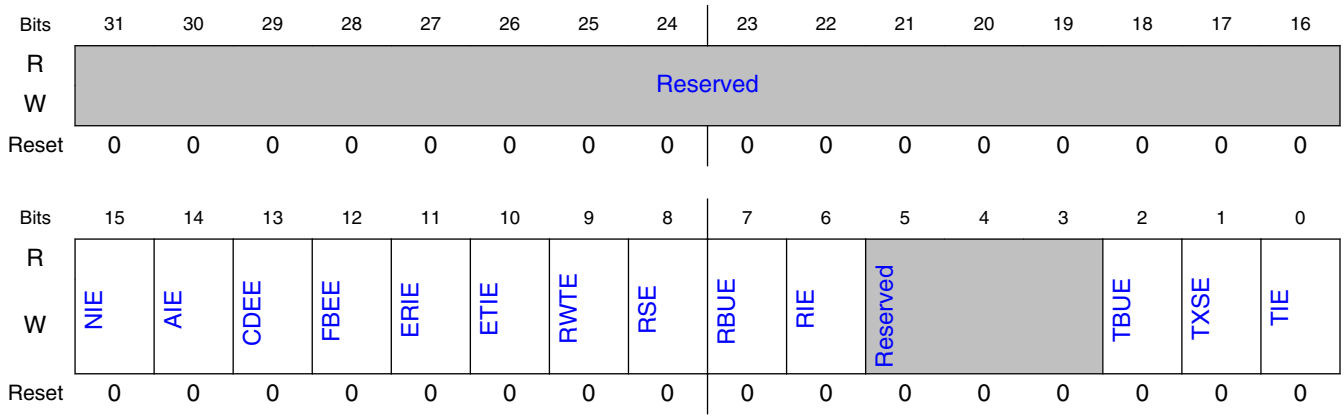
61.7.1.539.1 Offset

Register	Offset
DMA_CH3_INTERRUPT_ENABLE	12B4h

61.7.1.539.2 Function

The Channel 3 Interrupt Enable register enables the interrupts reported by the Status register.

61.7.1.539.3 Diagram



61.7.1.539.4 Fields

Field	Function
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH3_STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0b - Normal Interrupt Summary is disabled 1b - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled.

Table continues on the next page...

Field	Function
	0b - Abnormal Interrupt Summary is disabled 1b - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0b - Context Descriptor Error is disabled 1b - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0b - Fatal Bus Error is disabled 1b - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0b - Early Receive Interrupt is disabled 1b - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0b - Early Transmit Interrupt is disabled 1b - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0b - Receive Watchdog Timeout is disabled 1b - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0b - Receive Stopped is disabled 1b - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled.

Table continues on the next page...

Field	Function
	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0b - Receive Buffer Unavailable is disabled 1b - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0b - Receive Interrupt is disabled 1b - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0b - Transmit Buffer Unavailable is disabled 1b - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0b - Transmit Stopped is disabled 1b - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0b - Transmit Interrupt is disabled 1b - Transmit Interrupt is enabled

61.7.1.540 Channel 3 Receive Interrupt Watchdog Time (DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER)

61.7.1.540.1 Offset

Register	Offset
DMA_CH3_RX_INTERRUPT_WATCHDOG_TIMER	12B8h

61.7.1.540.2 Function

The Channel 3 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH3_Status register.

61.7.1.540.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.540.4 Fields

Field	Function
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.

Table continues on the next page...

Field	Function
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

61.7.1.541 Channel 3 Slot Function Control and Status (DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS)

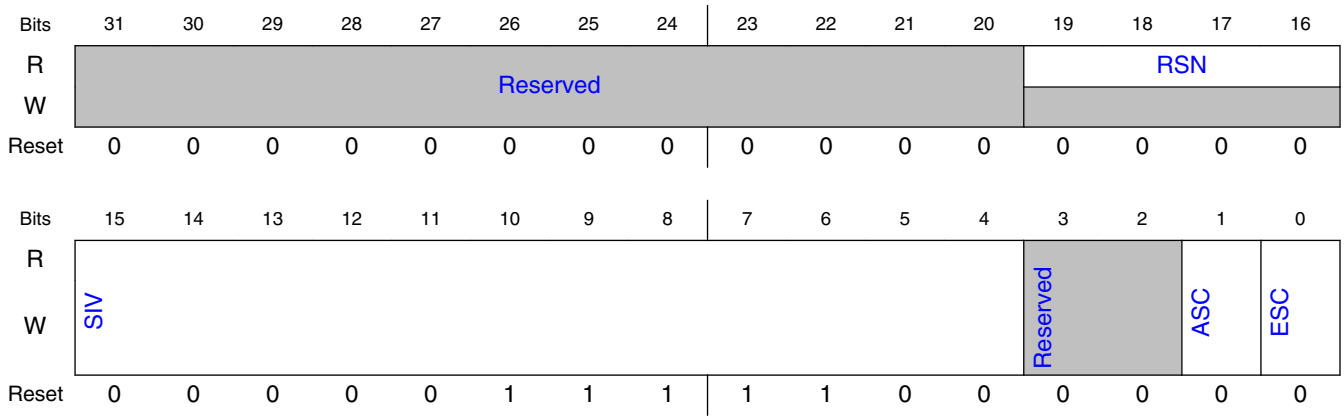
61.7.1.541.1 Offset

Register	Offset
DMA_CH3_SLOT_FUNCTION_CONTROL_STATUS	12BCh

61.7.1.541.2 Function

The Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

61.7.1.541.3 Diagram



61.7.1.541.4 Fields

Field	Function
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0b - Advance Slot Check is disabled 1b - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or

Field	Function
	- ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0b - Slot Comparison is disabled 1b - Slot Comparison is enabled

61.7.1.542 Channel 3 Current Application Transmit Descriptor (DMA_CH3_CURRENT_APP_TXDESC)

61.7.1.542.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXDESC	12C4h

61.7.1.542.2 Function

The Channel 3 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

61.7.1.542.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.542.4 Fields

Field	Function
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.543 Channel 3 Current Application Receive Descriptor (DMA_CH3_CURRENT_APP_RXDESC)

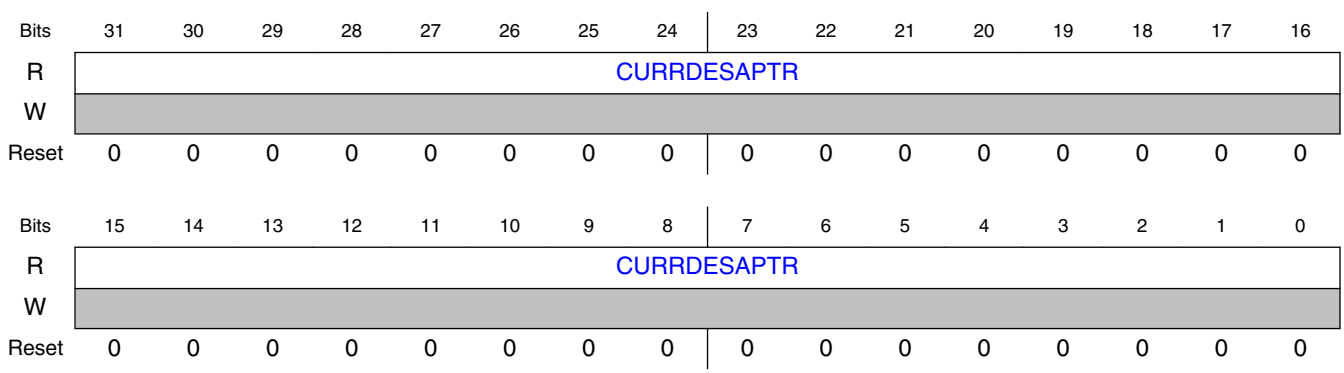
61.7.1.543.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXDESC	12CCh

61.7.1.543.2 Function

The Channel 3 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

61.7.1.543.3 Diagram



61.7.1.543.4 Fields

Field	Function
31-0	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer

Field	Function
CURRDESAPTR	The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.544 Channel 3 Current Application Transmit Buffer Address (DMA_CH3_CURRENT_APP_TXBUFFER)

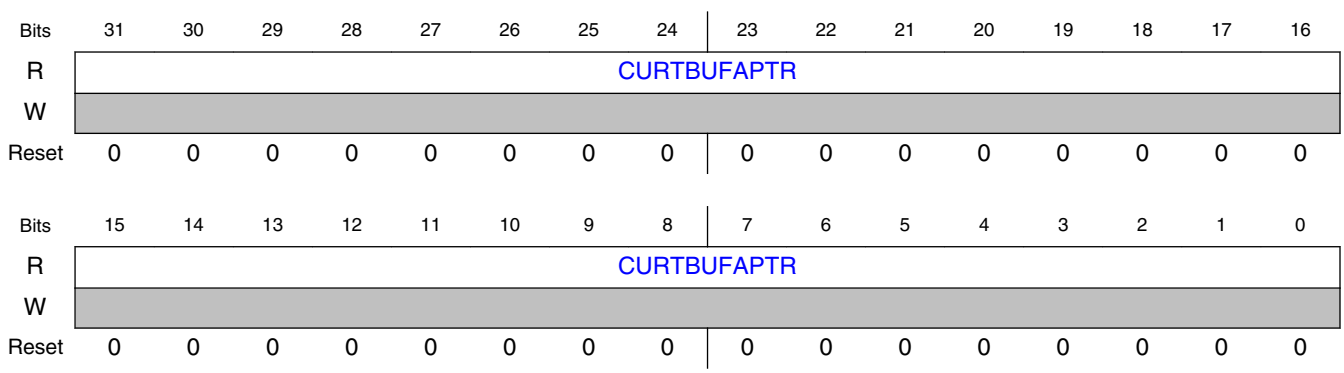
61.7.1.544.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_TXBUFFER	12D4h

61.7.1.544.2 Function

The Channel 3 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

61.7.1.544.3 Diagram



61.7.1.544.4 Fields

Field	Function
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.545 Channel 3 Current Application Receive Buffer Address (DMA_CH3_CURRENT_APP_RXBUFFER)

61.7.1.545.1 Offset

Register	Offset
DMA_CH3_CURRENT_APP_RXBUFFER	12DCh

61.7.1.545.2 Function

The Channel 3 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

61.7.1.545.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.545.4 Fields

Field	Function
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.546 DMA Channel 3 Status (DMA_CH3_STATUS)

61.7.1.546.1 Offset

Register	Offset
DMA_CH3_STATUS	12E0h

61.7.1.546.2 Function

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

61.7.1.546.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.546.4 Fields

Field	Function
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20

Table continues on the next page...

Field	Function
	-- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH3_INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Normal Interrupt Summary status not detected 1b - Normal Interrupt Summary status detected

Table continues on the next page...

Field	Function
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH3_INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Abnormal Interrupt Summary status not detected 1b - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Context Descriptor Error status not detected 1b - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Fatal Bus Error status not detected 1b - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory.

Table continues on the next page...

Field	Function
	In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Receive Interrupt status not detected 1b - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Transmit Interrupt status not detected 1b - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0b - Receive Watchdog Timeout status not detected 1b - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Process Stopped status not detected 1b - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Buffer Unavailable status not detected 1b - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state.

Table continues on the next page...

Field	Function
	Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Interrupt status not detected 1b - Receive Interrupt status detected
5-3 —	Reserved
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Buffer Unavailable status not detected 1b - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Process Stopped status not detected 1b - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Interrupt status not detected 1b - Transmit Interrupt status detected

61.7.1.547 Channel 3 Missed Frame Counter (DMA_CH3_MISS_FRAME_CNT)

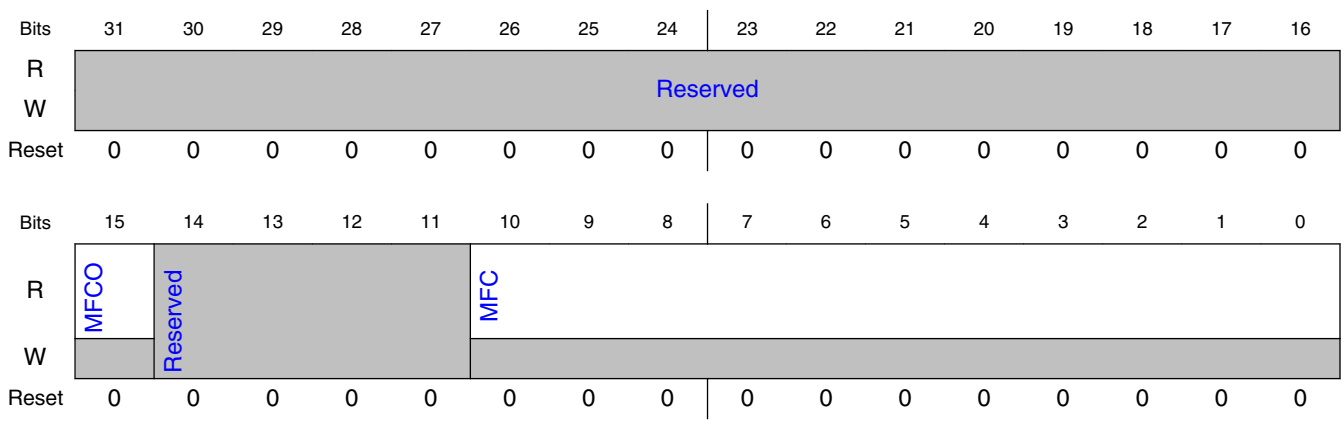
61.7.1.547.1 Offset

Register	Offset
DMA_CH3_MISS_FRAME_CNT	12E4h

61.7.1.547.2 Function

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH3_RX_CONTROL register.

61.7.1.547.3 Diagram



61.7.1.547.4 Fields

Field	Function
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Miss Frame Counter overflow not occurred 1b - Miss Frame Counter overflow occurred
14-11 —	Reserved

Table continues on the next page...

Field	Function
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH3_RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.548 Channel 3 RXP Frames Accepted Counter (DMA_CH3_RXP_ACCEPT_CNT)

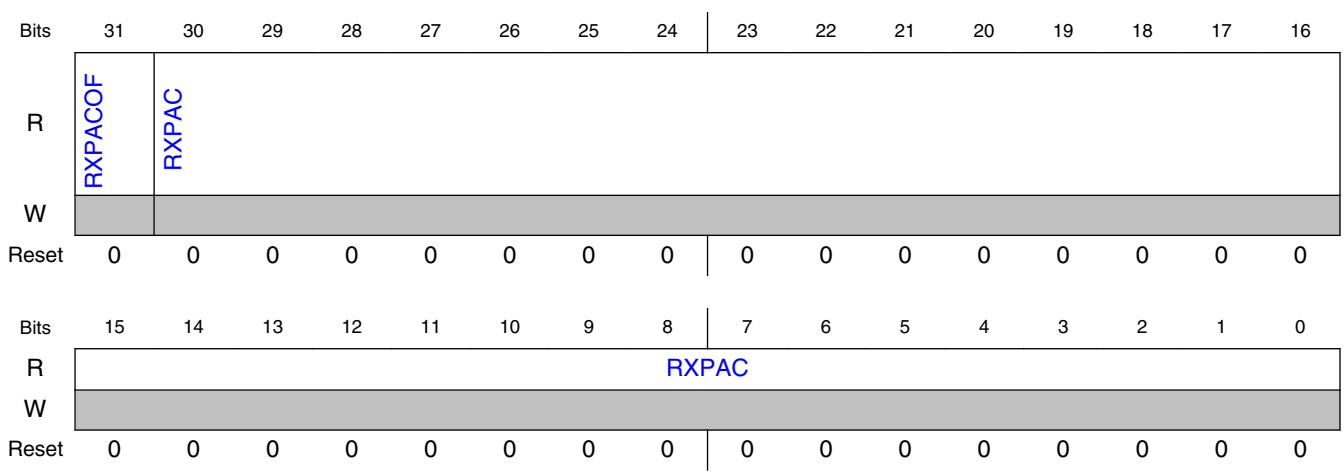
61.7.1.548.1 Offset

Register	Offset
DMA_CH3_RXP_ACCEPT_CNT	12E8h

61.7.1.548.2 Function

The DMA_CH3_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

61.7.1.548.3 Diagram



61.7.1.548.4 Fields

Field	Function
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Accept Counter overflow not occurred 1b - Rx Parser Accept Counter overflow occurred
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

61.7.1.549 Channel 4 Control (DMA_CH4_CONTROL)

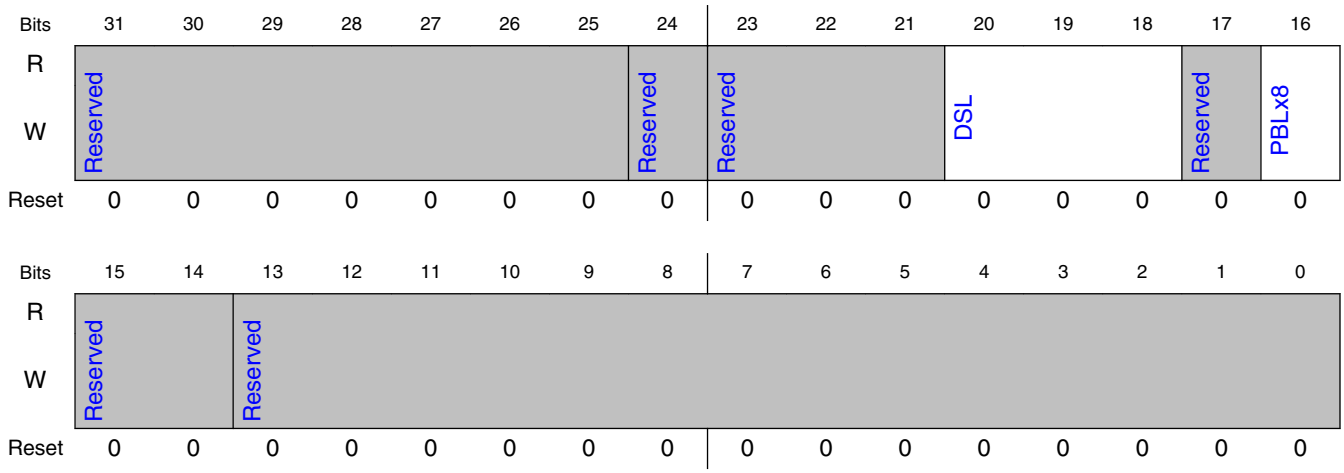
61.7.1.549.1 Offset

Register	Offset
DMA_CH4_CONTROL	1300h

61.7.1.549.2 Function

The DMA Channel 4 Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

61.7.1.549.3 Diagram



61.7.1.549.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23-21 —	Reserved
20-18 DSL	Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17 —	Reserved
16 PBLx8	8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times. 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH4_TX_CONTROL and Bits[21:16] in DMA_CH4_RX_CONTROL is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0b - 8xPBL mode is disabled 1b - 8xPBL mode is enabled
15-14 —	Reserved

Table continues on the next page...

Field	Function
13-0 —	Reserved

61.7.1.550 DMA Channel 4 Transmit Control (DMA_CH4_TX_CONTROL)

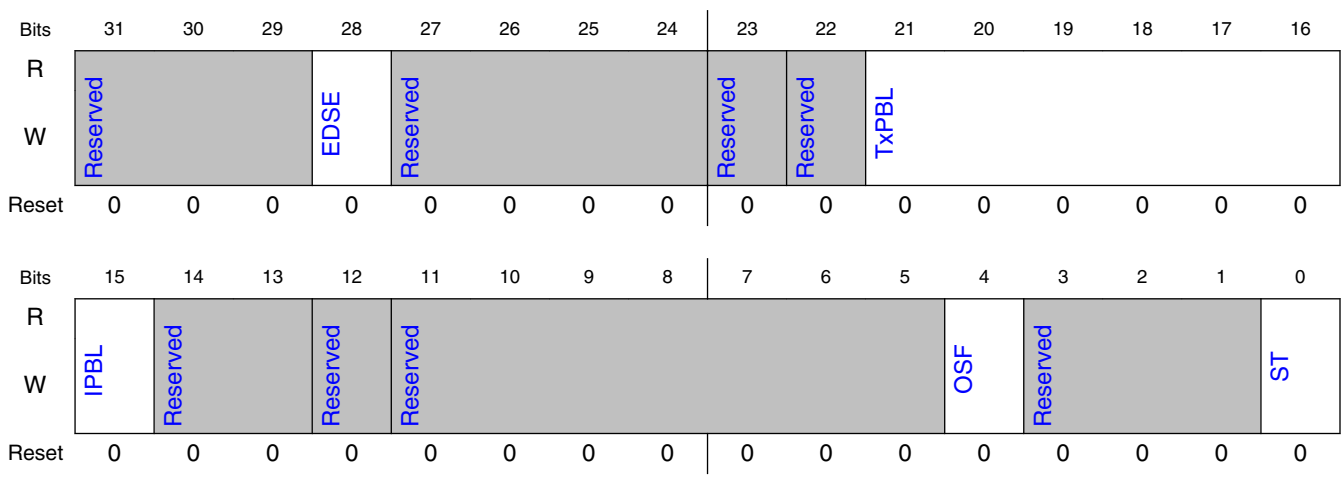
61.7.1.550.1 Offset

Register	Offset
DMA_CH4_TX_CONTROL	1304h

61.7.1.550.2 Function

The DMA Channel 4 Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

61.7.1.550.3 Diagram



61.7.1.550.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Field	Function
—	
28 EDSE	Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. Enhanced Descriptor Enable When this bit is set, the corresponding channel uses Enhanced Descriptors that are 32 Bytes for both Normal and Context Descriptors. When reset, the corresponding channel uses the descriptors that are 16 Bytes. 0b - Enhanced Descriptor is disabled 1b - Enhanced Descriptor is enabled
27-24 —	Reserved
23 —	Reserved
22 —	Reserved
21-16 TxPBL	Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in CONTROL register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of TXQ4_OPERATION_MODE register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 IPBL	Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0b - Ignore PBL Requirement is disabled 1b - Ignore PBL Requirement is enabled
14-13 —	Reserved
12 —	Reserved

Table continues on the next page...

Field	Function
11-5 —	Reserved
4 OSF	Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0b - Operate on Second Packet disabled 1b - Operate on Second Packet enabled
3-1 —	Reserved
0 ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the base address of the Transmit list set by the DMA_CH4_TXDESC_LIST_ADDRESS register. - The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH4_STATUS register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH4_TXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH4_TXDESC_LIST_ADDRESS register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0b - Stop Transmission Command 1b - Start Transmission Command

61.7.1.551 DMA Channel 4 Receive Control (DMA_CH4_RX_CONTROL)

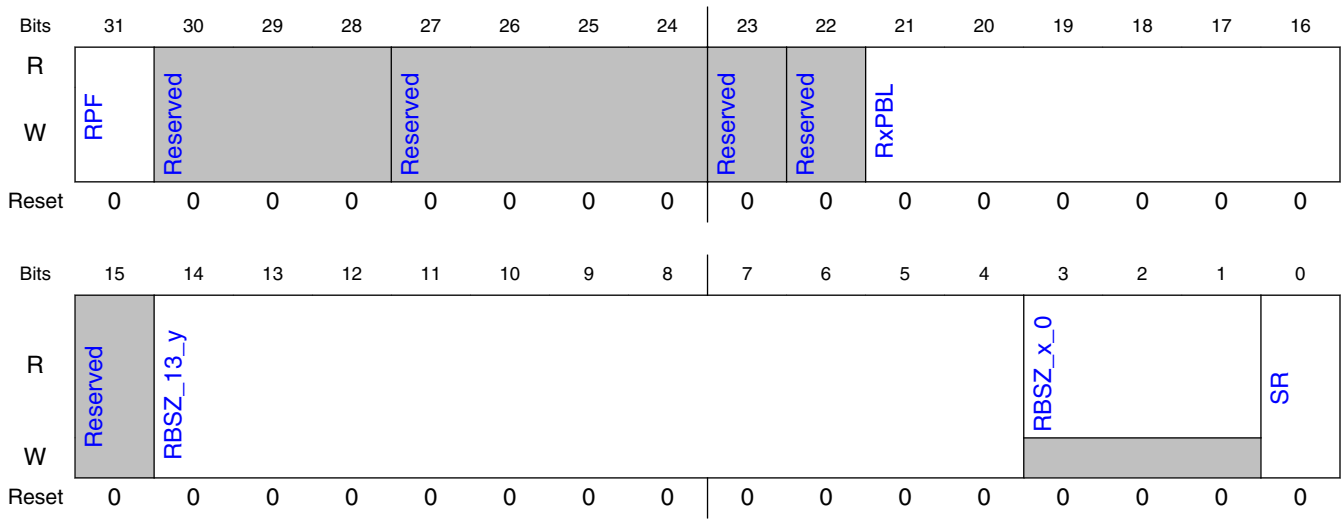
61.7.1.551.1 Offset

Register	Offset
DMA_CH4_RX_CONTROL	1308h

61.7.1.551.2 Function

The DMA Channel 4 Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

61.7.1.551.3 Diagram



61.7.1.551.4 Fields

Field	Function
31 RPF	Rx Packet Flush. Rx Packet Flush. When this bit is set to 1, then DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the DWC_ether_qos not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0b - Rx Packet Flush is disabled 1b - Rx Packet Flush is enabled
30-28 —	Reserved
27-24 —	Reserved
23 —	Reserved
22	Reserved

Table continues on the next page...

Field	Function
—	
21-16 RxPBL	Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH4_CONTROL register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of RXQ4_OPERATION_MODE register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15 —	Reserved
14-4 RBSZ_13_y	Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).
3-1 RBSZ_x_0	Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).
0 SR	Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets. The DMA tries to acquire descriptor from either of the following positions: - The current position in the list This is the address set by the DMA_CH4_RXDESC_LIST_ADDRESS register. - The position at which the Rx process was previously stopped

Field	Function
	If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH4_STATUS register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH4_RXDESC_LIST_ADDRESS register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0b - Stop Receive 1b - Start Receive

61.7.1.552 Channel 4 Tx Descriptor List Address (DMA_CH4_TXDESC_LIST_ADDRESS)

61.7.1.552.1 Offset

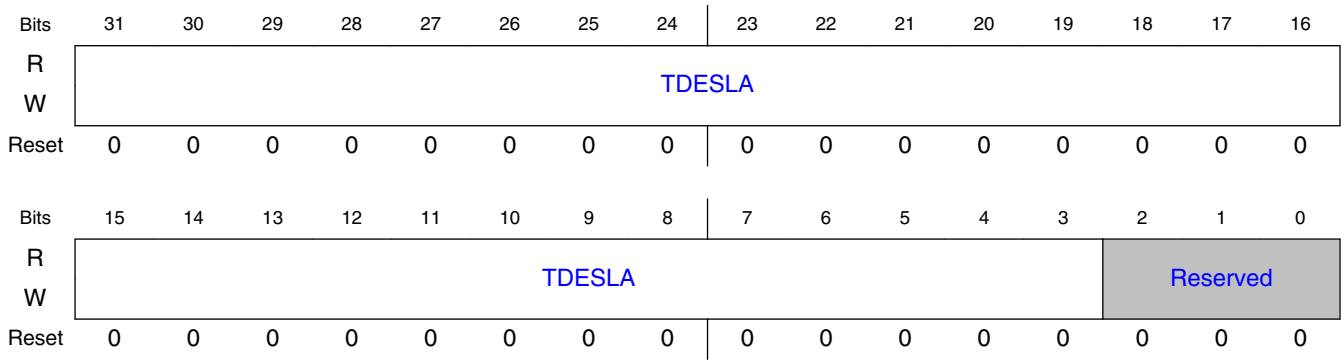
Register	Offset
DMA_CH4_TXDESC_LIST_ADDRESS	1314h

61.7.1.552.2 Function

The Channel 4 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in TX_CONTROL register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

61.7.1.552.3 Diagram



61.7.1.552.4 Fields

Field	Function
31-3 TDESLA	Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.553 Channel 4 Rx Descriptor List Address (DMA_CH4_RXDESC_LIST_ADDRESS)

61.7.1.553.1 Offset

Register	Offset
DMA_CH4_RXDESC_LIST_ADDRESS	131Ch

61.7.1.553.2 Function

The Channel 4 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in RX_CONTROL register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

61.7.1.553.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDESLA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDESLA													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.553.4 Fields

Field	Function
31-3 RDESLA	Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0	Reserved

Field	Function
—	

61.7.1.554 Channel 4 Tx Descriptor Tail Pointer (DMA_CH4_TXDESC_TAIL_POINTER)

61.7.1.554.1 Offset

Register	Offset
DMA_CH4_TXDESC_TAIL_POINTER	1320h

61.7.1.554.2 Function

The Channel 4 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.554.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TDTP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDTP													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.554.4 Fields

Field	Function
31-3 TDTP	Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration:

Table continues on the next page...

Field	Function
	- 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.555 Channel 4 Rx Descriptor Tail Pointer (DMA_CH4_RXDESC_TAIL_POINTER)

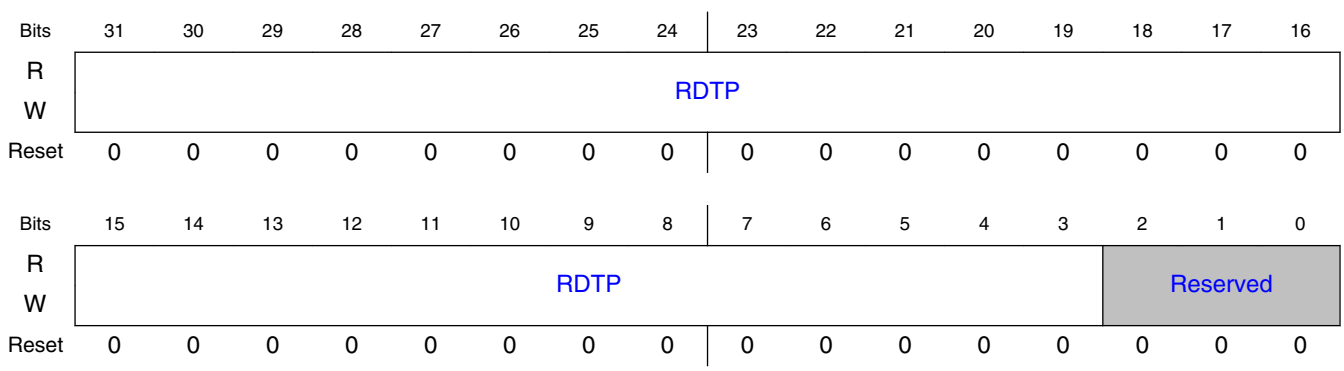
61.7.1.555.1 Offset

Register	Offset
DMA_CH4_RXDESC_TAIL_POINTER	1328h

61.7.1.555.2 Function

The Channel 4 Rx Descriptor Tail Pointer points to an offset from the base and indicates the location of the last valid descriptor.

61.7.1.555.3 Diagram



61.7.1.555.4 Fields

Field	Function
31-3 RDTP	Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. Receive Descriptor Tail Pointer

Table continues on the next page...

Field	Function
	This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: - 31:2 for 32-bit configuration - 31:3 for 64-bit configuration - 31:4 for 128-bit configuration
2-0 —	Reserved

61.7.1.556 Channel 4 Tx Descriptor Ring Length (DMA_CH4_TXDESC_RING_LENGTH)

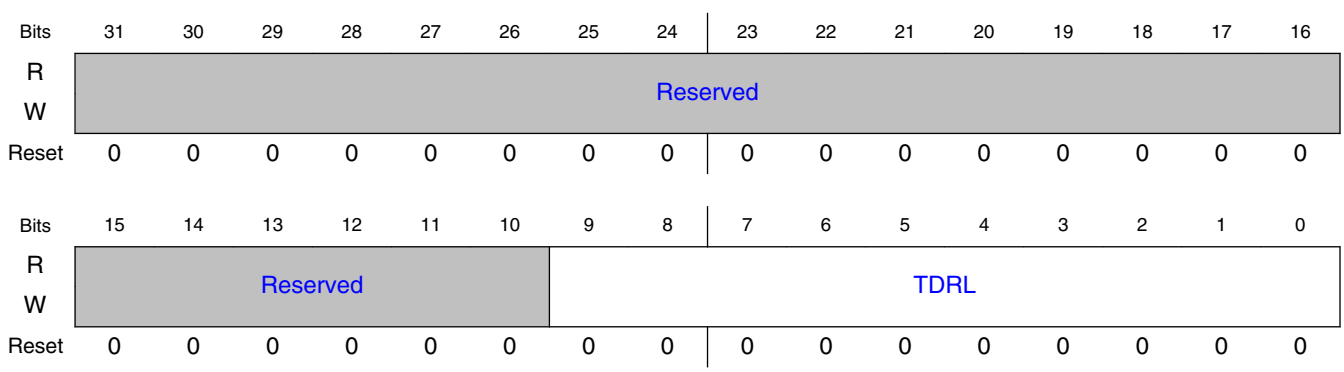
61.7.1.556.1 Offset

Register	Offset
DMA_CH4_TXDESC_RING_LENGTH	132Ch

61.7.1.556.2 Function

The Channel 4 Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

61.7.1.556.3 Diagram



61.7.1.556.4 Fields

Field	Function
31-10 —	Reserved
9-0 TDRL	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. We recommend a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.557 DMA Channel 4 Receive Control 2 register (DMA_CH4_RX_CONTROL2)

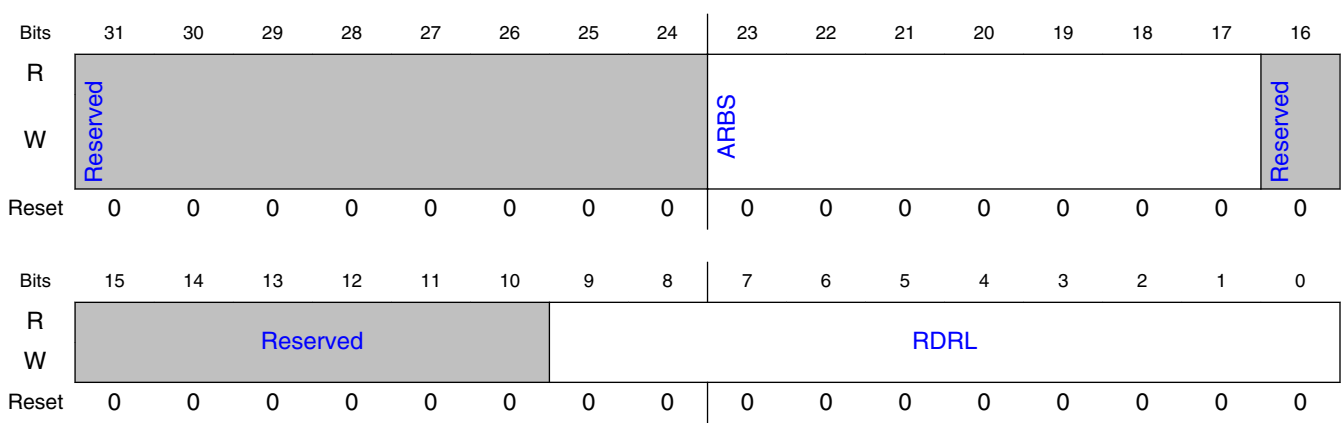
61.7.1.557.1 Offset

Register	Offset
DMA_CH4_RX_CONTROL2	1330h

61.7.1.557.2 Function

The Channel 4 Receive Control register controls Rx Descriptor Ring Length and Alternate Rx Buffer Size.

61.7.1.557.3 Diagram



61.7.1.557.4 Fields

Field	Function
31-24 —	Reserved
23-17 ARBS	Alternate Receive Buffer Size Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on DMA_CH2_RX_CONTROL[RBSZ]. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively).
16-10 —	Reserved
9-0 RDRL	Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

61.7.1.558 Channel 4 Interrupt Enable (DMA_CH4_INTERRUPT_ENABLE)

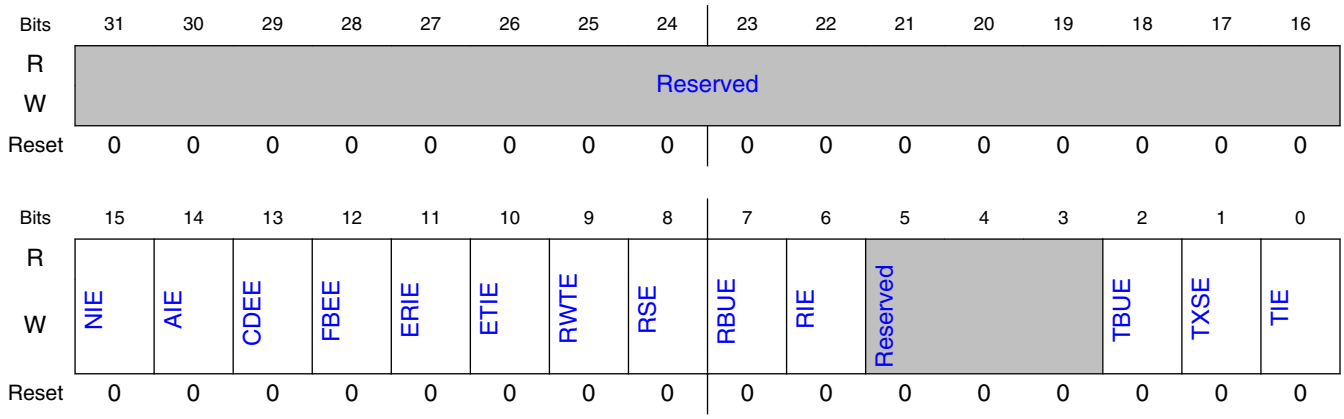
61.7.1.558.1 Offset

Register	Offset
DMA_CH4_INTERRUPT_ENABLE	1334h

61.7.1.558.2 Function

The Channel 4 Interrupt Enable register enables the interrupts reported by the Status register.

61.7.1.558.3 Diagram



61.7.1.558.4 Fields

Field	Function
31-16 —	Reserved
15 NIE	Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0b - Normal Interrupt Summary is disabled 1b - Normal Interrupt Summary is enabled
14 AIE	Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the STATUS register: - Bit 1: Transmit Process Stopped - Bit 7: Rx Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 9: Receive Watchdog Timeout - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled.

Table continues on the next page...

Field	Function
	0b - Abnormal Interrupt Summary is disabled 1b - Abnormal Interrupt Summary is enabled
13 CDEE	Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0b - Context Descriptor Error is disabled 1b - Context Descriptor Error is enabled
12 FBEE	Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0b - Fatal Bus Error is disabled 1b - Fatal Bus Error is enabled
11 ERIE	Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0b - Early Receive Interrupt is disabled 1b - Early Receive Interrupt is enabled
10 ETIE	Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0b - Early Transmit Interrupt is disabled 1b - Early Transmit Interrupt is enabled
9 RWTE	Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0b - Receive Watchdog Timeout is disabled 1b - Receive Watchdog Timeout is enabled
8 RSE	Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled. 0b - Receive Stopped is disabled 1b - Receive Stopped is enabled
7 RBUE	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled.

Table continues on the next page...

Field	Function
	Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0b - Receive Buffer Unavailable is disabled 1b - Receive Buffer Unavailable is enabled
6 RIE	Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0b - Receive Interrupt is disabled 1b - Receive Interrupt is enabled
5-3 —	Reserved
2 TBUE	Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0b - Transmit Buffer Unavailable is disabled 1b - Transmit Buffer Unavailable is enabled
1 TXSE	Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0b - Transmit Stopped is disabled 1b - Transmit Stopped is enabled
0 TIE	Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0b - Transmit Interrupt is disabled 1b - Transmit Interrupt is enabled

61.7.1.559 Channel 4 Receive Interrupt Watchdog Timer (DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER)

61.7.1.559.1 Offset

Register	Offset
DMA_CH4_RX_INTERRUPT_WATCHDOG_TIMER	1338h

61.7.1.559.2 Function

The Channel 4 Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH4_Status register.

61.7.1.559.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														RWTU	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RWT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.559.4 Fields

Field	Function
31-18 —	Reserved
17-16 RWTU	Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. - 2'b00: 256 - 2'b01: 512 - 2'b10: 1024 - 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.

Table continues on the next page...

Field	Function
15-8 —	Reserved
7-0 RWT	Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CH4_STATUS register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

61.7.1.560 Channel 4 Slot Function Control and Status (DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS)

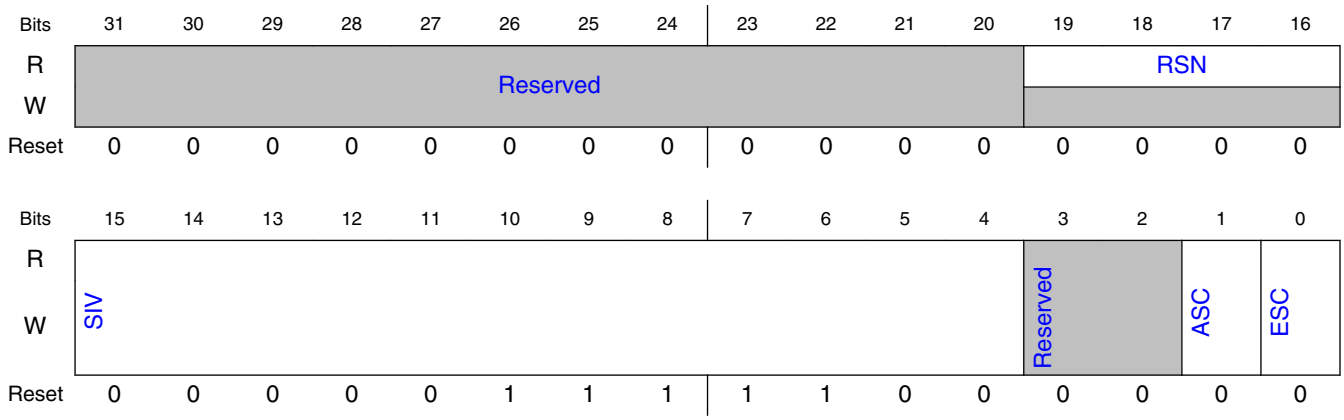
61.7.1.560.1 Offset

Register	Offset
DMA_CH4_SLOT_FUNCTION_CONTROL_STATUS	133Ch

61.7.1.560.2 Function

The Channel 4 Slot Function Control and Status register contains the control bits for slot function and the status for Transmit path.

61.7.1.560.3 Diagram



61.7.1.560.4 Fields

Field	Function
31-20 —	Reserved
19-16 RSN	Reference Slot Number This field gives the current value of the reference slot number in the DMA. Reference Slot Number This field gives the current value of the reference slot number in the DMA. It is used for slot comparison.
15-4 SIV	Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. Slot Interval Value This field controls the period of the slot interval in which the TxDMA fetches the scheduled packets. A value of 0 specifies the slot interval of 1 us while the maximum value 4095 specifies the slot interval of 4096us. The default/reset value is 0x07C which corresponds to slot interval of 125us
3-2 —	Reserved
1 ASC	Advance Slot Check When set, this bit enables the DMA to fetch the data from the buffer when the slot number (SLOTNUM) programmed in the Tx descriptor is - equal to the reference slot number given in the RSN field or - ahead of the reference slot number by up to two slots This bit is applicable only when the ESC bit is set. 0b - Advance Slot Check is disabled 1b - Advance Slot Check is enabled
0 ESC	Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. Enable Slot Comparison When set, this bit enables the checking of the slot numbers programmed in the Tx descriptor with the current reference given in the RSN field. The DMA fetches the data from the corresponding buffer only when the slot number is - equal to the reference slot number or

Field	Function
	- ahead of the reference slot number by one slot When reset, this bit disables the checking of the slot numbers. The DMA fetches the data immediately after the descriptor is processed. Note: The UFO (UDP Fragmentation over IPv4)/TSO/USO should not be enabled along with TBS/AVB Slot number check. The UFO/TSO/USO involves multiple packets/segments/fragments transmission for single packet received from application and the slot number check are applicable for fetching of only first segment/fragment. As a result it might be difficult for software to specify slot number for subsequent packets. 0b - Slot Comparison is disabled 1b - Slot Comparison is enabled

61.7.1.561 Channel 4 Current Application Transmit Descriptor (DMA_CH4_CURRENT_APP_TXDESC)

61.7.1.561.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXDESC	1344h

61.7.1.561.2 Function

The Channel 4 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

61.7.1.561.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.561.4 Fields

Field	Function
31-0 CURTDESAPTR	Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.562 Channel 4 Current Application Receive Descriptor (DMA_CH4_CURRENT_APP_RXDESC)

61.7.1.562.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXDESC	134Ch

61.7.1.562.2 Function

The Channel 4 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

61.7.1.562.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRDESAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.562.4 Fields

Field	Function
31-0	Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. Application Receive Descriptor Address Pointer

Field	Function
CURRDESAPTR	The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.563 Channel 4 Current Application Transmit Buffer Address (DMA_CH4_CURRENT_APP_TXBUFFER)

61.7.1.563.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_TXBUFFER	1354h

61.7.1.563.2 Function

The Channel 4 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

61.7.1.563.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURTBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.563.4 Fields

Field	Function
31-0 CURTBUFAPTR	Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

61.7.1.564 Channel 4 Current Application Receive Buffer Address (DMA_CH4_CURRENT_APP_RXBUFFER)

61.7.1.564.1 Offset

Register	Offset
DMA_CH4_CURRENT_APP_RXBUFFER	135Ch

61.7.1.564.2 Function

The Channel 4 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

61.7.1.564.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CURRBUFAPTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.564.4 Fields

Field	Function
31-0 CURRBUFAPTR	Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

61.7.1.565 DMA Channel 4 Status (DMA_CH4_STATUS)

61.7.1.565.1 Offset

Register	Offset
DMA_CH4_STATUS	1360h

61.7.1.565.2 Function

The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of STATUS register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

61.7.1.565.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										REB			TEB		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Reserved			TBU	TPS	TI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

61.7.1.565.4 Fields

Field	Function
31-22 —	Reserved
21-19 REB	Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 21 -- 1'b1: Error during data transfer by Rx DMA -- 1'b0: No Error during data transfer by Rx DMA - Bit 20 -- 1'b1: Error during descriptor access

Table continues on the next page...

Field	Function
	-- 1'b0: Error during data buffer access - Bit 19 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18-16 TEB	Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. - Bit 18 -- 1'b1: Error during data transfer by Tx DMA -- 1'b0: No Error during data transfer by Tx DMA - Bit 17 -- 1'b1: Error during descriptor access -- 1'b0: Error during data buffer access - Bit 16 -- 1'b1: Error during read transfer -- 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
15 NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 0: Transmit Interrupt - Bit 2: Transmit Buffer Unavailable - Bit 6: Receive Interrupt - Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in INTERRUPT_ENABLE register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Normal Interrupt Summary status not detected 1b - Normal Interrupt Summary status detected
14 AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit.

Table continues on the next page...

Field	Function
	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the INTERRUPT_ENABLE register: - Bit 1: Transmit Process Stopped - Bit 7: Receive Buffer Unavailable - Bit 8: Receive Process Stopped - Bit 10: Early Transmit Interrupt - Bit 12: Fatal Bus Error - Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Abnormal Interrupt Summary status not detected 1b - Abnormal Interrupt Summary status detected
13 CDE	Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Context Descriptor Error status not detected 1b - Context Descriptor Error status detected
12 FBE	Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Fatal Bus Error status not detected 1b - Fatal Bus Error status detected
11 ERI	Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer. The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Receive Interrupt status not detected

Table continues on the next page...

Field	Function
	1b - Early Receive Interrupt status detected
10 ETI	Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ETIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Early Transmit Interrupt status not detected 1b - Early Transmit Interrupt status detected
9 RWT	Receive Watchdog Timeout This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received. 0b - Receive Watchdog Timeout status not detected 1b - Receive Watchdog Timeout status detected
8 RPS	Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Process Stopped status not detected 1b - Receive Process Stopped status detected
7 RBU	Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Buffer Unavailable status not detected 1b - Receive Buffer Unavailable status detected
6 RI	Receive Interrupt This bit indicates that the packet reception is complete. Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Receive Interrupt status not detected 1b - Receive Interrupt status detected
5-3 —	Reserved

Table continues on the next page...

Field	Function
2 TBU	Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_CH4_DEBUG_STATUS0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Buffer Unavailable status not detected 1b - Transmit Buffer Unavailable status detected
1 TPS	Transmit Process Stopped This bit is set when the transmission is stopped. Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Process Stopped status not detected 1b - Transmit Process Stopped status detected
0 TI	Transmit Interrupt This bit indicates that the packet transmission is complete. Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0b - Transmit Interrupt status not detected 1b - Transmit Interrupt status detected

61.7.1.566 Channel 4 Missed Frame Counter (DMA_CH4_MISS_FRAME_CNT)

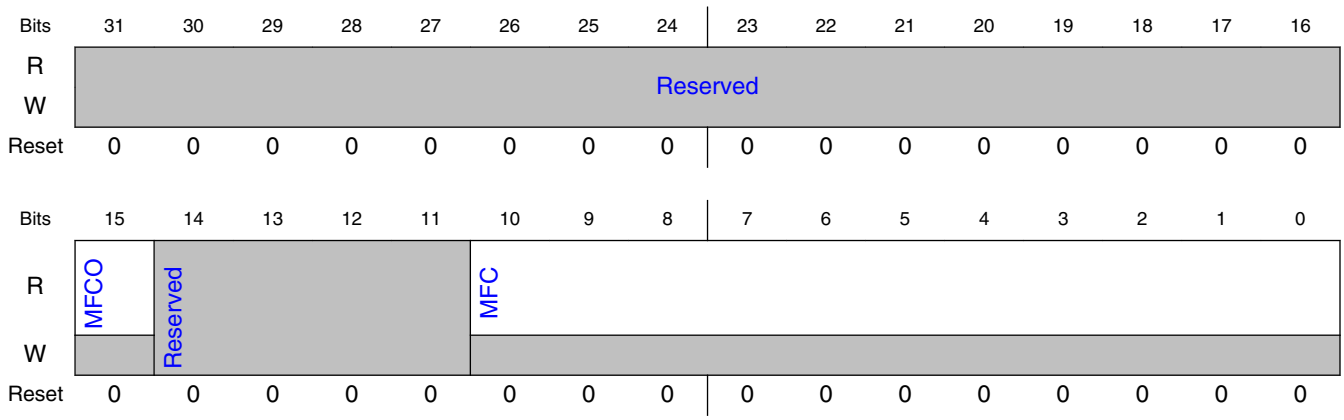
61.7.1.566.1 Offset

Register	Offset
DMA_CH4_MISS_FRAME_CNT	1364h

61.7.1.566.2 Function

This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH4_RX_CONTROL register.

61.7.1.566.3 Diagram



61.7.1.566.4 Fields

Field	Function
31-16 —	Reserved
15 MFCO	Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Miss Frame Counter overflow not occurred 1b - Miss Frame Counter overflow occurred
14-11 —	Reserved
10-0 MFC	Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in RX_CONTROL register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

61.7.1.567 Channel 4 RXP Frames Accepted Counter (DMA_CH4_RXP_ACCEPT_CNT)

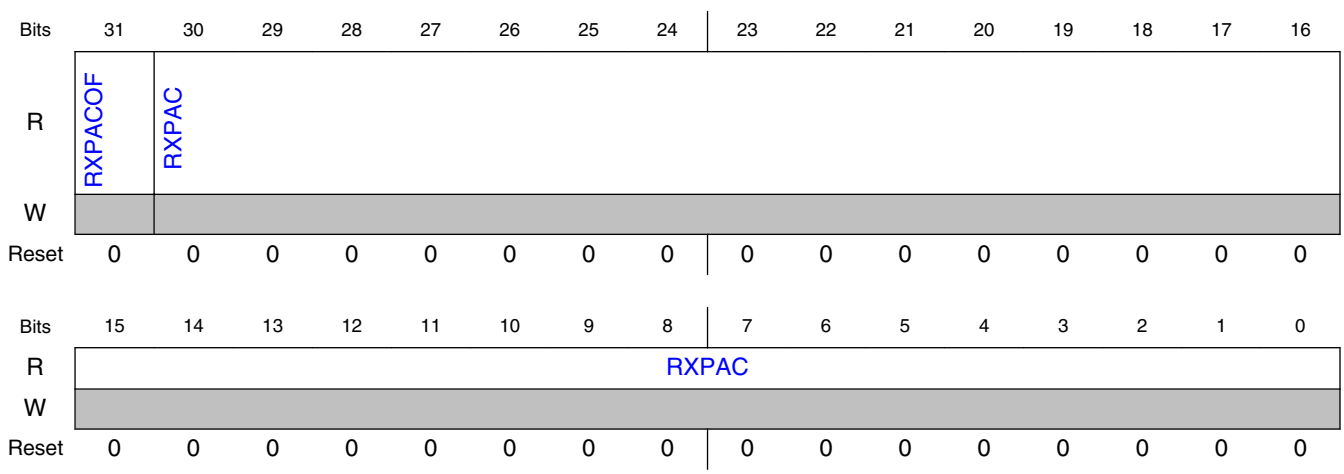
61.7.1.567.1 Offset

Register	Offset
DMA_CH4_RXP_ACCEPT_CNT	1368h

61.7.1.567.2 Function

The DMA_CH4_RXP_Accept_Cnt register provides the count of the number of frames accepted by Rx Parser.

61.7.1.567.3 Diagram



61.7.1.567.4 Fields

Field	Function
31 RXPACOF	Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Rx Parser Accept Counter Overflow Bit When set, this bit indicates that the RXPAC Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0b - Rx Parser Accept Counter overflow not occurred 1b - Rx Parser Accept Counter overflow occurred

Table continues on the next page...

Field	Function
30-0 RXPAC	Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. Rx Parser Accept Counter This 31-bit counter is implemented whenever a Rx Parser Accept a packet due to AF =1. The counter is cleared when the register is read.

Chapter 62
Universal Serial Bus Controller (USB)

62.1 Chip-specific USB information

Table 62-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

62.2 Overview

The USB controller block provides high performance USB functionality that conforms to the *Universal Serial Bus Specification*, Rev. 2.0 (Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips; 2000), and the *On-The-Go and Embedded Host Supplement to the USB Revision* 2.0 Specification (Hewlett-Packard Company, Intel Corporation, LSI Corporation, Microsoft Corporation, Renesas Electronics Corporation, ST-Ericsson; 2012).

The USB controller consists of two independent USB controller cores: two On-The-Go (OTG) controller cores. Each controller core supports UTMI interfaces according to its feature. See Features for more details. Both the controller cores are single-port cores. For the OTG cores, there is only one port. The port can be used as either a downstream or an upstream port.

62.2.1 Block Diagram

The following figure is a block diagram of USB.

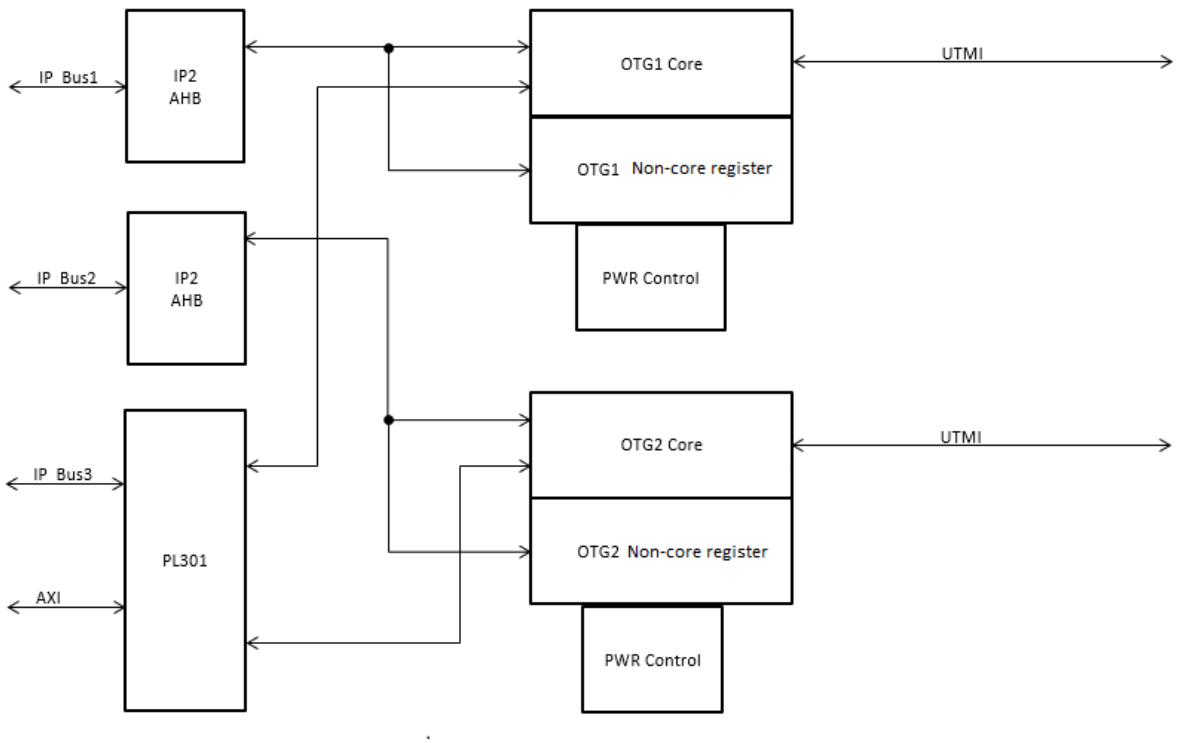


Figure 62-1. USB block diagram

62.2.2 Features

There are two USB 2.0 controller cores in this chip:

- Controller Core 0 is also named 'OTG1 Core'; its connected port is named 'OTG1 port'.
- Controller Core 1 is also named 'OTG2 Core'; its connected port is named 'OTG2 port'.

The following list provides features of each controller core.

- USB 2.0 Controller Core 0
  - High-Speed/Full-Speed/Low-Speed OTG core
  - HS/FS/LS UTMI compliant interface

    - High Speed, Full Speed and Low Speed operation in HOST mode (with UTMI transceiver)
    - High Speed, and Full Speed operation in Peripheral mode (with UTMI transceiver)
    - Hardware support for OTG signaling, session request protocol, and host negotiation protocol
    - 8 bidirectional endpoints
    - Support charger detection
- USB 2.0 Controller Core 1
    - High-Speed/Full-Speed/Low-Speed OTG core
    - HS/FS/LS UTMI compliant interface
    - High Speed, Full Speed and Low Speed operation in Host mode (with UTMI transceiver)
    - High Speed, and Full Speed operation in Peripheral mode (with UTMI transceiver)
    - Hardware support for OTG signaling, session request protocol, and host negotiation protocol
    - 8 bidirectional endpoints
- Low-power mode with local and remote wake-up capability
- Embedded DMA controller in each core

62.3 Functional Description

These sections describe the functionality of the various building blocks of the USB.

62.3.1 USB 2.0 Controller Core 0

The USB 2.0 Controller 0 is an instantiation of an EHCI-compatible core which supports high-, full-, and low-speed operation.

In Host mode, this controller core supports high-, full-, and low-speed operation. In Device mode, it supports high- and full-speed operation.

62.3.1.1 Host Mode

The controller supports direct connection of a HS/FS/LS device with on-chip UTMI transceiver.

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a USB 2.0 high speed hub has been implemented within the DMA and protocol engine blocks to support connection to full and low speed devices.

62.3.1.2 Peripheral (Device) Mode

- 8 bidirectional endpoints
- High/full-speed operation
- Support of HNP, and SRP
- Remote wake-up capability

62.3.2 USB 2.0 Controller Core 1

USB 2.0 Controller Core 1 is an instantiation of EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation

62.3.3 Modes of Operation

The USB has two main modes of operation: normal mode and low power mode .

Each USB OTG controller core can operate in High Speed operation (480 Mbps), Full Speed operation (12 Mbps) and Low Speed operation (1.5 Mbps).

This chapter explains the operation modes.

62.3.3.1 Normal Mode

The OTG controller core can operate in Host mode and Device (Peripheral) mode.

Each USB controller core has its corresponding port, which can work in one or more interface modes.

NOTE

Each controller supports only the interface type listed below. Selecting a different interface type in the PORTSC_PTS field results in unpredictable behavior and may cause the system to hang.

- OTG1 port

    - This port supports on-chip UTMI transceiver only.
- OTG2 port
    - This port supports on-chip UTMI transceiver only.

62.3.3.2 Low-Power Mode

Each USB controller core has a low-power mode (Suspend mode) to save power consumption.

As described in the USB 2.0 specification, the device can go into the Suspend state after it sees a constant Idle state on the upstream facing port. The OTG controller core enters Suspend mode after 3 ms of inactivity on the port when it is in Device Operation mode. Host controllers, including the OTG controller in Host mode, do not suspend automatically but can be placed in Suspend mode by software.

Either the local Arm platform or the remote USB Host/Peripheral can initiate a wake-up sequence to resume USB communication. For details about Suspend/Resume, see Power Control.

62.3.3.2.1 Power Control

The USB controller supports suspend and wake-up functionality.

The power control block allows for placing the transceiver in USB low power mode when USB bus is IDLE, and supports local and remote wake-up to bring the transceiver out of USB low power mode when needed. Additionally, the power control block can wake-up the Arm platform from core sleep mode by generating an interrupt.

62.3.3.2.1.1 Entering Low Power Suspend Mode

In Host operation mode, low power suspend mode is entered as follows:

1. Clear the ASE and PSE bits in USB_USBCMD, and wait until the AS and PS bits in USB_USBSTS become "0".
2. Set the "SUSPEND" bit in USB_PORTSC1
3. Set VBUSVALID_TO_SESSVALID in USBPHYx_USB1_VBUS_DETECT
4. Set the "PHCD" bit in USB_PORTSC1
5. Set all PWD bits in USBPHYx_PWD
6. Set CLKGATE in USBPHYx_CTRL

NOTE

Step 4,5,6 shall be done in atomic operation. That is, interrupt should be disabled during these three steps.

For device operation mode, low power suspend mode is entered as follows:

1. After Host drive is IDLE for 3ms, an SLI interrupt is issued (the "DCSUSPEND" or "SLI" bit in USB_USBSTS)
2. Set VBUSVALID_TO_SESSVALID in USBPHYx_USB1_VBUS_DETECT
3. Set the "PHCD" bit on USB_PORTSC1
4. Set all PWD bits in USBPHYx_PWD
5. Set CLKGATE in USBPHYx_CTRL

NOTE

Step 3,4,5 shall be done in atomic operation. That is, interrupt should be disabled during these three steps.

62.3.3.2.1.2 Wake-Up Events

The power control block monitors the USB bus when the USB core is in the USB suspend state.

Depending on whether the core is on Host or Device mode, a number of wake-up conditions are monitored. Upon detection of a wake-up condition, an interrupt will be generated to Arm platform if the related wake-up interrupt enable bit is set.

USB wake-up interrupt also re-activates the Arm platform clocks if they were stopped during the suspend.

62.3.3.2.1.2.1 Host Mode Events

The host controller wakes up on the following events:

- Remote Wake-up Request

A peripheral can request the host to reactivate the bus by driving wake-up signaling on the DM/DP lines. The power control block sends a wake-up request to the USB core when a J-K transition on DM/DP line is detected.

- Wake-Up On Overcurrent

If Wake-Up On Overcurrent is enabled (WKOC bit in the USB core register PORTSC1 is set '1'), the power control block sends a wake-up request to the USB core when an overcurrent event is detected.

- Wake-Up On Disconnect

If Wake-Up On Disconnect is enabled (WKDC bit in the USB core register PORTSC1 is set '1'), the power control block sends a wake-up request to the USB core when a disconnection event is detected (J-SE0/K-SE0 transition on DM/DP line).

- Wake-Up On Connect

If a Wake-Up On Connect is enabled (WKCN bit in the USB core register PORTSC1 is set '1'), the power control sub-block sends a wake-up request to the USB core when the connection event is detected (SE0-J/SE0-K transition on DM/DP line).

For a detailed description of register bits WKOC, WKDC, WKCN, please see USB_nPORTSC1.

62.3.4 Interrupts

62.3.4.1 USB Core Interrupts

Each USB core uses one dedicated vector in the Interrupt Table. The vector numbers associated with each of the cores can be found in the Interrupt section.

With the exception of the wake-up interrupts, all of the interrupt sources are controlled in the USB Cores. Refer to the USB_nUSBINTR for details.

62.3.4.2 USB Wake-Up Interrupts

Each USB Core has an associated wake-up interrupt. The wake-up interrupts are generated outside of the USB controller cores, but using the same vector as the corresponding USB controller cores interrupt.

These interrupts are generated by the Power Control blocks which run on the 32 KHz standby clock. The wake-up interrupt is designed to work even when the USB and Arm platform clocks are disabled, such that a wake-up condition on the USB bus can re-activate the Arm platform clocks.

Because the wake-up interrupt is generated and cleared on a 32 KHz clock, this interrupt request responds very slowly to clear actions. For this reason, the software must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously as this is clocked by the Arm platform clock. The software should wait for at least three 32 KHz clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is only used during low power modes of the USB, it is sufficient to enable the wake-up interrupt just prior to entering the USB suspend mode.

62.3.5 Clocks

The following table describes the clock sources for USB. Please see the clock control chapter for clock setting, configuration and gating information.

Table 62-2. USB Clocks

Clock	Description
ipg_ahb_clk	AHB bus clock
ipg_clk_s	Peripheral access clock
ipg_clk_s_pl301	PL301 peripheral clock

62.4 External Signals

The table found here describes the external signals of USB. See the chip-specific information to know the external signals supported by your device.

Table 62-3. USB External Signals

Signal	Description
OTGn_DN	DN OTG Signal
OTGn_DP	DP OTG Signal
OTGn_ID	ID signal
OTGn_OC	OTG External input for VBUS overcurrent detection
OTGn_PWR	To control PMIC to supply VBUS voltage

62.5 USB Operation Model

This section describes the detailed application knowledge for OTG1 and OTG2 ports.

62.5.1 Register Interface

Configuration, control and status registers are divided into three categories, identification, capability and operational registers.

NOTE

USB controller registers support only DWORD (32-bit) access.

- Identification registers are used to declare the slave interface presence along with the complete set of the hardware configuration parameters.
- Static, read only capability registers define the software limits, restrictions, and capabilities of the host/device controller.
- Operational registers are dynamic control or status registers that may be read only, read/write, or read/write-to-clear. The following sections define the use of these registers.

EHCI registers are listed alongside device registers to show the complementary nature of host and device control.

The following table describes the Interface register sets.

Table 62-4. Interface Register Sets

Offset	Register Set	Explanation
000h-07Ch	Identification Registers	Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.
100h-124h	Capability Registers	Capability registers specify the limits, restrictions, and capabilities of a host/device controller implementation. These values are used as parameters to the host/device controller driver.
080h-0FCh 140h-1FCh	Operational Registers	Operational registers are used by the system software to control and monitor the operational state of the host/device controller.

62.5.1.1 Configuration, Control and Status Register Set

The following table describes the Device/Host capability registers.

NOTE

Depending on implementation, "*x*" can have the following values: UOG1, UOG2.

Table 62-5. Device/Host Capability Registers

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
000h	4	USB_*x*_ID	Identification Register	O	O
004h	4	USB_*x*_HWGENERAL	General Hardware Parameters	O	O
008h	4	USB_*x*_HWHOST	Host Hardware Parameters	X	O
00Ch	4	USB_*x*_HWDEVICE	Device Hardware Parameters	O	X
010h	4	USB_*x*_HWTXBUF	TX Buffer Hardware Parameters	O	O
014h	4	USB_*x*_HWRXBUF	RX Buffer Hardware Parameters	O	O
018-07Fh		-	Reserved		

Table continues on the next page...

Table 62-5. Device/Host Capability Registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
080h	4	USB_*x*_GPTIMER0LD	General Purpose Timer #0 Load Register	O	O
084h	4	USB_*x*_GPTIMER0CTRL	General Purpose Timer #0 Control Register	O	O
088h	4	USB_*x*_GPTIMER1LD	General Purpose Timer #1 Load Register	O	O
08Ch	4	USB_*x*_GPTIMER1CTRL	General Purpose Timer #1 Control Register	O	O
090h	4	USB_*x*_SBUSCFG	System Bus Interface Configuration Register	O	O
094-09Fh		-	Reserved		
100h	1	USB_*x*_CAPLENGTH	Capability Register Length	O	O
101h		-	Reserved		
102h	2	USB_*x*_HCIVERSION	Host Controller Interface Version Number	X	O
104h	4	USB_*x*_HCSPARAMS	Host Controller Structural Parameters	X	O
108h	4	USB_*x*_HCCPARAMS	Host Controller Capability Parameters	X	O
10C-11Fh		-	Reserved		
120h	2	USB_*x*_DCIVERSION	Device Controller Interface Version Number	O	X
122h	2	-	Reserved		
124h	4	USB_*x*_DCCPARAMS	Device Controller Capability Parameters	O	X
128-13Fh		-	Reserved		
140h	4	USB_*x*_USBCMD	USB Command Register	O	O
144h	4	USB_*x*_USBSTS	USB Status Register	O	O
148h	4	USB_*x*_USBINTR	USB Interrupt Enable Register	O	O
14Ch	4	USB_*x*_FRINDEX	USB Frame Index	O	O
150h	4	-	Reserved		
154h	4	USB_*x*_PERIODICLISTBASE	Frame List Base Address	X	O
		USB_*x*_DEVICEADDR	USB Device Address	O	X
158h	4	USB_*x*_ASYNCLISTADDR	Next Asynchronous List Address	X	O
	4	USB_*x*_ENDPOINTLISTADDR	Address at Endpoint list in memory	O	X
15Ch	4	-	Reserved		
160h	4	USB_*x*_BURSTSIZE	Programmable Burst Size	O	O
164h	4	USB_*x*_TXFILLTUNING	Host Transmit Pre-Buffer Packet Tuning	X	O
168h	4	-	Reserved		
170h	4	-	Reserved		
178h	4	USB_*x*_ENDPTNAK	Endpoint NAK register	O	X
17Ch	4	USB_*x*_ENDPTNAKEN	Endpoint NAK Enable register	O	X
180h	4	USB_*x*_CONFIGFLAG	Configured Flag Register	X	O
184h	4	USB_*x*_PORTSC1	Port Status/Control Register 1	O	O
188-1A3h		-	Reserved		

Table continues on the next page...

Table 62-5. Device/Host Capability Registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
1A4h	4	USB_*x*_OTGSC	On-The-Go Status/Control Register (OTG only)	O	O
1A8h	4	USB_*x*_USBMODE	USB Controller Operating Mode	O	O
1ACh	4	USB_*x*_ENDPTSETUPSTAT	Endpoint Setup Status	O	X
1B0h	4	USB_*x*_ENDPTPRIME	Endpoint Initialization	O	X
1B4h	4	USB_*x*_ENDPTFLUSH	Endpoint De-Initialization	O	X
1B8h	4	USB_*x*_ENDPTSTATUS	Endpoint Status	O	X
1BCh	4	USB_*x*_ENDPTCOMPLETE	Endpoint Complete	O	X
1C0 1C4 ... 1DCh	64	USB_*x*_ENDPTCTRL0 USB_*x*_ENDPTCTRL1 .... USB_*x*_ENDPTCTRL7	Endpoint Control Register 0-7	O	X

NOTE

"O" means the register is available in host/device operation mode;

"X" means the register is reserved in host/device operation mode

62.5.1.2 Identification Registers

Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.

62.5.1.3 OTG Operations

62.5.1.3.1 Register Bits

In the previous section, the Register interface has behaviors described for device mode and behaviors described for host mode. However, for OTG operations it is necessary to perform tasks independent of the controller mode.

NOTE

The only way to transit the controller mode out of host or device mode is with the controller reset bit. Therefore, it is also

necessary for the OTG tasks to be performed independent of a controller reset as well as independent of the controller mode.

The following figure shows the controller mode.

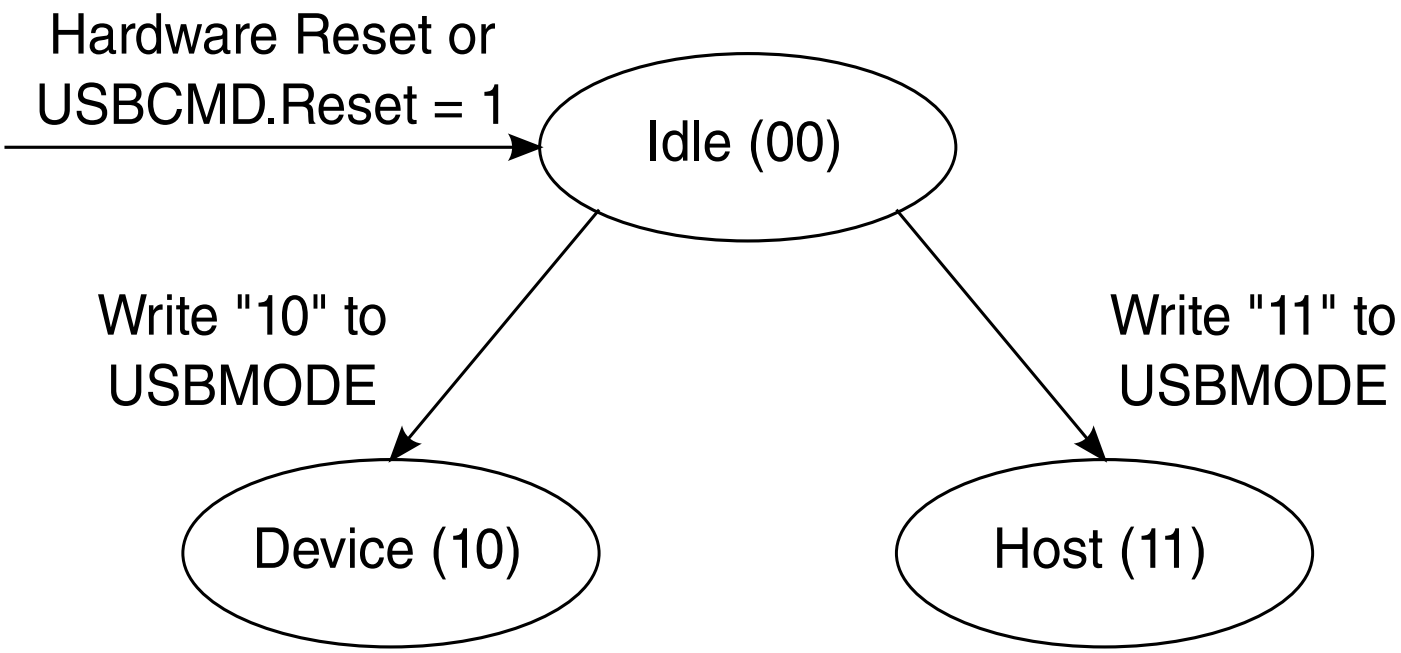


Figure 62-2. Controller Mode

To this end, listed below are the register bits that are used for OTG operations, which are independent of the controller mode and are also not affected by a write to the reset bit in the USBCMD register:

All Identification Registers

All Device/Host Capability Registers

OTGSC: All bits

PORTSC1:

- Physical Interface Select
- Physical Interface Serial Select
- Physical Interface Data Width
- Physical Interface Low Power
- Physical Interface Wake Signals
- Port Indicators
- Port Power

62.5.2 Host Data Structures

This section defines the interface data structures used to communicate control, status, and data between Host Controller Driver (HCD) (software) and the Enhanced Host Controller (hardware).

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of a Periodic Schedule, Periodic Frame List, Asynchronous Schedule, Isochronous Transaction Descriptors, Split-transaction Isochronous Transfer Descriptors, Queue Heads, and Queue Element Transfer Descriptors.

The periodic frame list is the root of all periodic (isochronous and interrupt transfer type) transfers for the host controller. The asynchronous list is the root for all the bulk and control transfers. Isochronous data streams are managed using Isochronous Transaction Descriptors. Isochronous split-transaction data streams are managed with Split-transaction Isochronous Transfer Descriptors. All Interrupt, Control, and Bulk data streams are managed via queue heads and Queue Element Transfer Descriptors. These data structures are optimized to reduce the total memory footprint of the schedule and to reduce (on average) the number of memory accesses needed to execute a USB transaction.

Note that software must ensure that no interface data structure reachable by the EHCI host controller spans a 4 K-page boundary.

The data structures defined in this section are (from the host controller's perspective) a mix of read-only and read/writeable fields. The host controller must preserve the read-only fields on all data structure writes.

62.5.2.1 Periodic Frame List

This schedule is for all periodic transfers (isochronous and interrupt). The periodic schedule is referenced from the operational registers space using the USB_PERIODICLISTBASE address register and the USB_FRINDEX register.

The periodic schedule is based on an array of pointers called the Periodic Frame List.

The USB_PERIODICLISTBASE address register is combined with the USB_FRINDEX register to produce a memory pointer into the frame list. The Periodic Frame List implements a sliding (1 msec) window of work over time.

The following figure shows the organization of periodic schedule.

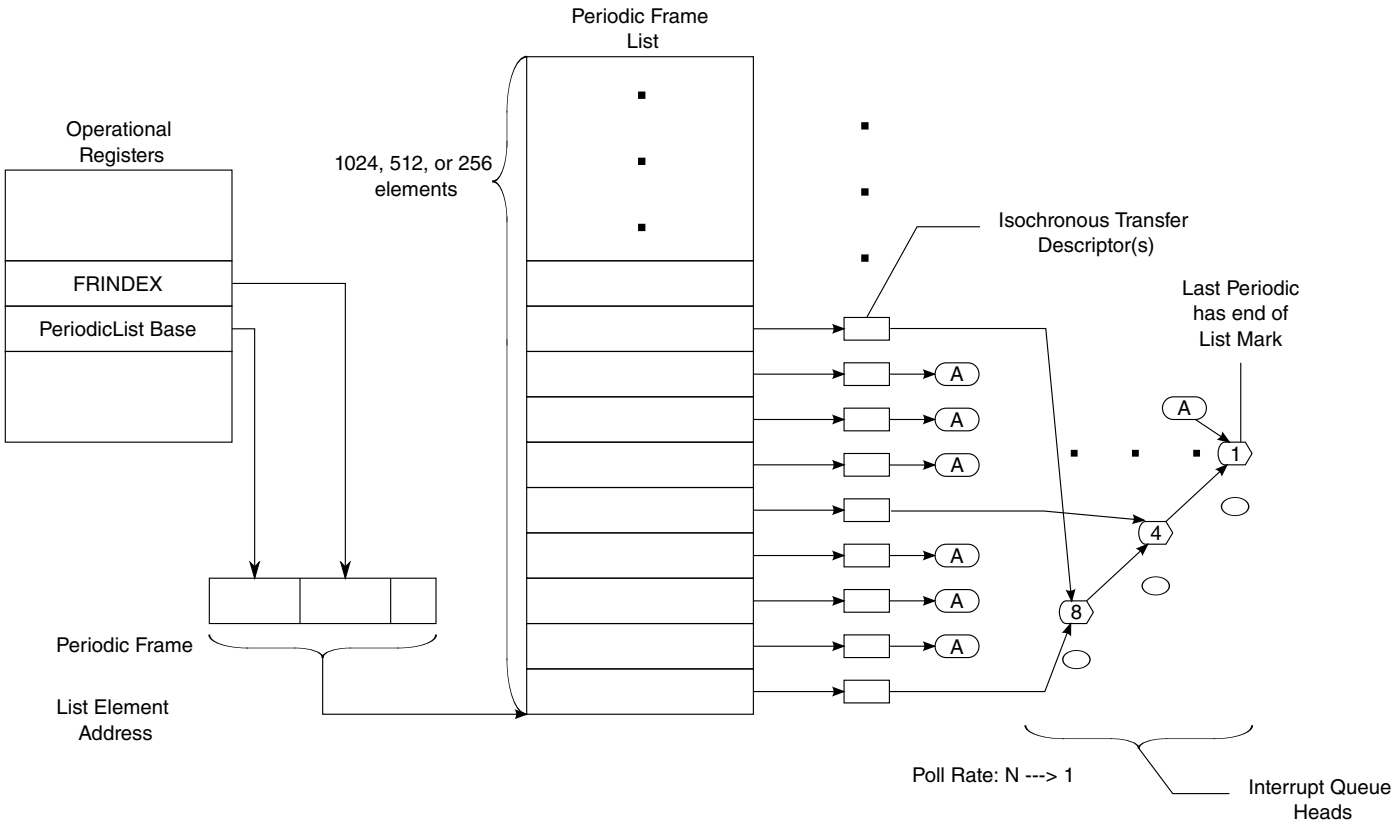


Figure 62-3. Periodic Schedule Organization

Split transaction Interrupt, Bulk and Control are also managed using queue heads and queue element transfer descriptors.

The periodic frame list is a 4 K-page aligned array of Frame List Link pointers. The length of the frame list may be programmable. The programmability of the periodic frame list is exported to system software via the USB_HCCPARAMS register. If non-programmable, the length is 1024 elements. If programmable, the length can be selected by system software as one of 8, 16, 32, 64, 128, 256, 512, or 1024 elements. An implementation must support all three sizes. Programming the size (that is, the number of elements) is accomplished by system software writing the appropriate value into Frame List Size field in the USB_USBCMD register.

Frame List Link pointers direct the host controller to the first work item in the frame's periodic schedule for the current micro-frame. The link pointers are aligned on DWord boundaries within the Frame List.

The table below illustrates the format of the Frame list element pointer.

Table 62-6. Format of Frame List Element Pointer

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	

Table continues on the next page...

Table 62-6. Format of Frame List Element Pointer (continued)

Frame List Link Pointer	0	Typ		03-00H

Frame List Link pointers always reference memory objects that are 32-byte aligned. The referenced object may be an isochronous transfer descriptor for high-speed devices, a split-transaction isochronous transfer descriptor (for full-speed isochronous endpoints), or a queue head (used to support high-, full- and low-speed interrupt). System software should not place non-periodic schedule items into the periodic schedule. The least significant bits in a frame list pointer are used to key the host controller as to the type of object the pointer is referencing.

The least significant bit is the T-Bit (bit 0). When this bit is set to a one, the host controller never uses the value of the frame list pointer as a physical memory pointer. The Typ field is used to indicate the exact type of data structure being referenced by this pointer. The value encodings are.

Table 62-7. Typ Field Value Definitions

Value	Meaning
00b	Isochronous Transfer Descriptor
01b	Queue Head
10b	Split Transaction Isochronous Transfer Descriptor.
11b	Frame Span Traversal Node.

62.5.2.2 Asynchronous List Queue Head Pointer

The Asynchronous Transfer List (based at the USB_ASYNCLISTADDR register) is where all of the control and bulk transfers are managed.

Host controllers use this list only when it reaches the end of the periodic list, the periodic list is disabled, or the periodic list is empty.

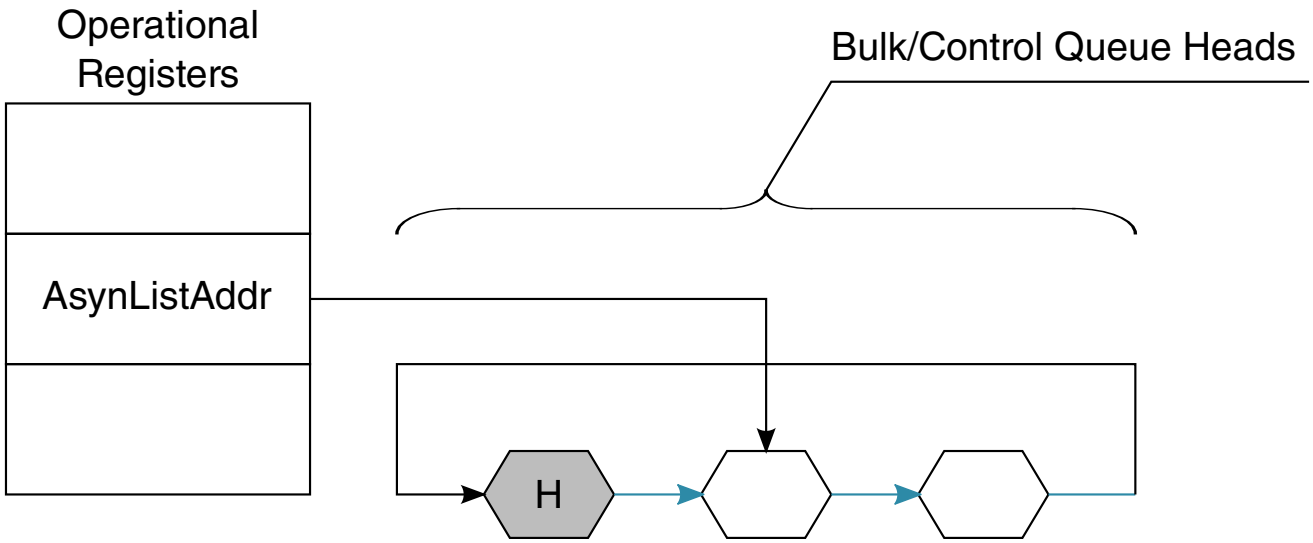


Figure 62-4. Asynchronous Schedule Organization

The Asynchronous list is a simple circular list of queue heads. The USB_ASYNCLISTADDR register is simply a pointer to the next queue head. This implements a pure round-robin service for all queue heads linked into the asynchronous list.

62.5.2.3 Isochronous (High-Speed) Transfer Descriptor (iTD)

The format of an isochronous transfer descriptor is shown in the table below.

This structure is used only for high-speed isochronous endpoints. All other transfer types should use queue structures. Isochronous TDs must be aligned on a 32-byte boundary.

Table 62-8. Isochronous Transaction Descriptor (iTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0		Typ		T	03-00 H
Status				Transaction 0 Length												IOC	PG*			Transaction 0 Offset*													07-04 H
Status				Transaction 1 Length												IOC	PG*			Transaction 1 Offset*													0B-08H
Status				Transaction 2 Length												IOC	PG*			Transaction 2 Offset*													0F-0CH
Status				Transaction 3 Length												IOC	PG*			Transaction 3 Offset*													13-10 H

Table continues on the next page...

Table 62-8. Isochronous Transaction Descriptor (iTD) (continued)

Status	Transaction 4 Length					IOC	PG*	Transaction 4 Offset*								17-14H
Status	Transaction 5 Length					IOC	PG*	Transaction 5 Offset*								1B-18H
Status	Transaction 6 Length					IOC	PG*	Transaction 6 Offset*								1F-1CH
Status	Transaction 7 Length					IOC	PG*	Transaction 7 Offset*								23-20H
Buffer Pointer (Page 0)								EndPt		R	Device Address					27-24H
Buffer Pointer (Page 1)								I/O	Maximum Packet Size							2B-28H
Buffer Pointer (Page 2)								-							Mult	2F-2CH
Buffer Pointer (Page 3)								-								33-30H
Buffer Pointer (Page 4)								-								37-34H
Buffer Pointer (Page 5)								-								3B-38H
Buffer Pointer (Page 6)								-								3F-3CH

Host Controller Read/Write

Host Controller Read Only

These fields may be modified by the host controller if the I/O field indicates an OUT.

62.5.2.3.1 Next Link Pointer

The first DWord of an iTD is a pointer to the next schedule data structure.

The following table describes the Next Schedule Element pointer field.

Table 62-9. Next Schedule Element Pointer

Bit	Description
31-5 Link Pointer (LP)	These bits correspond to memory address signals [31:5], respectively. This field points to another Isochronous Transaction Descriptor (iTD/siTD) or Queue Head (QH).
4-3	These bits are reserved and their value has no effect on operation. Software should initialize this field to zero.

Table continues on the next page...

Table 62-9. Next Schedule Element Pointer (continued)

Reserved	
2-1 QH/(s)iTD Select (Typ)	This field indicates to the Host Controller whether the item referenced is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0 Terminate (T)	1= Link Pointer field is not valid. 0= Link Pointer field is valid.

62.5.2.3.2 iTD Transaction Status and Control List

DWords 1 through 8 are eight slots of transaction control and status.

Each transaction description includes:

- Status results field
- Transaction length (bytes to send for OUT transactions and bytes received for IN transactions).
- Buffer offset. The PG and Transaction X Offset fields are used with the buffer pointer list to construct the starting buffer address for the transaction.

The host controller uses the information in each transaction description plus the endpoint information contained in the first three DWords of the Buffer Page Pointer list, to execute a transaction on the USB.

The following table describes iTD Transaction Status and Control fields.

Table 62-10. iTD Transaction Status and Control

Bit	Description	
31-28 Status	This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	31	Active. Set to one by software to enable the execution of an isochronous transaction by the Host Controller. When the transaction associated with this descriptor is completed, the Host Controller sets this bit to zero indicating that a transaction for this element should not be executed when it is next encountered in the schedule.
	30	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, no action is necessary.
	29	Babble Detected. Set to one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.

Table continues on the next page...

Table 62-10. iTD Transaction Status and Control (continued)

Bit	Description	
	28	Transaction Error (XactErr). Set to one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit may only be set for isochronous IN transactions.
27-16 Transaction X Length	For an OUT, this field is the number of data bytes the host controller sends during the transaction. The host controller is not required to update this field to reflect the actual number of bytes transferred during the transfer. For an IN, the initial value of the endpoint to deliver. During the status update, the host controller writes back the field is the number of bytes the host expects the number of bytes successfully received. The value in this register is the actual byte count (0‡zero length data, 1‡one byte, 2‡two bytes, etc.). The maximum value this field may contain is 0xC00 (3072).	
15 Interrupt On Complete (IOC)	If this bit is set to one, it specifies that when this transaction completes, the Host Controller should issue an interrupt at the next interrupt threshold.	
14-12 Page Select (PG)	These bits are set by software to indicate which of the buffer page pointers the offset field in this slot should be concatenated to produce the starting memory address for this transaction. The valid range of values for this field is 0 to 6.	
11-0 Transaction X Offset	This field is a value that is an offset, expressed in bytes, from the beginning of a buffer. This field is concatenated onto the buffer page pointer indicated in the adjacent *PG* field to produce the starting buffer address for this transaction.	

62.5.2.3.3 iTD Buffer Page Pointer List (Plus)

DWords 9-15 of an isochronous transaction descriptor are nominally page pointers (4 K aligned) to the data buffer for this transfer descriptor. This data structure requires the associated data buffer to be contiguous (relative to virtual memory), but allows the physical memory pages to be non-contiguous.

Seven page pointers are provided to support the expression of eight isochronous transfers. The seven pointers allow for 3 (transactions) * 1024 (maximum packet size) * 8 (transaction records) (24576 bytes) to be moved with this data structure, regardless of the alignment offset of the first page.

Because each pointer is a 4 K aligned page pointer, the least significant 12 bits in several of the page pointers are used for other purposes.

The tables below illustrate the field descriptions.

Table 62-11. iTD Buffer Pointer Page 0 (Plus)

Bit	Description
31-12 Buffer Pointer (Page 0)	This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-8 Endpoint Number (Endpt)	This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.

Table continues on the next page...

Table 62-11. iTD Buffer Pointer Page 0 (Plus) (continued)

Bit	Description
7 Reserved	Bit reserved for future use and should be initialized by software to zero.
6-0 Device Address	This field selects the specific device serving as the data source or sink.

Table 62-12. iTD Buffer Pointer Page 1 (Plus)

Bit	Description
31-12 Buffer Pointer (Page 1)	This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11 Direction (I/O)	0 = OUT; 1 = IN. This field encodes whether the high-speed transaction should use an IN or OUT PID.
10-0 Maximum Packet Size	This directly corresponds to the maximum packet size of the associated endpoint *(wMaxPacketSize)*. This field is used for high-bandwidth endpoints where more than one transaction is issued per transaction description (per micro-frame). This field is used with the *Multi* field to support high-bandwidth pipes. This field is also used for all IN transfers to detect packet babble. Software should not set a value larger than 1024 (400h). Any value larger yields undefined results.

Table 62-13. iTD Buffer Pointer Page 2 (Plus)

Bit	Description
31-12 Buffer Pointer	This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-2 Reserved	This bit reserved for future use and should be set to zero.
1-0 Multi	This field is used to indicate to the host controller the number of transactions that should be executed per transaction description (per micro-frame). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro- frame. 10b Two transactions to be issued for this endpoint per micro- frame. 11b Three transactions to be issued for this endpoint per micro- frame.

Table 62-14. iTD Buffer Pointer Page 3-6

Bit	Description
31-12 Buffer Pointer	This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-0 Reserved	These bits reserved for future use and should be set to zero.

62.5.2.4 Split Transaction Isochronous Transfer Descriptor (siTD)

All Full-speed isochronous transfers through the internal transaction translator are managed using the siTD data structure. This data structure satisfies the operational requirements for managing the split transaction protocol.

The following table shows the Split Transaction Isochronous Transfer Descriptor (siTD).

Table 62-15. Split Transaction Isochronous Transfer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Next Link Pointer																											0		Typ		T	03-00
I/O	Port Number							-	Hub Addr							Reserved				EndPt				-	Device Address							07-04[1]
Reserved																µFrame C-mask								µFrame S-mask								0B-08[1]
ioc	P	Reserved				Total Bytes to Transfer										µFrame C-prog-mask								Status								0F-0C[2]
Buffer Pointer (Page 0)																				Current Offset												13-10[2]
Buffer Pointer (Page 1)																				Reserved							TP		T-count			17-14[2]
Back Pointer																											0				T	1B-18

1. 04-0B: Static Endpoint State
2. 0C-13: Transfer results

Host Controller Read/Write

Host Controller Read Only

62.5.2.4.1 Next Link Pointer

DWord0 of a siTD is a pointer to the next schedule data structure.

The following table describes the Next Link Pointer fields.

Table 62-16. Next Link Pointer

Bit	Description
31-5	Next Link Pointer (LP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.

Table continues on the next page...

Table 62-16. Next Link Pointer (continued)

Bit	Description
4-3	Reserved. These bits must be written as zeros.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD/siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1 = Link Pointer field is not valid. 0 = Link Pointer is valid.

62.5.2.4.2 siTD Endpoint Capabilities/Characteristics

DWords 1 and 2 specify static information about the full-speed endpoint, the addressing of the parent Companion Controller, and micro-frame scheduling control.

The tables below describe the Endpoint and transaction translator characteristics and micro-frame schedule control fields.

Table 62-17. Endpoint and Transaction Translator Characteristics

Bit	Description
31	Direction (I/O).0 = OUT; 1 = IN. This field encodes whether the full-speed transaction should be an IN or OUT.
30-24	Port Number. This field is the port number of the recipient Transaction Translator.
23	Reserved. Bit reserved and should be set to zero.
22-16	Hub Address. This field holds the device address of the Companion Controllers' hub.
15-12	Reserved. Field reserved and should be set to zero.
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit is reserved for future use. It should be set to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 62-18. Micro-frame Schedule Control

Bit	Description
31-16	Reserved. This field reserved for future use. It should be set to zero.
15-8	Split Completion Mask (mFrame C-Mask). This field (along with the *Active* and *SplitX- state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute complete-split transactions. When the criteria for using this field is met, an all zeros value has undefined behavior. The host

Table continues on the next page...

Table 62-18. Micro-frame Schedule Control (continued)

Bit	Description
	controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame C-Mask* field is a one, then this siTD is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Split Start Mask (mFrame S-mask). This field (along with the *Active* and *SplitX-state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute start-split transactions. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame S-mask* field is a one, then this siTD is a candidate for transaction execution. An all zeros value in this field, in combination with existing periodic frame list has undefined results.

62.5.2.4.3 siTD Transfer State

DWords 3-6 are used to manage the state of the transfer.

The following table describes siTD transfer state fields.

Table 62-19. siTD Transfer Status and Control

Bit	Description
31	Interrupt On Complete (ioc). 0 = Do not interrupt when transaction is complete. 1 = Do interrupt when transaction is complete. When the host controller determines that the split transaction has completed it asserts a hardware interrupt at the next interrupt threshold.
30	Page Select (P). Used to indicate which data page pointer should be concatenated with the *CurrentOffset* field to construct a data buffer pointer (0 selects *Page 0* pointer and 1 selects *Page 1*). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).
29-26	Reserved. This field reserved for future use and should be set to zero.
25-16	Total Bytes To Transfer. This field is initialized by software to the total number of bytes expected in this transfer. Maximum value is 1023 (3FFh)
15-8	µFrame Complete-split Progress Mask (C-prog-Mask). This field is used by the host controller to record which split-completes has been executed.
7-0: Status—This field records the status of the transaction executed by the host controller for this slot. It is a bit vector with the encoding shown in the following rows.	
7	Active. Set to one by software to enable the execution of an isochronous split transaction by the Host Controller.
6	ERR. Set to a one by the Host Controller when an ERR response is received from the Companion Controller.
5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). In the case of an under run, the Host Controller transmits an incorrect CRC (thus invalidating the data at the endpoint). If an overrun condition occurs, no action is necessary.
4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.
3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit is set only for IN transactions.
2	Missed Micro-Frame. The host controller detected that a host-induced hold- off caused the host controller to miss a required complete-split transaction.
1	Split Transaction State (SplitXstate). The bit encodings are:

Table continues on the next page...

Table 62-19. siTD Transfer Status and Control (continued)

Bit	Description
	Value Meaning 00b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint when a match is encountered in the S-mask. 01b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint when a match is encountered in the C-mask.
0	Reserved. Bit reserved for future use and should be set to zero.

62.5.2.4.4 siTD Buffer Pointer List (plus)

DWords 4 and 5 are the data buffer page pointers for the transfer. This structure supports one physical page cross. The most significant 20 bits of each DWord in this section are the 4 K (page) aligned buffer pointers.

The least significant 12 bits of each DWord are used as additional transfer state. The following table describes the siTD buffer pointer fields.

Table 62-20. Buffer Page Pointer List (plus)

Bit	Description
31-12	Buffer Pointer List. Bits [31:12] of DWords 4 and 5 are 4 K page aligned physical memory addresses. These bits correspond to physical address bits [31:12] respectively. The lower 12 bits in each pointer are defined and used as specified below. The field *P* (see siTD Transfer State) specifies the *current* active pointer.
Bits 11-0 (Page 0)	Current Offset—The 12 least significant bits of the Page 0 pointer are the current byte offset for the current page pointer (as selected with the page indicator bit (*P* field). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).
Bits 11-0 (Page 1)—The least significant bits of the Page 1 pointer are split into three subfields as shown in the following rows.	
11-5 (Page 1)	Reserved
4-3 (Page 1)	Transaction position (TP). This field is used with T-count to determine whether to send *all*, *first*, *middle*, or *last* with each outbound transaction payload. System software must initialize this field with the appropriate starting value. The host controller must correctly manage this state during the lifetime of the transfer. The bit encodings are: Value Meaning 00b All. The entire full-speed transaction data payload is in this transaction (that is, less than or equal to 188 bytes). 01b Begin. This is the first data payload for a full- speed that is greater than 188 bytes. 10B Mid. This is the *middle* payload for a full-speed OUT transaction that is larger than 188 bytes. 11b End. This is the *last* payload for a full-speed OUT transaction that was larger than 188 bytes.
2-0 (Page 1)	Transaction count (T-Count). Software initializes this field with the number of OUT start-splits this transfer requires. Any value larger than 6 is undefined.

62.5.2.4.5 siTD Back Link Pointer

DWord 6 of a siTD is simply another schedule link pointer. This pointer is always zero, or references a siTD, and it cannot reference any other schedule data structure.

The following table describes the siTD back link pointer fields.

Table 62-21. siTD Back Link Pointer

Bit	Description
31-5	siTD Back Pointer. This field is a physical memory pointer to a siTD.
4-1	Reserved. This field is reserved for future use. It should be set to zero.
0	Terminate (T). 1 = siTD Back Pointer field is not valid. 0 = siTD Back Pointer field is valid.

62.5.2.5 Queue element transfer descriptor (qTD)

This data structure is only used with a queue head. It describes one or more USB transactions to transfer up to 20480 (5*4096) bytes.

The structure contains two structure pointers used for queue advancement, a DWord of transfer state, and a five-element array of data buffer pointers.

It is 32 bytes and must be physically contiguous.

The buffer associated with this transfer must be virtually contiguous. The buffer may start on any byte boundary; however, for optimal utilization of on-chip busses it is recommended to align the buffers on a 32-byte boundary. A separate buffer pointer list element must be used for each physical page in the buffer, regardless of whether the buffer is physically contiguous.

Host controller updates (host controller writes) to stand-alone qTDs only occur during transfer retirement. References in the following bit field definitions of updates to the qTD are to the qTD portion of a queue head.

The following table shows the queue element transfer descriptor data structure.

Table 62-22. Queue element transfer descriptor data structure

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Next qTD Pointer																											0				T	03-00

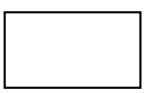
Table continues on the next page...

Table 62-22. Queue element transfer descriptor data structure (continued)

Alternate Next qTD Pointer						0	T	07-04
dt	Total Bytes to Transfer	ioc	C_Page	Cerr	PID Code	Status		0B-08[1]
Buffer Pointer (page 0)				Current Offset				0F-0C[1]
Buffer Pointer (page 1)				Reserved				13-10
Buffer Pointer (page 2)				Reserved				17-14
Buffer Pointer (page 3)				Reserved				1B-18
Buffer Pointer (page 4)				Reserved				1F-1C

1. 08-0F: Transfer Results

Host Controller Read/Write



Host Controller Read Only

Queue Element Transfer Descriptors must be aligned on 32-byte boundaries.

62.5.2.5.1 Next qTD Pointer

The first DWord of an element transfer descriptor is a pointer to another transfer element descriptor.

The following table describes Next qTD pointer fields.

Table 62-23. qTD Next Element Transfer Pointer (DWord 0)

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed. The field corresponds to memory address signals[31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

62.5.2.5.2 Alternate Next qTD Pointer

The second DWord of a queue element transfer descriptor is used to support hardware-only advance of the data stream to the next transfer descriptor on short packet. To be more explicit the host controller always uses this pointer when the current qTD is retired due to short packet.

The following table describes the TD Alternate Next Element Transfer Pointer field descriptions.

Table 62-24. TD Alternate Next Element Transfer Pointer (DWord 1)

Bit	Description
31-5	Alternate Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed in the event that the current qTD execution encounters a short packet (for an IN transaction). The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

62.5.2.5.3 qTD Token

The third DWord of a queue element transfer descriptor contains most of the information the host controller requires to execute a USB transaction (the remaining endpoint-addressing information is specified in the queue head).

NOTE

The field descriptions forward reference fields defined in the queue head. Where necessary, these forward references are preceded with a QH notation.

The following table describes the TD Token fields.

Table 62-25. TD Token (DWord 2)

Bit	Description
31 Data Toggle	This is the data toggle sequence bit. The use of this bit depends on the setting of the *Data Toggle Control* bit in the queue head.
30-16 Total Bytes to Transfer	This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction, only on the successful completion of the transaction. The maximum value software may store in this field is 5 * 4K (5000H). This is the maximum number of bytes 5 page pointers can access. If the value of this field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the host controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for OUT transfers that *Total Bytes To Transfer* be an even multiple of QHD.Maximum Packet Length. If software builds such a transfer descriptor for an OUT transfer, the last transaction is always less than QHD.Maximum Packet Length.

Table continues on the next page...

Table 62-25. TD Token (DWord 2) (continued)

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td></td><td colspan="2">Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16 K(4000H).</td></tr>
<tr><td>15
Interrupt On Complete (IOC)</td><td colspan="2">If this bit is set to a one, it specifies that when this qTD is completed, the Host Controller should issue an interrupt at the next interrupt threshold.</td></tr>
<tr><td>14-12
Current Page (C_Page)</td><td colspan="2">This field is used as an index into the qTD buffer pointer list. Valid values are in the range 0H to 4H. The host controller is not required to write this field back when the qTD is retired.</td></tr>
<tr><td>11-10
Error Counter (CERR)</td><td colspan="2">This field is a 2-bit down counter that keeps track of the number of consecutive Errors detected while executing this qTD. If this field is programmed with a non-zero value during set-up, the Host Controller decrements the count and writes it back to the qTD if the transaction fails. If the counter counts from one to zero, the Host Controller marks the qTD inactive, sets the Halted bit to a one, and error status bit for the error that caused CERR to decrement to zero. An interrupt is generated if the USB Error Interrupt Enable bit in the USBINTR register is set to a one. If HCD programs this field to zero during set-up, the Host Controller does not count errors for this qTD and there is no limit on the retries of this qTD. Note that write-backs of intermediate execution state are to the queue head overlay area, not the qTD.
Transaction Error - Decrement
Data Buffer Error - No Decrement[3]
Stalled - No Decrement[1]
Babble Detected - No Decrement[1]
No Error - No Decrement[2]
<table>
<tr><td>Error</td><td>Decrement Counter</td></tr>
<tr><td>1</td><td>Detection of Babble or Stall automatically halts the queue head. Thus, count is not decremented</td></tr>
<tr><td>2</td><td>If the EPS field indicates a HS device or the queue head is in the Asynchronous Schedule (and PIDCode indicates an IN or OUT) and a bus transaction completes and the host controller does not detect a transaction error, then the host controller should reset CERR to extend the total number of errors for this transaction. For example, CERR should be reset with maximum value (3) on each successful completion of a transaction. The host controller must never reset this field if the value at the start of the transaction is 00b.
See Split Transaction Interrupt for CERR adjustment rules when the EPS field indicates a FS or LS device and the queue head is in the Periodic Schedule. See Asynchronous - Do Complete Split for CERR adjustment rules when the EPS field indicates a FS or LS device, the queue head is in the Asynchronous schedule and the PIDCode indicates a SETUP.</td></tr>
<tr><td>3</td><td>Data buffer errors are host problems. They don't count against the device's retries.</td></tr>
<tr><td colspan="2">NOTE: Software must not program CERR to a value of zero when the EPS field is programmed with a value indicating a Full- or Low-speed device. This combination could result in undefined behavior.</td></tr>
</table></td></tr>
<tr><td>9-8
PID Code</td><td colspan="2">This field is an encoding of the token, which should be used for transactions associated with this transfer descriptor. Encodings are:
<table>
<tr><td>00b</td><td>OUT Token generates token (E1H)</td></tr>
<tr><td>01b</td><td>IN Token generates token (69H)</td></tr>
</table></td></tr>
</table>

Table continues on the next page...

Table 62-25. TD Token (DWord 2) (continued)

Bit	Description	
	10b	SETUP Token generates token (2DH) (undefined if endpoint is an interrupt, the queue head is non-zero) transfer type, for example, µ*Frame S-mask* field in.
	11b	Reserved
7-0 Status	This field is used by the Host Controller to communicate individual command execution states back to HCD. This field contains the status of the last transaction performed on this qTD. The bit encodings are:	
	Bit	Status Field Description
	7	Active. Set to one by software to enable the execution of transactions by the Host Controller.
	6	Halted. Set to one by the Host Controller during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this qTD. This can be caused by babble, the error counter counting down to zero, or reception of the STALL handshake from the device during a transaction. Any time that a transaction results in the Halted bit being set to a one, the Active bit is also set to zero.
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, the Host Controller forces a timeout condition on the USB, invalidating the transaction at the source. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction. In addition to setting this bit, the Host Controller also sets the *Halted* bit to a one. Because "babble" is considered a fatal error for the transfer, setting the Halted bit to a one insures that no more transactions occur because of this descriptor.
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	2	Missed Micro-Frame. This bit is ignored unless the *QH.EPS* field indicates a full- or low-speed endpoint and the queue head is in the periodic list. This bit is set when the host controller detected that a host-induced hold-off caused the host controller to miss a required complete-split transaction. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	1	Split Transaction State (SplitXstate). This bit is ignored by the host controller unless the *QH.EPS* field indicates a full- or low-speed endpoint. When a Full- or Low-speed device, the host controller uses this bit to track the state of the split-transaction. The functional requirements of the host controller for managing this state bit and the split transaction protocol depends on whether the endpoint is in the periodic or asynchronous schedule. The bit encodings are: Value Meaning 0b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint. 1b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint.

Table continues on the next page...

Table 62-25. TD Token (DWord 2) (continued)

Bit	Description	
	0	Ping State (P)/ERR. If the *QH.EPS* field indicates a High-speed device and the *PID_Code* indicates an OUT endpoint, then this is the state bit for the Ping protocol. The bit encodings are: Value Meaning 0b Do OUT. This value directs the host controller to issue an OUT PID to the endpoint. 1b Do Ping. This value directs the host controller to issue a PING PID to the endpoint. If the *QH.EPS* field does not indicate a High-speed device, then this field is used as an error indicator bit. It is set to a one by the host controller whenever a periodic split-transaction receives an ERR handshake.

62.5.2.5.4 qTD Buffer Page Pointer List

The last five DWords of a queue element transfer descriptor is an array of physical memory address pointers. These pointers reference the individual pages of a data buffer.

System software initializes Current Offset field to the starting offset into the current page, where current page is selected through the value in the C_Page field.

The following table describes the qTD Buffer Pointer(s) (DWords 3-7) fields.

Table 62-26. qTD Buffer Pointer(s) (DWords 3-7)

Bit	Description
31-12	Buffer Pointer List. Each element in the list is a 4 K page aligned physical memory address. The lower 12 bits in each pointer are reserved (except for the first one), as each memory pointer must reference the start of a 4 K page. The field C_Page specifies the current active pointer. When the transfer element descriptor is fetched, the starting buffer address is selected using C_Page (similar to an array index to select an array element). If a transaction spans a 4K buffer boundary, the host controller must detect the page-span boundary in the data stream, increment C_Page and advance to the next buffer pointer in the list, and conclude the transaction through the new buffer pointer.
11-0	Current Offset (Reserved). This field is reserved in all pointers except the first one (for example Page 0). The host controller should ignore all reserved bits. For the page 0 current offset interpretation, this field is the byte offset into the current page (as selected by C_Page). The host controller is not required to write this field back when the qTD is retired. Software should ensure the Reserved fields are initialized to zero.

62.5.2.6 Queue Head

The table located in this section shows the Queue Head structure layout.

The following table shows the queue head structure layout.

Table 62-27. Queue Head Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Queue Head Horizontal Link Pointer																											0		Typ		T	03-00
RL			C	Maximum Packet Length												H	dtc	EP		EndPt				I	Device Address							07-04[1]
Mult		Port Number[2]						Hub Addr[2]								μFrame C-mask[2]								μFrame S-mask								0B-08[1]
Current qTD Pointer																											0					0F-0C
Next qTD Pointer																											0				T	13-10[3]
Alternate Next qTD pointer																											NakCnt				T	17-14[4]
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								1B-18
Buffer Pointer (Page 0)																				Current Offset												1F-1C
Buffer Pointer (Page 1)																				Reserved				C-prog-mask[2]								23-20
Buffer Pointer (Page 2)																				S-bytes[2]										FrameTag[2]		27-24[4]
Buffer Pointer (Page 3)																				Reserved												2B-28
Buffer Pointer (Page 4)																				Reserved												2F-2C[3]

1. 04-0B: Static endpoint state.
2. These fields are used exclusively to support split transactions to USB 2.0 hubs
3. 10-2F: Transfer overlay.
4. 14-27: Transfer results.

Host Controller Read/Write

Host Controller Read Only

62.5.2.6.1 Queue Head Horizontal Link Pointer

The first DWord of a Queue Head contains a link pointer to the next data object to be processed after any required processing in this queue has been completed, as well as the control bits defined below.

This pointer may reference a queue head or one of the isochronous transfer descriptors. It must not reference a queue element transfer descriptor.

The following table describes the Queue head DWord 0 fields.

Table 62-28. Queue Head DWord 0

Bit	Description
31-5	Queue Head Horizontal Link Pointer (QHLP). This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD Select (Typ). This field indicates to the hardware whether the item referenced by the link pointer is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Last QH (pointer is invalid). 0=Pointer is valid. If the queue head is in the context of the periodic list, a one bit in this field indicates to the host controller that this is the end of the periodic list. This bit is ignored by the host controller when the queue head is in the Asynchronous schedule. Software must ensure that queue heads reachable by the host controller always have valid horizontal link pointers.

62.5.2.6.2 Queue Head Endpoint Capabilities/Characteristics

The second and third DWords of a Queue Head specifies static information about the endpoint. This information does not change over the lifetime of the endpoint.

There are three types of information in this region:

- Endpoint Characteristics. These are the USB endpoint characteristics including addressing, maximum packet size, and endpoint speed.
- Endpoint Capabilities. These are adjustable parameters of the endpoint. They effect how the endpoint data stream is managed by the host controller.
- Split Transaction Characteristics. This data structure is used to manage full- and low-speed data streams for bulk, control, and interrupt via split transactions to USB2.0 Hub Transaction Translator. There are additional fields used for addressing the hub and scheduling the protocol transactions (for periodic).

The host controller must not modify the bits in this region.

The following table describes the Endpoint characteristics: Queue head DWord 1 fields.

Table 62-29. Endpoint Characteristics: Queue Head DWord 1

Bit	Description
31-28	Nak Count Reload (RL). This field contains a value, which is used by the host controller to reload the Nak Counter field.

Table continues on the next page...

Table 62-29. Endpoint Characteristics: Queue Head DWord 1 (continued)

27	Control Endpoint Flag (C). If the *QH.EPS* field indicates the endpoint is not a high-speed device, and the endpoint is a control endpoint, then software must set this bit to a one. Otherwise, it should always set this bit to zero.	
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (*wMaxPacketSize*). The maximum value this field may contain is 0x400 (1024).	
15	Head of Reclamation List Flag (H). This bit is set by System Software to mark a queue head as being the head of the reclamation list.	
14	Data Toggle Control (DTC). This bit specifies where the host controller should get the initial data toggle on an overlay transition. 0b Ignore DT bit from incoming qTD. Host controller preserves DT bit in the queue head. 1b Initial data toggle comes from incoming qTD DT bit. Host controller replaces DT bit in the queue head from the DT bit in the qTD.	
13-12	Endpoint Speed (EPS). This is the speed of the associated endpoint. Bit combinations are:	
	Value	Meaning
	00b	Full-Speed (12 Mbits/sec)
	01b	Low-Speed (1.5 Mbits/sec)
	10b	High-Speed (480 Mbits/sec)
	11b	Reserved
	This field must not be modified by the host controller.	
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.	
7	Inactivate on Next Transaction (I). This bit is used by system software to request that the host controller set the Active bit to zero. See Rebalancing the periodic schedule , for full operational details. This field is only valid when the queue head is in the Periodic Schedule and the *EPS* field indicates a Full or Low-speed endpoint. Setting this bit to one when the queue head is in the Asynchronous Schedule or the *EPS* field indicates a high-speed device yields undefined results.	
6-0	Device Address. This field selects the specific device serving as the data source or sink.	

The table below describes the Endpoint capabilities: Queue head DWord 2 field descriptions.

Table 62-30. Endpoint Capabilities: Queue Head DWord 2

Bit	Description
31-30	High-Bandwidth Pipe Multiplier (Mult). This field is a multiplier used to key the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. The host controller makes the simplifying assumption that software properly initializes this field (regardless of location of queue head in the schedules or other run time parameters). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro-frame. 10b Two transactions to be issued for this endpoint per micro-frame. 11b Three transactions to be issued for this endpoint per micro-frame.

Table continues on the next page...

Table 62-30. Endpoint Capabilities: Queue Head DWord 2 (continued)

29-23	Port Number. This field is ignored by the host controller unless the *EPS* field indicates a full- or low-speed device. The value is the port number identifier on the USB 2.0 Hub (for hub at device address *Hub Addr* below), below which the full- or low-speed device associated with this endpoint is attached. This information is used in the split-transaction protocol.
22-16	Hub Addr. This field is ignored by the host controller unless the *EPS* field indicates a full-or low-speed device. The value is the USB device address of the USB 2.0 Hub below which the full- or low-speed device associated with this endpoint is attached. This field is used in the split-transaction protocol.
15-8	Split Completion Mask (µFrame C-Mask). This field is ignored by the host controller unless the *EPS* field indicates this device is a low- or full-speed device and this queue head is in the periodic list. This field (along with the *Active* and *SplitX-state* fields) is used to determine during which micro-frames the host controller should execute a complete-split transaction. When the criteria for using this field are met, a zero value in this field has undefined behavior. This field is used by the host controller to match against the three low-order bits of the FRINDEX register. If the FRINDEX register bits decode to a position where the µ*Frame C- Mask* field is a one, then this queue head is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Interrupt Schedule Mask (µFrame S-mask). This field is used for all endpoint speeds. Software should set this field to a zero when the queue head is on the asynchronous schedule. A non-zero value in this field indicates an interrupt endpoint. The host controller uses the value of the three low-order bits of the FRINDEX register as an index into a bit position in this bit vector. If the µ*Frame S-mask* field has a one at the indexed bit position then this queue head is a candidate for transaction execution. If the *EPS* field indicates the endpoint is a high-speed endpoint, then the transaction executed is determined by the *PID_Code* field contained in the execution area. This field is also used to support split transaction types: Interrupt (IN/OUT). This condition is true when this field is non-zero and the *EPS* field indicates this is either a full- or low-speed device. A zero value in this field, in combination with existing in the periodic frame list has undefined results.

62.5.2.6.3 Transfer Overlay-Queue Head

The nine DWords in this area represent a transaction working space for the host controller. The general operational model is that the host controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it follows the Queue Head Horizontal Link Pointer to the next queue head. The host controller will never follow the Next Transfer Queue Element or Alternate Queue Element pointers unless it is actively attempting to advance the queue. For the duration of the transfer, the host controller keeps the incremental status of the transfer in the overlay area. When the transfer is complete, the results are written back to the original queue element.

The DWord3 of a Queue Head contains a pointer to the source qTD currently associated with the overlay. The host controller uses this pointer to write back the overlay area into the source qTD after the transfer is complete.

The following table describes the current qTD link pointer field descriptions.

Table 62-31. Current qTD Link Pointer

Bit	Description

Table continues on the next page...

Table 62-31. Current qTD Link Pointer (continued)

31-5	Current Element Transaction Descriptor Link Pointer. This field contains the address Of the current transaction being processed in this queue and corresponds to memory address signals [31:5], respectively.
4-0	Reserved (R). These bits are ignored by the host controller when using the value as an address to write data. The actual value may vary depending on the usage.

The DWords 4-11 of a queue head are the transaction overlay area. This area has the same base structure as a Queue Element Transfer Descriptor. The queue head utilizes the reserved fields of the page pointers to implement tracking the state of split transactions.

This area is characterized as an overlay because when the queue is advanced to the next queue element, the source queue element is merged onto this area. This area serves as execution cache for the transfer.

The table below describes the Host-controller rules for bits in overlay.

Table 62-32. Host-Controller Rules for Bits in Overlay (DWords 5, 6, 8 and 9)

DWord	Bit	Description
5	4-1	Nak Counter (NakCnt)µRW. This field is a counter the host controller decrements whenever a transaction for the endpoint associated with this queue head results in a Nak or Nyet response. This counter is reloaded from *RL* before a transaction is executed during the first pass of the reclamation list (relative to an Asynchronous List Restart condition). It is also loaded from RL during an overlay.
6	31	Data Toggle. The *Data Toggle Control* controls whether the host controller preserves this bit when an overlay operation is performed.
6	15	Interrupt On Complete (IOC). The IOC control bit is always inherited from the source qTD when the overlay operation is performed.
6	11-10	Error Counter (C_ERR). This two-bit field is copied from the qTD during the overlay and written back during queue advancement.
6	0	Ping State (P)/ERR. If the *EPS* field indicates a high-speed endpoint, then this field should be preserved during the overlay operation.
8	7-0	Split-transaction Complete-split Progress (C-prog-mask). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	4-0	Split-transaction Frame Tag (Frame Tag). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	11-5	S-bytes. Software must ensure that the *S-bytes* field in a *qTD* is zero before activating the *qTD*. This field is used to keep track of the number of bytes sent or received during an IN or OUT split transaction.

62.5.2.7 Periodic Frame Span Traversal Node (FSTN)

This data structure is to be used only for managing Full- and Low-speed transactions that span a Host-frame boundary.

See Host Controller Operational Model for FSTNs for full operational details. Software must not use an FSTN in the Asynchronous Schedule. An FSTN in the Asynchronous schedule results in undefined behavior. Software must not use the FSTN feature with a host controller whose USB_HCIVERSION register indicates a revision implementation below 0096h. FSTNs are not defined for implementations before 0.96 and their use yields undefined results.

Table 62-33. Frame Span Traversal Node Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Addr
Normal Path Link Pointer																											0		Typ		T	03-00
Back Path Link Pointer																											0		Typ[1]		T	07-04

1. Must be set to indicate a queue head

Host Controller Read/Write

Host Controller Read Only

62.5.2.7.1 FSTN Normal Path Pointer

The first DWord of an FSTN contains a link pointer to the next schedule object. This object can be of any valid periodic schedule data type.

The following table describes the FSTN normal path pointer fields.

Table 62-34. FSTN Normal Path Pointer Field Descriptions

Bit	Description
31-5	Normal Path Link Pointer (NPLP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD/FSTN Select (Typ). This field indicates to the Host Controller whether the item referenced is a iTD/siTD, a QH or an FSTN. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (Frame Span Traversal Node)
0	Terminate (T). 1 = Link Pointer field is not valid. 0 = Link Pointer is valid.

62.5.2.7.2 FSTN Back Path Link Pointer

The second DWord of an FTSN node contains a link pointer to a queue head.

If the T-bit in this pointer is zero, then this FSTN is a Save-Place indicator. Its Typ field must be set by software to indicate the target data structure is a queue head. If the T-bit in this pointer is set to one, then this FSTN is the Restore indicator. When the T-bit is one, the host controller ignores the Typ field.

The following table describes the FSTN back path link pointer fields.

Table 62-35. FSTN Back Path Link Pointer Field Descriptions

Bit	Description
31-5	Back Path Link Pointer (BPLP). This field contains the address of a Queue Head. This field corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	Typ. Software must ensure this field is set to indicate the target data structure is a Queue Head. Any other value in this field yields undefined results.
0	Terminate (T). 1=Link Pointer field is not valid (that is the host controller must not use bits [31:5] as a valid memory address). This value also indicates that this FSTN is a Restore indicator. 0=Link Pointer is valid (that is the host controller may use bits [31:5] (in combination with the CTRLDSSEGMENT register if applicable) as a valid memory address). This value also indicates that this FSTN is a Save-Place indicator.

62.5.3 Host Operational Model

The general operational model is for the enhanced interface host controller hardware and enhanced interface host controller driver (generally referred to as system software).

Each significant operational feature of the EHCI host controller is discussed in a separate section. Each section presents the operational model requirements for the host controller hardware. Where appropriate, recommended system software operational models for features are also presented.

62.5.3.1 Host Controller Initialization

After initial power-on or HCReset (hardware or through HCReset bit in the USB_USBCMD register), all of the operational registers are at their default values. After a hardware reset, only the operational registers not contained in the Auxiliary power well are at their default values.

The following table describes the default values of operational registers.

Table 62-36. Default Values of Operational Register Space

Operational Register	Default Value (after Reset)
USB_USBCMD	00080000h (00080B00h, if *Asynchronous Schedule Park Capability is one*)
USB_USBSTS	00001000h
USB_USBINTR	00000000h
USB_FRINDEX	00000000h
USB_CTRLDSSEGMENT	00000000h
USB_PERIODICLISTBASE	Undefined
USB_ASYNCLISTADDR	Undefined
USB_CONFIGFLAG	00000000h
USB_PORTSC1	00002000h (w/*PPC* set to one); 00003000h (w/*PPC* set to zero)

To initialize the host controller, software should perform the following steps:

- Write the appropriate value to the USB_USBINTR register to enable the appropriate interrupts.
- Write the base address of the Periodic Frame List to the USB_PERIODICLIST BASE register. If no work items are in the periodic schedule, all elements of the Periodic Frame List should have their T-Bits set to one.
- Write the USB_USBCMD register to set the desired interrupt threshold, frame list size (if applicable) and turn the host controller ON through setting the Run/Stop bit.

At this point, the host controller is up and running and the port registers begin reporting device connects, and so on. System software can enumerate a port through the reset process (where the port is in the enabled state). At this point, the port is active with SOFs occurring down the enabled ports, but the schedules have not enabled. To communicate with devices through the asynchronous schedule, system software must write the USB_ASYNCLISTADDR register with the address of a control or bulk queue head. Software must then enable the asynchronous schedule by writing one to the Asynchronous Schedule Enable bit in the USB_USBCMD register. To communicate with devices through the periodic schedule, system software must enable the periodic schedule by writing one to the Periodic Schedule Enable bit in the USB_USBCMD register.

NOTE

The schedules can be turned on before the first port is reset (and enabled).

When the USB_USBCMD register is written, system software must ensure the appropriate bits are preserved, depending on the intended operation.

62.5.3.2 Port Routing and Control

The EHCI specification defines that a USB 2.0 Host controller is comprised of one high-speed host controller, which implements the EHCI programming interface and 0 to N USB 1.1 companion host controllers.

Companion host controllers (cHCs) may be implementations of either Universal or Open host controller specifications. This configuration is used to deliver the required full USB 2.0-defined port capability; for example, Low-, Full-, and High-speed capability for every port.

NOTE

The USB controllers on do not require nor support companion controllers to support Full and Low Speed device. Full and Low Speed devices are supported within the USB controller by emulating the functionality of a high-speed HUB. Therefore, no port routing is present in the controller. Please refer to Embedded Transaction Translator Function for details.

The following figure illustrates a simple block diagram of the port routing logic and its relationship to the high-speed and companion host controllers within a USB 2.0 host controller.

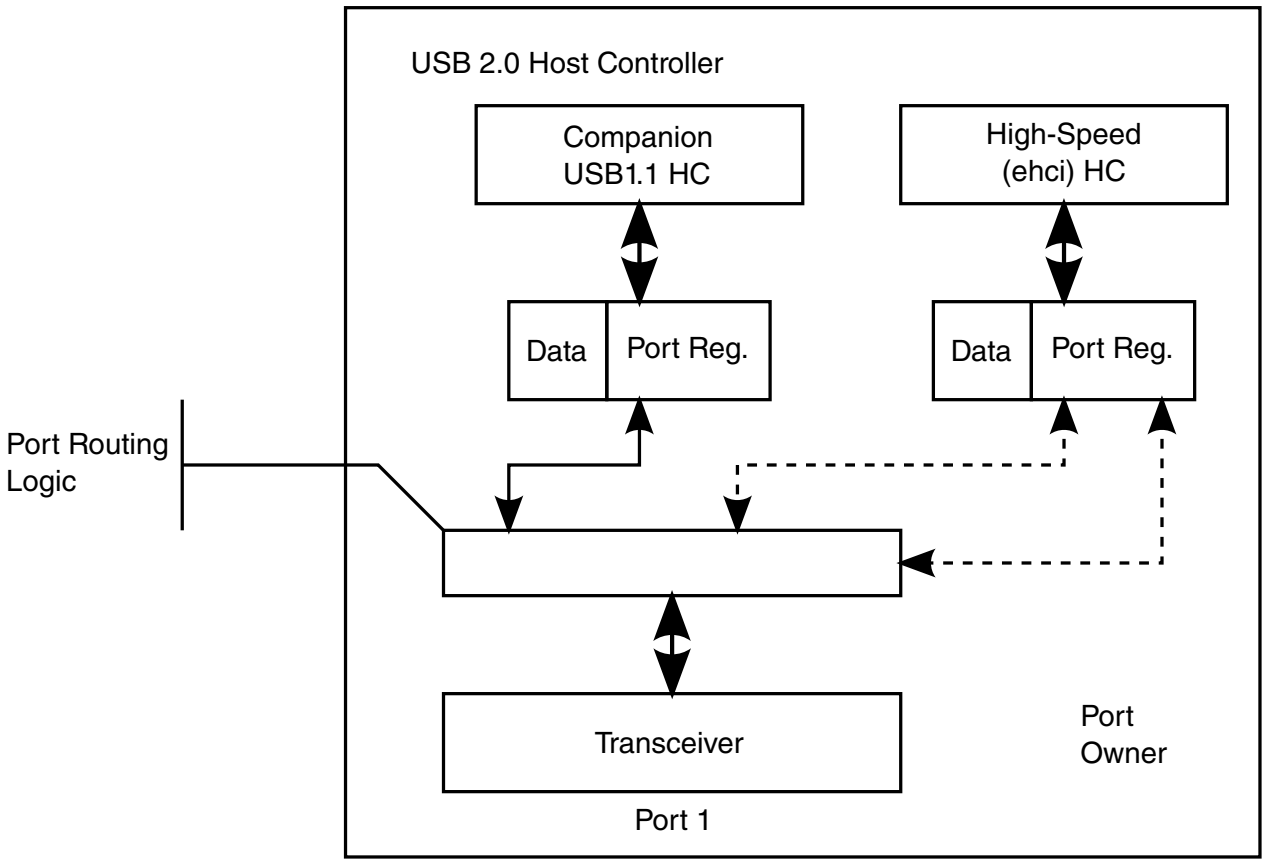


Figure 62-5. Example USB 2.0 Host Controller Port Routing Block Diagram

There exists one transceiver per physical port and each host controller block has its own port status and control registers. The EHCI controller has port status and control registers for every port. Each companion host controller has only the port control and status registers it is required to operate. Either the EHCI host controller or one companion host controller controls each transceiver. Routing logic lies between the transceiver, the port status and control registers.[1]

The port routing logic is controlled from signals originating in the EHCI host controller. The EHCI host controller has a global routing policy control field and per-port ownership control fields. The Configured Flag (CF) bit is the global routing policy control. At power-on or reset, the default routing policy is to the companion controllers (if they exist). If the system does not include a driver for the EHCI host controller and the host controller includes Companion Controllers, then the ports still work in Full- and Low-speed mode (assuming the system includes a driver for the companion controllers). In general, when the EHCI owns the ports, the companion host controllers' port registers do not see a connect indication from the transceiver. Similarly, when a companion host controller owns a port, the EHCI controller's port registers do not see a connect indication

1. The routing logic should not be implemented in the 480 MHz clock domain of the transceiver.

from the transceiver. The details on the rules for the port routing logic are described in the following sections. The USB 2.0 host controller must be implemented as a multi-function PCI device if the implementation includes companion controllers. The companion host controllers' function numbers must be less than the EHCI host controller function number. The EHCI host controller must be a larger function number with respect to the companion host controllers associated with this EHCI host controller. If a PCI device implementation contains only an EHCI controller (that is no companion controllers or other PCI functions), then the EHCI host controller must be function zero, in accordance with the PCI Specification. The N_CC field in the Structural Parameter register (HCSPARAMS) indicates whether the controller implementation includes companion host controllers. When N_CC has a non-zero value there exists companion host controllers. If N_CC has a value of zero, then the host controller implementation does not include companion host controllers. If the host controller root ports are exposed to attachment of full- or low-speed devices, the ports always fails the high-speed chirp during reset and the ports are not enabled. System software can notify the user of the illegal condition. This type of implementation requires a USB 2.0 hub be connected to a root port to provide full and low-speed device connectivity.

System software uses information in the host controller capability registers to determine how the ports are routed to the companion host controllers. See USB_nHCSPARAMS

62.5.3.2.1 Port Routing Control through EHCI Configured (CF) Bit

Each port in the USB 2.0 host controller are routed either to a single companion host controller or to the EHCI host controller.

The port routing logic is controlled by two mechanisms in the EHCI HC: a host controller global flag and per-port control. The Configured Flag (CF) bit, is used to globally set the policy of the routing logic. Each port register has a Port Owner control bit which allows the EHCI Driver to explicitly control the routing of individual ports. Whenever the CF bit transitions from zero to one (this transition is only available under program control) the port routing unconditionally routes all of the port registers to the EHCI HC (all Port Owner bits go to zero). While the CF-bit is one, the EHCI Driver controls individual ports' routing through the Port Owner control bit. Likewise, whenever the CF bit transitions from one to zero (as a result of Aux power application, HCRESET, or software writing zero to CF-bit), the port routing unconditionally routes all of the port registers to the appropriate companion HC. The default value for the EHCI HC's CF bit (after Aux power application or HCRESET) is zero.

The *view* of the port depends on the current owner. A Universal or Open companion host controller will see port register bits consistent with the appropriate specification. Port bit definitions that are required for EHCI host controllers are not visible to companion host controllers.

The following table summarizes the default routing for all the ports, based on the value of the EHCI HC's CF bit.

Table 62-37. Default Port Routing Depending on EHCI HC CF Bit

HS CF Bit	Default Port Ownership	Explanation
0B	Companion HCs	The companion host controllers own the ports and only Full- and Low-speed devices are supported in the system. The exact port assignments are implementation dependent. The ports behave only as Full- and Low-speed ports in this configuration
1B	EHCI HC	The EHCI host controller has default ownership over all of the ports. The routing logic inhibits device connect events from reaching the companion HCs' port status and control registers when the port owner is the EHCI HC.The EHCI HC has access to the additional port status and control bits defined in this specification (see PORTSC1 register). The EHCI HC can temporarily release control of the port to a companion HC by setting the *PortOwner* bit in the PORTSC1 register to one.

62.5.3.2.2 Port Routing Control through PortOwner and Disconnect Event

Manipulating the port routing through the CF-bit is an extreme process and not intended to be used during normal operation.

The normal mode of port ownership transferal is on the granularity of individual ports using the Port Owner bit in the EHCI HC's USB_PORTSC1 register (for hand-offs from EHCI to companion host controllers). Individual port ownership is returned to the EHCI controller when the port registers a device disconnect. When the disconnect is detected, the port routing logic immediately returns the port ownership to the EHCI controller. The companion host controller port register detects the device disconnect and operates normally.

Under normal operating conditions (assuming all HC drivers loaded and operational and the EHCI *CF-bit* is set to one), the typical port enumeration sequence proceeds as illustrated below:

- Initial condition is that EHCI is port owner. A device is connected causing the port to detect a connect, set the port connect change bit and issue a port-change interrupt (if enabled).
- EHCI Driver identifies the port with the new connect change bit asserted and sends a change report to the hub driver. Hub driver issues a GetPortStatus() request and identifies the connect change. It then issues a request to clear the connect change, followed by a request to reset and enable the port.
- When the EHCI Driver receives the request to reset and enable the port, it first checks the value reported by the LineStatus bits in the USB_PORTSC1 register. If they indicate the attached device is a full-speed device (for example, D+ is asserted), then the EHCI Driver sets the PortReset control bit to one (and sets the PortEnable

bit to zero) which begins the reset-process. Software times the duration of the reset, then terminates reset signaling by writing zero to the port reset bit. The reset process is actually complete when software reads zero in the PortReset bit. The EHCI Driver checks the PortOwner bit in the USB_PORTSC1 register. If set to one, the connected device is a high-speed device and EHCI Driver (root hub emulator) issues a change report to the hub driver and the hub driver continues to enumerate the attached device.

- At the time the EHCI Driver receives the port reset and enable request the LineStatus bits might indicate a low-speed device. Additionally, when the port reset process is complete, the PortEnable field may indicate that a full-speed device is attached. In either case the EHCI driver sets the PortOwner bit in the USB_PORTSC1 register to one to release port ownership to a companion host controller.
- When the EHCI Driver sets PortOwner bit to one, the port routing logic makes the connection state of the transceiver available to the companion host controller port register and removes the connection state from the EHCI HC port. The EHCI USB_PORTSC1 register observes and reports a disconnect event through the disconnect change bit. The EHCI Driver detects the connection status change (either by polling or by port change interrupt) and then sends a change report to the hub driver. When the hub driver requests that port-state, the EHCI Driver responds with a reset complete change set to one, a connect change set to one and a connect status set to zero. This information is derived directly from the EHCI port register. This allows the hub driver to assume the device was disconnected during reset. It acknowledges the change bits and wait for the next change event. While the EHCI controller does not own the port, it simply remains in a state where the port reports no device connected. The device-connect evaluation circuitry of the companion HC activates and detects the device, the companion Driver detects the connection and enumerates the port.

When a port is routed to a companion HC, it remains under the control of the companion HC until the device is disconnected from the root port (ignoring for now the scenario where EHCI's CF-bit transitions from1b to 0b). When a disconnect occurs, the disconnect event is detected by both the companion HC port control and the EHCI port ownership control. On the event, the port ownership is returned immediately to the EHCI controller. The companion HC stack detects the disconnect and acknowledges as it would in an ordinary standalone implementation. Subsequent connects is detected by the EHCI port register and the process repeats.

62.5.3.2.3 Example Port Routing State Machine

The following figure illustrates an example of how the port ownership should be managed. The following sections describe the entry conditions to each state.

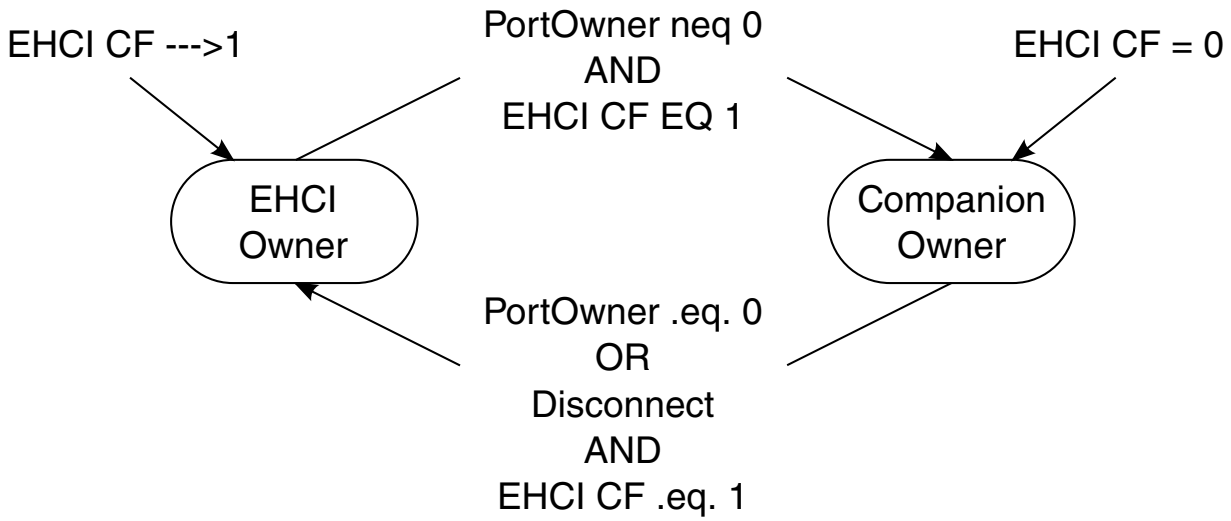


Figure 62-6. Port Owner Handoff State Machine

62.5.3.2.3.1 EHCI HC Owner

Entry to this state occurs when one of the following events occur:

- When the EHCI HC's Configure Flag (CF) bit in the USB_CONFIGFLAG register transitions from zero to one. This signals the fact that the system has a host controller driver for the EHCI HC and that all ports in the USB 2.0 host controller must default route to the EHCI controller.
- When the port is owned by a companion HC and the device is disconnected from the port. The EHCI port routing control logic is notified of the disconnect, and returns port routing to the EHCI controller. The connection state of the companion HC goes immediately to the disconnected state (with appropriate side effect to connect change, enable and enable change). The companion HC driver acknowledges the disconnect by setting the connect status change bit to zero. This allows the companion HC's driver to interact with the port completely through the disconnect process.
- When system software writes zero to the PortOwner bit in the USB_PORTSC1 register. This allows software to take ownership of a port from a companion host controller. When this occurs, the routing logic to the companion HC effectively signals a disconnect to the companion HC's port status and control register.

62.5.3.2.3.2 Companion HC Owner

Entry to this state occurs whenever one of the following events occur:

- When the PortOwner field transitions from zero to one.
- When the HS-mode HC's Configure Flag (CF) is equal to zero.

On entry to this state, the routing logic allows the companion HC port register to detect a device connect. Normal port enumeration proceeds.

62.5.3.2.4 Port Power

The Port Power Control (PPC) bit in the USB_HCSPARAMS register indicates whether the USB 2.0 host controller has port power control (see HCSPARAMS register). When this bit is zero, then the host controller does not support software control of port power switches. When in this configuration, the port power is always available and the companion host controllers must implement functionality consistent with port power always on. When the *PPC* bit is one, then the host controller implementation includes port power switches. Each available switch has an output enable, which is referred to in this discussion as PortPowerOutputEnable (PPE). PPE is controlled based on the state of the combination bits PPC bit, EHCI Configured (CF)-bit and individual Port Power (PP) bits.

The following table describes the summary behavioral model.

Table 62-38. Port Power Enable Control Rules

CF	CHC[1] (PP)	EHC[2] (PP)	Owner	PPE[3]	Description
0	0	X	CHC	0	When the EHCI controller is not configured, the port is owned by the companion host controller. When the companion HC's port power select is off, then the port power is off.
0	1	X	CHC	1	Similar to previous entry. When the companion HC's port power select is on, then the port power is on.
1	0	0	CHC	0	Port owner has port power turned off, the power to port is off.
1	0	0	EHC	0	Port owner has port power turned off, the power to port is off.
1	0	1	EHC	1	Port owner has port power on, so power to port is on.
1	0	1	CHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	EHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	EHC	1	Port owner has port power on, so power to port is on.

1. CHC (Companion Host Controller).
2. EHC (EHCI Host Controller).
3. PPE (Port Power Enable). This bit actually turns on the port power switch (if one exists).

62.5.3.2.5 Port Reporting Over-Current

Host controllers are by definition power providers on USB. Whether the ports are considered high- or low-powered is a platform implementation issue. Each EHCI USB_PORTSC1 register has an over-current status and over-current change bit.

The functionality of these bits are specified in the USB Specification Revision 2.0.

The over current detection and limiting logic usually resides outside the host controller logic. This logic may be associated with one or more ports. When this logic detects an over-current condition it is made available to both the companion and EHCI ports. The effect of an over-current status on a companion host controller port is beyond the scope of this document.

The over-current condition effects the following bits in the USB_PORTSC1 register on the EHCI port:

- Over-current Active bits are set to one. When the over-current condition goes away, the Over-current Active bit transitions from one to zero.
- Over-current Change bits are set to one. On every transition of the Over-current Active bit the host controller sets the Over-current Change bit to one. Software sets the Over-current Change bit to zero by writing one to this bit.
- Port Enabled/Disabled bit is set to zero. When this change bit gets set to one, then the Port Change Detect bit in the USB_USBSTS register is set to one.
- Port Power (PP) bits may optionally be set to zero. There is no requirement in USB that a power provider shut off power in an over current condition. It is sufficient to limit the current and leave power applied. When the Over-current Change bit transitions from zero to one, the host controller also sets the Port Change Detect bit in the USB_USBSTS register to one. In addition, if the Port Change Interrupt Enable bit in the USB_USBINTR register is one, then the host controller issues an interrupt to the system. Refer to Table 62-39 for summary behavior for over-current detection when the host controller is halted (suspended from a device component point of view).

62.5.3.3 Suspend/Resume-Host Operational Model

The EHCI host controller provides an equivalent suspend and resume model as that defined for individual ports in a USB 2.0 Hub.

Control mechanisms are provided to allow system software to suspend and resume individual ports. The mechanisms allow the individual ports to be resumed completely through software initiation. Other control mechanisms are provided to parameterize the host controller's response (or sensitivity) to external resume events. In this discussion, host-initiated, or software initiated resumes are called Resume Events/Actions. Bus-initiated resume events are called wake-up events. The classes of wake-up events are:

- Remote-wake-up enabled device asserts resume signaling. In similar kind to USB 2.0 Hubs, EHCI controllers must always respond to explicit device resume signaling and wake-up the system (if necessary).
- Port connect and disconnect and over-current events. Sensitivity to these events can be turned on or off by using the per-port control bits in the USB_PORTSC1 registers.

Selective suspend is a feature supported by every USB_PORTSC1 register. It is used to place specific ports into a suspend mode. This feature is used as a functional component for implementing the appropriate power management policy implemented in a particular operating system. When system software intends to suspend the entire bus, it should selectively suspend all enabled ports, then shut off the host controller by setting the Run/Stop bit in the USB_USBCMD register to zero. The EHCI sub-block can then be placed into a lower device state through the PCI power management interface (see Appendix A, Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com).

When a wake event occurs, the system resumes operation and system software eventually set the Run/Stop bit to one and resume the suspended ports. Software must not set the Run/Stop bit to one until it is confirmed that the clock to the host controller is stable. This is usually confirmed in a system implementation in that all of the clocks in the system are stable before the Arm platform is restarted. So, by definition, if software is running, clocks in the system are stable and the Run/Stop bit in the USB_USBCMD register can be set to one. Minimum system software delays are also defined in the PCI Power Management Specification. Refer to PCI Power Management Specification for more information.

62.5.3.3.1 Port Suspend/Resume

System software places individual ports into suspend mode by writing one into the appropriate USB_PORTSC1 Suspend bit. Software must only set the Suspend bit when the port is in the enabled state (Port Enabled bit is one) and the EHCI is the port owner (PortOwner bit is zero).

The host controller may evaluate the Suspend bit immediately or wait until a micro-frame or frame boundary occurs. If evaluated immediately, the port is not suspended until the current transaction (if one is executing) completes. Therefore, there may be several micro-frames of activity on the port until the host controller evaluates the Suspend bit. The host controller must evaluate the Suspend bit at least every frame boundary.

System software can initiate a resume on a selectively suspended port by writing one to the Force Port Resume bit. Software should not attempt to resume a port unless the port reports that it is in the suspended state (see USB_nPORTSC1). If system software sets Force Port Resume bit to one when the port is not in the suspended state, the resulting behavior is undefined. In order to assure proper USB device operation, software must wait for at least 10 ms after a port indicates that it is suspended (Suspend bit is one) before initiating a port resume through the Force Port Resume bit. When Force Port Resume bit is one, the host controller sends resume signaling down the port. System software times the duration of the resume (nominally 20 ms) then sets the Force Port Resume bit to zero. When the host controller receives the write to transition Force Port Resume to zero, it completes the resume sequence as defined in the USB specification, and sets both the Force Port Resume and Suspend bits to zero. Software-initiated port resumes do not affect the Port Change Detect bit in the USB_USBSTS register nor do they cause an interrupt if the Port Change Interrupt Enable bit in the USB_USBINTR register is one. An external USB event may also initiate a resume. The wake events are defined above. When a wake event occurs on a suspended port, the resume signaling is detected by the port and the resume is reflected downstream within 100 µsec. The port's Force Port Resume bit is set to one and the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit in the USB_USBINTR register is one the host controller issues a hardware interrupt.

System software observes the resume event on the port, delays a port resume time (nominally 20 ms), then terminates the resume sequence by writing zero to the Force Port Resume bit in the port. The host controller receives the write of zero to Force Port Resume, terminates the resume sequence and sets Force Port Resume and Suspend port bits to zero. Software can determine that the port is enabled (not suspended) by sampling the USB_PORTSC1 register and observing that the Suspend and Force Port Resume bits are zero. Software must ensure that the host controller is running (that is HCHalted bit in the USB_USBSTS register is zero), before terminating a resume by writing zero to a port's Force Port Resume bit. If HCHalted is one when Force Port Resume is set to zero, then SOFs do not occur down the enabled port and the device returns to suspend mode in a maximum of 10 msec.

The table below summarizes the wake-up events. Whenever a resume event is detected, the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit is one in the USB_USBINTR register, the host controller generates an interrupt on the resume event. Software acknowledges the resume event interrupt by clearing the Port Change Detect status bit in the USB_USBSTS register.

Table 62-39. Behavior During Wake-up Events

Port Status and Signaling Type	Signaled Port Response	Device State	
		D0	Not D0
Port disabled, resume K-State received	No Effect	N/A	N/A
Port suspended, resume K-State received	Resume reflected downstream on signaled port. Force Port Resume status bit in USB_PORTSC1 register is set to one. Port Change Detect bit in USB_USBSTS register set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is one. A disconnect is detected.	Depending in the initial port state, theUSB_ PORTSC1 Connected Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is zero. A disconnect is detected.	Depending on the initial port state, the USB_PORTSC1 Connect and Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is not connected and the port's WKCNNT_E bit is one. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is not connected and the port's WKCNNT_E bit is zero. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is connected and the port's WKOC_E bit is one. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one	[1], [2]	[2]
Port is connected and the port's WKOC_E bit is zero. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

[1] Hardware interrupt issued if Port Change Interrupt Enable bit in the USB_USBINTR register is one.

[2] PME# asserted if enabled (Note: PME Status must always be set to one).

[3] PME# not asserted.

62.5.3.4 Schedule Traversal Rules

The host controller executes transactions for devices using a simple, shared-memory schedule.

The schedule is comprised of a few data structures, organized into two distinct lists. The data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic and hardware / software complexity.

System software maintains two schedules for the host controller: a periodic schedule and an asynchronous schedule. The root of the periodic schedule is the USB_PERIODICLISTBASE register (see USB_nPERIODICLISTBASE) DEVICEADDR. The USB_PERIODICLISTBASE register is the physical memory base address of the periodic frame list. The periodic frame list is an array of physical memory pointers. The objects referenced from the frame list must be valid schedule data structures as defined in Host Data Structures. In each micro-frame, if the periodic schedule is enabled (see Periodic scheduling threshold) then the host controller must execute from the periodic schedule before executing from the asynchronous schedule. It only executes from the asynchronous schedule after it encounters the end of the periodic schedule. The host controller traverses the periodic schedule by constructing an array offset reference from the USB_PERIODICLISTBASE and the USB_FRINDEX registers (see the following figure). It fetches the element and begins traversing the graph of linked schedule data structures.

The end of the periodic schedule is identified by a next link pointer of a schedule data structure having its T-bit set to one. When the host controller encounters a T-Bit set to one during a horizontal traversal of the periodic list, it interprets this as an End-Of-Periodic-List mark. This causes the host controller to cease working on the periodic schedule and transitions immediately to traversing the asynchronous schedule. After the transition, the host controller executes from the asynchronous schedule until the end of the micro-frame.

The following figure illustrates the derivation of pointer into frame list array.

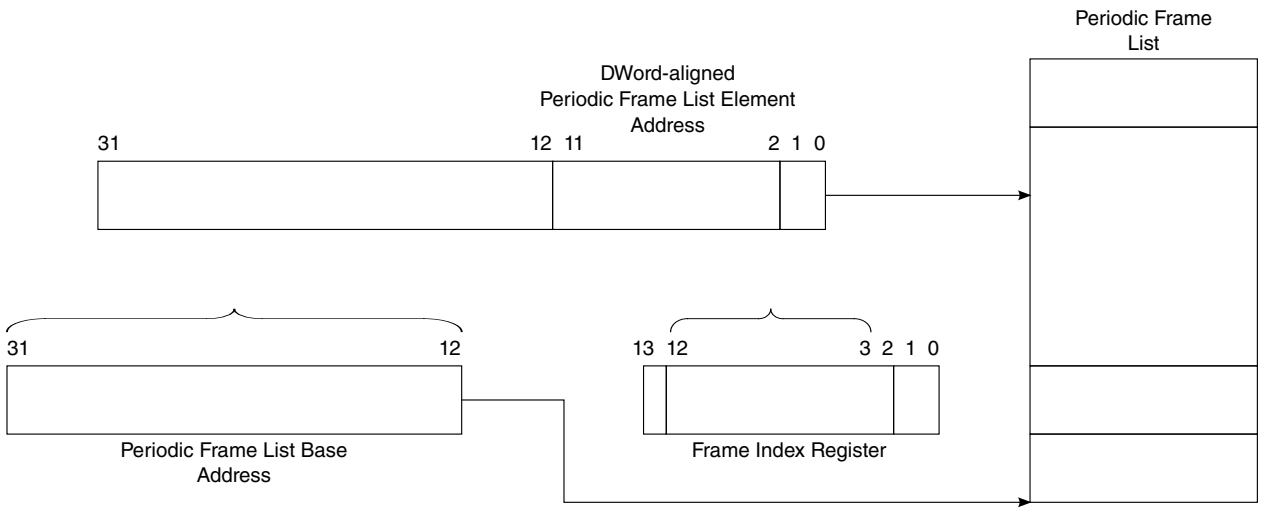


Figure 62-7. Derivation of Pointer into Frame List Array

When the host controller determines that it is the time to execute from the asynchronous list, it uses the operational register USB_ASYNCLISTADDR to access the asynchronous schedule, see the figure below.

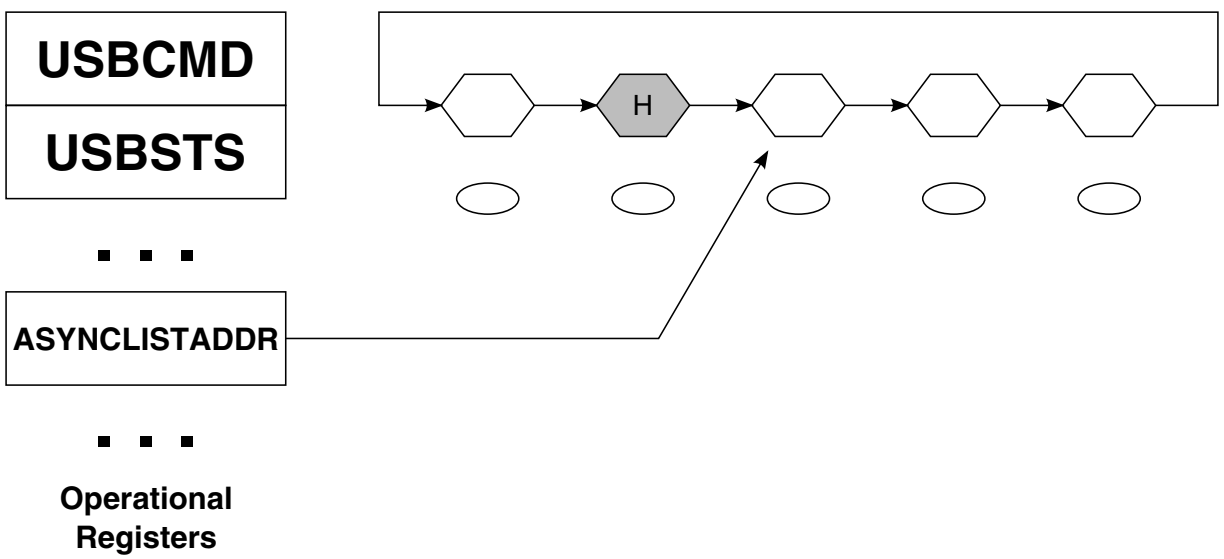


Figure 62-8. General Format of Asynchronous Schedule List

The USB_ASYNCLISTADDR register contains a physical memory pointer to the next queue head. When the host controller makes a transition to executing the asynchronous schedule, it begins by reading the queue head referenced by the USB_ASYNCLISTADDR register. Software must set queue head horizontal pointer T-bits to zero for queue heads in the asynchronous schedule. See Asynchronous Schedule for complete operational details.

62.5.3.4.1 Example - Preserving Micro-Frame Integrity

One of the requirements of a USB host controller is to maintain Frame Integrity. This means that the HC must preserve the micro-frame boundaries.

For example, SOF packets must be generated on time (within the specified allowable jitter), and High-speed EOF1,2 thresholds must be enforced. The end of micro-frame timing points EOF1 and EOF2 are clearly defined in the USB Specification Revision 2.0. One implication of this responsibility is that the HC must ensure that it does not start transactions that do not complete before the end of the micro-frame. More precisely, no transactions should be started by the host controller, which do not complete in their entirety before the EOF1 point. In order to enforce this rule, the host controller must check each transaction before it starts to ensure that it completes before the end of the micro-frame.

So, what exactly needs to be involved in this check? Fundamentally, the transaction data payload, plus bit stuffing, plus transaction overhead must be taken into consideration. It is possible to be extremely accurate on how much time the next transaction takes. Take OUTs for an example. The host controller must fetch all of the OUT data from memory in order to send it onto the USB bus. A host controller implementation could pre-fetch all of the OUT data, and pre-compute the actual number of bits in the token and data packets. In addition, the system knows the depth of the target endpoint, so it could closely estimate turnaround time for handshake. In addition, the host controller knows the size of a handshake packet. Pre-computing effects of bit stuffing and summing up the other overhead numbers can allow the host controller to know exactly whether there is enough bus time, before EOF1 to complete the OUT transaction. To accomplish this particular approach takes an inordinate amount of time and hardware complexity.

The alternative is to make a reasonable guess whether the next transaction can be started. An example approximation algorithm is described below. This example algorithm relies on the EHCI policy that periodic transactions are scheduled first in the micro-frame. It is a reasonable assumption that software never over-commits the micro-frame to periodic transactions greater than the specification allowable 80%. In the available remaining 20% bandwidth, the host controller has some ability (in this example) to decide whether or not to execute a transaction. The result of this algorithm is that sometimes, under some circumstances a transaction is not executed that could have been executed. However, under all circumstances, a transaction is never started unless there is enough time in the frame to complete the transaction.

62.5.3.4.1.1 Transaction Fit - A Best-Fit Approximation Algorithm

A curve is calculated which represents the latest start time for every packet size, at which software schedules the start of a periodic transaction.

This curve is the 80% bandwidth curve. Another curve is calculated which is the absolute, latest permitted start time for every packet size. This curve represents the absolute latest time, that a transaction of each packet size can be started and completed, in the micro-frame. A plot of these two curves are illustrated in Figure 62-9. The plot Y-axis represents the number of byte-times left in a frame.

The space between the 80% and the Last Start plots is bandwidth reclamation area. In this algorithm the host controller may skip transactions during this time if it is prudent.

The Best-Fit Approximation method plots a function ($f(x)$) between the 80% and Last Start curves. The function $f(x)$ adds a constant to every transaction's maximum packet size and the result compared with the number of bytes left in the frame. The constant represents an approximation of the effects of bit stuffing and protocol overhead. The host controller starts transactions whose results land above the function curve. The host controller will not start transactions whose results land below the function curve.

The following figure illustrates the Best-Fit Approximation.

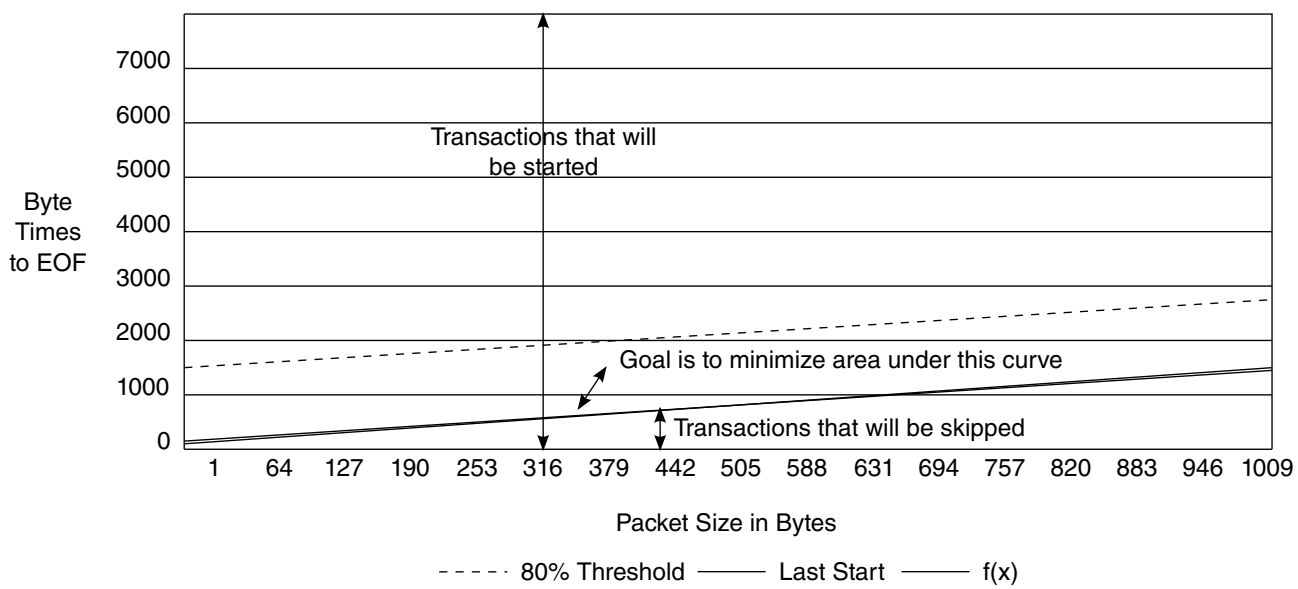


Figure 62-9. Best Fit Approximation

The LastStart line was calculated in this example to assume the absolute worst-case bus overhead per transaction. The particular transaction used is a start-split, zero-length OUT transaction with a handshake. Summaries of the component parts are listed in the table below. The component times were derived from the protocol timings defined in the USB Specification Revision 2.0.

Table 62-40. Example Worse-case Transaction Timing Components

Component	Bit time	Byte Time	Explanation

Table continues on the next page...

Table 62-40. Example Worse-case Transaction Timing Components (continued)

Split Token	76	9.5	Split token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Number of bit times required between consecutive host packets.
Token	67	8.375	Token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Token as defined in USB core specification. Includessync, token, eop, and so on.
Data Packet (0 data bytes)	66.7	8.34	Zero-length data packet. Includes sync, PID, crc16, eop, and so on.
Turnaround time	721	90.125	Time for packet initiator (Host) to see the beginning of a response to a transmitted packet.
Handshake packet	48	6	Handshake packet as defined in USB core specification. Includes sync, PID, eop, and so on.
		144	Total

The exact details of the function (f(x)) are up to the particular implementation. However, it should be obvious that the goal is to minimize the area under the curve between the approximation function and the Last Start curve, without dipping below the LastStart line, while at the same time keeping the check as simple as possible for hardware implementation. The f(x) in Figure 62-9 was constructed using the following pseudo-code test on each transaction size data point. This algorithm assumes that the host controller keeps track of the remaining bits in the frame.

```
Alorithm CheckTransactionWillFit (MaximumPacketSize, HC_BytesLeftInFrame)
Begin
Local Temp = MaximumPacketSize + 192
Local rvalue = TRUE
If MaximumPacketSize >= 128 then
     Temp += 128
End If
If Temp > HC_BytesLeftInFrame then
     Rvalue = FALSE
End If
Return rvalue
End
```

This algorithm takes two inputs, the current maximum packet size of the transaction and the hardware counter of the number of bytes left in the current micro-frame. It unconditionally adds a simple constant of 192 to the maximum packet size to account for a first-order effect of transaction overhead and bit stuffing. If the transaction size is greater than or equal to 128 bytes, then an additional constant of 128 is added to the running sum to account for the additional worst-case bit stuffing of payloads larger than 128. An inflection point was inserted at 128 because the $f(x)$ plot was getting close to the LastStart line.

62.5.3.5 Periodic Schedule Frame Boundaries vs Bus Frame Boundaries

The USB Specification Revision 2.0 requires that the frame boundaries (SOF frame number changes) of the high-speed bus and the full- and low-speed bus(s) below USB 2.0 Hubs be strictly aligned.

Super-imposed on this requirement is that USB 2.0 Hubs manage full- and low-speed transactions through a micro-frame pipeline (see start- (SS) and complete- (CS) splits illustrated in the following figure). A simple, direct projection of the frame boundary model into the host controller interface schedule architecture creates tension (complexity for both hardware and software) between the frame boundaries and the scheduling mechanisms required to service the full- and low-speed transaction translator periodic pipelines.

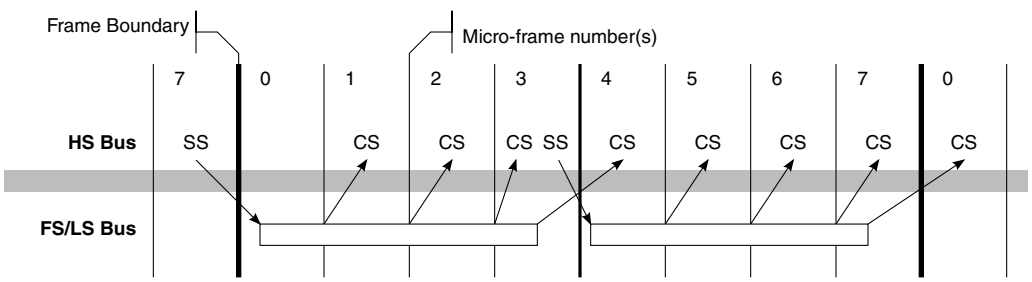


Figure 62-10. Frame Boundary Relationship between HS bus and FS/LS Bus

The simple projection, as the above figure illustrates, introduces frame-boundary wrap conditions for scheduling on both the beginning and end of a frame. In order to reduce the complexity for hardware and software, the host controller is required to implement one micro-frame phase shift for its view of frame boundaries. The phase shift eliminates the beginning of frame and frame-wrap scheduling boundary conditions.

The implementation of this phase shift requires that the host controller use one register value for accessing the periodic frame list and another value for the frame number value included in the SOF token. These two values are separate, but tightly coupled. The periodic frame list is accessed through the Frame List Index Register (USB_FRINDEX) documented in FRINDEX and initially illustrated in Schedule Traversal Rules. Bits FRINDEX[2:0], represent the micro-frame number. The SOF value is coupled to the value of FRINDEX[13:3]. Both FRINDEX[13:3] and the SOF value are increment based on FRINDEX[2:0]. It is required that the SOF value be delayed from the FRINDEX value by one micro-frame. The one micro-frame delay yields host controller periodic schedule and bus frame boundary relationship as illustrated in the following figure. This adjustment allows software to trivially schedule the periodic start and complete-split

transactions for full-and low-speed periodic endpoints, using the natural alignment of the periodic schedule interface. The reasons for selecting this phase-shift are beyond the scope of this specification.

The following figure illustrates how periodic schedule data structures relate to schedule frame boundaries and bus frame boundaries. To aid the presentation, two terms are defined: The host controller's view of the 1 msec boundaries is called H-Frames. The high-speed bus's view of the 1 msec boundaries is called B-Frames.

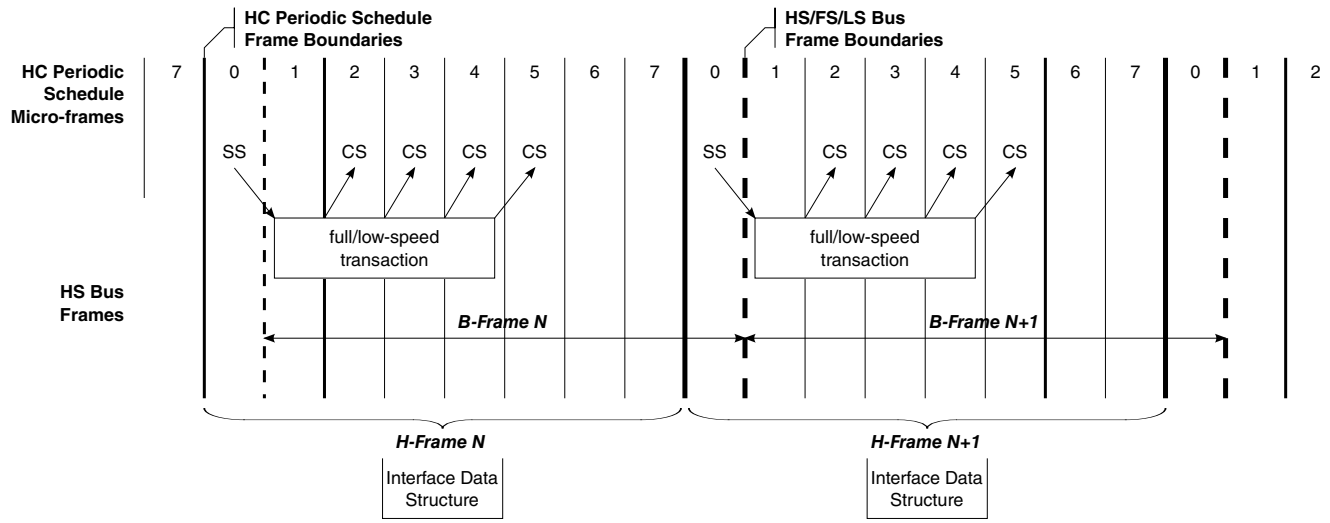


Figure 62-11. Relationship of Periodic Schedule Frame Boundaries to Bus Frame Boundaries

H-Frame boundaries for the host controller correspond to increments of FRINDEX[13:3]. Micro-frame numbers for the H-Frame are tracked by FRINDEX[2:0]. B-Frame boundaries are visible on the high-speed bus through changes in the SOF token's frame number. Micro-frame numbers on the high-speed bus are only derived from the SOF token's frame number (that is the high-speed bus sees eight SOFs with the same frame number value). H-Frames and B-Frames have the fixed relationship (that is B-Frames lag H-Frames by one micro-frame time) illustrated in the figure above. The host controller's periodic schedule is naturally aligned to H-Frames. Software schedules transactions for full- and low-speed periodic endpoints relative the H-Frames. The result is these transactions execute on the high-speed bus at exactly the right time for the USB 2.0 Hub periodic pipeline. As described in FRINDEX, the SOF Value can be implemented as a shadow register (in this example, called SOFV), which lags the FRINDEX register bits [13:3] by one micro-frame count. This lag behavior can be accomplished by incrementing FRINDEX[13:3] based on carry-out on the 7 to 0 increment of FRINDEX[2:0] and incrementing SOFV based on the transition of 0 to 1 of FRINDEX[2:0].

Software is allowed to write to FRINDEX. FRINDEX provides the requirements that software should adhere when writing a new value in FRINDEX.

The table below illustrates the required relationship between the value of FRINDEX and the value of SOFV.

Table 62-41. Operation of FRINDEX and SOFV (SOF Value Register)

Current			Next		
FRINDEX[F]	SOFV	FRINDEX[mF]	FRINDEX[F]	SOFV	FRINDEX[mF]
N	N	111b	N+1	N	000b
N+1	N	000b	N+1	N+1	001b
N+1	N+1	001b	N+1	N+1	010b
N+1	N+1	010b	N+1	N+1	011b
N+1	N+1	011b	N+1	N+1	100b
N+1	N+1	100b	N+1	N+1	101b
N+1	N+1	101b	N+1	N+1	110b
N+1	N+1	110b	N+1	N+1	111b

NOTE

Where [F] = [13:3]; [μF] = [2:0]

62.5.3.6 Periodic Schedule

The periodic schedule traversal is enabled or disabled through the Periodic Schedule Enable bit in the USB_USBCMD register. If the Periodic Schedule Enable bit is set to zero, then the host controller simply does not try to access the periodic frame list through the USB_PERIODICLISTBASE register. Likewise, when the Periodic Schedule Enable bit is one, then the host controller does use the USB_PERIODICLISTBASE register to traverse the periodic schedule.

The host controller will not react to modifications to the Periodic Schedule Enable immediately. In order to eliminate conflicts with split transactions, the host controller evaluates the Periodic Schedule Enable bit only when FRINDEX[2:0] is zero. System software must not disable the periodic schedule if the schedule contains an active split transaction work item that spans the 000b micro-frame. These work items must be removed from the schedule before the Periodic Schedule Enable bit is written to zero.

The Periodic Schedule Status bit in the USB_USBSTS register indicates status of the periodic schedule. System software enables (or disables) the periodic schedule by writing one (or zero) to the Periodic Schedule Enable bit in the USB_USBCMD register. Software then can poll the Periodic Schedule Status bit to determine when the periodic schedule has made the desired transition. Software must not modify the Periodic Schedule Enable bit unless the value of the Periodic Schedule Enable bit equals that of the Periodic Schedule Status bit.

The periodic schedule is used to manage all isochronous and interrupt transfer streams. The base of the periodic schedule is the periodic frame list. Software links schedule data structures to the periodic frame list to produce a graph of scheduled data structures. The graph represents an appropriate sequence of transactions.

The following figure illustrates isochronous transfers (using iTDs and siTDs) with a period of one are linked directly to the periodic frame list. Interrupt transfers (are managed with queue heads) and isochronous streams with periods other than one are linked following the period-one iTD/siTDs. Interrupt queue heads are linked into the frame list ordered by poll rate. Longer poll rates are linked first (for example, closest to the periodic frame list), followed by shorter poll rates, with queue heads with a poll rate of one, on the very end.

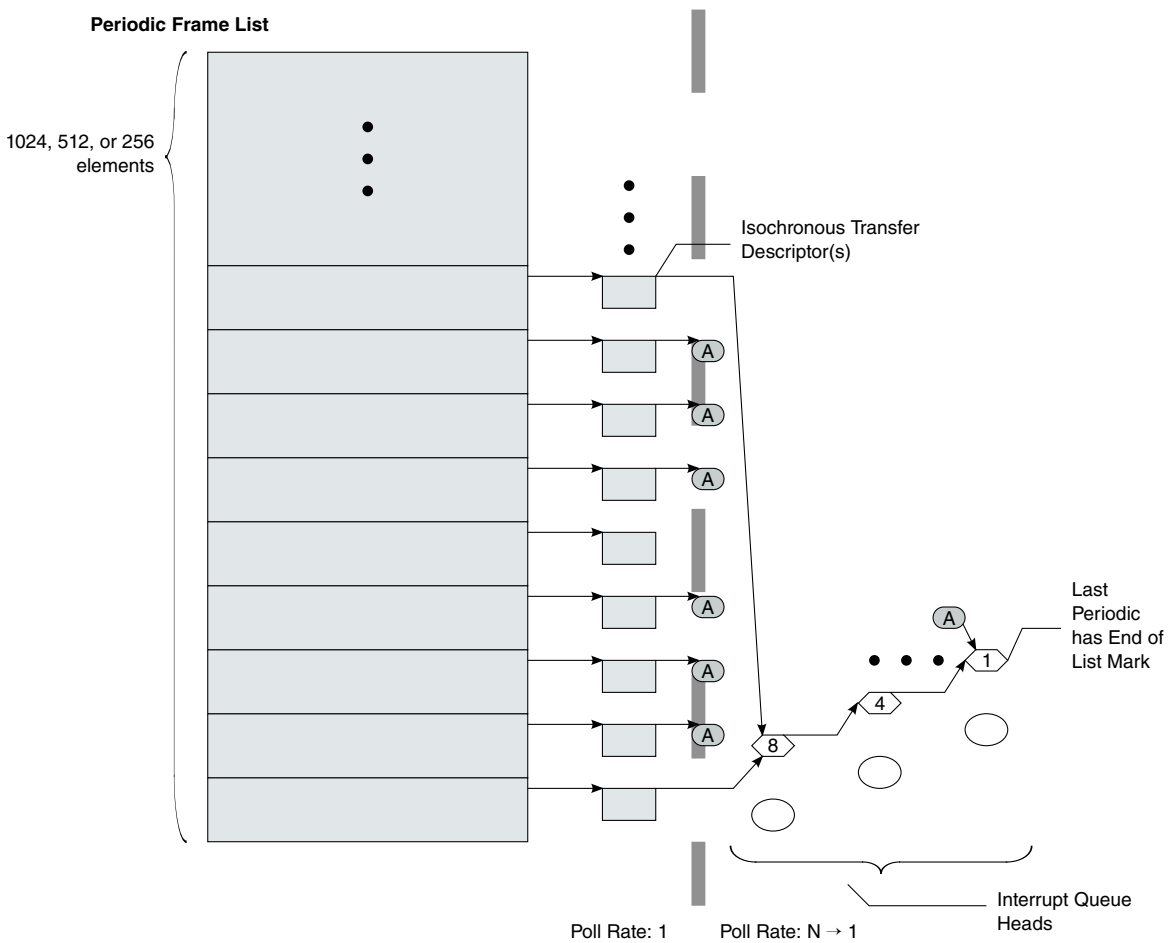


Figure 62-12. Example Periodic Schedule

62.5.3.7 Managing Isochronous Transfers Using iTDs

The structure of an iTD is presented in Isochronous (High-Speed) Transfer Descriptor (iTD). The four distinct sections to an iTD:

- The first field is the Next Link Pointer. This field is for schedule linkage purposes only.
- Transaction description array. This area is an eight-element array. Each element represents control and status information for one micro-frame's worth of transactions for a single high-speed isochronous endpoint.
- The buffer page pointer array is a 7-element array of physical memory pointers to data buffers. These are 4 K aligned pointers to physical memory.
- Endpoint capabilities. This area utilizes the unused low-order 12 bits of the buffer page pointer array. The fields in this area are used across all transactions executed for this iTD, including endpoint addressing, transfer direction, maximum packet size and high-bandwidth multiplier.

62.5.3.7.1 Host Controller Operational Model for iTDs

The host controller uses FRINDEX register bits [12:3] to index into the periodic frame list. This means that the host controller visits each frame list element eight consecutive times before incrementing to the next periodic frame list element. Each iTD contains eight transaction descriptions, which map directly to FRINDEX register bits [2:0]. Therefore, each transaction descriptor corresponds to one micro-frame. Each iTD can span 8 micro-frames worth of transactions.

When the host controller fetches an iTD, it uses FRINDEX register bits [2:0] to index into the transaction description array.

If the active bit in the Status field of the indexed transaction description is set to zero, the host controller ignores the iTD and follows the Next pointer to the next schedule data structure.

When the indexed active bit is one, the host controller continues to parse the iTD. It stores the indexed transaction description and the general endpoint information (device address, endpoint number, maximum packet size, and so on.). It also uses the Page Select (PG) field to index the buffer pointer array, storing the selected buffer pointer and the next sequential buffer pointer. For example, if PG field is 0, then the host controller stores Page 0 and Page 1.

The host controller constructs a physical data buffer address by concatenating the current buffer pointer (as selected using the current transaction description's PG field) and the transaction description's Transaction Offset field. The host controller uses the endpoint addressing information and I/O-bit to execute a transaction to the appropriate endpoint.

When the transaction is complete, the host controller clears the active bit and writes back any additional status information to the Status field in the currently selected transaction description.

The data buffer associated with the iTD must be virtually contiguous memory. Seven page pointers are provided to support eight high-bandwidth transactions regardless of the starting packet's offset alignment into the first page. A starting buffer pointer (physical memory address) is constructed by concatenating the page pointer (for example, page 0 pointer) selected by the active transaction descriptions' PG (for example, value: 00B) field with the transaction offset field. As the transaction moves data, the host controller must detect when an increment of the current buffer pointer crosses a page boundary. When this occurs the host controller simply replaces the current buffer pointer's page portion with the next page pointer (for example, page 1 pointer) and continues to move data. The size of each bus transaction is determined by the value in the Maximum Packet Size field. An iTD supports high-bandwidth pipes through the Mult (multiplier) field. When the Mult field is 1, 2, or 3, the host controller executes the specified number of Maximum Packet sized bus transactions for the endpoint in the current micro-frame. In other words, the Mult field represents a transaction count for the endpoint in the current micro-frame. If the Mult field is zero, the operation of the host controller is undefined. The transfer description is used to service all transactions indicated by the Mult field.

For OUT transfers, the value of the Transaction X Length field represents the total bytes to be sent during the micro-frame. The Mult field must be set by software to be consistent with Transaction X Length and Maximum Packet Sixe. The host controller sends the bytes in Maximum Packet Size'd portions. After each transaction, the host controller decrements its local copy of Transaction X Length by Maximum Packet Size. The number of bytes the host controller sends is always Maximum Packet Size or Transaction X Length, whichever is less. The host controller advances the transfer state in the transfer description, updates the appropriate record in the iTD and moves to the next schedule data structure. The maximum sized transaction supported is 3 x 1024 bytes.

For IN transfers, the host controller issues Mult transactions. It is assumed that software has properly initialized the iTD to accommodate all of the possible data. During each IN transaction, the host controller must use Maximum Packet Size to detect packet babble errors. The host controller keeps the sum of bytes received in the Transaction X Length field. After all transactions for the endpoint have completed for the micro-frame, Transaction X Length contains the total bytes received. If the final value of Transaction X Length is less than the value of Maximum Packet Size, then less data than was allowed for was received from the associated endpoint. This short packet condition does not set the USBINT bit in the USB_USBSTS register to one. The host controller will not detect this condition. If the device sends more than Transaction X Length or Maximum Packet Size bytes (whichever is less), then the host controller sets the Babble Detected bit to one and set the Active bit to zero. Note, that the host controller is not required to update the

iTD field Transaction X Length in this error scenario. If the Mult field is greater than one, then the host controller automatically executes the value of Mult transactions. The host controller will not execute all Mult transactions if:

- The endpoint is an OUT and Transaction X Length goes to zero before all the Mult transactions have executed (ran out of data), or
- The endpoint is an IN and the endpoint delivers a short packet, or an error occurs on a transaction before Mult transactions have been executed. The end of micro-frame may occur before all of the transaction opportunities have been executed. When this happens, the transfer state of the transfer description is advanced to reflect the progress that was made, the result written back to the iTD and the host controller proceeds to processing the next micro-frame. Refer to Appendix D for a table summary of the host controller required behavior for all the high-bandwidth transaction cases.

62.5.3.7.2 Software Operational Model for iTDs

A client buffer request to an isochronous endpoint may span 1 to N micro-frames. When N is larger than one, system software may have to use multiple iTDs to read or write data with the buffer (if N is larger than eight, it must use more than one iTD).

The following figure illustrates the simple model of how a client buffer is mapped by system software to the periodic schedule (that is the periodic frame list and a set of iTDs). On the right is the client description of its request. The description includes a buffer base address plus additional annotations to identify which portions of the buffer should be used with each bus transaction. In the middle is the iTD data structures used by the system software to service the client request. Each iTD can be initialized to service up to 24 transactions, organized into eight groups of up to three transactions each. Each group maps to one micro-frame's worth of transactions. The EHCI controller does not provide per-transaction results within a micro-frame. It treats the per-micro-frame transactions as a single logical transfer. On the left is the host controller's frame list. System software establishes references from the appropriate locations in the frame list to each of the appropriate iTDs. If the buffer is large, then system software can use a small set of iTDs to service the entire buffer. System software can activate the transaction description records (contained in each iTD) in any pattern required for the particular data stream.

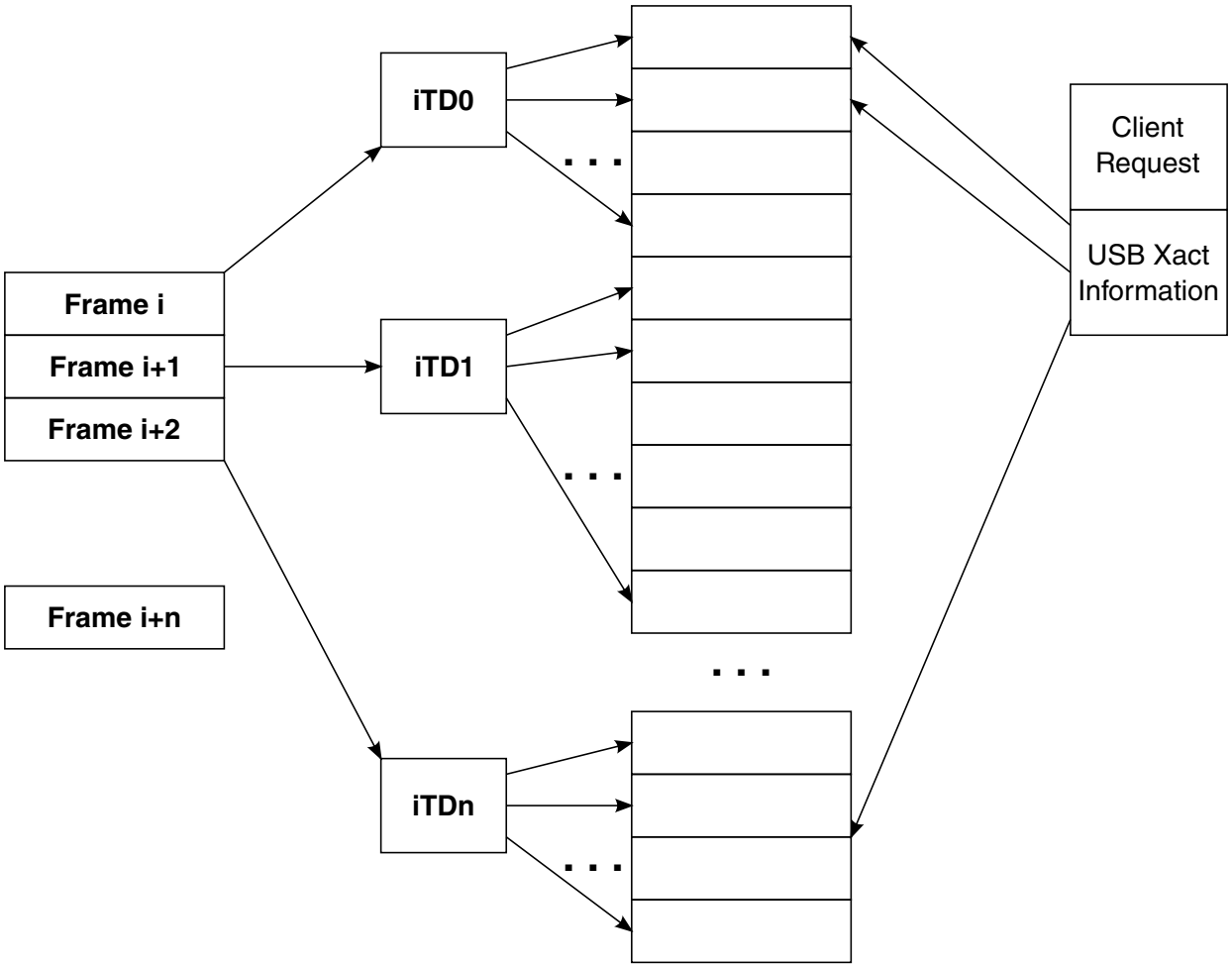


Figure 62-13. Example Association of iTDs to Client Request Buffer

As noted above, the client request includes a pointer to the base of the buffer and offsets into the buffer to annotate which buffer sections are to be used on each bus transaction that occurs on this endpoint. System software must initialize each transaction description in an iTD to ensure it uses the correct portion of the client buffer. For example, for each transaction description, the PG field is set to index the correct physical buffer page pointer and the Transaction Offset field is set relative to the correct buffer pointer page (for example, the same one referenced by the PG field). When the host controller executes a transaction it selects a transaction description record based on FRINDEX[2:0]. It then uses the current Page Buffer Pointer (as selected by the PG field) and concatenates to the transaction offset field. The result is a starting buffer address for the transaction. As the host controller moves data for the transaction, it must watch for a page wrap condition and properly advance to the next available Page Buffer Pointer. System software must not use the Page 6 buffer pointer in a transaction description where the length of the transfer wraps a page boundary. Doing so yields undefined behavior. The host controller hardware is not required to 'alias' the page selector to Page zero. USB 2.0 isochronous

endpoints can specify a period greater than one. Software can achieve the appropriate scheduling by linking iTDs into the appropriate frames (relative to the frame list) and by setting appropriate transaction description elements active bits to one.

62.5.3.7.2.1 Periodic scheduling threshold

The Isochronous Scheduling Threshold field in the USB_HCCPARAMS capability register is an indicator to system software as to how the host controller pre-fetches and effectively caches schedule data structures.

It is used by system software when adding isochronous work items to the periodic schedule. The value of this field indicates to system software the minimum distance it can update isochronous data (relative to the current location of the host controller execution in the periodic list) and still have the host controller process them.

The iTD and siTD data structures each describe 8 micro-frames worth of transactions. The host controller is allowed to cache one (or more) of these data structures in order to reduce memory traffic. Three basic caching models that account for the fact the isochronous data structures span 8 micro-frames. The three caching models are: no caching, micro-frame caching and frame caching.

When software is adding new isochronous transactions to the schedule, it always performs a read of the USB_FRINDEX register to determine the current frame and micro-frame the host controller is currently executing. Of course, there is no information about where in the micro-frame the host controller is, so a constant uncertainty-factor of one micro-frame has to be assumed. Combining the knowledge of where the host controller is executing with the knowledge of the caching model allows the definition of simple algorithms for how closely software can reliably work to the executing host controller.

No caching is indicated with a value of zero in the Isochronous Scheduling Threshold field. The host controller may pre-fetch data structures during a periodic schedule traversal (per micro-frame) but always dumps any accumulated schedule state at the end of the micro-frame. At the appropriate time relative to the beginning of every micro-frame, the host controller always begins schedule traversal from the frame list. Software can use the value of the USB_FRINDEX register (plus the constant 1 uncertainty-factor) to determine the approximate position of the executing host controller. When no caching is selected, software can add an isochronous transaction as near as 2 micro-frames in front of the current executing position of the host controller.

Frame caching is indicated with a non-zero value in bit [7] of the Isochronous Scheduling Threshold field. In the frame-caching model, system software assumes that the host controller caches one (or more) isochronous data structures for an entire frame (8 micro-frames). Software uses the value of the USB_FRINDEX register (plus the constant 1

uncertainty) to determine the current micro-frame/frame (assume modulo 8 arithmetic in adding the constant 1 to the micro-frame number). For any current frame N, if the current micro-frame is 0 to 6, then software can safely add isochronous transactions to Frame N + 1. If the current micro-frame is 7, then software can add isochronous transactions to Frame N + 2.

Micro-frame caching is indicated with a non-zero value in the least-significant 3 bits of the Isochronous Scheduling Threshold field. System software assumes the host controller caches one or more periodic data structures for the number of micro-frames indicated in the Isochronous Scheduling Threshold field. For example, if the count value were 2, then the host controller keeps a window of 2 micro-frames worth of state (current micro-frame, plus the next) on-chip. On each micro-frame boundary, the host controller releases the current micro-frame state and begins accumulating the next micro-frame state.

62.5.3.8 Asynchronous Schedule

The Asynchronous schedule traversal is enabled or disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

If the Asynchronous Schedule Enable bit is set to zero, then the host controller simply does not try to access the asynchronous schedule through the USB_ASYNCLISTADDR register. Likewise, when the Asynchronous Schedule Enable bit is one, then the host controller does use the USB_ASYNCLISTADDR register to traverse the asynchronous schedule. Modifications to the Asynchronous Schedule Enable bit are not necessarily immediate. Rather the new value of the bit is taken into consideration the next time the host controller needs to use the value of the USB_ASYNCLISTADDR register to get the next queue head.

The Asynchronous Schedule Status bit in the USB_USBSTS register indicates status of the asynchronous schedule. System software enables (or disables) the asynchronous schedule by writing one (or zero) to the Asynchronous Schedule Enable bit in the USB_USBCMD register. Software then can poll the Asynchronous Schedule Status bit to determine when the asynchronous schedule has made the desired transition. Software must not modify the Asynchronous Schedule Enable bit unless the value of the Asynchronous Schedule Enable bit equals that of the Asynchronous Schedule Status bit.

The asynchronous schedule is used to manage all Control and Bulk transfers. Control and Bulk transfers are managed using queue head data structures. The asynchronous schedule is based at the USB_ASYNCLISTADDR register. The default value of the USB_ASYNCLISTADDR register after reset is undefined and the schedule is disabled when the Asynchronous Schedule Enable bit is zero.

Software may only write this register with defined results when the schedule is disabled. For example, Asynchronous Schedule Enable bit in the USB_USBCMD and the Asynchronous Schedule Status bit in the USB_USBSTS register are zero. System software enables execution from the asynchronous schedule by writing a valid memory address (of a queue head) into this register. Then software enables the asynchronous schedule by setting the Asynchronous Schedule Enable bit to one. The asynchronous schedule is actually enabled when the Asynchronous Schedule Status bit is one.

When the host controller begins servicing the asynchronous schedule, it begins by using the value of the USB_ASYNCLISTADDR register. It reads the first referenced data structure and begins executing transactions and traversing the linked list as appropriate. When the host controller completes processing the asynchronous schedule, it retains the value of the last accessed queue head's horizontal pointer in the USB_ASYNCLISTADDR register. Next time the asynchronous schedule is accessed, this is the first data structure that is serviced. This provides round-robin fairness for processing the asynchronous schedule.

A host controller completes processing the asynchronous schedule when one of the following events occur:

- The end of a micro-frame occurs.
- The host controller detects an empty list condition (see Empty Asynchronous Schedule Detection)
- The schedule has been disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

The queue heads in the asynchronous list are linked into a simple circular list as shown in Figure 62-8. Queue head data structures are the only valid data structures that may be linked into the asynchronous schedule. An isochronous transfer descriptor (iTD or siTD) in the asynchronous schedule yields undefined results.

The maximum packet size field in a queue head is sized to accommodate the use of this data structure for all non-isochronous transfer types. The USB Specification, Revision 2.0 specifies the maximum packet sizes for all transfer types and transfer speeds. System software should always parameterize the queue head data structures according to the core specification requirements.

62.5.3.8.1 Adding Queue Heads to Asynchronous Schedule

This is a software requirement section.

There are two independent events for adding queue heads to the asynchronous schedule. The first is the initial activation of the asynchronous list. The second is inserting a new queue head into an activated asynchronous list.

Activation of the list is simple. System software writes the physical memory address of a queue head into the USB_ASYNCLISTADDR register, then enables the list by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to one.

When inserting a queue head into an active list, software must ensure that the schedule is always coherent from the host controllers' point of view. This means that the system software must ensure that all queue head pointer fields are valid. For example, qTD pointers have T-Bits set to one or reference valid qTDs and the Horizontal Pointer references a valid queue head data structure. The following algorithm represents the functional requirements:

```
InsertQueueHead (pQHeadCurrent, pQueueHeadNew)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadCurrent is a pointer to a queue head that is
          -- already in the active list
          -- pQHeadNew is a pointer to the queue head to be added
          --
          -- This algorithm links a new queue head into a existing
          -- list
          --
          pQueueHeadNew.HorizontalPointer  =  pQueueHeadCurrent.HorizontalPointer
          pQueueHeadCurrent.HorizontalPointer =  physicalAddressOf(pQueueHeadNew)
End InsertQueueHead
```

62.5.3.8.2 Removing Queue Heads from Asynchronous Schedule

This is a software requirement section.

There are two independent events for removing queue heads from the asynchronous schedule. The first is shutting down (deactivating) the asynchronous list. The second is extracting a single queue head from an activated list.

Software deactivates the asynchronous schedule by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to zero. Software can determine when the list is idle when the Asynchronous Schedule Status bit in the USB_USBSTS register is zero. The normal mode of operation is that software removes queue heads from the asynchronous schedule without shutting it down. Software must not remove an active queue head from the schedule. Software should first deactivate all active qTDs, wait for the queue head to go inactive, then remove the queue head from the asynchronous list. Software removes a queue head from the asynchronous list through the following algorithm. As illustrated, the unlinking is quite easy. Software merely must ensure all of the link pointers reachable by the host controller are kept consistent.

```
UnlinkQueueHead (pQHeadPrevious, pQueueHeadToUnlink, pQHeadNext)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadPrevious is a pointer to a queue head that
          -- references the queue head to remove
          -- pQHeadToUnlink is a pointer to the queue head to be
```

```
        -- removed
        -- pQheadNext is a pointer to a queue head still in the
        -- schedule. Software provides this pointer with the
        -- following strict rules:
        --          if the host software is one queue head, then
        --          pQHeadNext must be the same as
        --          QueueheadToUnlink.HorizontalPointer. If the host
        --          software is unlinking a consecutive series of
        --          queue heads, QHeadNext must be set by software to
        --          the  queue head remaining in the schedule.
        --
        -- This algorithm unlinks a queue head from a circular list
        --
        pQueueHeadPrevious.HorizontalPointer =  pQueueHeadToUnlink.HorizontalPointer
                            pQueueHeadToUnlink.HorizontalPointer = pQHeadNext
End UnlinkQueueHead
```

If software removes the queue head with the H-bit set to one, it must select another queue head still linked into the schedule and set its H-bit to one. This should be completed before removing the queue head. The requirement is that software keep one queue head in the asynchronous schedule, with its H-bit set to one. At the point software has removed one or more queue heads from the asynchronous schedule, it is unknown whether the host controller has a cached pointer to them. Similarly, it is unknown how long the host controller might retain the cached information, as it is implementation dependent and may be affected by the actual dynamics of the schedule load. Therefore, once software has removed a queue head from the asynchronous list, it must retain the coherency of the queue head (link pointers, and so on). It cannot disturb the removed queue heads until it knows that the host controller does not have a local copy of a pointer to any of the removed data structures.

The method software uses to determine when it is safe to modify a removed queue head is to handshake with the host controller. The handshake mechanism allows software to remove items from the asynchronous schedule, then execute a simple, lightweight handshake that is used by software as a key that it can free (or reuse) the memory associated the data structures it has removed from the asynchronous schedule.

The handshake is implemented with three bits in the host controller. The first bit is a command bit (Interrupt on Async Advance Doorbell bit in the USB_USBCMD register) that allows software to inform the host controller that something has been removed from its asynchronous schedule. The second bit is a status bit (Interrupt on Async Advance bit in the USB_USBSTS register) that the host controller sets after it has released all on-chip state that may potentially reference one of the data structures just removed. When the host controller sets this status bit to one, it also sets the command bit to zero. The third bit is an interrupt enable (Interrupt on Async Advance bit in the USB_USBINTR register) that is matched with the status bit. If the status bit is one and the interrupt enable bit is one, then the host controller asserts a hardware interrupt.

The figure below illustrates a general example. In this example, consecutive queue heads (B and C) are unlinked from the schedule using the algorithm above. Before the unlink operation, the host controller has a copy of queue head A.

The unlink algorithm requires that as software unlinks each queue head, the unlinked queue head is loaded with the address of a queue head that remains in the asynchronous schedule.

When the host controller observes that doorbell bit being set to one, it makes a note of the local reachable schedule information. In this example, the local reachable schedule information includes both queue heads (A and B). It is sufficient that the host controller can set the status bit (and clear the doorbell bit) as soon as it has traversed beyond current reachable schedule information (that is traversed beyond queue head (B) in this example). The following figure illustrates the generic queue head unlink scenario.

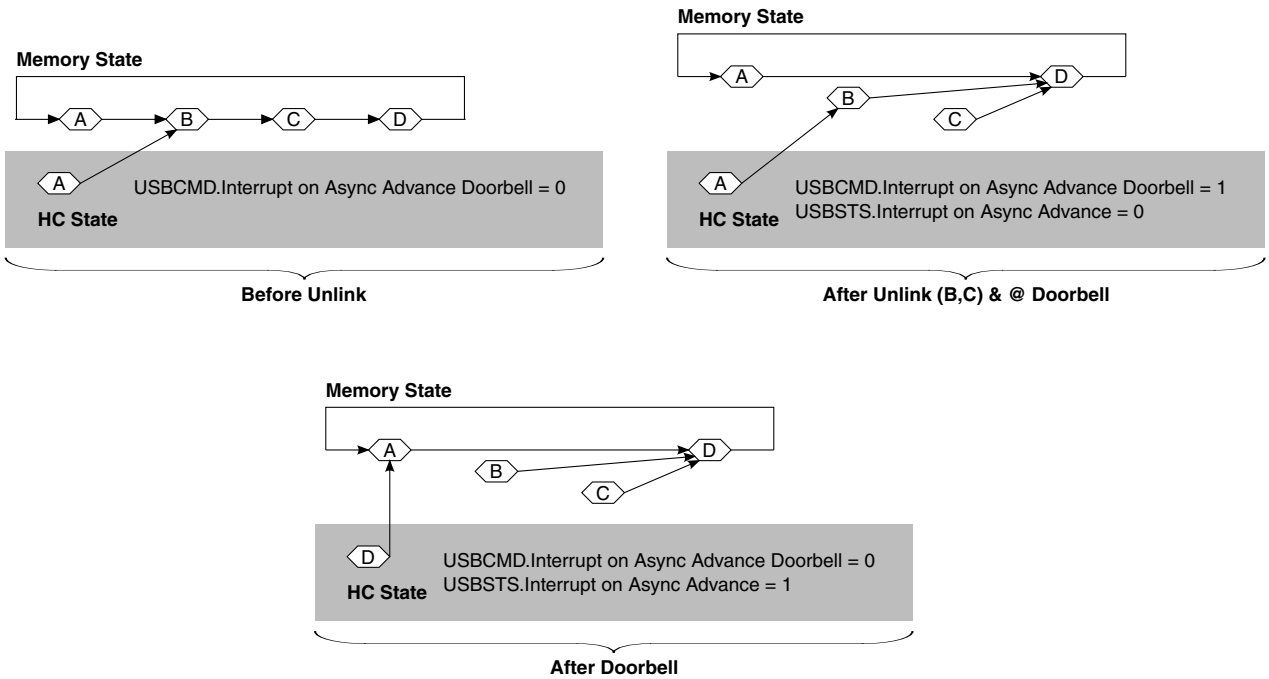


Figure 62-14. Generic Queue Head Unlink Scenario

Alternatively, a host controller implementation is allowed to traverse the entire asynchronous schedule list (for example, observed the head of the queue (twice)) before setting the Advance on Async status bit to one.

Software may re-use the memory associated with the removed queue heads after it observes the Interrupt on Async Advance status bit is set to one, following assertion of the doorbell. Software should acknowledge the Interrupt on Async Advance status as indicated in the USB_USBSTS register, before using the doorbell handshake again.

62.5.3.8.3 Empty Asynchronous Schedule Detection

The Enhanced Host Controller Interface uses two bits to detect when the asynchronous schedule is empty.

The queue head data structure (see Queue Head) defines an *H-bit* in the queue head, which allows software to mark a queue head as being the *head* of the reclaim list. The Enhanced Host Controller Interface also keeps a 1-bit flag in the USB_USBSTS register (*Reclamation*) that is set to zero when the Enhanced Interface Host Controller observes a queue head with the H-bit set to one. The reclamation flag in the status register is set to one when any USB transaction from the asynchronous schedule is executed (or whenever the asynchronous schedule starts, see Asynchronous schedule traversal: *Start Event*).

If the Enhanced Host Controller Interface ever encounters an *H-bit* of one and a *Reclamation* bit of zero, the EHCI controller simply stops traversal of the asynchronous schedule.

An example illustrating the H-bit in a schedule is shown in the following figure.

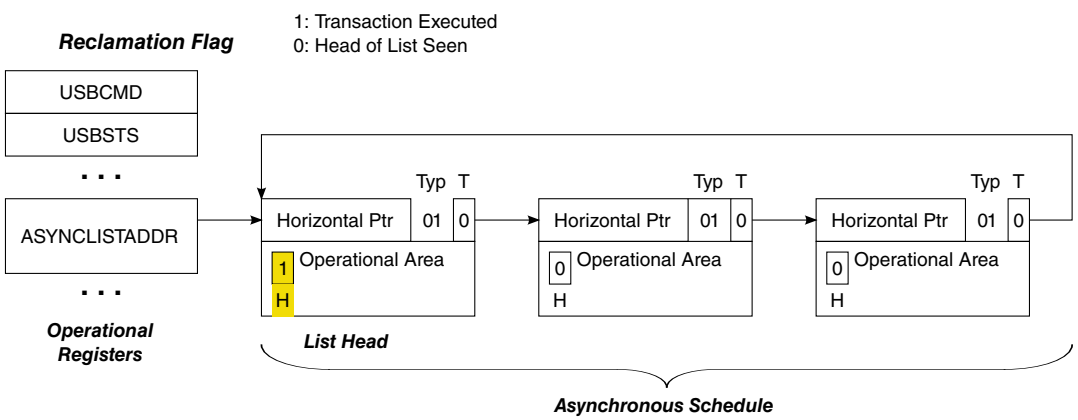


Figure 62-15. Asynchronous Schedule List w/Annotation to Mark Head of List

Software must ensure there is at most one queue head with the *H-bit* set to one, and that it is always coherent with respect to the schedule.

62.5.3.8.4 Restarting Asynchronous Schedule Before EOF

There are many situations where the host controller will detect an empty list *long* before the end of the micro-frame.

It is important to remember that under many circumstances the schedule traversal has stopped due to Nak/Nyet responses from all endpoints.

An example of particular interest is when a start-split for a bulk endpoint occurs early in the micro-frame. Given the EHCI simple traversal rules, the complete-split for that transaction may Nak/Nyet out very quickly. If it is the only item in the schedule, then the host controller ceases traversal of the Asynchronous schedule very early in the micro-frame. In order to provide reasonable service to this endpoint, the host controller should issue the complete-split before the end of the current micro-frame, instead of waiting

until the next micro-frame. When the reason for host controller idling asynchronous schedule traversal is because of empty list detection, it is mandatory the host controller implement a 'waking' method to resume traversal of the asynchronous schedule. An example method is described below.

62.5.3.8.4.1 Example Method for Restarting Asynchronous Schedule Traversal

The reason for idling the host controller when the list is empty is to keep the host controller from unnecessarily occupying too much memory bandwidth. The question is: *how long should the host controller stay idle before restarting*?

The answer in this example is based on deriving a manifest constant, which is the amount of time the host controller will stay idle before restarting traversal. In this example, the manifest constant is called *AsyncSchedSleepTime*, and has a value of 10 µsec. The value is derived based on the analysis in Example Derivation for *AsyncSchedSleepTime*, The traversal algorithm is simple:

- Traverse the Asynchronous schedule until the either an End-Of-micro-Frame event occurs, or an empty list is detected. If the event is an End-of-micro-Frame, go attempt to traverse the Periodic schedule. If the event is an empty list, then set a sleep timer and go to a *schedule sleep* state.
- When the sleep timer expires, set working context to the Asynchronous Schedule start condition and go to *schedule active* state. The start context allows the HC to reload *Nakcnt* fields, and so on. So the HC has a chance to run for more than one iteration through the schedule.

This process simply repeats itself each micro-frame. The figure below illustrates a sample state machine to manage the active and sleep states of the Asynchronous Schedule traversal policy. There are three states: Actively traversing the Asynchronous schedule, Sleeping, and Not Active. The last two are similar in terms of interaction with the Asynchronous schedule, but the Not Active state means that the host controller is busy with the Periodic schedule or the Asynchronous schedule is not enabled. The Sleeping state is specifically a special state where the host controller is just waiting for a period of time before resuming execution of the Asynchronous schedule.

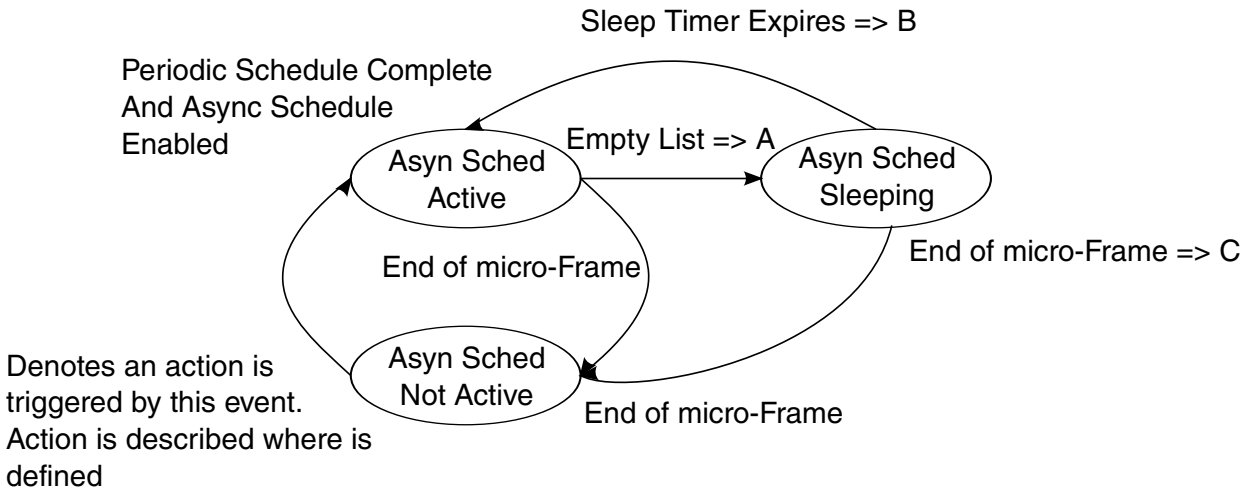


Figure 62-16. Example State Machine for Managing Asynchronous Schedule Traversal

The actions referred to in the figure above are defined in the following table.

Table 62-42. Asynchronous Schedule SM Transition Actions

Action	Action Description Label
A	On detection of the empty list, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.
B	When the *AsynchronousTraversalSleepTimer* expires, the host controller sets the *Reclamation* bit in the USBSTS register to one and moves the Nak Counter reload state machine to WaitForListHead (see Nak Count Reload Control).
C	The host controller cancels the sleep timer (*AsynchronousTraversalSleepTimer*).

62.5.3.8.4.2 Async Sched Not Active

This is the initial state of the traversal state machine after a host controller reset. The traversal state machine does not leave this state when the *Asynchronous Schedule Enable* bit in the USB_USBCMD register is zero.

This state is entered from Async Sched Active or Async Sched Sleeping states when the end-of-micro-frame event is detected.

62.5.3.8.4.3 Async Sched Active

This state is entered from the Async Sched Not Active state when the periodic schedule is not active. It is also entered from the Async Sched Sleeping states when the *AsyncrhonousTraversalSleepTimer* expires. On every transition into this state, the host controller sets the *Reclamation* bit in the USB_USBSTS register to one.

While in this state, the host controller continually traverses the asynchronous schedule until either the end of micro-frame or an empty list condition is detected.

62.5.3.8.4.4 Async Sched Sleeping

The state is entered from the Async Sched Active state when a schedule empty condition is detected. On entry to this state, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.

62.5.3.8.4.5 Example Derivation for *AsyncSchedSleepTime*

The derivation is based on analysis of what work the host controller could be doing next.

It assumes the host controller does not keep any state about what work is possibly pending in the asynchronous schedule. The schedule could contain any mix of the possible combinations of high- full- or low-speed control and bulk requests.

The table below summarizes some of the typical 'next transactions' that could be in the schedule, and the amount of time (for example *footprint*, or *wall clock*) the transaction takes to complete.

Table 62-43. Typical Low-/Full-speed Transaction Times

Transaction Attributes		Footprint (time)	Description
Speed	HS	11.9 ms	Maximum foot print for a worst-case, full-sized bulk data transaction.
Size	512	9.45 ms	Maximum footprint for an approximate best-case, full-sized bulk data transaction.
Type	Bulk		
Speed	FS	~50 ms	Approximate typical for full-sized bulk data. An 8-byte low-speed is about 2x, or between 90 and 100 ms.
Size	64		
Type	Bulk		
Speed	FS	~12 ms	Approximate typical for 8-byte bulk/control (that is setup)
Size	8		
Type	Cntrl		

A *AsyncSchedSleepTime* value of 10 µs provides a reasonable relaxation of the system memory load and still provides a good level of service for the various transfer types and payload sizes. For example, say we detect an empty list after issuing a start-split for a 64-byte full-speed bulk request. Assuming this is the only thing in the list, the host controller gets the results of the full-speed transaction from the hub during the fifth complete-split request. If the full-speed transaction was an IN and it nak'd, the 10 µs sleep period would allow the host controller to get the NAK results on the first complete-split.

62.5.3.8.5 Asynchronous schedule traversal: *Start Event*

Once the HC has *idled* itself through the empty schedule detection (Section 0), it will naturally *activate* and begin processing from the Periodic Schedule at the beginning of each micro-frame.

In addition, it may have idled itself early in a micro-frame. When this occurs (idles early in the micro-frame) the HC must occasionally *re-activate* during the micro-frame and traverse the asynchronous schedule to determine whether any progress can be made. The requirements and method for this restart are described in Restarting Asynchronous Schedule Before EOF . Asynchronous schedule *Start Event*s are defined to be:

- Whenever the host controller transitions from the periodic schedule to the asynchronous schedule. If the periodic schedule is disabled and the asynchronous schedule is enabled, then the beginning of the micro-frame is equivalent to the transition from the periodic schedule, or
- The asynchronous schedule traversal restarts from a sleeping state (see Restarting Asynchronous Schedule Before EOF).

62.5.3.8.6 Reclamation Status Bit (USBSTS Register)

The operation of the empty asynchronous schedule detection feature (see Empty Asynchronous Schedule Detection) depends on the proper management of the *Reclamation* bit in the USB_USBSTS register. The host controller tests for an empty schedule just after it fetches a new queue head while traversing the asynchronous schedule (see Fetch Queue Head).

The host controller is required to set the *Reclamation* bit to one whenever an asynchronous schedule traversal *Start Event*, as documented in Asynchronous schedule traversal: *Start Event*, occurs. The *Reclamation* bit is also set to one whenever the host controller executes a transaction while traversing the asynchronous schedule (see Execute Transaction). The host controller sets the *Reclamation* bit to zero whenever it finds a queue head with its *H-bit* set to one. Software should only set a queue head's *H-bit* if the queue head is in the asynchronous schedule. If software sets the *H-bit* in an interrupt queue head to one, the resulting behavior is undefined. The host controller may set the *Reclamation* bit to zero when executing from the periodic schedule.

62.5.3.9 Operational Model for Nak Counter

This section describes the operational model for the *NakCnt* field defined in a queue head.

See Queue Head Initialization for more information. Software should not use this feature for interrupt queue heads. This rule is not required to be enforced by the host controller.

USB protocol has built-in flow control through the Nak response by a device. There are several scenarios, beyond the Ping feature, where an endpoint may naturally Nak or Nyet the majority of the time. An example is the host controller management of the split transaction protocol for control and bulk endpoints. All bulk endpoints (High- or Full-speed) are serviced through the same asynchronous schedule. The time between the *Start-split* transaction and the first *Complete-split* transaction could be very short (that is like when the endpoint is the only one in the asynchronous schedule). The hub NYETs (effectively Naks) the *Complete-split* transaction until the classic transaction is complete. This could result in the host controller thrashing memory, repeatedly fetching the queue head and executing the transaction to the Hub, which does not complete until after the transaction on the classic bus completes.

The two component fields in a queue head to support the throttling feature: a counter field (*NakCnt*), and a counter reload field (*RL*). *NakCnt* is used by the host controller as one of the criteria to determine whether or not to execute a transaction to the endpoint. The two operational modes associated with this counter:

- Not Used- This mode is set when the *RL* field is zero. The host controller ignores the *NakCnt* field for any execution of transactions through a queue head with an *RL* field of zero. Software must use this selection for interrupt endpoints.
- Nak Throttle Mode- This mode is selected when the *RL* field is non-zero. In this mode, the value in the *NakCnt* field represents the maximum number of Nak or Nyet responses the host controller tolerates on each endpoint. In this mode, the HC decrements the *NakCnt* field based on the token/handshake criteria listed in the table below. The host controller must reload *NakCnt* when the endpoint successfully moves data (for example, policy to reward device for moving data).

The following table describes the NakCnt field adjustment rules.

Table 62-44. NakCnt Field Adjustment Rules

Token	Handshake	
	Handshake NAK	NYET
IN/PING	decrement *NakCnt*	N/A (protocol error)
OUT	decrement *NakCnt*	No Action[1] Start
Split	decrement *NakCnt*	N/A (protocol error)
Complete Split	No Action	Decrement *NakCnt*

1. Recommended behavior on this response is to reload *NakCnt*

In summary, system software enables the counter by setting the reload field (*RL*) to a non-zero value. The host controller may execute a transaction if *NakCnt* is non-zero. The host controller does not execute a transaction if *NakCnt* is zero. The reload mechanism is described in detail in Nak Count Reload Control .

NOTE

When all queue heads in the Asynchronous Schedule either exhausts all transfers or all NakCnt's go to zero, then the host controller detects an empty Asynchronous Schedule and idle schedule traversal (see Empty Asynchronous Schedule Detection).

Any time the host controller begins a new traversal of the Asynchronous Schedule, a *Start Event* is assumed, see Asynchronous schedule traversal: *Start Event*. Every time a Start-Event occurs, the Nak Count reload procedure is enabled.

62.5.3.9.1 Nak Count Reload Control

When the host controller reaches the *Execute Transaction* state for a queue head (meaning that it has an active operational state), it checks to determine whether the *NakCnt* field should be reloaded from *RL* (see Execute Transaction). If the answer is yes, then *RL* is copied into *NakCnt*. After the reload or if the reload is not active, the host controller evaluates whether to execute the transaction.

The host controller must reload nak counters (*NakCnt* see Queue Head) in queue heads during the first pass through the reclamation list after an asynchronous schedule Start Event (see Asynchronous schedule traversal: *Start Event* for the definition of the Start Event). The Asynchronous Schedule should have at most one queue head marked as the head (see Figure 62-15).

The following figure illustrates an example state machine that satisfies the operational requirements of the host controller detecting the first pass through the Asynchronous Schedule. This state machine is maintained internal to the host controller and is only used to gate reloading of the nak counter during the queue head traversal state: Execute Transaction (see the figure below). The host controller does not perform the nak counter reload operation if the RL field (see Queue Head) is set to zero.

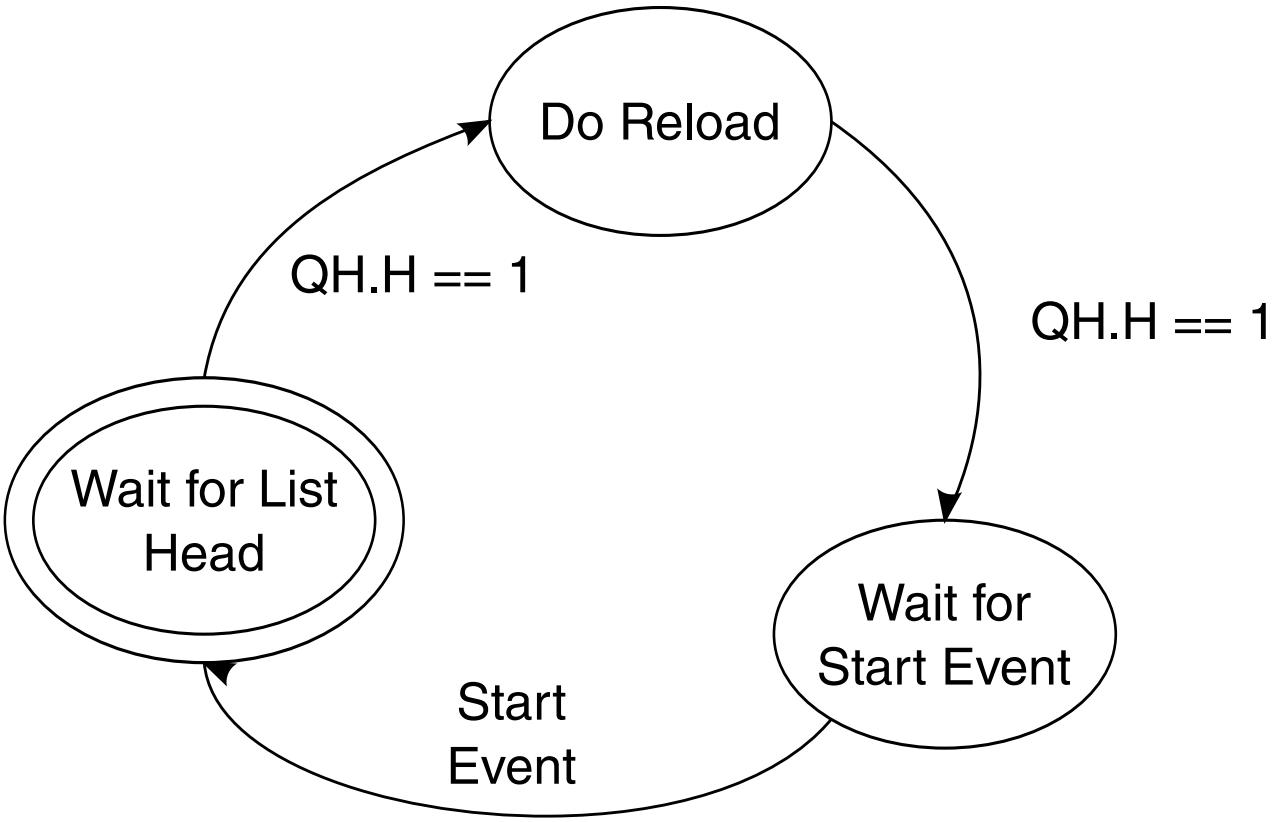


Figure 62-17. Example HC State Machine for Controlling Nak Counter Reloads

62.5.3.9.1.1 Wait for List Head

This is the initial state.

The state machine enters this state from Wait for Start Event when a start event as defined in Asynchronous schedule traversal: *Start Event* occurs.

The purpose of this state is to wait for the first observation of the head of the Asynchronous Schedule.

This occurs when the host controller fetches a queue head whose *H-bit* is set to one.

62.5.3.9.1.2 Do Reload

This state is entered from the Wait for List Head state when the host controller fetches a queue head with the *H-bit* set to one. While in this state, the host controller performs nak counter reloads for every queue head visited that has a non-zero nak reload value (*RL*) field.

62.5.3.9.1.3 Wait for Start Event

This state is entered from the *Do Reload* state when a queue head with the *H-bit* set to one is fetched. While in this state, the host controller does not perform nak counter reloads.

62.5.3.10 Managing Control/Bulk/Interrupt Transfers through Queue Heads

This section presents an overview of how the host controller interacts with queuing data structures.

Queue heads use the Queue Element Transfer Descriptor (qTD) structure. One queue head is used to manage the data stream for one endpoint. The queue head structure contains static endpoint characteristics and capabilities. It also contains a working area from where individual bus transactions for an endpoint are executed (see Overlay area defined in Table 62-27). Each qTD represents one or more bus transactions, which is defined in the context of this specification as a *transfer*.

The general processing model for the host controller's use of a queue head is simple:

- read a queue head,
- execute a transaction from the overlay area,
- write back the results of the transaction to the overlay area,
- move to the next queue head.

If the host controller encounters errors during a transaction, the host controller sets one (or more) of the error reporting bits in the queue head's *Status* field. The *Status* field accumulates all errors encountered during the execution of a qTD (for example, the error bits in the queue head *Status* field are 'sticky' until the transfer (qTD) has completed). This state is always written back to the source qTD when the transfer is complete. On transfer (for example, buffer or halt conditions) boundaries, the host controller must auto-advance (without software intervention) to the next qTD. Additionally, the hardware must be able to halt the queue so no additional bus transactions occurs for the endpoint and the host controller does not advance the queue.

An example host controller operational state machine of a queue head traversal is illustrated in the following figure. This state machine is a model for how a host controller should traverse a queue head. The host controller must be able to advance the queue from the *Fetch QH* state in order to avoid all hardware/software race conditions. This simple mechanism allows software to simply link qTDs to the queue head and *activate* them, then the host controller always *find* them if/when they are reachable. The figure below illustrates the Host Controller Queue Head Traversal State Machine.

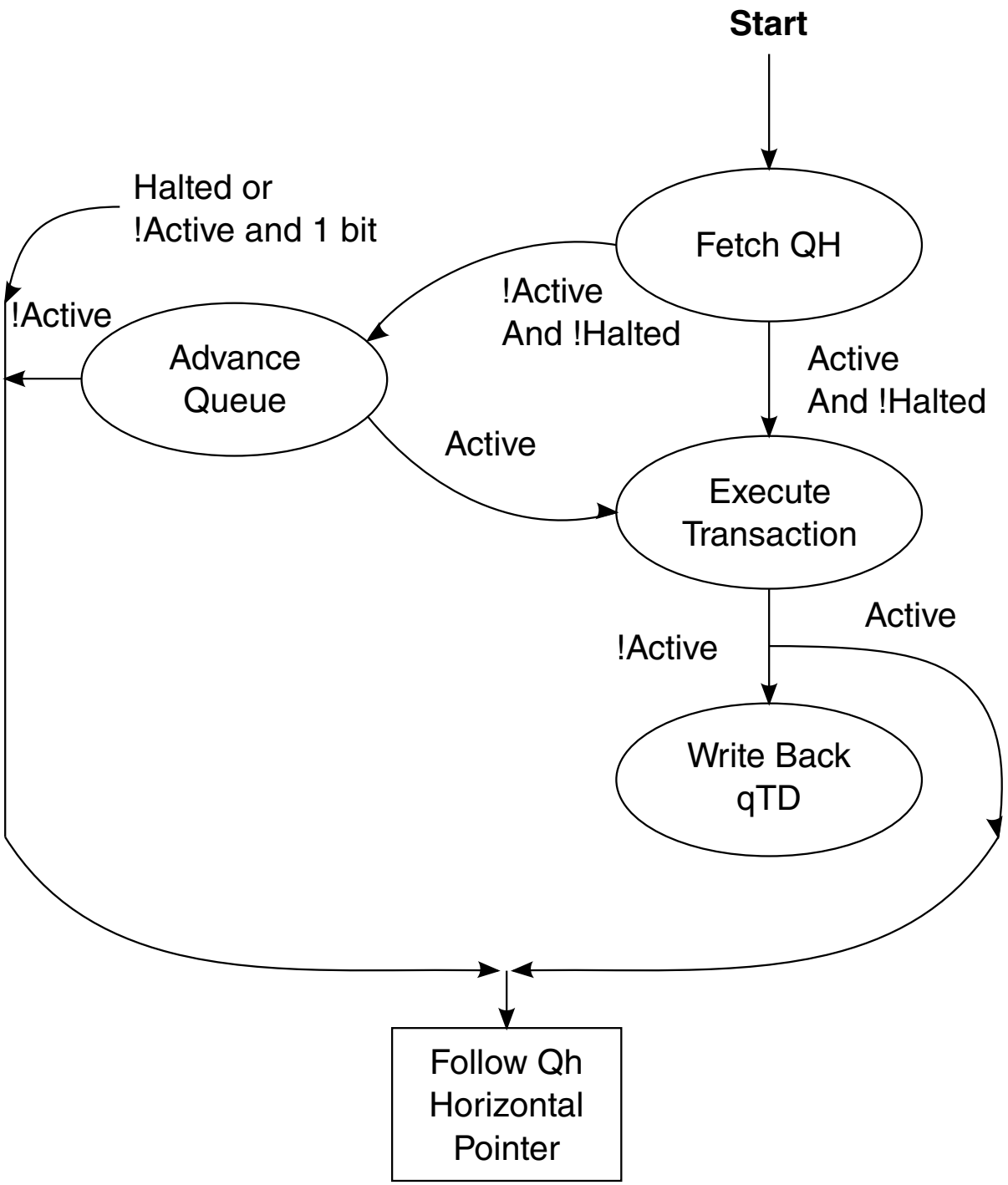


Figure 62-18. Host Controller Queue Head Traversal State Machine

This traversal state machine applies to all queue heads, regardless of transfer type or whether split transactions are required. The following sections describe each state. Each state description describes the entry criteria. The Execute Transaction state (see Execute

Transaction) describes the basic requirements for all endpoints. Split Transactions for Asynchronous Transfers and Split Transaction Interrupt describe details of the required extensions to the Execute Transaction state for endpoints requiring split transactions.

NOTE

Prior to software placing a queue head into either the periodic or asynchronous list, software must ensure the queue head is properly initialized. Minimally, the queue head should be initialized to the following (see Section Queue Head for layout of a queue head):

Valid static endpoint state.

- For the very first use of a queue head, software may zero-out the queue head transfer overlay, then set the *Next qTD Pointer* field value to reference a valid qTD.

62.5.3.10.1 Fetch Queue Head

A queue head can be referenced from the physical address stored in the ASYNCLISTADDR Register. Additionally, it may be referenced from the *Next LinkPointer* field of an iTD, siTD, FSTN or another Queue Head. If the referencing link pointer has the *Typ* field set to indicate a queue head, it is assumed to reference a queue head structure as defined in Queue Head.

While in this state, the host controller performs operations to implement empty schedule detection (see Empty Asynchronous Schedule Detection) and Nak Counter reloads (see Operational Model for Nak Counter). After the queue head has been fetched, the host controller conducts the following queries for empty schedule detection:

- If queue head is not an interrupt queue head (that is *S-mask* is zero), and
- The *H-bit* is one, and
- The *Reclamation* bit in the USBSTS register is zero.

When these criteria are met, the host controller stops traversing the asynchronous list (as described in Empty Asynchronous Schedule Detection). When the criteria are not met, the host controller continues schedule traversal. If the queue head is not an interrupt and the *H-bit* is one and the *Reclamation* bit is one, then the host controller sets the *Reclamation* bit in the USBSTS register to zero before completing this state. The operations for reloading of the Nak Counter are described in detail in Operational Model for Nak Counter.

This state is complete when the queue head has been read on-chip.

62.5.3.10.2 Advance Queue

To advance the queue, the host controller must find the next qTD, adjust pointers, perform the overlay and write back the results to the queue head.

This state is entered from the FetchQHD state if the overlay *Active* and *Halt* bits are set to zero. On entry to this state, the host controller determines which next pointer to use to fetch a qTD, fetches a qTD and determines whether or not to perform an overlay.

NOTE

If the *I-bit* is one and the *Active* bit is zero, the host controller immediately skips processing of this queue head, exits this state and uses the horizontal pointer to the next schedule data structure. If the field *Bytes to Transfer* is not zero and the *T-bit* in the *Alternate Next qTD Pointer* is set to zero, then the host controller uses the *Alternate Next qTD Pointer*. Otherwise, the host controller uses the *NextqTD Pointer*. If *NextqTD Pointer's T-bit* is set to one, then the host controller exits this state and uses the horizontal pointer to the next schedule data structure.

Using the selected pointer the host controller fetches the referenced qTD. If the fetched qTD has its *Active* bit set to one, the host controller moves the pointer value used to reach the qTD (*Next* or *Alternate Next*) to the *Current qTD Pointer* field, then performs the overlay. If the fetched qTD has its *Active* bit set to zero, the host controller aborts the queue advance and follows the queue head's horizontal pointer to the next schedule data structure.

The host controller performs the overlay based on the following rules:

- The value of the data toggle (*dt*) field in the overlay area depends on the value of the *data toggle control* (*dtc*) bit (see Table 62-29).
- If the *EPS* field indicates the endpoint is a high-speed endpoint, the *Ping* state field is preserved by the host controller. The value of this field is not changed as a result of the overlay.
- *C-prog-mask* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *Frame Tag* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *NakCnt* field in the overlay area is loaded from the *RL* field in the queue head's Static Endpoint State.
- All other areas of the overlay are set by the incoming qTD.

The host controller exits this state when it has committed the write to the queue head.

62.5.3.10.3 Execute Transaction

The host controller enters this state from the Fetch Queue Head state only if the *Active* bit in *Status* field of the queue head is set to one.

On entry to this state, the host controller executes a few pre-operations, then checks some pre-condition criteria before committing to executing a transaction for the queue head.

The pre-operations performed and the pre-condition criteria depend on whether the queue head is an interrupt endpoint. The host controller can determine that a queue head is an interrupt queue head when the queue head's *S-mask* field contains a non-zero value. It is the responsibility of software to ensure the *S-mask* field is appropriately initialized based on the transfer type. There are other criteria that must be met if the *EPS* field indicates that the endpoint is a low- or full-speed endpoint, see Split Transactions for Asynchronous Transfers and Split Transaction Interrupt .

62.5.3.10.3.1 Interrupt Transfer Pre-condition Criteria

If the queue head is for an interrupt endpoint (for example, non-zero *S-mask* field), then the FRINDEX[2:0] field must identify a bit in the *S-mask* field that has one in it.

For example, an *S-mask* value of 00100000b would evaluate to true only when FRINDEX[2:0] is equal to 101b. If this condition is met then the host controller considers this queue head for a transaction.

62.5.3.10.3.2 Asynchronous Transfer Pre-operations and Pre-condition Criteria

If the queue head is not for an interrupt endpoint (for example, zero *S-mask* field), then the host controller performs one pre-operation and then evaluates one pre-condition criteria.

The pre-operation is:

Checks the Nak counter reload state (Operational Model for Nak Counter). It may be necessary for the host controller to reload the Nak Counter field. The reload is performed at this time.

The pre-condition evaluated is:

- Whether or not the *NakCnt* field has been reloaded, the host controller checks the value of the *NakCnt* field in the queue head. If *NakCnt* is non-zero, or if the *Reload Nak Counter* field is zero, then the host controller considers this queue head for a transaction.

62.5.3.10.3.3 Transfer Type Independent Pre-operations

Regardless of the transfer type, the host controller always performs at least one pre-operation and evaluates one pre-condition. The pre-operation is:

- A host controller internal transaction (down) counter *qHTransactionCounter* is loaded from the queue head's *Mult* field. A host controller implementation is allowed to ignore this for queue heads on the asynchronous list. It is mandatory for interrupt queue heads. Software should ensure that the *Mult* field is set appropriately for the transfer type.

The pre-conditions evaluated are:

- The host controller determines whether there is enough time in the micro-frame to complete this transaction (see Transaction Fit - A Best-Fit Approximation Algorithm for an example evaluation method). If there is not enough time to complete the transaction, the host controller exits this state.
- If the value of *qHTransactionCounter* for an interrupt endpoint is zero, then the host controller exits this state.

When the pre-operations are complete and pre-conditions are met, the host controller sets the *Reclamation* bit in the USBSTS register to one and then begins executing one or more transactions using the endpoint information in the queue head. The host controller iterates *qHTransactionCounter* times in this state executing transactions. After each transaction is executed, *qHTransactionCounter* is decremented by one. The host controller exits this state when one of the following events occurs:

- The *qHTransactionCounter* decrements to zero, or
- The endpoint responds to the transaction with any handshake other than an ACK,[4] or
- The transaction experiences a transaction error, or
- The *Active* bit in the queue head goes to zero, or
- There is not enough time in the micro-frame left to execute the next transaction(see Transaction Fit - A Best-Fit Approximation Algorithm) for example method for implementing the frame boundary test).

NOTE

For a high-bandwidth interrupt OUT endpoint, the host controller may optionally immediately retry the transaction if it fails.

The results of each transaction is recorded in the on-chip overlay area. If data was successfully moved during the transaction, the transfer state in the overlay area is advanced. To advance queue head's transfer state, the *Total Bytes to Transfer* field is decremented by the number of bytes moved in the transaction, the data toggle bit (*dt*) is toggled, the current page offset is advanced to the next appropriate value (for example,

advanced by the number of bytes successfully moved), and the *C_Page* field is updated to the appropriate value (if necessary). See Buffer Pointer List Use for Data Streaming with qTDs .

NOTE

The *Total Bytes To Transfer* field may be zero when all the other criteria for executing a transaction are met. When this occurs, the host controller executes zero-length transaction to the endpoint. If the *PID_Code* field indicates an IN transaction and the device delivers data, the host controller detects a packet babble condition, set the *babble* and *halted* bits in the *Status* field, set the *Active* bit to zero, write back the results to the source qTD, then exit this state.

In the event an IN token receives a data PID mismatch response, the host controller must ignore the received data (for example not advance the transfer state for the bytes received). Additionally, if the endpoint is an interrupt IN, then the host controller must record that the transaction occurred (for example, decrement *qHTransactionCounter*). It is recommended (but not required) the host controller continue executing transactions for this endpoint if the resultant value of *qHTransactionCounter* is greater than one.

If the response to the IN bus transaction is a Nak (or Nyet) and *RL* is non-zero, *NakCnt* is decremented by one. If *RL* is zero, then no write-back by the host controller is required (for a transaction receiving a Nak or Nyet response and the value of *CErr* did not change). Software should set the *RL* field to zero if the queue head is an interrupt endpoint. Host controller hardware is not required to enforce this rule or operation.

After the transaction has finished and the host controller has completed the post processing of the results (advancing the transfer state and possibly *NakCnt*, the host controller writes back the results of the transaction to the queue head's overlay area in main memory).

The number of bytes moved during an IN transaction depends on how much data the device endpoint delivers. The maximum number of bytes a device can send is *MaximumPacket Size*. The number of bytes moved during an OUT transaction is either *Maximum Packet Length* bytes or *Total Bytes to Transfer*, whichever is less.

If there was a transaction error during the transaction, the transfer state (as defined above) is not advanced by the host controller. The *CErr* field is decremented by one and the status field is updated to reflect the type of error observed. Transaction errors are summarized in Transaction Error .

The following events causes the host controller to clear the *Active* bit in the queue head's overlay status field. When the *Active* bit transitions from one to zero, the transfer in the overlay is considered complete. The reason for the transfer completion (clearing the *Active* bit) determines the next state.

- *CErr* field decrements to zero. When this occurs the *Halted* bit is set to one and *Active* is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The device responds to the transaction with a STALL PID. When this occurs, the *Halted* bit is set to one and the *Active* bit is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The *Total Bytes to Transfer* field is zero after the transaction completes.
  - For a zero length transaction, it was zero before the transaction was started. When this condition occurs, the *Active* bit is set to zero.
- The PID code is an IN, and the number of bytes moved during the transaction is less than the *Maximum Packet Length*. When this occurs, the *Active* bit is set to zero and a short packet condition exists. The short-packet condition is detected during the Advance Queue state. Refer to Split Transactions for additional rules for managing low- and full-speed transactions.

With the exception of a NAK response (when *RL* field is zero), the host controller always writes the results of the transaction back to the overlay area in main memory. This includes when the transfer completes. For a high-speed endpoint, the queue head information written back includes minimally the following fields: The *PID Code* field indicates an IN and the device sends more than the expected number of bytes (for example *Maximum Packet Length* or *Total Bytes to Transfer* bytes, whichever is less) (for example a packet babble). This results in the host controller setting the *Halted* bit to one.

- NakCnt, dt, Total Bytes to Transfer, C_Page, Status, CERR, and Current Offset

For a low- or full-speed device the queue head information written back also includes the fields:

- C-prog-mask, FrameTag and S-bytes.

The duration of this state depends on the time it takes to complete the transaction(s) and the status write to the overlay is committed.

62.5.3.10.3.4 Halting a Queue Head

A halted endpoint is defined only for the transfer types that are managed through queue heads (control, bulk and interrupt).

The following events indicate that the endpoint has reached a condition where no more activity can occur without intervention from the driver:

- An endpoint may return a STALL handshake during a transaction,
- A transaction had three consecutive error conditions, or
- A Packet Babble error occurs on the endpoint.

When any of these events occur (for a queue head) the Host Controller halts the queue head and set the USBERRINT status bit in the USB_n_USBSTS register to one. To halt the queue head, the *Active* bit is set to zero and the *Halted* bit is set to one. There may be other error status bits that are set when a queue is halted. The host controller always writes back the overlay area to the source qTD when the transfer is complete, regardless of the reason (normal completion, short packet or halt). The host controller does not advance the transfer state on a transaction that results in a *Halt* condition (for example no updates necessary for *Total Bytes to Transfer*, *C_Page*, *Current Offset,* and *dt*). The host controller must update *CErr* as appropriate. When a queue head is halted, the *USB Error Interrupt* bit in the USB_n_USBSTS register is set to one. If the *USB Error Interrupt Enable* bit in the USB_n_USBINTR register is set to one, a hardware interrupt is generated at the next interrupt threshold.

62.5.3.10.3.5 Asynchronous Schedule Park Mode

Asynchronous Schedule Park mode is a special execution mode that can be enabled by system software, where the host controller is permitted to execute more than one bus transaction from a high-speed queue head in the Asynchronous schedule before continuing horizontal traversal of the Asynchronous schedule.

This feature has no effect on queue heads or other data structures in the Periodic schedule. This feature is similar in intent as the *Mult* feature that is used in the Periodic schedule. Where-as the *Mult* feature is a characteristic that is tunable for each endpoint; park-mode is a policy that is applied to all high-speed queue heads in the asynchronous schedule. It is essentially the specification of an iterator for consecutive bus transactions to the same endpoint. All of the rules for managing bus transactions and the results of those as defined in Execute Transaction apply. This feature merely specifies how many consecutive times the host controller is permitted to execute from the same queue head before moving to the next queue head in the Asynchronous List. This feature should allow the host controller to attain better bus utilization for those devices that are capable of moving data at maximum rate, while at the same time providing a fair service to all endpoints.

A host controller exports its capability to support this feature to system software by setting the *Asynchronous Schedule Park Capability* bit in the USB_n_HCCPARAMs register to one. This information keys system software that the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USB_n_USBCMD register are modifiable. System software enables the feature by writing a one to the *Asynchronous Schedule Park Mode Enable* bit.

When park-mode is not enabled (for example *Asynchronous Schedule Park Mode Enable* bit in the USB_n_USBCMD register is zero), the host controller must not execute more than one bus transaction per high-speed queue head, per traversal of the asynchronous schedule. When park-mode is enabled, the host controller must not apply the feature to a queue head whose *EPS* field indicates a Low/Full-speed device (for example only one bus transaction is allowed from each Low/Full-speed queue head per traversal of the asynchronous schedule). Park-mode may only be applied to queue heads in the Asynchronous schedule whose *EPS* field indicates that it is a high-speed device.

The host controller must apply park mode to queue heads whose *EPS* field indicates a high-speed endpoint. The maximum number of consecutive bus transactions a host controller may execute on a high-speed queue head is determined by the value in the *Asynchronous Schedule Park Mode Count* field in the USB_n_USBCMD register. Software must not set *Asynchronous Schedule Park Mode Enable* bit to one and also set *Asynchronous Schedule Park Mode Count* field to zero. The resulting behavior is not defined. An example behavioral example describes the operational requirements for the host controller implementing park-mode. This feature does not affect how the host controller handles the bus transaction as defined in Execute Transaction . It only effects how many consecutive bus transactions for the current queue head can be executed. All boundary conditions, error detection and reporting applies as usual. This feature is similar in concept to the use of the *Mult* field for high-bandwidth Interrupt for queue heads in the Periodic Schedule.

The host controller effectively loads an internal down-counter *PM-Count* from *Asynchronous Schedule Park Mode Count* when *Asyncrhonous Schedule Park Mode Enable* bit is one, and a high-speed queue head is first fetched and meets all the criteria for executing a bus transaction. After the bus transaction, *PM-Count* is decremented. The host controller may continue to execute bus transactions from the current queue head until *PM-Count* goes to zero, an error is detected, the buffer for the current transfer is exhausted or the endpoint responds with a flow-control or STALL handshake.

The following table summarizes the responses that effect whether the host controller continues with another bus transaction for the current queue head.

Table 62-45. Actions for Park Mode, based on Endpoint Response and Residual Transfer State

PID	Endpoint Response	Transfer State after Transaction		Action
		PM-Count	Bytes to Transfer	
IN	DATA[0,1] w/Maximum Packet sized data	Not zero	Not Zero	Allowed to perform another bus transaction.[1, 2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't care	Move to next QH.

Table continues on the next page...

Table 62-45. Actions for Park Mode, based on Endpoint Response and Residual Transfer State (continued)

	DATA[0,1] w/short packet	Don't care	Don't care	Retire qTD and move to next QH.
	NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH.
OUT	ACK	Not zero	Not Zero	Allowed to perform another bus transaction.[2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't' care	Move to next QH.
	NYET, NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH
PING	ACK	Not Zero	Not Zero	Allowed to perform another bus transaction. [2]
	NAK	Don't care	Don't care	Move to next QH
	STALL, XactErr	Don't care	Don't care	Move to next QH

1. The host controller may continue to execute bus transactions from the current high-speed queue head (if *PM-Count* is not equal to zero), if a PID mismatch is detected (for example expected DATA1 and received DATA0, or visa-versa).
2. This specification does not *require* that the host controller execute another bus transaction when *PM-Count* is non-zero. Implementations are encouraged to make appropriate complexity and performance trade-offs.

62.5.3.10.4 Write Back qTD

This state is entered from the Execute Transaction state when the *Active* bit is set to zero.

The source data for the write-back is the transfer results area of the queue head overlay area (see Table 62-45).

The host controller uses the *Current qTD Pointer* field as the target address for the qTD.

The queue head transfer result area is written back to the transfer result area of the target qTD. This state is also referred to as: qTD retirement. The fields that must be written back to the source qTD include *Total Bytes to Transfer*, *Cerr*, and *Status*.

The duration of this state depends on when the qTD write-back is committed.

62.5.3.10.5 Follow Queue Head Horizontal Pointer

The host controller must use the horizontal pointer in the queue head to the next schedule data structure when any of the following conditions exist:

- If the *Active* bit is one on exit from the Execute Transaction state, or
- When the host controller exits the Write Back qTD state, or
- If the Advance Queue state fails to advance the queue because the target qTD is not active, or
- If the *Halted* bit is one on exit from the Fetch QH state.

There is no functional requirement that the host controller wait until the current transaction is complete before using the horizontal pointer to read the next linked data structure. However, it must wait until the current transaction is complete before executing the next data structure.

62.5.3.10.6 Buffer Pointer List Use for Data Streaming with qTDs

A qTD has an array of buffer pointers, which is used to reference the data buffer for a transfer. This specification requires that the buffer associated with the transfer be *virtually contiguous*.

This means: if the buffer spans more than one physical page, it must obey the following rules (the figure below illustrates an example):

- The first portion of the buffer must begin at some offset in a page and extend through the end of the page.
- The remaining buffer cannot be allocated in small chunks scattered around memory. For each 4 K chunk beyond the first page, each buffer portion matches to a full 4 K page. The final portion, which may only be large enough to occupy a portion of a page, must start at the top of the page and be contiguous within that page.

The buffer pointer list in the qTD is long enough to support a maximum transfer size of 20 K bytes. This case occurs when all five buffer pointers are used and the first offset is zero. A qTD handles a 16 Kbyte buffer with any starting buffer alignment.

The host controller uses the field *C_Page* field as an index value to determine which buffer pointer in the list should be used to start the current transaction. The host controller uses a different buffer pointer for each physical page of the buffer. This is always true, even if the buffer is physically contiguous.

The host controller must detect when the current transaction spans a page boundary and automatically move to the next available buffer pointer in the page pointer list. The next available pointer is reached by incrementing *C_Page* and pulling the next page pointer from the list. Software must ensure there are sufficient buffer pointers to move the amount of data specified in the *Bytes to Transfer* field.

The following figure illustrates a nominal example of how System software would initialize the buffer pointers list and the *C_Page* field for a transfer size of 16383 bytes. *C_Page* is set to zero. The upper 20-bits of Page 0 references the start of the physical page. *Current Offset* (the lower 12-bits of queue head Dword 7) holds the offset in the page for example 2049 (for example 4096-2047). The remaining page pointers are set to reference the beginning of each subsequent 4 K page.

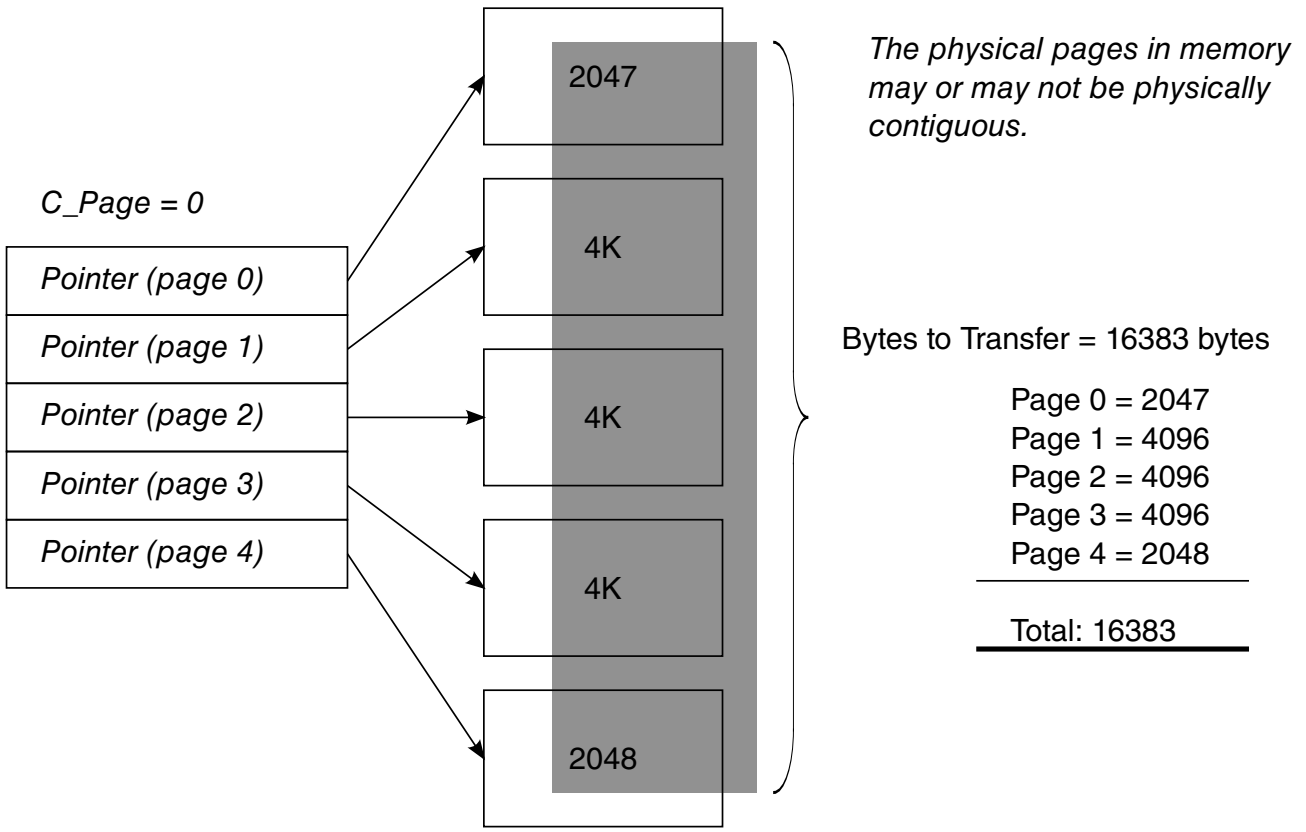


Figure 62-19. Example Mapping of qTD Buffer Pointers to Buffer Pages

For the first transaction on the qTD (assuming a 512-byte transaction), the host controller uses the first buffer pointer (page 0 because *C_Page* is set to zero) and concatenates the *Current Offset* field. The 512 bytes are moved during the transaction, the *Current Offset* and *Total Bytes to Transfer* are adjusted by 512 and written back to the queue head working area.

During the 4th transaction, the host controller needs 511 bytes in page 0 and one byte in page 1. The host controller increments *C_Page* (to 1) and use the page 1 pointer to move the final byte of the transaction. After the 4th transaction, the active page pointer is the page 1 pointer and *Current Offset* has rolled to one, and both are written back to the overlay area. The transactions continue for the rest of the buffer, with the host controller automatically moving to the next page pointer (that is *C_Page*) when necessary. The three conditions for how the host controller handles *C_Page*:

- The current transaction does not span a page boundary. The value of *C_Page* is not adjusted by the host controller.

- The current transaction does span a page boundary. The host controller must detect the page cross condition and advance to the next buffer while streaming data to/from the USB.
- The current transaction completes on a page boundary (that is the last byte moved for the current transaction is the last byte in the page for the current page pointer). The host controller must increment *C_Page* before writing back status for the transaction.

NOTE

The only valid adjustment the host controller may make to *C_Page* is to increment by one.

62.5.3.10.7 Adding Interrupt Queue Heads to the Periodic Schedule

The link path(s) from the periodic frame list to a queue head establishes in which frames a transaction can be executed for the queue head. Queue heads are linked into the periodic schedule so they are polled at the appropriate rate.

System software sets a bit in a queue head's *S-Mask* to indicate which micro-frame within 1 msec period a transaction should be executed for the queue head. Software must ensure that all queue heads in the periodic schedule have *S-Mask* set to a non-zero value. An *S-mask* with zero value in the context of the periodic schedule yields undefined results.

If the desired poll rate is greater than one frame, system software can use a combination of queue head linking and *S-Mask* values to spread interrupts of equal poll rates through the schedule so that the periodic bandwidth is allocated and managed in the most efficient manner possible. Some examples are illustrated in the following table.

Table 62-46. Example Periodic Reference Patterns for Interrupt Transfers with 2ms Poll Rate

Frame # Reference Sequence	Description
0, 2, 4, 6, 8, and so on *S-Mask* = 01h	A queue head for the *bInterval* of 2 msec (16 micro-frames) is linked into the periodic schedule so that it is reachable from the periodic frame list locations indicated in the previous column. In addition, the *S-Mask* field in the queue head is set to 01h, indicating that the transaction for the endpoint should be executed on the bus during micro-frame 0 of the frame.
0, 2, 4, 6, 8, and so on *S-Mask* = 02h	Another example of a queue head with a *bInterval* of 2 msec is linked into the periodic frame list at exactly the same interval as the previous example. However, the *S-Mask* is set to 02h indicating that the transaction for the endpoint should be executed on the bus during micro-frame 1 of the frame.

62.5.3.10.8 Managing Transfer Complete Interrupts from Queue Heads

The host controller sets an interrupt to be signaled at the next interrupt threshold when the completed transfer (qTD) has an *Interrupt on Complete* (*IOC*) bit set to one, or whenever a transfer (qTD) completes with a short packet.

If system software needs multiple qTDs to complete a client request (that is like a control transfer) the intermediate qTDs do not require interrupts. System software may only need a single interrupt to notify it that the complete buffer has been transferred. System software may set IOC's to occur more frequently. A motivation for this may be that it wants early notification so that interface data structures can be re-used in a timely manner.

62.5.3.11 Ping Control

USB 2.0 defines an addition to the protocol for high-speed devices called Ping. Ping is required for all USB 2.0 High-speed bulk and control endpoints.

Ping is not allowed for a split-transaction stream. This extension to the protocol eliminates the bad side-effects of Naking OUT endpoints. The *Status* field has a *Ping State* bit, which the host controller uses to determine the *next* actual PID it uses in the next transaction to the endpoint (see the table below).

The Ping State bit is only managed by the host controller for queue heads that meet the following criteria:

- Queue head is not an interrupt and
- *EPS* field equals High-Speed and
- *PIDCode* field equals OUT

The following table illustrates the state transition table for the host controller's responsibility for maintaining the PING protocol. Refer to Chapter 8 in the USB Specification Revision 2.0 for detailed description on the Ping protocol.

Table 62-47. Ping Control State Transition Table

Event			
Current	Host	Device	Next
Do Ping	PING	Nak	Do Ping
Do Ping	PING	Ack	Do OUT
Do Ping	PING	XactErr [1]	Do Ping
Do Ping	PING	Stall	N/C [2] Do
OUT	OUT	Nak	Do Ping
Do OUT	OUT	Nyet	Do Ping

Table continues on the next page...

Table 62-47. Ping Control State Transition Table (continued)

Do OUT	OUT	Ack	Do OUT
Do OUT	OUT	XactErr[1]	Do Ping
Do OUT	OUT	Stall	N/C [2]

1. Transaction Error (XactErr) is any time the host misses the handshake.
2. No transition change required for the Ping State bit. The Stall handshake results in the endpoint being halted (for example Active set to zero and Halt set to one). Software intervention is required to restart queue. 3 A Nyet response to an OUT means that the device has accepted the data, but cannot receive any more at this time. Host must advance the transfer state and additionally, transition the Ping State bit to Do Ping. The Ping State bit has the following encoding:

Table 62-48. Ping State bit Encoding

Value	Meaning
0B	Do OUT The host controller uses an OUT PID during the next bus transaction to this endpoint.
1B	Do Ping The host controller uses a PING PID during the next bus transaction to this endpoint.

The defined ping protocol (see USB 2.0 Specification, Chapter 8) allows the host to be *imprecise* on the initialization of the ping protocol (that is start in *Do OUT* when we don't know whether there is space on the device or not). The host controller manages the *Ping State* bit. System software sets the initial value in the queue head when it initializes a queue head. The host controller preserves the *Ping State* bit across all queue advancements. This means that when a new qTD is written into the queue head overlay area, the previous value of the *Ping State* bit is preserved.

62.5.3.12 Split Transactions

USB 2.0 defines extensions to the bus protocol for managing USB 1.x data streams through USB 2.0 Hubs.

This section describes how the host controller uses the interface data structures to manage data streams with full- and low-speed devices, connected below USB 2.0 hub, utilizing the split transaction protocol.

Refer to USB 2.0 Specification for the complete definition of the split transaction protocol. Full- and Low-speed devices are enumerated identically as high-speed devices, but the transactions to the Full- and Low-speed endpoints use the split-transaction protocol on the high-speed bus. The split transaction protocol is an encapsulation of (or wrapper around) the Full- or Low-speed transaction. The high-speed wrapper portion of the protocol is addressed to the USB 2.0 Hub and Transaction Translator below which the Full- or Low-speed device is attached.

The EHCI interface uses dedicated data structures for managing full-speed isochronous data streams (see Split Transaction Isochronous Transfer Descriptor (siTD)). Control, Bulk and Interrupt are managed using the queuing data structures (see Queue Head). The interface data structures need to be programmed with the device address and the Transaction Translator number of the USB 2.0 Hub operating as the Low-/Full-speed host controller for this link. The following sections describe the details of how the host controller must process and manage the split transaction protocol.

62.5.3.12.1 Split Transactions for Asynchronous Transfers

A queue head in the asynchronous schedule with an *EPS* field indicating a full-or low-speed device indicates to the host controller that it must use split transactions to stream data for this queue head.

All full-speed bulk and full-, low-speed control are managed through queue heads in the asynchronous schedule.

Software must initialize the queue head with the appropriate device address and port number for the transaction translator that is serving as the full/low-speed host controller for the links connecting the endpoint. Software must also initialize the split transaction state bit (*SplitXState*) to Do-Start-Split. Finally, if the endpoint is a control endpoint, then system software must set the *Control Transfer Type* (*C*) bit in the queue head to one. If this is not a control transfer type endpoint, the *C* bit must be initialized by software to be zero. This information is used by the host controller to properly set the Endpoint Type (ET) field in the split transaction bus token. When the *C* bit is zero, the split transaction token's ET field is set to indicate a bulk endpoint. When the *C* bit is one, the split transaction token's ET field is set to indicate a control endpoint. Refer to Chapter 8 of USB Specification Revision 2.0 for details.

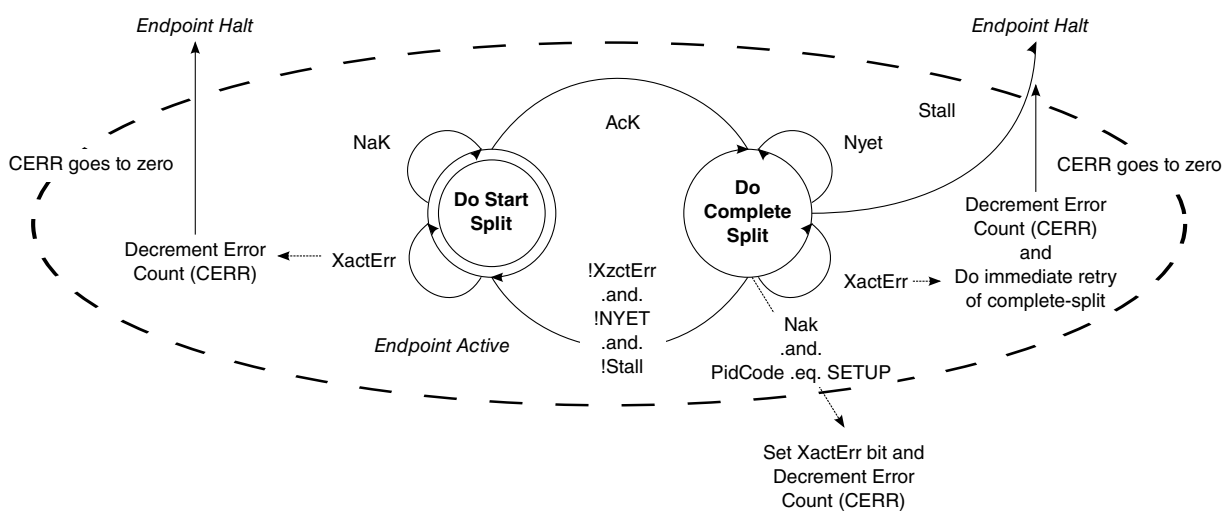


Figure 62-20. Host Controller Asynchronous Schedule Split-Transaction State Machine

62.5.3.12.1.1 Asynchronous - Do Start Split

This is the state which software must initialize a full- or low-speed asynchronous queue head. This state is entered from the Do Complete Split state only after a complete-split transaction receives a valid response from the transaction translator that is not a Nyet handshake.

For queue heads in this state, the host controller executes a start-split transaction to the appropriate transaction translator. If the bus transaction completes without an error and *PidCode* indicates an IN or OUT transaction, then the host controller reloads the error counter (*CErr*). If it is a successful bus transaction and the *PidCode* indicates a SETUP, the host controller does not reload the error counter. If the transaction translator responds with a Nak, the queue head is left in this state, and the host controller proceeds to the next queue head in the asynchronous schedule.

If the host controller times out the transaction (no response, or bad response) the host controller decrements *Cerr* and proceeds to the next queue head in the asynchronous schedule.

62.5.3.12.1.2 Asynchronous - Do Complete Split

This state is entered from the Do Start Split state only after a start-split transaction receives an Ack handshake from the transaction translator.

For queue heads in this state, the host controller executes a complete-split transaction to the appropriate transaction translator. If the transaction translator responds with a Nyet handshake, the queue head is left in this state, the error counter is reset and the host controller proceeds to the next queue head in the asynchronous schedule. When a Nyet handshake is received for a bus transaction where the queue head's *PidCode* indicates an IN or OUT, the host controller reloads the error counter (*CErr*). When a Nyet handshake is received for a complete-split bus transaction where the queue head's *PidCode* indicates a SETUP, the host controller must not adjust the value of *CErr*.

Independent of *PIDCode*, the following responses have the effects:

- Transaction Error (XactErr). Timeout or data CRC failure, and so on. The error counter (*Cerr*) is decremented by one and the complete split transaction is *immediately* retried (if possible). If there is not enough time in the micro-frame to execute the retry, the host controller MUST ensure that the next time the host controller begins executing from the Asynchronous schedule, it must begin executing from this queue head. If another start-split (for some other endpoint) is sent to the transaction translator before the complete-split is really completed, the transaction translator could dump the results (which were never delivered to the host). This is why the core specification states the retries must be immediate. A method to

accomplish this behavior is to not advance the asynchronous schedule. When the host controller returns to the asynchronous schedule in the next micro-frame, the first transaction from the schedule is the retry for this endpoint.

If *Cerr* went to zero, the host controller must halt the queue.

- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced and the state is exited. If the *PidCode* is a SETUP, then the Nak response is a protocol error. The *XactErr* status bit is set to one and the *CErr* field is decremented.
- STALL. The target endpoint responded with a STALL handshake. The host controller sets the *halt* bit in the status byte, retires the qTD but does not attempt to advance the queue.

If the *PidCode* indicates an IN, then any of following responses are expected:

- DATA0/1. On reception of data, the host controller ensures the PID matches the expected data toggle and checks CRC. If the packet is *good*, the host controller advances the state of the transfer, for example move the data pointer by the number of bytes received, decrement *BytesToTransfer* field by the number of bytes received, and toggle the *dt* bit. The host controller then exit this state. The response and advancement of transfer may trigger other processing events, such as retirement of the qTD and advancement of the queue.

If the data sequence PID does not match the expected, the data is ignored, the transfer state is not advanced and this state is exited. If the *PidCode* indicates an OUT/SETUP, then any of following responses are expected:

- ACK. The target endpoint accepted the data, so the host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount and the data toggle bit (*dt*) is toggled. The host controller then exit this state.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Managing Control/Bulk/Interrupt Transfers through Queue Heads).

62.5.3.12.2 Split Transaction Interrupt

Split-transaction Interrupt-IN/OUT endpoints are managed through the same data structures used for high-speed interrupt endpoints. They both co-exist in the periodic schedule.

Queue heads/qTDs offer the set of features required for reliable data delivery, which is characteristic to interrupt transfer types. The split-transaction protocol is managed completely within this defined functional transfer framework. For example, for a high-speed endpoint, the host controller visits a queue head, execute a high-speed transaction (if criteria are met) and advance the transfer state (or not) depending on the results of the entire transaction. For low- and full-speed endpoints, the details of the *execution* phase are different (that is takes more than one bus transaction to complete), but the remainder of the operational framework is intact. This means that the transfer advancement, and so on, occurs as defined in Managing Control/Bulk/Interrupt Transfers through Queue Heads, but only occurs on the completion of a split transaction.

62.5.3.12.2.1 Split Transaction Scheduling Mechanisms for Interrupt

Full- and low-speed Interrupt queue heads have an *EPS* field indicating full- or low-speed and have a non-zero *S-mask* field.

The host controller can detect this combination of parameters and assume the endpoint is a periodic endpoint. Low- and full-speed interrupt queue heads require the use of the split transaction protocol. The host controller sets the Endpoint Type (ET) field in the split token to indicate the transaction is an interrupt. These transactions are managed through a transaction translator's periodic pipeline. Software should not set these fields to indicate the queue head is an interrupt unless the queue head is used in the periodic schedule.

System software manages the per/transaction translator periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each endpoint occurs. The characteristics of the transaction translator are such that the high-speed transaction protocol must execute during explicit micro-frames, or the data or response information in the pipeline is lost.

The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule and queue head data structure. The S and ^{C}X labels indicate micro-frames where software can schedule start-splits and complete splits (respectively).

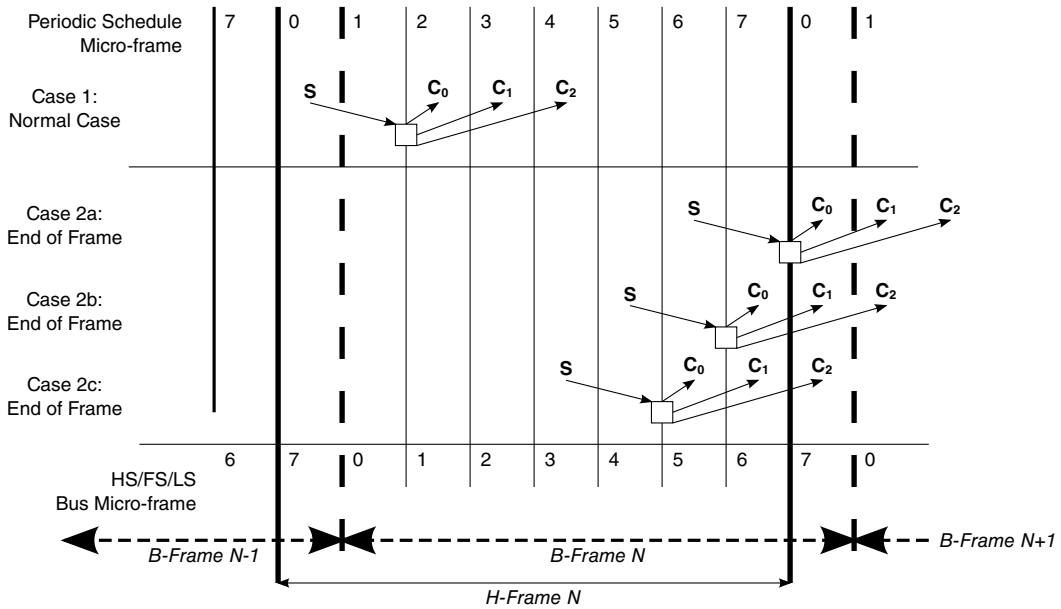


Figure 62-21. Split Transaction, Interrupt Scheduling Boundary Conditions

The scheduling cases are:

- Case 1: The normal scheduling case is where the entire split transaction is completely bounded by a frame (*H-Frame* in this case).
- Case 2a through Case 2c: The USB 2.0 Hub pipeline rules states clearly, when and how many complete-splits must be scheduled to account for earliest to latest execution on the full/low-speed link. The complete-splits may span the *H-Frame* boundary when the start-split is in micro-frame 4 or later. When this occurs, the *H-Frame* to *B-Frame* alignment requires that the queue head be reachable from consecutive periodic frame list locations. System software cannot build an efficient schedule that satisfies this requirement unless it uses FSTNs.

The figure below illustrates the general layout of the periodic schedule.

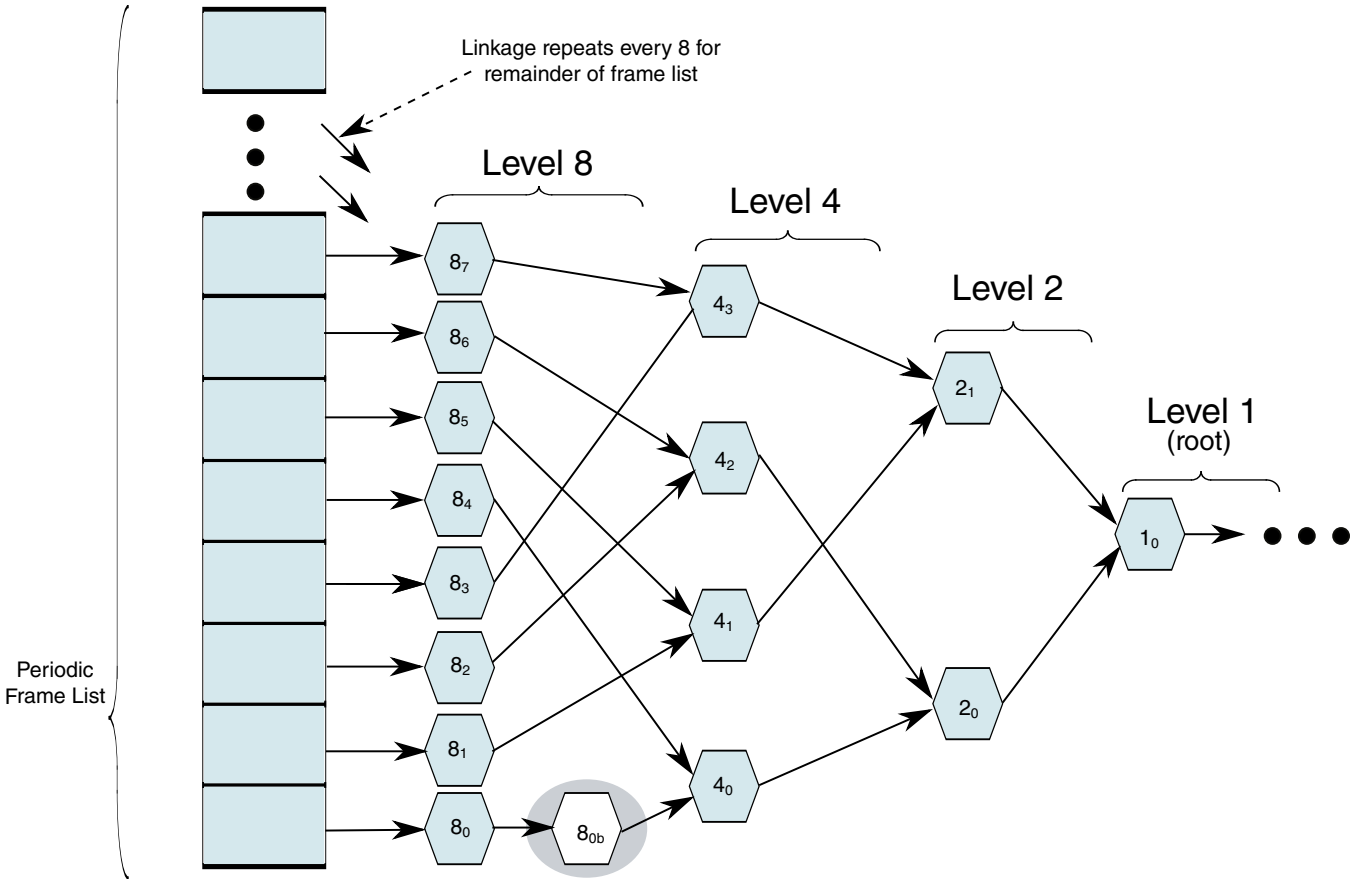


Figure 62-22. General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading

The periodic frame list is effectively the leaf level a binary tree, which is always traversed leaf to root. Each level in the tree corresponds to a 2^N poll rate. Software can efficiently manage periodic bandwidth on the USB by *spreading* interrupt queue heads that have the same poll rate requirement across all the available paths from the frame list. For example, system software can schedule eight poll rate 8 queue heads and account for them once in the high-speed bus bandwidth allocation.

When an endpoint is allocated an execution footprint that spans a frame boundary, the queue head for the endpoint must be reachable from consecutive locations in the frame list. An example would be if 8_{0b} where such an endpoint. Without additional support on the interface, to get 8_{0b} reachable at the correct time, software would have to link 8_1 to 8_{0b}. It would then have to move 4_1 and everything linked after into the same path as 4_0. This upsets the integrity of the binary tree and disallows the use of the spreading technique.

FSTN data structures are used to preserve the integrity of the binary-tree structure and enable the use of the spreading technique. Host Controller Operational Model for FSTNs defines the hardware and software operational model requirements for using FSTNs.

The following queue head fields are initialized by system software to instruct the host controller when to execute portions of the split-transaction protocol:

- *SplitXState*. This is single bit residing in the *Status* field of a queue head (see Table 62-25). This bit is used to track the current state of the split transaction.
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 62-21, case one, the *S-mask* would have a value of 00000001b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Start, and the current micro-frame as indicated by FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 62-21, case one, the *C-mask* would have a value of 00011100b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Complete, and the current micro-frame as indicated by FRINDEX[2:0] is 2, 3, or 4, then execute a complete-split transaction. It is software's responsibility to ensure that the translation between *H-Frames* and *B-Frames* is correctly performed when setting bits in *S-mask* and *C-mask*

62.5.3.12.2.2 Host Controller Operational Model for FSTNs

The FSTN data structure is used to manage Low/Full-speed interrupt queue heads that need to be reached from consecutive frame list locations (that is boundary cases 2a through 2c).

An FSTN is essentially a *back pointer*, similar in intent to the back pointer field in the siTD data structure (see siTD Back Link Pointer).

This feature provides software a simple primitive to save a schedule position, redirect the host controller to traverse the necessary queue heads in the previous frame, then restore the original schedule position and complete normal traversal.

The four components to the use of FSTNs:

- FSTN data structure.
- A *Save Place* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to zero.

- A *Restore* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to one.
- Host controller FSTN traversal rules.

When the host controller encounters an FSTN during micro-frames 2 through 7 it simply follows the node's *Normal Path Link Pointer* to access the next schedule data structure.

NOTE

The FSTN's *Normal Path Link Pointer.T-bit* may set to one, which the host controller must interpret as the end of periodic list mark.

When the host controller encounters a *Save-Place* FSTN in micro-frames 0 or 1, it saves the value of the *Normal Path Link Pointer* and set an internal flag indicating that it is executing in *Recovery Path* mode. *Recovery Path* mode modifies the host controller's rules for how it traverses the schedule and limits which data structures is considered for execution of bus transactions. The host controller continues executing in *Recovery Path* mode until it encounters a *Restore* FSTN or it determines that it has reached the end of the micro-frame (see details in the list below).

The rules for schedule traversal and limited execution while in *Recovery Path* mode are:

- Always follow the *Normal Path Link Pointer* when it encounters an FSTN that is a *Save-Place* indicator. The host controller must not recursively follow *Save-Place* FSTNs. Therefore, while executing in *Recovery Path* mode, it must never follow an FSTN's *Back Path Link Pointer*.
- Do not process an siTD or, iTD data structure. Simply follow its *Next Link Pointer*.
- Do not process a QH (Queue Head) whose *EPS* field indicates a high-speed device. Simply follow its *Horizontal Link Pointer*.
- When a QH's *EPS* field indicates a Full/Low-speed device, the host controller considers only it for execution if its *SplitXState* is DoComplete (note: this applies whether the *PID Code* indicates an IN or an OUT). See Execute Transaction and Tracking Split Transaction Progress for Interrupt Transfers for a complete list of additional conditions that must be met in general for the host controller to issue a bus transaction.
  - The host controller must not execute a Start-split transaction while executing in *Recovery Path* mode. See Periodic Isochronous - Do Complete Split for special handling when in *Recovery Path* mode.
- Stop traversing the *recovery path* when it encounters an FSTN that is a *Restore* indicator. The host controller unconditionally uses the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* when returning to the normal path traversal. The

host controller must clear the context of executing a *Recovery Path* when it restores schedule traversal to the *Save-Place* FSTN's *Normal Path Link Pointer*.

- If the host controller determines that there is not enough time left in the micro-frame to complete processing of the periodic schedule, it abandons traversal of the recovery path, and clears the context of executing a recovery path. The result is that at the start of the next consecutive micro-frame, the host controller starts traversal at the frame list.

An example traversal of a periodic schedule that includes FSTNs is illustrated in the following figure.

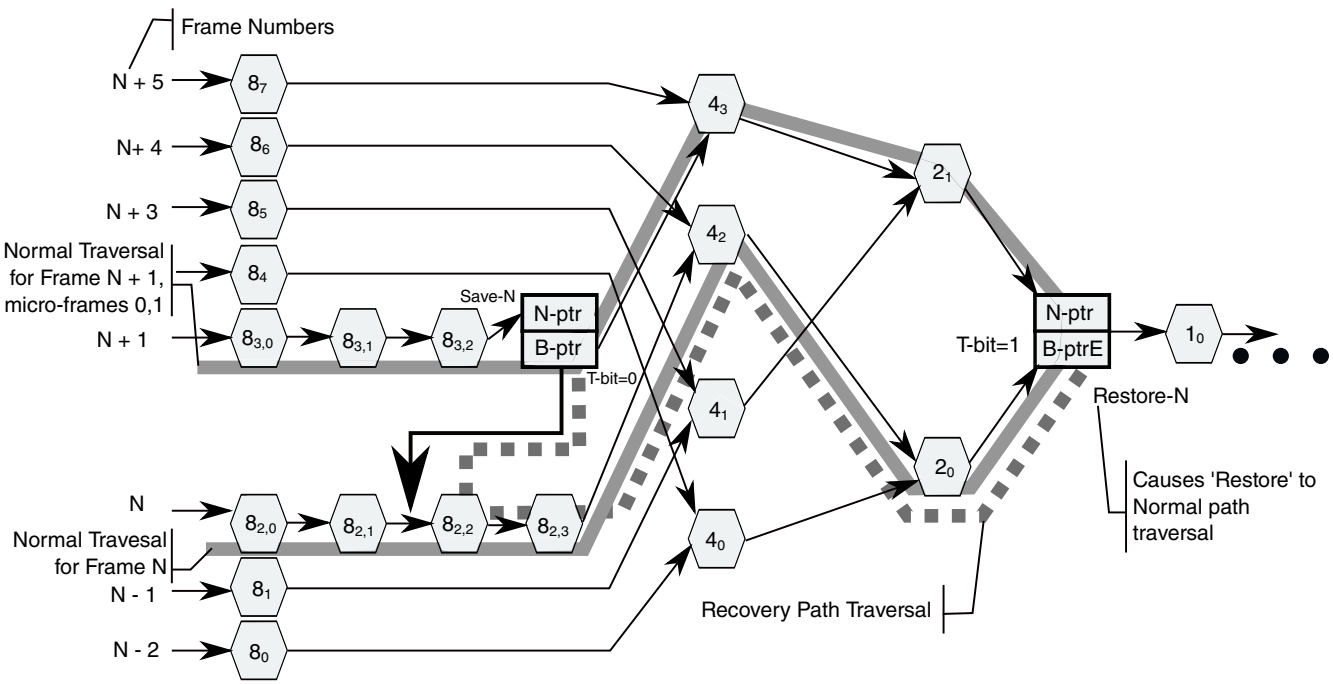


Figure 62-23. Example Host Controller Traversal of Recovery Path via FSTNs

In frame N+1 (micro-frames 0 and 1), when the host controller encounters Save-Path FSTN (Save-N), it observes that Save-N.Back Path Link Pointer.T-bit is zero (definition of a Save-Path indicator). The host controller saves the value of Save-N.Normal Path Link Pointer and follows Save-N.Back Path Link Pointer. At the same time, it sets an internal flag indicating that it is now in *Recovery Path* mode (the recovery path is annotated in the figure above with a large dashed line). The host controller continues traversing data structures on the recovery path and executing only those bus transactions as noted above, on the recovery path until it reaches Restore FSTN (Restore-N). Restore-N.Back Path Link Pointer.T-bit is set to one (definition of a Restore indicator), so the host controller exits *Recovery Path* mode by clearing the internal *Recovery Path* mode flag and commences (restores) schedule traversal using the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* (for example Save-N.Normal Path Link Pointer). The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, $8_{2.2}$, $8_{2.3}$, 4_2,

2_0, Restore-N, 4_3, 2_1, Restore-N, 1_0 …}. The nodes on the recovery-path are in bold. In frame N (micro-frames 0-7), for this example, the host controller traverses all of the schedule data structures utilizing the *Normal Path Link Pointers* in any FSTNs it encounters. This is because the host controller has not yet encountered a *Save-Place* FSTN so it not executing in *Recovery Path* mode. When it encounters the *Restore* FSTN, (Restore-N), during micro-frames 0 and 1, it uses Restore-N.Normal Path Link Pointer to traverse to the next data structure (that is normal schedule traversal). This is because the host controller must use a Restore FSTN's *Normal Path Link Pointer* when not executing in a *Recovery-Path* mode. The nodes traversed during frame N include: {$8_{2.0}$, $8_{2.1}$, $8_{2.2}$, $8_{2.3}$, 4_2, 2_0, Restore-N, 1_0 …}.

In frame N+1 (micro-frames 2-7), when the host controller encounters Save-Path FSTN Save-N, it unconditionally follows Save-N.Normal Path Link Pointer. The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, 4_3, 2_1, Restore-N, 1_0 …}.

62.5.3.12.2.3 Software Operational Model for FSTNs

Software must create a consistent, coherent schedule for the host controller to traverse.

When using FSTNs, system software must adhere to the following rules:

- Each *Save-Place* indicator requires a matching *Restore* indicator.
  - The *Save-Place* indicator is an FSTN with a valid *Back Path Link Pointer and T-bit* equal to zero.
    - *Back Path Link Pointer.Typ* field must be set to indicate the referenced data structure is a queue head. The *Restore* indicator is an FSTN with its *Back Path Link Pointer.T-bit* set to one.
  - A *Restore* FSTN may be matched to one or more *Save-Place* FSTNs. For example, if the schedule includes a poll-rate 1 level, then system software only needs to place a *Restore* FSTN at the beginning of this list in order to match all possible *Save-Place* FSTNs.
- If the schedule does not have elements linked at a poll-rate level of one, and one or more *Save-Place* FSTNs are used, then System Software must ensure the *Restore* FSTN's *Normal Path Link Pointer*'s *T-bit* is set to one, as this is used to mark the end of the periodic list.
- When the schedule does have elements linked at a poll rate level of one, a *Restore* FSTN must be the first data structure on the poll rate one list. All traversal paths from the frame list converge on the poll-rate one list. System software must ensure that *Recovery Path* mode is exited before the host controller is allowed to traverse the poll rate level one list.
- A *Save-Place* FSTN's *Back Path Link Pointer* must reference a queue head data structure. The referenced queue head must be reachable from the previous frame list

location. In other words, if the *Save-Place* FSTN is reachable from frame list offset N, then the FSTN's *Back Path Link Pointer* must reference a queue head that is reachable from frame list offset N-1.

Software should make the schedule as efficient as possible. What this means in this context is that software should have no more than one *Save-Place* FSTN reachable in any single frame. Note there is times when two (or more, depending on the implementation) could exist as full/low-speed footprints change with bandwidth adjustments. This could occur, for example when a bandwidth re-balance causes system software to move the *Save-Place* FSTN from one poll rate level to another. During the transition, software must preserve the integrity of the previous schedule until the new schedule is in place.

62.5.3.12.2.4 Tracking Split Transaction Progress for Interrupt Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where data is lost.

For interrupt-IN transfers, data is lost when it makes it into the USB 2.0 hub, but the USB 2.0 host system is unable to get it from the USB 2.0 Hub and into the system before it expires from the transaction translator pipeline.

When a lost data condition is detected, the queue must be halted, thus signaling system software to recover from the error. A data-loss condition exists whenever a start-split is issued, accepted and successfully executed by the USB 2.0 Hub, but the complete-splits get unrecoverable errors on the high-speed link, or the complete-splits do not occur at the correct times. One reason complete-splits might not occur at the right time would be due to host-induced system hold-offs that cause the host controller to miss bus transactions because it cannot get timely access to the schedule in system memory.

The same condition can occur for an interrupt-OUT, but the result is not an endpoint halt condition, but rather effects only the progress of the transfer. The queue head has the following fields to track the progress of each split transaction. These fields are used to keep incremental state about which (and when) portions have been executed.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets one of the *C-prog*-mask bits for each complete-split executed. The bit position is determined by the micro-frame number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a

complete-split transaction. If the previous complete-splits have not been executed then it means one (or more) have been skipped and data has potentially been lost.
- *FrameTag*. This field is used by the host controller during the complete-split portion of the split transaction to tag the queue head with the frame number (*H-Frame* number) when the next complete split must be executed.
- *S-bytes*. This field can be used to store the number of data payload bytes sent during the start-split (if the transaction was an OUT). The *S-bytes* field must be used to accumulate the data payload bytes received during the complete-splits (for an IN).

62.5.3.12.2.5 Split Transaction Execution State Machine for Interrupt

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

As with asynchronous Full- and Low-speed endpoints, a split-transaction state machine is used to manage the split transaction sequence.

Aside from the fields defined in the queue head for scheduling and tracking the split transaction, the host controller calculates one internal mechanism that is also used to manage the split transaction. The internal calculated mechanism is:

- *cMicroFrameBit* is a single-bit encoding of the current micro-frame number. It is an eight-bit value calculated by the host controller at the beginning of every micro-frame. It is calculated from the three least significant bits of the *FRINDEX* register (that is, *cMicroFrameBit* = (1 shifted-left(*FRINDEX*[2:0])). The *cMicroFrameBit* has at most one bit asserted, which always corresponds to the current micro-frame number. For example, if the current micro-frame is 0, then *cMicroFrameBit* will equal 00000001b. The variable *cMicroFrameBit* is used to compare against the *S-mask* and *C-mask* fields to determine whether the queue head is marked for a start- or complete-splt transaction for the current micro-frame.

The following figure illustrates the state machine for managing a complete interrupt split transaction. There are two phases to each split transaction. The first is a single start-split transaction, which occurs when the *SplitXState* is at Do_Start and the single bit in *cMicroFrameBit* has a corresponding bit active in *QH.S-mask*. The transaction translator does not acknowledge the receipt of the periodic start-split, so the host controller unconditionally transitions the state to Do_Complete. Due to the available jitter in the transaction translator pipeline, there will be more than one complete-split transaction scheduled by software for the Do_Complete state. This translates simply to the fact that there are multiple bits set to a one in the *QH.C-mask* field.

The host controller keeps the queue head in the Do_Complete state until the split transaction is complete (see definition below), or an error condition triggers the *three-strikes-rule* (for example, after the host tries the same transaction three times, and each

encounters an error, the host controller will stop retrying the bus transaction and halt the endpoint, thus requiring system software to detect the condition and perform system-dependent recovery).

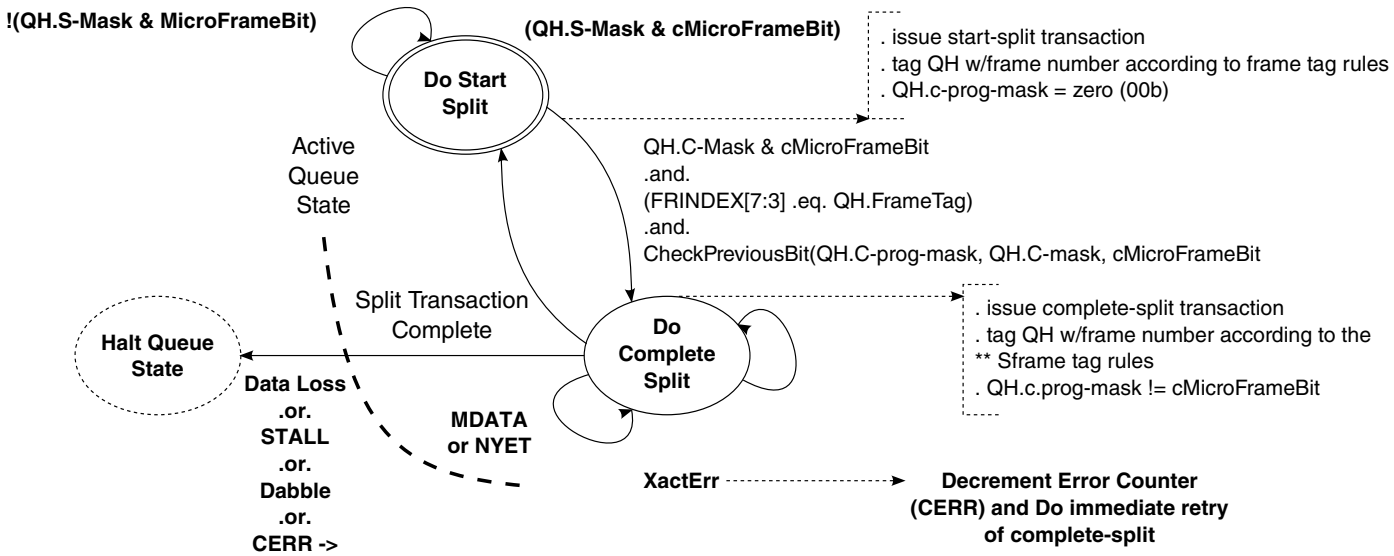


Figure 62-24. Split Transaction State Machine for Interrupt

See Previous Section for the frame tag management rules.

Periodic Interrupt - Do Start Split

This is the state software must initialize a full- or low-speed interrupt queue head *StartXState* bit. This state is entered from the Do_Complete Split state only after the split transaction is complete. This occurs when one of the following events occur: The transaction translator responds to a complete-split transaction with one of the following:

- NAK. A NAK response is a propagation of the full- or low-speed endpoint's NAK response.
- ACK. An ACK response is a propagation of the full- or low-speed endpoint's ACK response. Only occurs on an OUT endpoint.
- DATA 0/1. Only occurs for INs. Indicates that this is the last of the data from the endpoint for this split transaction.
- ERR. The transaction on the low-/full-speed link below the transaction translator had a failure (for example, timeout, bad CRC, etc.).
- NYET (and Last). The host controller issued the last complete-split and the transaction translator responded with a NYET handshake. This means that the start-split was not correctly received by the transaction translator, so it never executed a transaction to the full- or low-speed endpoint, see Section Periodic Isochronous - Do Complete Split for the definition of 'Last'.

Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it performs the following test to determine whether to execute a start-split.

- *QH.S-mask* is bit-wise anded with *cMicroFrameBit*.

If the result is non-zero, then the host controller will issue a start-split transaction. If the *PIDCode* field indicates an IN transaction, the host controller must zero-out the *QH.S-bytes* field. After the split-transaction has been executed, the host controller sets up state in the queue head to track the progress of the complete-split phase of the split transaction. Specifically, it records the expected frame number into *QH.FrameTag* field (see Section), set *C-prog-mask* to zero (00h), and exits this state. Note that the host controller must not adjust the value of *CErr* as a result of completion of a start-split transaction.

Periodic Interrupt - Do Complete Split

This state is entered unconditionally from the Do Start Split state after a start-split transaction is executed on the bus. Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it checks to determine whether a complete-split transaction should be executed now.

There are four tests to determine whether a complete-split transaction should be executed.

- Test A. *cMicroFrameBit* is bit-wise anded with *QH.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame.
- Test B. *QH.FrameTag* is compared with the current contents of *FRINDEX*[7:3]. An equal indicates a match.
- Test C. The complete-split progress bit vector is checked to determine whether the previous bit is set, indicating that the previous complete-split was appropriately executed. An example algorithm for this test is provided below:

```
Algorithm Boolean CheckPreviousBit(QH.C-prog-mask, QH.C-mask, cMicroFrameBit)
Begin
-- Return values:
-- TRUE - no error
-- FALSE - error
--
Boolean rvalue = TRUE;
previousBit = cMicroframeBit logical-rotate-right(1)
-- Bit-wise anding previousBit with C-mask indicates
-- whether there was an intent
-- to send a complete split in the previous micro-frame. So,
-- if the
-- 'previous bit' is set in C-mask, check C-prog-mask to
-- make sure it
-- happened.
If (previousBit bitAND QH.C-mask)then
                                If not(previousBit bitAND QH.C-prog-mask) then
                      rvalue = FALSE;
          End if
End If
-- If the C-prog-mask already has a one in this bit position,
```

```
-- then an aliasing
-- error has occurred. It will probably get caught by the
-- FrameTag Test, but
-- at any rate it is an error condition that as detectable here
-- should not allow
-- a transaction to be executed.
If (cMicroFrameBit bitAND QH.C-prog-mask) then
 rvalue = FALSE;
End if
return (rvalue)
End Algorithm
```

- Test D. Check to see if a start-split should be executed in this micro-frame. Note this is the same test performed in the Do Start Split state (see Section Periodic Isochronous - Do Start Split). Whenever it evaluates to TRUE and the controller is NOT processing in the context of a *Recovery Path* mode, it means a start-split should occur in this micro-frame. Test D and Test A evaluating to TRUE at the same time is a system software error. Behavior is undefined.

If (A .and. B .and. C .and. not(D)) then the host controller will execute a complete-split transaction. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. On completion of the complete-split transaction, the host controller records the result of the transaction in the queue head and sets *QH.FrameTag* to the expected *H-Frame* number (see Section). The effect to the state of the queue head and thus the state of the transfer depends on the response by the transaction translator to the complete-split transaction. The following responses have the effects (note that any responses that result in decrementing of the *CErr* will result in the queue head being halted by the host controller if the result of the decrement is zero):

- NYET (and Last). On each NYET response, the host controller checks to determine whether this is the last complete-split for this split transaction. Last is defined in this context as the condition where all of the scheduled complete-splits have been executed. If it is the last complete-split (with a NYET response), then the transfer state of the queue head is not advanced (never received any data) and this state exited. The transaction translator must have responded to all the clompete-splits with NYETs, meaning that the start-split issued by the host controller was not received. The start-split should be retried at the next poll period.
- The test for whether this is the Last complete split can be performed by XOR *QH.C-mask* with *QH.C-prog-mask*. If the result is all zeros then all complete-splits have been executed. When this condition occurs, the *XactErr* status bit is set to a one and the *CErr* field is decremented.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask* and *FrameTag*) and stay in this state. The host controller must not adjust *CErr* on this response.

- Transaction Error (XactErr). Timeout, data CRC failure, etc. The *CErr* field is decremented and the *XactErr* bit in the *Status* field is set to a one. The complete split transaction is *immediately* retried (if *Cerr* is non-zero).If there is not enough time in the micro-frame to complete the retry and the endpoint is an IN, or *CErr* is decremented to a zero from a one, the queue is halted. If there is not enough time in the micro-frame to complete the retry and the endpoint is an OUT and *CErr* is not zero, then this state is exited (that is, return to Do Start Split). This results in a retry of the entire OUT split transaction, at the next poll period. Refer to Chapter 11 Hubs (specifically the section full- and low-speed Interrupts) in the USB Specification Revision 2.0 for detailed requirements on why these errors must be immediately retried.
- ACK. This can only occur if the target endpoint is an OUT. The target endpoint ACK'd the data and this response is a propagation of the endpoint ACK up to the host controller. The host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount. And the data toggle bit (*dt*) is toggled. The host controller will then exit this state for this queue head. The host controller must reload *CErr* with maximum value on this response. Advancing the transfer state may cause other process events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- MDATA. This response will only occur for an IN endpoint. The transaction translator responded with zero or more bytes of data and an MDATA PID. The incremental number of bytes received is accumulated in *QH.S-bytes*. The host controller must not adjust *CErr* on this response.
- DATA0/1. This response may only occur for an IN endpoint. The number of bytes received is added to the accumulated byte count in *QH.S-bytes*. The state of the transfer is advanced by the result and the host controller will exit this state for this queue head.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- If the data sequence PID does not match the expected, the entirety of the data received in this split transaction is ignored, the transfer state is not advanced and this state is exited.
- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced, and this state is exited. The host controller must reload *CErr* with maximum value on this response.

- ERR. There was an error during the full- or low-speed transaction. The ERR status bit is set to a one, *Cerr* is decremented, the state of the transfer is not advanced, and this state is exited.
- STALL. The queue is halted (an exit condition of the Execute Transaction state). The status field bits: *Active* bit is set to zero and the *Halted* bit is set to a one and the qTD is retired. Responses which are not enumerated in the list or which are received out of sequence are illegal and may result in undefined host controller behavior. The other possible combinations of tests A, B, C, and D may indicate that data or response was lost. The table below lists the possible combinations and the appropriate action.

Table 62-49. Interrupt IN/OUT Do Complete Split State Execution Criteria

Condition	Action	Description
not(A) not(D)	Ignore QHD	Neither a start nor complete-split is scheduled for the current micro-frame.Host controller should continue walking the schedule.
A not(C)	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	Progress bit check failed. These means a complete-split has been missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A not(B) C	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	*QH.FrameTag* test failed. This means that exactly one or more *H-Frames* have been skipped. This means complete-splits and have missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A B C not(D)	Execute complete-split	This is the non-error case where the host controller executes a complete-split transaction.
D	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	This is a degenerate case where the start-split was issued, but all of the complete-splits were skipped and all possible intervening opportunities to detect the missed data failed to fire. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one. Note: When executing in the context of a *Recovery Path* mode, the host controller is allowed to process the queue head and take the actions indicated above, or it may wait until the queue head is visited in the

Table 62-49. Interrupt IN/OUT Do Complete Split State Execution Criteria

		normal processing mode. Regardless, the host controller must not execute a start-split in the context of a executing in a *Recovery Path* mode.

Managing QH.FrameTag Field

The *QH.FrameTag* field in a queue head is completely managed by the host controller. The rules for setting *QH.FrameTag* are simple:

- Rule 1: If transitioning from Do Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is 6 *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates split transactions whose start-split and complete-splits are in different *H-Frames* (case 2a, see Figure 62-21).
- Rule 2: If the current value of *FRINDEX*[2:0] is 7, *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates staying in Do Complete Split for cases 2a, 2b, and 2c (Figure 62-21).
- Rule 3: If transitioning from Do_Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is not 6, or currently in Do Complete Split and the current value of (*FRINDEX*[2:0]) is not 7, *FrameTag* is set to *FRINDEX*[7:3]. This accommodates all other cases (Figure 62-21).

62.5.3.12.2.6 Rebalancing the periodic schedule

System software must occasionally adjust a periodic queue head's S-mask and C-mask fields during operation.

This need occurs when adjustments to the periodic schedule create a new bandwidth budget and one or more queue head's are assigned new execution footprints (that is, new S-mask and C-mask values).

It is imperative that System software must not update these masks to new values in the midst of a split transaction. In order to avoid any race conditions with the update, the EHCI host controller provides a simple assist to system software. System software sets the *Inactivate-on-next-Transaction* (*I*) bit to a one to signal the host controller that it intends to update the S-mask and C-mask on this queue head. System software will then wait for the host controller to observe the *I-bit* is a one and transition the *Active* bit to a zero. The rules for how and when the host controller sets the *Active* bit to zero are enumerated below:

- If the *Active* bit is a zero, no action is taken. The host controller does not attempt to advance the queue when the *I-bit* is a one.
- If the *Active* bit is a one and the *SplitXState* is DoStart (regardless of the value of *S-mask*), the host controller will simply set *Active* bit to a zero. The host controller is

not required to write the transfer state back to the *current* qTD. Note that if the *S-mask* indicates that a start-split is scheduled for the current micro-frame, the host controller must not issue the start-split bus transaction. It must set the *Active* bit to zero.

System software must save transfer state before setting the *I-bit* to a one. This is required so that it can correctly determine what transfer progress (if any) occurred after the *I-bit* was set to a one and the host controller executed its final bus-transaction and set *Active* to a zero.

After system software has updated the S-mask and C-mask, it must then reactivate the queue head. Because the *Active* bit and the *I-bit* cannot be updated with the same write, system software needs to use the following algorithm to coherently re-activate a queue head that has been stopped via the *I-bit*.

1. Set the *Halted* bit to a one, then
2. Set the *I-bit* to a zero, then
3. Set the *Active* bit to a one and the *Halted* bit to a zero in the same write.

Setting the *Halted* bit to a one inhibits the host controller from attempting to advance the queue between the time the *I-bit* goes to a zero and the *Active* bit goes to a one.

62.5.3.12.3 Split Transaction Isochronous

Full-speed isochronous transfers are managed using the split-transaction protocol through a USB 2.0 transaction translator in a USB2.0 Hub. The EHCI controller utilizes siTD data structure to support the special requirements of isochronous split-transactions.

This data structure uses the scheduling model of isochronous TDs (iTD, Section Isochronous (High-Speed) Transfer Descriptor (iTD)) (see Section Managing Isochronous Transfers Using iTDs for the operational model of iTDs) with the contiguous data feature provided by queue heads. This simple arrangement allows a single isochronous scheduling model and adds the additional feature that all data received from the endpoint (per split transaction) must land into a contiguous buffer.

62.5.3.12.3.1 Split Transaction Scheduling Mechanisms for Isochronous

Full-speed isochronous transactions are managed through a transaction translator's periodic pipeline. As with full- and low-speed interrupt, system software manages each transaction translator's periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each full-speed isochronous endpoint occur.

The requirements described in Section Split Transaction Scheduling Mechanisms for Interrupt apply. The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule. The ^{S}X and ^{C}X labels indicate micro-frames where software can schedule start- and complete-splits (respectively). The *H-Frame* boundaries are marked with a large, solid bold vertical line. The *B-Frame* boundaries are marked with a large, bold, dashed line. The bottom of the figure illustrates the relationship of an siTD to the *H-Frame*.

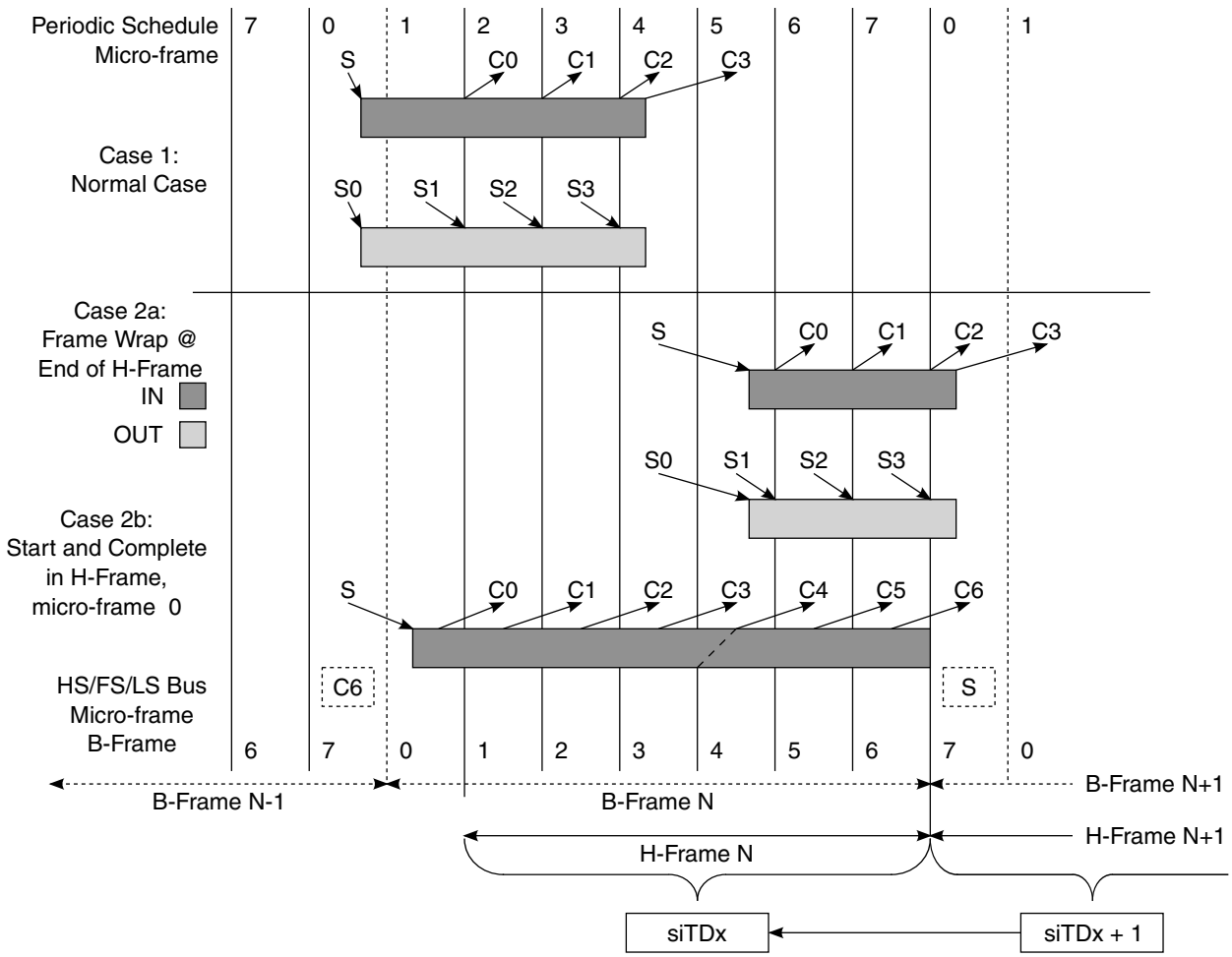


Figure 62-25. Split Transaction, Isochronous Scheduling Boundary Conditions

When the endpoint is an isochronous OUT, there are only start-splits, and no complete-splits. When the endpoint is an isochronous IN, there is at most one start-split and one to *N* complete-splits. The scheduling boundary cases are:

- *Case 1*: The entire split transaction is completely bounded by an *H-Frame*. For example: the start-splits and complete-splits are all scheduled to occur in the same *H-Frame*.

- *Case 2a*: This boundary case is where one or more (at most two) complete-splits of a split transaction IN are scheduled across an *H-Frame* boundary. This can only occur when the split transaction has the possibility of moving data in *B-Frame,* micro-frames 6 or 7 (*H-Frame* micro-frame 7 or 0). When an *H-Frame* boundary wrap condition occurs, the scheduling of the split transaction spans more than one location in the periodic list.(For example, it takes two siTDs in adjacent periodic frame list locations to fully describe the scheduling for the split transaction.)
- Although the scheduling of the split transaction may take two data structures, all of the complete-splits for each full-speed IN isochronous transaction must use only one data pointer. For this reason, siTDs contain a back pointer, the use of which is described below.
- Software must never schedule full-speed isochronous OUTs across an *H-Frame* boundary.
- *Case 2b*: This case can only occur for a very large isochronous IN. It is the only allowed scenario where a start-split and complete-split for the same endpoint can occur in the same micro-frame. Software must enforce this rule by scheduling the large transaction first. Large is defined to be anything larger than 579 byte maximum packet size.

A subset of the same mechanisms employed by full- and low-speed interrupt queue heads are employed in siTDs to schedule and track the portions of isochronous split transactions. The following fields are initialized by system software to instruct the host controller when to execute portions of the split transaction protocol.

- *SplitXState*. This is a single bit residing in the *Status* field of an siTD (see Figure 62-26). This bit is used to track the current state of the split transaction. The rules for managing this bit are described in Section Split Transaction Execution State Machine for Interrupt.
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 62-25, case one, the *S-mask* would have a value of 00000001b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Start Split, and the current micro-frame as indicated by USB_n_FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 62-25, case one, the *C-mask* would have a value of 00111100b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do

Complete Split, and the current micro-frame as indicated by *USB_n_FRINDEX*[2:0] is 2, 3, 4, or 5, then execute a complete-split transaction.

- *Back Pointer*. This field in a siTD is used to complete an IN split-transaction using the previous *H-Frame*'s siTD. This is only used when the scheduling of the complete-splits span an *H-Frame* boundary.

There exists a one-to-one relationship between a high-speed isochronous split transaction (including all start- and complete-splits) and one full-speed isochronous transaction. An siTD contains (amongst other things) buffer state and split transaction scheduling information. An siTD's buffer state always maps to one full-speed isochronous data payload. This means that for any full-speed transaction payload, a single siTD's data buffer must be used. This rule applies to both IN an OUTs. An siTD's scheduling information usually also maps to one high-speed isochronous split transaction. The exception to this rule is the *H-Frame* boundary wrap cases mentioned above.

The siTD data structure describes at most, one frame's worth of high-speed transactions and that description is strictly bounded within a frame boundary. The figure below illustrates some examples. On the top are examples of the full-speed transaction footprints for the boundary scheduling cases described above. In the middle are time-frame references for both the *B-Frames* (HS/FS/LS Bus) and the *H-Frames*. On the bottom is illustrated the relationship between the scope of an siTD description and the time references. Each *H-Frame* corresponds to a single location in the periodic frame list. The implication is that each siTD is reachable from a single periodic frame list location at a time.

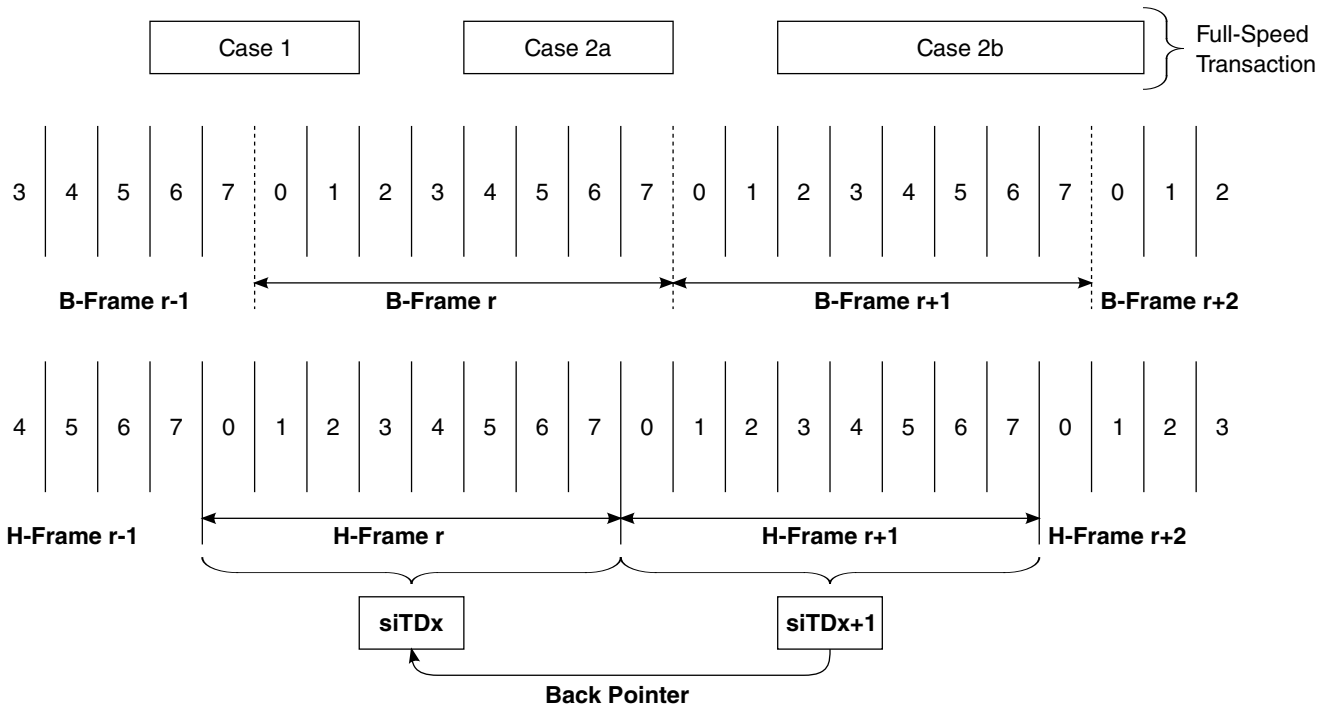


Figure 62-26. siTD Scheduling Boundary Examples

Each case is described below:

- *Case 1*: One siTD is sufficient to describe and complete the isochronous split transaction because the whole isochronous split transaction is tightly contained within a single *H-Frame*.
- *Case 2a, 2b*: Although both INs and OUTs can have these footprints, OUTs always take only one siTD to schedule. However, INs (for these boundary cases) require two siTDs to complete the scheduling of the isochronous split transaction $siTD_X$ is used to always issue the start-split and the first *N* complete-splits. The full-speed transaction (for these cases) can deliver data on the full-speed bus segment during micro-frame 7 of $H\text{-}Frame_{Y+1}$, or micro-frame 0 of $H\text{-}Frame_{Y+2}$. The complete splits are scheduled using $siTD_{X+2}$ (not shown). The complete-splits to extract this data must use the buffer pointer from $siTD_{X+1}$. The only way for the host controller to reach $siTD_{X+1}$ from $H\text{-}Frame_{Y+2}$is to use $siTD_{X+2}$'s back pointer. The host controller rules for when to use the back pointer are described is Section Periodic Isochronous - Do Complete Split .

Software must apply the following rules when calculating the schedule and linking the schedule data structures into the periodic schedule:

- Software must ensure that an isochronous split-transaction is started so that it will complete before the end of the *B-Frame*.
- Software must ensure that for a single full-speed isochronous endpoint, there is never a start-split and complete-split in *H-Frame, micro-frame 1*. This is mandated as a rule so that case 2a and case 2b can be discriminated. According to the core USB specification, the long isochronous transaction illustrated in Case 2b, could be scheduled so that the start-split was in micro-frame 1 of *H-Frame* N and the last complete-split would need to occur in micro-frame 1 of H-Frame N+1. However, it is impossible to discriminate between cases 2a and case 2b, which has significant impact on the complexity of the host controller.

62.5.3.12.3.2 Tracking Split Transaction Progress for Isochronous Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where device to host data is lost. Isochronous endpoints do not employ the concept of a halt on error, however the host is required to identify and report per-packet errors observed in the data stream. This includes schedule traversal problems (skipped micro-frames), timeouts and corrupted data received.

In similar kind to interrupt split-transactions, the portions of the split transaction protocol must execute in the micro-frames they are scheduled. The queue head data structure used to manage full- and low-speed interrupt has several mechanisms for tracking when portions of a transaction have occurred. Isochronous transfers use siTDs, for their transfers, and the data structures are only reachable via the schedule in the exact micro-frame in which they are required (so all the mechanism employed for tracking in queue heads is not required for siTDs). Software has the option of reusing siTD several times in the complete periodic schedule. However, it must ensure that the results of split transaction N are consumed and the siTD reinitialized (activated) before the host controller gets back to the siTD (in a future micro-frame).

Split-transaction isochronous OUTs utilize a low-level protocol to indicate which portions of the split transaction data have arrived. Control over the low-level protocol is exposed in an siTD via the fields *Transaction Position* (*TP*) and *Transaction Count* (*T-count*). If the entire data payload for the OUT split transaction is larger than 188 bytes, there will be more than one start-split transaction, each of which require proper annotation. If host hold-offs occur, then the sequence of annotations received from the host will not be complete, which is detected and handled by the transaction translator. See Section Periodic Isochronous - Do Start Split for a description on how these fields are used during a sequence of start-split transactions.

The fields *siTD.T-Count* and *siTD.TP* are used by the host controller to drive and sequence the transaction position annotations. It is the responsibility of system software to properly initialize these fields in each siTD. Once the budget for a split-transaction

isochronous endpoint is established, *S-mask*, *T-Count*, and *TP* initialization values for all the siTD associated with the endpoint are constant. They remain constant until the budget for the endpoint is recalculated by software and the periodic schedule adjusted.

For IN-endpoints, the transaction translator simply annotates the response data packets with enough information to allow the host controller to identify the last data. As with split transaction Interrupt, it is the host controller's responsibility to detect when it has missed an opportunity to execute a complete-split. The following field in the siTD is used to track and detect errors in the execution of a split transaction for an IN isochronous endpoint.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets a bit for each complete-split executed. The bit position is determined by the micro-frame (USB_n_FRINDEX[2:0]) number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a complete-split transaction. If the previous complete-splits have not been executed, then it means one (or more) have been skipped and data has potentially been lost. System software is required to initialize this field to zero before setting an siTD's *Active* bit to a one.

If a transaction translator returns with the final data before all of the complete-splits have been executed, the state of the transfer is advanced so that the remaining complete-splits are not executed. Refer to Section Asynchronous - Do Complete Split for a description on how the state of the transfer is advanced. It is important to note that an IN siTD is retired based solely on the responses from the Transaction Translator to the complete-split transactions. This means, for example, that it is possible for a transaction translator to respond to a complete-split with an MDATA PID. The number of bytes in the MDATA's data payload could cause the siTD field *Total Bytes to Transfer* to decrement to zero. This response can occur, before all of the scheduled complete-splits have been executed. In other interface, data structures (for example, high-speed data streams through queue heads), the transition of *Total Bytes to Transfer* to zero signals the end of the transfer and results in setting of the *Active* bit to zero. However, in this case, the result has not been delivered by the Transaction Translator and the host must continue with the next complete-split transaction to extract the residual transaction state. This scenario occurs because of the pipeline rules for a Transaction Translator (see Chapter 11 of the Universal Serial Bus Revision 2.0). In summary the periodic pipeline rules require that on a micro-frame boundary, the Transaction Translator will hold the final two bytes received (if it has not seen an End Of Packet (EOP)) in the full-speed bus pipe stage and give the

remaining bytes to the high-speed pipeline stage. At the micro-frame boundary, the Transaction Translator could have received the entire packet (including both CRC bytes) but not received the packet EOP. In the next micro-frame, the Transaction Translator will respond with an MDATA and send all of the data bytes (with the two CRC bytes being held in the full-speed pipeline stage). This could cause the siTD to decrement its *Total Bytes to Transfer* field to zero, indicating it has received all expected data. The host must still execute one more (scheduled) complete-split transaction in order to extract the results of the full-speed transaction from the Transaction Translator (for example, the Transaction Translator may have detected a CRC failure, and this result must be forwarded to the host).

If the host experiences hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous OUT, then the protocol to the transaction translator will not be consistent and the transaction translator will detect and react to the problem. Likewise, for host hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous IN, the *C-prog-mask* is used by the host controller to detect errors. However, if the host experiences a hold-off that causes it to skip all of an siTD, or an siTD expires during a host hold off (for example, a hold-off occurs and the siTD is no longer reachable by the host controller in order for it to report the hold-off event), then system software must detect that the siTDs have not been processed by the host controller (that is, state not advanced) and report the appropriate error to the client driver.

62.5.3.12.3.3 Split Transaction Execution State Machine for Isochronous

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

If the *Active* bit in the *Status* byte is a zero, the host controller will ignore the siTD and continue traversing the periodic schedule. Otherwise the host controller will process the siTD as specified below. A split transaction state machine is used to manage the split-transaction protocol sequence. The host controller uses the fields defined in Section Tracking Split Transaction Progress for Interrupt Transfers , plus the variable *cMicroFrameBit* defined in Section Split Transaction Execution State Machine for Interrupt to track the progress of an isochronous split transaction. The figure below illustrates the state machine for managing an siTD through an isochronous split transaction. Bold, dotted circles denote the state of the *Active* bit in the *Status* field of a siTD. The Bold, dotted arcs denote the transitions between these states. Solid circles denote the states of the split transaction state machine and the solid arcs denote the transitions between these states. Dotted arcs and boxes reference actions that take place either as a result of a transition or from being in a state.

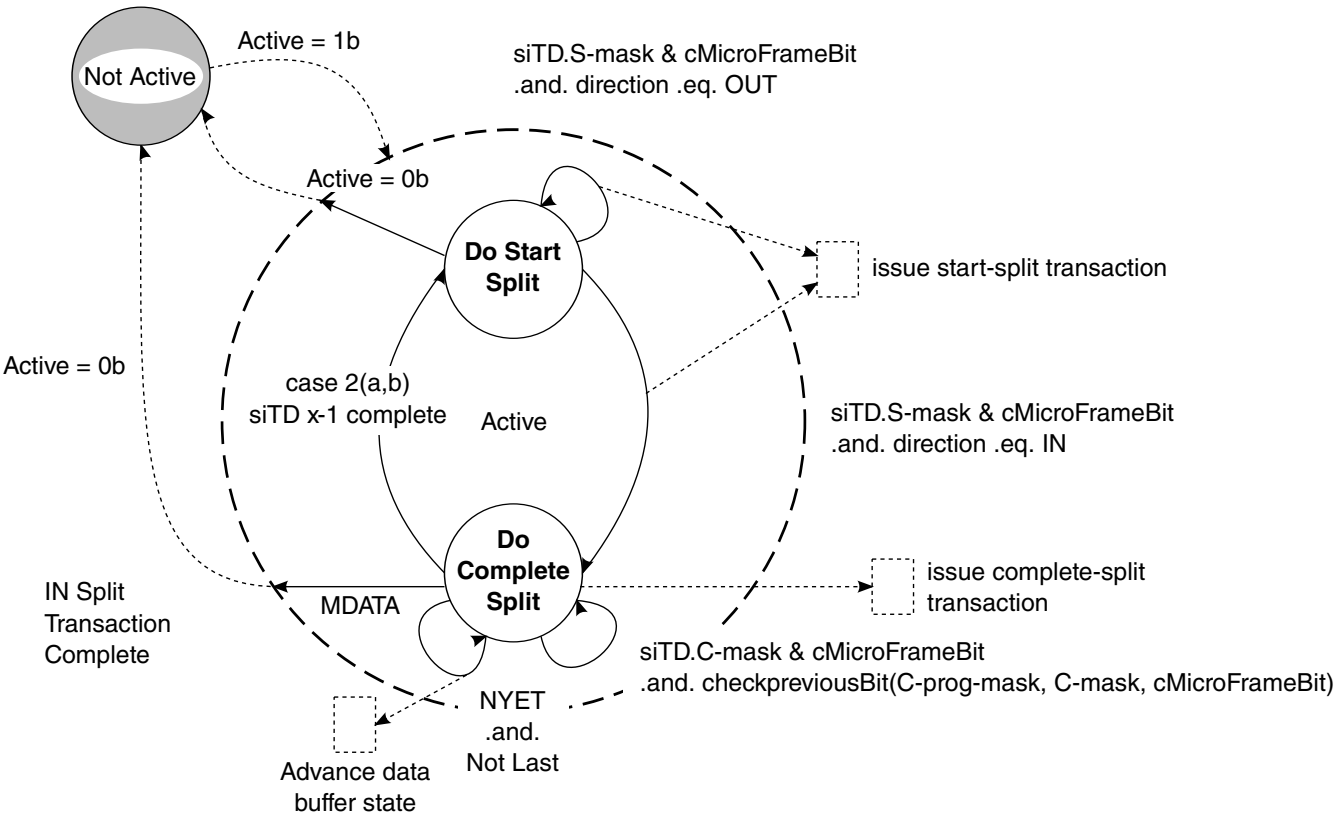


Figure 62-27. Split Transaction State Machine for Isochronous

62.5.3.12.3.4 Periodic Isochronous - Do Start Split

Isochronous split transaction OUTs use only this state.

An siTD for a split-transaction isochronous IN is either initialized to this state, or the siTD transitions to this state from Do Complete Split when a case 2a (IN) or 2b scheduling boundary isochronous split-transaction completes.

Each time the host controller reaches an active siTD in this state, it checks the *siTD.S-mask* against *cMicroFrameBit*. If there is a one in the appropriate position, the siTD will execute a start-split transaction. By definition, the host controller cannot *reach* an siTD at the wrong time. If the *I/O* field indicates an IN, then the start-split transaction includes only the extended token plus the full-speed token. Software must initialize the *siTD.Total Bytes To Transfer* field to the number of bytes expected. This is usually the maximum packet size for the full-speed endpoint. The host controller exits this state when the start-split transaction is complete.

The remainder of this section is specific to an isochronous OUT endpoint (that is, the *I/O* field indicates an OUT). When the host controller executes a start-split transaction for an isochronous OUT it includes a data payload in the start-split transaction. The memory

buffer address for the data payload is constructed by concatenating *siTD.Current Offset* with the page pointer indicated by the page selector field (*siTD.P*). A zero in this field selects Page 0 and a 1 selects Page 1. During the start-split for an OUT, if the data transfer crosses a page boundary during the transaction, the host controller must detect the page cross, update the *siTD.P*-bit from a zero to a one, and begin using the *siTD.Page 1* with *siTD.Current Offset* as the memory address pointer. The field *siTD.TP* is used to annotate each start-split transaction with the indication of which part of the split-transaction data the current payload represents (ALL, BEGIN, MID, END). In all cases the host controller simply uses the value in *siTD.TP* to mark the start-split with the correct transaction position code.

T-Count is always initialized to the number of start-splits for the current frame. *TP* is always initialized to the first required transaction position identifier. The scheduling boundary case (see Figure 62-26) is used to determine the initial value of *TP*. The initial cases are summarized in the following table.

Table 62-50. Initial Conditions for OUT siTD's TP and T-count Fields

Case	T-count	TP	Description
1, 2a	=1	ALL	When the OUT data payload is less than (or equal to) 188 bytes, only one start-split is required to move the data. The one start-split must be marked with an ALL.
1, 2a	!=1	BEGIN	When the OUT data payload is greater than 188 bytes more than one start-split must be used to move the data. The initial start-split must be marked with a BEGIN.

After each start-split transaction is complete, the host controller updates *T-Count* and *TP* appropriately so that the next start-split is correctly annotated.

The table below illustrates all of the *TP* and *T-count* transitions, which must be accomplished by the host controller.

.

Table 62-51. Transaction Position (TP)/Transaction Count (T-Count) Transition Table

TP	T-count next	TP next	Description
ALL	0	N/A	Transition from ALL, to done.
BEGIN	1	END	Transition from BEGIN to END. Occurs when *T-count* starts at 2.
BEGIN	!=1	MID	Transition from BEGIN to MID. Occurs when *T-count* starts at greater than 2.
MID	!=1	MID	*TP* stays at MID while *T-count* is not equal to 1 (that is, greater than 1). This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 3.
MID	1	END	Transition from MID to END. This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 2.

The start-split transactions do not receive a handshake from the transaction translator, so the host controller always advances the transfer state in the siTD after the bus transaction is complete. To advance the transfer state the following operations take place:

- The *siTD.Total Bytes To Transfer* and the *siTD.Current Offset* fields are adjusted to reflect the number of bytes transferred.
- The *siTD.P* (page selector) bit is updated appropriately.
- The *siTD.TP* and *siTD.T-count* fields are updated appropriately as defined in Table 62-51.

These fields are then written back to the memory based siTD. The *S-mask* is fixed for the life of the current budget. As mentioned above, *TP* and *T-count* are set specifically in each siTD to reflect the data to be sent from this siTD. Therefore, regardless of the value of *S-mask*, the actual number of start-split transactions depends on *T-count* (or equivalently, *Total Bytes to Transfer*). The host controller must set the *Active* bit to a zero when it detects that all of the schedule data has been sent to the bus. The preferred method is to detect when *T-Count* decrements to zero as a result of a start-split bus transaction. Equivalently, the host controller can detect when *Total Bytes to Transfer* decrements to zero. Either implementation must ensure that if the initial condition is *Total Bytes to Transfer* equal to zero and *T-count* is equal to a one, then the host controller will issue a single start-split, with a zero-length data payload. Software must ensure that *TP*, *T-count* and *Total Bytes to Transfer* are set to deliver the appropriate number of bus transactions from each siTD. An inconsistent combination will yield undefined behavior.

If the host experiences hold-offs that cause the host controller to skip start-split transactions for an OUT transfer, the state of the transfer will not progress appropriately. The transaction translator will observe protocol violations in the arrival of the start-splits for the OUT endpoint (that is, the transaction position annotation will be incorrect as received by the transaction translator).

Example scenarios are described in Section Split Transaction for Isochronous - Processing Examples.

A host controller implementation can optionally track the progress of an OUT split transaction by setting appropriate bits in the *siTD.C-prog-mask* as it executes each scheduled start-split. The *checkPreviousBit*() algorithm defined in Periodic Isochronous - Do Complete Split can be used prior to executing each start-split to determine whether start-splits were skipped. The host controller can use this mechanism to detect missed micro-frames. It can then set the siTD's *Active* bit to zero and stop execution of this siTD. This saves on both memory and high-speed bus bandwidth.

62.5.3.12.3.5 Periodic Isochronous - Do Complete Split

This state is only used by a split-transaction isochronous IN endpoint.

This state is entered unconditionally from the Do Start State after a start-split transaction is executed for an IN endpoint. Each time the host controller visits an siTD in this state, it conducts a number of tests to determine whether it should execute a complete-split transaction. The individual tests are listed below. The sequence they are applied depends on which micro-frame the host controller is currently executing which means that the tests might not be applied until after the siTD referenced from the back pointer has been fetched.

- Test A. *cMicroFrameBit* is bit-wise anded with *siTD.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame. This test is always applied to a newly fetched siTD that is in this state.
- Test B. The *siTD.C-prog-mask* bit vector is checked to determine whether the previous complete splits have been executed. An example algorithm is below (this is slightly different than the algorithm used in Section Periodic Isochronous - Do Start Split). The sequence in which this test is applied depends on the current value of USB_n_FRINDEX[2:0]. If USB_n_FRINDEX[2:0] is 0 or 1, it is not applied until the back pointer has been used. Otherwise it is applied immediately.

```
Algorithm Boolean CheckPreviousBit(siTD.C-prog-mask, siTD.C-mask, cMicroFrameBit)
Begin
          Boolean rvalue = TRUE;
          previousBit = cMicroFrameBit rotate-right(1)
          -- Bit-wise anding previousBit with C-mask indicates whether there was an intent
          -- to send a complete split in the previous micro-frame. So, if the
          -- 'previous bit' is set in C-mask, check C-prog-mask to make sure it
          -- happened.
          if previousBit bitAND siTD.C-mask then
                    if not (previousBit bitAND siTD.C-prog-mask) then
                              rvalue = FALSE
                    End if
          End if
          Return rvalue
End Algorithm
```

If Test A is true and USB_n_FRINDEX[2:0] is zero or one, then this is a case 2a or 2b scheduling boundary (see Figure 62-25). See Section Periodic Isochronous - Do Start Split for details in handling this condition.

If Test A and Test B evaluate to true, then the host controller will execute a complete-split transaction using the transfer state of the current siTD. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. The transfer state is advanced based on the completion status of the complete-split transaction. To advance the transfer state of an IN siTD, the host controller must:

- Decrement the number of bytes received from *siTD.Total Bytes To Transfer,*
- Adjust *siTD.Current Offset* by the number of bytes received,

- Adjust *siTD.P* (page selector) field if the transfer caused the host controller to use the next page pointer, and
- Set any appropriate bits in the *siTD.Status* field, depending on the results of the transaction.

Note that if the host controller encounters a condition where *siTD.Total Bytes To Transfer* is zero, and it receives more data, the host controller must not write the additional data to memory. The *siTD.Status.Active* bit must be set to zero and the *siTD.Status.Babble Detected* bit must be set to a one. The fields *siTD.Total Bytes To Transfer*, *siTD.Current Offset*, and *siTD.P* (page selector) are not required to be updated as a result of this transaction attempt.

The host controller must accept (assuming good data packet CRC and sufficient room in the buffer as indicated by the value of *siTD.Total Bytes To Transfer*) MDATA and DATA0/1 data payloads up to and including 192 bytes. A host controller implementation may optionally set *siTD.Status Active* to a zero and *siTD.Status.Babble Detected* to a one when it receives and MDATA or DATA0/1 with a data payload of more than 192 bytes. The following responses have the noted effects:

- ERR. The full-speed transaction completed with a time-out or bad CRC and this is a reflection of that error to the host. The host controller sets the *ERR* bit in the *siTD.Status* field and sets the *Active* bit to a zero.
- Transaction Error (XactErr). The complete-split transaction encounters a Timeout, CRC16 failure, etc. The *siTD.Status* field *XactErr* field is set to a one and the complete-split transaction must be retried immediately. The host controller must use an internal error counter to count the number of retries as a counter field is not provided in the siTD data structure. The host controller will not retry more than two times. If the host controller exhausts the retries or the end of the micro-frame occurs, the *Active* bit is set to zero.
- DATAx (0 or 1). This response signals that the final data for the split transaction has arrived. The transfer state of the siTD is advanced and the *Active* bit is set to a zero. If the *Bytes To Transfer* field has not decremented to zero (including the reception of the data payload in the DATAx response), then less data than was expected, or allowed for was actually received. This *short packet* event does not set the USBINT status bit in the USBSTS register to a one. The host controller will not detect this condition.
- NYET (and Last). On each NYET response, the host controller also checks to determine whether this is the last complete-split for this split transaction. Last was defined in Section Periodic Isochronous - Do Complete Split . If it is the last complete-split (with a NYET response), then the transfer state of the siTD is not advanced (never received any data) and the *Active* bit is set to a zero. No bits are set in the *Status* field because this is essentially a skipped transaction. The transaction translator must have responded to all the scheduled clompete-splits with NYETs,

meaning that the start-split issued by the host controller was not received. This result should be interpreted by system software as if the transaction was completely skipped. The test for whether this is the last complete split can be performed by XORing C-mask with C-prog-mask. A zero result indicates that all complete-splits have been executed.

- MDATA (and Last). See above description for testing for Last. This can only occur when there is an error condition. Either there has been a babble condition on the full-speed link, which delayed the completion of the full-speed transaction, or software set up the *S-mask* and/or *C-mask*s incorrectly. The host controller must set *XactErr* bit to a one and the *Active* bit is set to a zero.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask*) and stay in this state.
- MDATA (and not Last). The transaction translator responds with an MDATA when it has partial data for the split transaction. For example, the full-speed transaction data payload spans from micro-frame X to X+1 and during micro-frame X, the transaction translator will respond with an MDATA and the data accumulated up to the end of micro-frame X. The host controller advances the transfer state to reflect the number of bytes received.

If Test A succeeds, but Test B fails, it means that one or more of the complete-splits have been skipped. The host controller sets the *Missed Micro-Frame* status bit and sets the *Active* bit to a zero.

62.5.3.12.3.6 Complete-Split for Scheduling Boundary Cases 2a, 2b

Boundary cases 2a and 2b (INs only) (see Figure 62-25) require that the host controller use the transaction state context of the previous siTD to finish the split transaction. The table below enumerates the transaction state fields.

Table 62-52. Summary siTD Split Transaction State

Buffer State	Status	Execution Progress
Total Bytes To Transfer P (page select) Current Offset TP (transaction position) T-count (transaction count)	All bits in the status field	C-prog-mask

NOTE

TP and *T-count* are used only for Host to Device (OUT) endpoints.

If software has budgeted the schedule of this data stream with a frame wrap case, then it must initialize the *siTD.Back Pointer* field to reference a valid siTD and will have the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer*

field set to a zero. Otherwise, software must set the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer* field to a one. The host controller's rules for interpreting when to use the *siTD.Back Pointer* field are listed below. These rules apply only when the siTD's *Active* bit is a one and the *SplitXState* is Do Complete Split.

- When cMicroFrameBit is a 1h and the siTDX.Back Pointer.T-bit is a zero, or
- If cMicroFrameBit is a 2h and siTDX.S-mask[0] is a zero

When either of these conditions apply, then the host controller must use the transaction state from $siTD_{X-1}$.

In order to access $siTD_{X-1}$, the host controller reads on-chip the siTD referenced from $siTD_X$.*Back Pointer*.

The host controller must save the entire state from $siTD_X$ while processing $siTD_{X-1}$. This is to accommodate for case 2b processing. The host controller must not recursively walk the list of *siTD.Back Pointers*.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Complete Split), then both Test A and Test B are applied as described above. If these criteria to execute a complete-split are met, the host controller executes the complete split and evaluates the results as described above. The transaction state (see Table 62-52) of $siTD_{X-1}$ is appropriately advanced based on the results and written back to memory. If the resultant state of $siTD_{X-1}$'s *Active* bit is a one, then the host controller returns to the context of $siTD_X$, and follows its next pointer to the next schedule item. No updates to $siTD_X$ are necessary.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Start Split), then the host controller must set *Active* bit to a zero and *Missed Micro-Frame* status bit to a one and the resultant status written back to memory.

If $siTD_{X-1}$*'sActive* bit is a zero, (because it was zero when the host controller first visited $siTD_{X-1}$ via $siTD_X$*'s* back pointer, it transitioned to zero as a result of a detected error, or the results of $siTD_{X-1}$*'s* complete-split transaction transitioned it to zero), then the host controller returns to the context of $siTD_X$ and transitions its *SplitXState* to Do Start Split. The host controller then determines whether the case 2b start split boundary condition exists (that is, if *cMicroframeBit* is a 1b and $siTD_X$.*S-mask*[0] is a 1b). If this criterion is met the host controller immediately executes a start-split transaction and appropriately advances the transaction state of $siTD_X$, then follows $siTD_X$.*Next Pointer* to the next schedule item. If the criterion is not met, the host controller simply follows $siTD_X$.*Next Pointer* to the next schedule item. Note that in the case of a 2b boundary case, the split-transaction of $siTD_{X-1}$ will have its *Active* bit set to zero when the host controller returns

to the context of $siTD_X$. Also, note that software should not initialize an siTD with *C-mask* bits 0 and 1 set to a one and an *S-mask* with bit zero set to a one. This scheduling combination is not supported and the behavior of the host controller is undefined.

62.5.3.12.3.7 Split Transaction for Isochronous - Processing Examples

There is an important difference between how the hardware/software manages the isochronous split transaction state machine and how it manages the asynchronous and interrupt split transaction state machines.

The asynchronous and interrupt split transaction state machines are encapsulated within a single queue head. The progress of the data stream depends on the progress of each split transaction. In some respects, the split-transaction state machine is sequenced via the Execute Transaction queue head traversal state machine (see Figure 62-18).

Isochronous is a pure time-oriented transaction/data stream. The interface data structures are optimized to efficiently describe transactions that need to occur at specific times. The isochronous split-transaction state machine must be managed across these time-oriented data structures. This means that system software must correctly describe the scheduling of split-transactions across more than one data structure.

Then the host controller must make the appropriate state transitions at the appropriate times, in the correct data structures.

For example, the table below illustrates a couple of frames worth of scheduling required to schedule a case 2a full-speed isochronous data stream.

Table 62-53. Example Case 2a - Software Scheduling siTDs for an IN Endpoint

siTDX		Micro-Frames								Initial
#	Masks	0	1	2	3	4	5	6	7	SplitXState
X	S-Mask	-	-	-	-	1	-	-	-	Do Start Split
	C-Mask	1	1	-	-	-	-	1	1	
X+1	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+2	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+3	S-Mask	Repeats previous pattern								Do Complete Split
	C-Mask									

This example shows the first three siTDs for the transaction stream. Because this is the case-2a frame-wrap case, *S-masks* of all siTDs for this endpoint have a value of 10h (a one bit in micro-frame 4) and *C-mask* value of C3h (one-bits in micro-frames 0,1, 6 and 7). Additionally, sofware ensures that the *Back Pointer* field of each siTD references the appropriate siTD data structure (and the *Back PointerT-bits* are set to zero).

The initial *SplitXState* of the first siTD is Do Start Split. The host controller will visit the first siTD eight times during frame X. The C-mask bits in micro-frames 0 and 1 are ignored because the state is Do Start Split. During micro-frame 4, the host controller determines that it can run a start-split (and does) and changes *SplitXState* to Do Complete Split. During micro-frames 6 and 7, the host controller executes complete-splits. Notice the siTD for frame X+1 has its *SplitXState* initialized to Do Complete Split. As the host controller continues to traverse the schedule during *H-Frame* X+1, it will visit the second siTD eight times. During micro-frames 0 and 1 it will detect that it must execute complete-splits.

During *H-Frame* X+1, micro-frame 0, the host controller detects that $siTD_{X+1}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+1}$ and fetches $siTD_X$. It executes the complete split transaction using the transaction state of $siTD_X$. If the $siTD_X$ split transaction is complete, siTD's *Active* bit is set to zero and results written back to $siTD_X$. The host controller retains the fact that $siTD_X$ is retired and transitions the *SplitXState* in the $siTD_{X+1}$ to Do Start Split. At this point, the host controller is prepared to execute the start-split for $siTD_{X+1}$ when it reaches micro-frame 4. If the split-transaction completes early (transaction-complete is defined in Section Periodic Isochronous - Do Complete Split), that is, before all the scheduled complete-splits have been executed, the host controller will transition $siTD_X$.*SplitXState* to Do Start Split early and naturally skip the remaining scheduled complete-split transactions. For this example, $siTD_{X+1}$ does not receive a DATA0 response until *H-Frame* X+2, micro-frame 1.

During *H-Frame X+2,* micro-frame 0, the host controller detects that $siTD_{X+2}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. As described above, it executes another split transaction, receives an MDATA response, updates the transfer state, but does not modify the *Active* bit. The host controller returns to the context of $siTD_{X+2}$, and traverses its next pointer without any state change updates to $siTD_{X+2}$. S

During *H-Frame X+2*, micro-frame 1, the host controller detects $siTD_{X+2}$'s *S-mask[0]* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. It executes another complete-split transaction, receives a DATA0 response, updates the transfer state and sets the *Active* bit to a zero. It returns to the state of $siTD_{X+2}$ and changes its *SplitXState* to Do Start Split. At this point, the host controller is prepared to execute start-splits for $siTD_{X+2}$ when it reaches micro-frame 4.

62.5.3.13 Host Controller Pause

When the host controller's *HCHalted* bit in the USBSTS register is a zero, the host controller is sending SOF (Start OF Frame) packets down all enabled ports.

When the schedules are enabled, the EHCI host controller will access the schedules in main memory each micro-frame. This constant pinging of main memory is known to create Arm platform power management problems for mobile systems. Specifically, mobile systems aggressively manage the state of the Arm platform, based on recent history usage. In the more aggressive power saving modes, the Arm platform can disable its caches. Current PC architectures assume that bus-master accesses to main memory must be cache-coherent. So, when bus masters are busy touching memory, the Arm platform power management software can detect this activity over time and inhibit the transition of the Arm platform into its lowest power savings mode. USB controllers are bus-masters and the frequency at which they access their memory-based schedules keeps the Arm platform power management software from placing the Arm platform into its lowest power savings state.

USB Host controllers don't access main memory when they are suspended. However, there are a variety of reasons why placing the USB controllers into suspend won't work, but they are beyond the scope of this document. The base requirement is that the USB controller needs to be kept out of main memory, while at the same time, the USB bus is kept from going into suspend.

EHCI controllers provide a large-grained mechanism that can be manipulated by system software to change the memory access pattern of the host controller. System software can manipulate the schedule enable bits in the USBCMD register to turn on/off the scheduling traversal. A software heuristic can be applied to implement an on/off duty cycle that allows the USB to make reasonable progress and allow the Arm platform power management to get the Arm platform into its lowest power state. This method is not intended to be applied at all times to throttle USB, but should only be applied in very specific configurations and usage loads. For example, when only a keyboard or mouse is attached to the USB, the heuristic could detect times when the USB is attempting to move data only very infrequently and can adjust the duty cycle to allow the Arm platform to reach its low power state for longer periods of time. Similarly, it could detect increases in the USB load and adjust the duty cycle appropriately, even to the point where the schedules are never disabled. The assumption here is that the USB is moving data and the Arm platform will be required to process the data streams.

It is suggested that in order to provide a complete solution for the system, the companion host controllers should also provide a similar method to allow system software to inhibit the companion host controller from accessing its shared memory based data structures (schedule lists or otherwise).

62.5.3.14 Port Test Modes -Host Operational Model

EHCI host controllers must implement the port test modes Test J_State, Test K_State, Test_Packet, Test Force_Enable, and Test SE0_NAK as described in the USB Specification Revision 2.0.

The system is only allowed to test ports that are owned by the EHCI controller (for example, *CF-bit* is a one and *PortOwner* bit is a zero). System software is allowed to have at most one port in test mode at a time. Placing more than one port in test mode will yield undefined results. The required, per port test sequence is (assuming the *CF-bit* in the USB_n_CONFIGFLAG register is a one):

- Disable the periodic and asynchronous schedules by setting the *Asynchronous Schedule Enable* and *Periodic Schedule Enable* bits in the USBCMD register to a zero.
- Place all enabled root ports into the suspended state by setting the *Suspend* bit in each appropriate USB_n_PORTSC register to a one.
- Set the *Run/Stop* bit in the USBCMD register to a zero and wait for the *HCHalted* bit in the USBSTS register, to transition to a one. Note that an EHCI host controller implementation may optionally allow port testing with the *Run/Stop* bit set to a one. However, all host controllers must support port testing with *Run/Stop* set to a zero and *HCHalted* set to a one.
- Set the *Port Test Control* field in the port under test PORTSC register to the value corresponding to the desired test mode. If the selected test is Test_Force_Enable, then the *Run/Stop* bit in the USBCMD register must then be transitioned back to one, in order to enable transmission of SOFs out of the port under test.
- When the test is complete, system software must ensure the host controller is halted (*HCHalted* bit is a one) then it terminates and exits test mode by setting *HCReset* to a one.

62.5.3.15 Interrupts-Host Operational Model

The EHCI Host Controller hardware provides interrupt capability based on a number of sources.

There are several general groups of interrupt sources:

- Interrupts as a result of executing transactions from the schedule (success and error conditions),
- Host controller events (Port change events, etc.), and
- Host Controller error events

All transaction-based sources are maskable through the Host Controller's Interrupt Enable register (USBINTR). Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt.

During normal operation, interrupts may be immediate or deferred until the next interrupt threshold occurs. The interrupt threshold is a tunable parameter via the *Interrupt Threshold Control* field in the USBCMD register. The value of this register controls when the host controller will generate an interrupt on behalf of normal transaction execution. When a transaction completes during an interrupt interval period, the interrupt signaling the completion of the transfer will not occur until the interrupt threshold occurs. For example, the default value is eight micro-frames. This means that the host controller will not generate interrupts any more frequently than once every eight micro-frames.

Section Host System Error details effects of a host system error.

If an interrupt has been scheduled to be generated for the current interrupt threshold interval, the interrupt is not signaled until after the status for the last complete transaction in the interval has been written back to host memory. This may sometimes result in the interrupt not being signaled until the next interrupt threshold.

Initial interrupt processing is the same, regardless of the reason for the interrupt. When an interrupt is signaled by the hardware, Arm platform control is transferred to host controller's USB interrupt handler. The precise mechanism to accomplish the transfer is OS specific. For this discussion it is just assumed that control is received. When the interrupt handler receives control, its first action is to reads the USBSTS (USB Status Register). It then acknowledges the interrupt by clearing all of the interrupt status bits by writing ones to these bit positions. The handler then determines whether the interrupt is due to schedule processing or some other event. After acknowledging the interrupt, the handler (via an OS-specific mechanism), schedules a deferred procedure call (DPC) which will execute later. The DPC routine processes the results of the schedule execution. The precise mechanisms used are beyond the scope of this document.

Note: the host controller is not required to de-assert a currently active interrupt condition when software sets the interrupt enables (in the USBINR register) to a zero. The only reliable method software should use for acknowledging an interrupt is by transitioning the appropriate status bits in the USBSTS register from a one to a zero.

62.5.3.15.1 Transfer/Transaction Based Interrupts

These interrupt sources are associated with transfer and transaction progress. They are all dependent on the next interrupt threshold.

62.5.3.15.1.1 Transaction Error

A transaction error is any error that caused the host controller to think that the transfer did not complete successfully.

The table below lists the events/responses that the host can observe as a result of a transaction. The effects of the error counter and interrupt status are summarized in the following paragraphs. Most of these errors set the *XactErr* status bit in the appropriate interface data structure.

There is a small set of protocol errors that relate only when executing a queue head and fit under the umbrella of a WRONG PID error that are significant to explicitly identify. When these errors occur, the *XactErr* status bit in the queue head is set and the *CErr* field is decremented. When the *PIDCode* indicates a SETUP, the following responses are protocol errors and result in *XactErr* bit being set to a one and the *CErr* field being decremented.

- *EPS* field indicates a high-speed device and it returns a Nak handshake to a SETUP.
- *EPS* field indicates a high-speed device and it returns a Nyet handshake to a SETUP.
- *EPS* field indicates a low- or full-speed device and the complete-split receives a Nak handshake.

Table 62-54. Summary of Transaction Errors

Event / Result	Queue Head/qTD/iTD/siTD Side-effects		USB Status Register (USBSTS)
	Cerr	Status Field	USBERRINT
CRC	-1	XactErr set to a one.	1[1]
Timeout	-1	XactErr set to a one.	1[1]
Bad PID[2]	-1	XactErr set to a one.	1[1]
Babble	N/A	Section Serial Bus Babble	1
Buffer Error	N/A	Section Data Buffer Error	

1. If occurs in a queue head, then *USBERRINT* is asserted only when *CErr* counts down from a one to a zero. In addition the queue is halted, see Halting a Queue Head.
2. The host controller received a response from the device, but it could not recognize the PID as a valid PID.

62.5.3.15.1.2 Serial Bus Babble

When a device transmits more data on the USB than the host controller is expecting for this transaction, it is defined to be babbling. In general, this is called a *Packet Babble*.

When a device sends more data than the *Maximum Length* number of bytes, the host controller sets the *Babble Detected* bit to a one and halts the endpoint if it is using a queue head (see Halting a Queue Head). *Maximum Length* is defined as the minimum of *Total Bytes to Transfer* and *Maximum Packet Size*. The *CErr* field is not decremented for a packet babble condition (only applies to queue heads). A babble condition also exists if IN transaction is in progress at High-speed EOF2 point. This is called a frame babble. A frame babble condition is recorded into the appropriate schedule data structure. In addition, the host controller must disable the port to which the frame babble is detected.

The *USBERRINT* bit in the USB_n_USBSTS register is set to a one and if the *USB Error Interrupt Enable* bit in the USB_n_USBINTR register is a one, then a hardware interrupt is signaled to the system at the next interrupt threshold. The host controller must never start an OUT transaction that will babble across a micro-frame EOF.

NOTE

When a host controller detects a data PID mismatch, it must either: disable the packet babble checking for the duration of the bus transaction or do packet babble checking based solely on *Maximum Packet Size*. The USB core specification defines the requirements on a data receiver when it receives a data PID mismatch (for example, expects a DATA0 and gets a DATA1 or visa-versa). In summary, it must ignore the received data and respond with an ACK handshake, in order to advance the transmitter's data sequence.

The EHCI interface allows System software to provide buffers for a Control, Bulk or Interrupt IN endpoint that are not an even multiple of the maximum packet size specified by the device. Whenever a device misses an ACK for an IN endpoint, the host and device are out of synchronization with respect to the progress of the data transfer. The host controller may have advanced the transfer to a buffer that is less than maximum packet size. The device will re-send its maximum packet size data packet, with the original data PID, in response to the next IN token. In order to properly manage the bus protocol, the host controller must disable the packet babble check when it observes the data PID mismatch.

62.5.3.15.1.3 Data Buffer Error

This event indicates that an overrun of incoming data or a underrun of outgoing data has occurred for this transaction.

This would generally be caused by the host controller not being able to access required data buffers in memory within necessary latency requirements. These conditions are not considered transaction errors, and do not effect the error count in the queue head. When these errors do occur, the host controller records the fact the error occurred by setting the *Data Buffer Error* bit in the queue head, iTD or siTD.

If the data buffer error occurs on a non-isochronous IN, the host controller will not issue a handshake to the endpoint. This will force the endpoint to resend the same data (and data toggle) in response to the next IN to the endpoint.

If the data buffer error occurs on an OUT, the host controller must corrupt the end of the packet so that it cannot be interpreted by the device as a good data packet. Simply truncating the packet is not considered acceptable. An acceptable implementation option is to 1's complement the CRC bytes and send them. There are other options suggested in the Transaction Translator section of the USB Specification Revision 2.0.

62.5.3.15.1.4 USB Interrupt (Interrupt on Completion (IOC))

Transfer Descriptors (iTDs, siTDs, and queue heads (qTDs)) contain a bit that can be set to cause an interrupt on their completion. The completion of the transfer associated with that schedule item causes the USB Interrupt (USBINT) bit in the USB_n_USBSTS register to be set to a one.

In addition, if a short packet is encountered on an IN transaction associated with a queue head, then this event also causes USBINT to be set to a one. If the USB Interrupt Enable bit in the USB_n_USBINTR register is set to a one, a hardware interrupt is signaled to the system at the next interrupt threshold. If the completion is because of errors, the *USBERRINT* bit in the USB_n_USBSTS register is also set to a one.

62.5.3.15.1.5 Short Packet

Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer. Whenever a short packet completion occurs during a queue head execution, the *USBINT* bit in the USB_n_USBSTS register is set to a one.

If the *USB Interrupt Enable* bit is set in the USB_n_USBINTR register, a hardware interrupt is signaled to the system at the next interrupt threshold.

62.5.3.15.2 Host Controller Event Interrupts

These interrupt sources are independent of the interrupt threshold (with the one exception being the Interrupt on Async Advance, see Section Interrupt on Async Advance).

62.5.3.15.2.1 Port Change Events

Port registers contain status and status change bits. When the status change bits are set to a one, the host controller sets the *Port Change Detect* bit in the USBSTS register to a one.

If the *Port Change Interrupt Enable* bit in the USB_n_USBINTR register is a one, then the host controller will issue a hardware interrupt. The port status change bits include:

- Connect Status Change
- Port Enable/Disable Change
- Over-current Change
- Force Port Resume

62.5.3.15.2.2 Frame List Rollover

This event indicates that the host controller has wrapped the frame list. The current programmed size of the frame list effects how often this interrupt occurs.

If the frame list size is 1024, then the interrupt will occur every 1024 milliseconds, if it is 512, then it will occur every 512 milliseconds, etc. When a frame list rollover is detected, the host controller sets the *Frame List Rollover* bit in the USB.USBSTS register to a one. If the *Frame List Rollover Enable* bit in the USB.USBINTR register is set to a one, the host controller issues a hardware interrupt. This interrupt is not delayed to the next interrupt threshold.

62.5.3.15.2.3 Interrupt on Async Advance

This event is used for deterministic removal of queue heads from the asynchronous schedule. Whenever the host controller advances the on-chip context of the asynchronous schedule, it evaluates the value of the *Interrupt on Async Advance Doorbell* bit in the USB.USBCMD register. If it is a one, it sets the *Interrupt on Async Advance* bit in the USB.USBSTS register to a one. If the *Interrupton Async Advance Enable* bit in the USB.USBINTR register is a one, the host controller issues a hardware interrupt at the next interrupt threshold. A detailed explanation of this feature is described in Section Removing Queue Heads from Asynchronous Schedule .

62.5.3.15.2.4 Host System Error

The host controller is a bus master and any interaction between the host controller and the system may experience errors.

The type of host error may be catastrophic to the host controller (such as a Master Abort) making it impossible for the host controller to continue in a coherent fashion.In the presence of non-catastrophic host errors, such as parity errors, the host controller could potentially continue operation. The recommended behavior for these types of errors is to escalate it to a catastrophic error and halt the host controller. Host-based error must result in the following actions:

- The *Run/Stop* bit in the USB.USBCMD register is set to a zero.
- The following bits in the USB.USBSTS register are set:
  - *Host System Error* bit is to a one.
  - *HCHalted* bit is set to a one.
- If the *Host System Error Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. This interrupt is not delayed to the next interrupt threshold. The following table summarizes the required actions taken on the various host errors.

Table 62-55. Summary Behavior of EHCI Host Controller on Host System Errors

Cycle Type	Master Abort	Target Abort	Data Phase Parity
Frame list pointer fetch (read)	Fatal	Fatal	Fatal [o]
siTD fetch (read)	Fatal	Fatal	Fatal [o]
siTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
iTD fetch (read)	Fatal	Fatal	Fatal [o]
iTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
qTD fetch (read)	Fatal	Fatal	Fatal [o]
qHD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
Data write	Fatal [o]	Fatal [o]	Fatal [o]
Data read	Fatal	Fatal	Fatal [o]

Potentially, a host controller implementation could continue operation without a halt. However, the recommended behavior is to halt the host controller.

NOTE

After a *Host System Error*, Software must reset the host controller through *HCReset* in the USB.USBCMD register before re-initializing and restarting the host controller.

62.5.4 EHCI Deviation

For the purposes a dual-role Host/Device controller with support for On-The-Go applications, it is necessary to deviate from the EHCI specification. Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November

2000, Intel Corporation. http://www.intel.com. Device operation & On-The-Go operation is not specified in the EHCI and thus the implementation supported in this core is proprietary.

The host mode operation of the core is near EHCI compatible with few minor differences documented in this section.

The particulars of the deviations occur in the areas summarized here:

- Embedded Transaction Translator - Allows direct attachment of FS and LS devices in host mode without the need for a companion controller.
- Device operation - In host mode the device operational registers are generally disabled and thus device mode is mostly transparent when in host mode. However, there are a couple exceptions documented in the following sections.
- Embedded design interface - This core does not have a PCI Interface and therefore the PCI configuration registers described in the EHCI specification are not applicable.
- On-The-Go Operation - This design includes an On-The-Go controller for Port #1.

62.5.4.1 Embedded Transaction Translator Function

The OTG controller supports directly connected full and low speed devices without requiring a companion controller by including the capabilities of a USB 2.0 high speed hub transaction translator.

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a high speed hub has been implemented within the DMA and Protocol engine blocks. The embedded transaction translator function is an extension to EHCI interface, but makes use of the standard data structures and operational models that exist in the EHCI specification to support full and low speed devices.

62.5.4.1.1 Capability Registers

The following additions have been added to the capability registers to support the embedded Transaction Translator Function:

- N_TT added to USB.HCSPARAMS - Host Control Structural Parameters
- N_PTT added to USB.HCSPARAMS - Host Control Structural Parameters

62.5.4.1.2 Operational Registers

The following additions have been added to the operational registers to support the embedded TT:

- Addition of two-bit Port Speed (PSPD) to the PORTSC1 register.

62.5.4.1.3 Discovery-EHCI Deviation

In a standard EHCI controller design, the EHCI host controller driver detects a Full speed (FS) or Low speed (LS) device by noting if the port enable bit is set after the port reset operation.

The port enable will only be set in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate a High-Speed connection (that is, Chirp completes successfully).

Because this controller has an embedded Transaction Translator, the port enable will always be set after the port reset operation regardless of the result of the host device chirp result and the resulting port speed will be indicated by the PSPD field in USB.PORTSCx.

Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected Full and Low speed devices or hubs.

The change is a fundamental one in that is summarized in the following table.

Table 62-56. Summary of EHCI

Standard EHCI	EHCI with embedded Transaction Translator
After port enable bit is set following a connection and reset sequence, the device/hub is assumed to be HS.	After port enable bit is set following a connection and reset sequence, the device/hub speed is noted from USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub thus, all port-level control is performed through the Hub Class to the nearest Hub.	FS and LS device can be either downstream from a HS hub or directly attached. When the FS/LS device is downstream from a HS hub, then port-level control is done using the Hub Class through the nearest Hub. When a FS/LS device is directly attached, then port-level control is accomplished using USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub with HubAddr=X. [where HubAddr > 0 and HubAddr is the address of the Hub where the bus transitions from HS to FS/LS (ie. Split target hub)]	FS and LS device can be either downstream from a HS hub with HubAddr = X [HubAddr > 0] or directly attached [where HubAddr = 0 and HubAddr is the address of the Root Hub where the bus transitions from HS to FS/LS (ie. Split target hub is the root hub)]

62.5.4.1.4 Data Structures

The same data structures used for FS/LS transactions though a HS hub are also used for transactions through the Root Hub with sm embedded Transaction Translator.

Here it is demonstrated how the Hub Address and Endpoint Speed fields should be set for directly attached FS/LS devices and hubs:

1. QH (for direct attach FS/LS) - Async. (Bulk/Control Endpoints) Periodic (Interrupt)
   - Hub Address = 0
   - Transactions to direct attached device/hub.
     - QH.EPS = Port Speed
   - Transactions to a device downstream from direct attached FS hub.
     - QH.EPS = Downstream Device Speed

NOTE

When QH.EPS = 01 (LS) and PORTSCx.PSPD = 00 (FS), a LS-pre-pid will be sent before the transmitting LS traffic.

Maximum Packet Size must be less than or equal 64 or undefined behaviour may result.

2. siTD (for direct attach FS) - Periodic (ISO Endpoint)
   - All FS ISO transactions:
     - Hub Address = 0
     - siTD.EPS = 00 (full speed)
       - Maximum Packet Size must less than or equal to 1023 or undefined behaviour may result.

62.5.4.1.5 Operational Model

The operational models are well defined for the behavior of the Transaction Translator (see USB 2.0 specification. Universal Serial Bus Specification, Revision 2.0, April 2000, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. http://www.usb.org) and for the EHCI controller moving packets between system memory and a USB-HS hub. Because the embedded Transaction Translator exists within the host controller there is no physical bus between EHCI host controller driver and the USB FS/LS bus. These sections will briefly discuss the operational model for how the EHCI and Transaction Translator operational models are combined without the physical bus between. The following sections assume the reader is familiar with both the EHCI and USB 2.0 Transaction Translator operational models.

62.5.4.1.5.1 Micro- frame Pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the Host (H) and the Bus (B). The embedded Transaction Translator shall use the same pipeline algorithms specified in the USB 2.0 specification for a Hub-based Transaction Translator.

All periodic transfers always begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the micro-frame pipeline implemented in the embedded Transaction Translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 will be ready to execute on the bus in B-frame 0.

It is important to note that when programming the S-mask and C-masks in the EHCI data structures to schedule periodic transfers for the embedded Transaction Translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming the S-mask and C-mask for downstream Hub-based Transaction Translators.

Once periodic transfers are exhausted, any stored asynchronous transfer will be moved. Asynchronous transfers are opportunistic in that they shall execute whenever possible and their operation is not tied to H-frame and B-frame boundaries with the exception that an asynchronous transfer can not babble through the SOF (start of B-frame 0.)

62.5.4.1.5.2 Split State Machines

The start and complete split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded Transaction Translator.

Where a start or complete-split operation would occur by requesting the split to the HS hub, the start/complete split operation is simple an internal operation to the embedded Transaction Translator. The following table summarizes the conditions where handshakes are emulated from internal state instead of actual handshakes to HS split bus traffic.

Table 62-57. Summary of the Conditons of Handshakes[1]

Condition	Emulate TT Response
Start-Split: All asynchronous buffers full.	NAK
Start-Split: All periodic buffers full.	ERR
Start-Split: Success for start of Async. Transaction.	ACK
Start-Split: Start Periodic Transaction.	No Handshake (Ok)
Complete-Split: Failed to find transaction in queue.	Bus Time Out
Complete-Split: Transaction in Queue is Busy.	NYET
Complete-Split: Transaction in Queue is Complete.	[Actual Handshake from LS/FS device]

1. The un-shaded cells represent Start-Splits and the shaded cells represent Complete-Splits

62.5.4.1.5.3 Asynchronous Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.
- Transaction tracking for 2 data pipes.

62.5.4.1.5.3.1 USB 2.0 - 11.17.3

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.

62.5.4.1.5.3.2 USB 2.0 - 11.17.4

- Transaction tracking for 2 data pipes.

62.5.4.1.5.4 Periodic Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits
  - EOF
  - Idle for more than 4 micro-frames
- Transaction tracking for up to 16 data pipes.
- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

62.5.4.1.5.4.1 USB 2.0 - 11.18.6.[1-2]

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits
  - EOF
  - Idle for more than 4 micro-frames

62.5.4.1.5.4.2 USB 2.0 - 11.18.[7-8]

- Transaction tracking for up to 16 data pipes.
- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

62.5.4.1.5.5 Multiple Transaction Translators

The maximum number of embedded Transaction Translators that is currently supported is one as indicated by the N_TT field in the HCSPARAMS register.

62.5.4.2 Device Operation

The co-existence of a device operational controller within the host controller has little effect on EHCI compatibility for host operation except as noted in this section.

62.5.4.2.1 USB_USBMODE Register

Given that the dual-role controller is initialized in neither host nor device mode, the USB_nUSBMODE register must be programmed for host operation before the EHCI host controller driver can begin EHCI host operations.

62.5.4.2.2 Non-Zero Fields the Register File

Some of the reserved fields and reserved addresses in the capability registers and operational register have use in device mode, the following must be adhered to:

- Write operations to all EHCI reserved fields (some of which are device fields) with the operation registers should always be written to zero. This is an EHCI requirement of the device controller driver that must be adhered to.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusive for device are undefined in host mode .

62.5.4.2.3 SOF Interrupt

This SOF Interrupt used for device mode is shared as a free running 125us interrupt for host mode.

EHCI does not specify this interrupt but it has been added for convenience and as a potential software time base. See USB_nUSBSTS and USB_nUSBINTR registers.

62.5.4.3 Embedded Design Interface

This is an Embedded USB Host Controller as defined by the EHCI specification and thus does not implement the PCI configuration registers.

62.5.4.3.1 Frame Adjust Register

Given that the optional PCI configuration registers are not included in this implementation, there is no corresponding bit level timing adjustments like is provided by the Frame Adjust register in the PCI configuration registers. Starts of micro-frames are timed precisely to 125 us using the transceiver clock as a reference clock. That is, a 60 Mhz transceiver clock for 8-bit physical interfaces & full-speed serial interfaces or 30 Mhz transceiver clock for 16-bit physical interfaces.

62.5.4.4 Miscellaneous variations from EHCI

62.5.4.4.1 Programmable Physical Interface Behaviour

This design supports multiple Physical interfaces which can operate in differing modes when the core is configured with software programmable Physical Interface Modes. Software programmability allows the selection of the Physical interface part during the board design phase instead of during the chip design phase. The control bits for selecting the Physical Interface operating mode have been added to the USB_nPORTSC1 register providing a capability that is not defined by EHCI.

62.5.4.4.2 Discovery

62.5.4.4.2.1 Port Reset

The port connect methods specified by EHCI require setting the port reset bit in the USB_nPORTSC1 register for a duration of 10ms. Due to the complexity required to support the attachment of devices that are not high speed there are counter already present in the design that can count the 10ms reset pulse to alleviate the requirement of the software to measure this duration. Therefore, the basic connection is then summarized as the following:

- [Port Change Interrupt] Port connect change occurs to notify the host controller driver that a device has attached.
- Software shall write a '1' to reset the device.

- Software shall write a '0' to reset the device after 10 ms.
  - This step, which is necessary in a standard EHCI design, may be omitted with this implementation. Should the EHCI host controller driver attempt to write a '0' to the reset bit while a reset is in progress, the write will simple be ignored and the reset will continue until completion.
- [Port Change Interrupt] Port enable change occurs to notify the host controller that the device in now operational and at this point the port speed has been determined.

62.5.4.4.2.2 Port Speed Detection

After the port change interrupt indicates that a port is enabled, the EHCI stack should determine the port speed. Unlike the EHCI implementation, which will re-assign the port owner for any device that does not connect at High-Speed, this host controller supports direct attach of non High-Speed devices.

Therefore, the following differences are important regarding port speed detection:

- Port Owner is read-only and always reads 0.
- A 2-bit Port Speed indicator has been added to PORTSC to provide the current operating speed of the port to the host controller driver.
- A 1-bit High Speed indicator has been added to PORTSC to signify that the port is in High-Speed vs. Full/Low Speed - *This information is redundant with the 2-bit Port Speed indicator above.*

62.5.4.4.3 Port Test Mode

Port Test Control mode behaves fully as described in EHCI. An alternate host controller driver procedure is not necessary or supported.

62.5.5 Device Data Structures

This section defines the interface data structures used to communicate control, status, and data between Device Controller Driver (DCD) Software and the Device Controller.

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device Queue Heads and Transfer Descriptors.

NOTE

Software must ensure that no interface data structure reachable by the Device Controller spans a 4K-page boundary.

The data structures defined in the chapter are (from the device controller's perspective) a mix of read-only and read/ writable fields. The device controller must preserve the read-only fields on all data structure writes.

The figure below shows the organization of the EndPoint Queue Head.

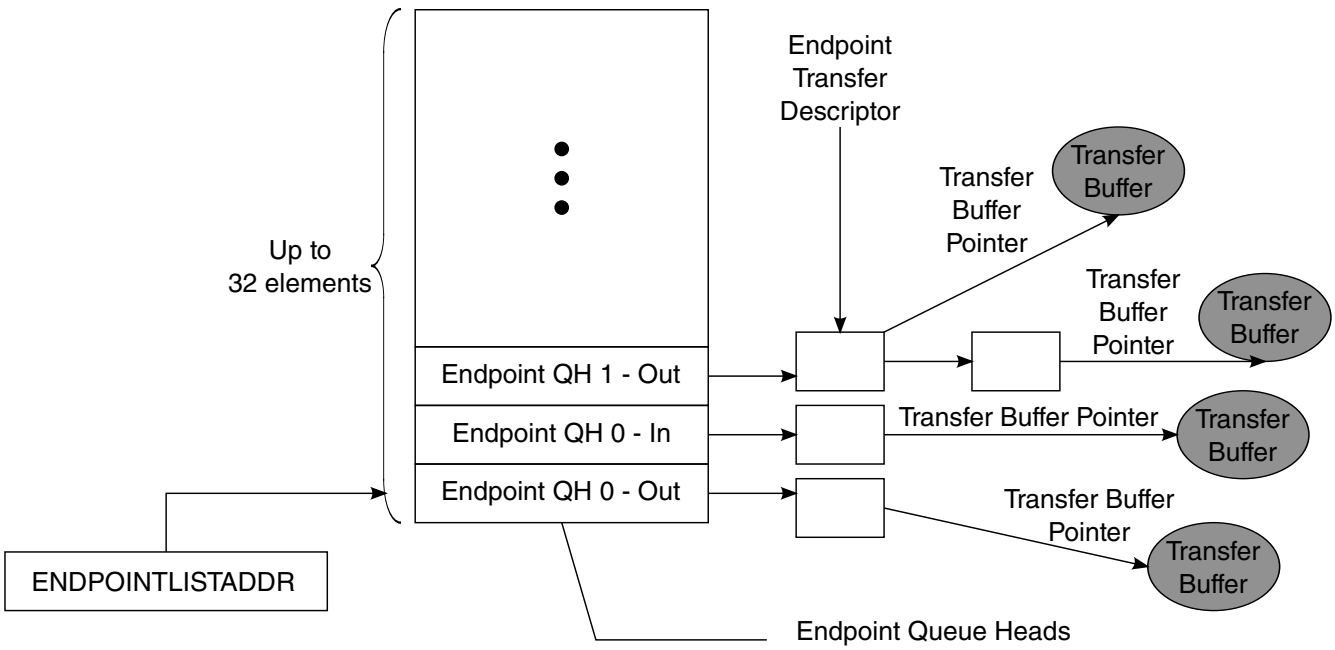


Figure 62-28. EndPoint Queue Head Organization

Endpoint queue heads are arranged in an array in a continuous area of memory pointed to by the USB.ENDPOINTLISTADDR pointer. The even-numbered device queue heads in the list support receive endpoints (OUT/SETUP) and the odd-numbered queue heads in the list are used for transmit endpoints (IN/INTERRUPT). That is, the device queue heads in the list with even offset support receiver endpoints (OUT) and the queue heads in the list with odd offset are used for transmit endpoint(IN). The device controller will index into this array based upon the endpoint number received from the USB bus. All information necessary to respond to transactions for all primed transfers is contained in this list so the Device Controller can readily respond to incoming requests without having to traverse a linked list.

NOTE

The Endpoint Queue Head List must be aligned to a 2k boundary.

62.5.5.1 Endpoint Queue Head (dQH)

The device Endpoint Queue Head (dQH) is where all transfers for a given endpoint are managed. The dQH is a 48-byte data structure, but must be aligned on 64-byte boundaries.

During priming of an endpoint, the dTD (device transfer descriptor) is copied into the overlay area of the dQH, which starts at the nextTD pointer DWord and continues through the end of the buffer pointers DWords. After a transfer is complete, the dTD status DWord is updated in the dTD pointed to by the currentTD pointer. While a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the Device Controller can access needed information with little minimal latency.

Table 62-58. Endpoint Queue Head (dQH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mult		zlt	0		Maximum Packet Length											ios	0														
Current dTD Pointer																											0				
Next dTD Pointer																											0				T[1]
0	Total Bytes															ioc	0			MultO		0		Status							
Buffer Pointer (Page 0)																				Current Offset											
Buffer Pointer (Page 1)																				Reserved											
Buffer Pointer (Page 2)																				Reserved											
Buffer Pointer (Page 3)																				Reserved											
Buffer Pointer (Page 4)[1]																				Reserved											
Reserved																															
Set-up Buffer Bytes 3…0																															
Set-up Buffer Bytes 7…4																															

1. Transfer overlay starts at T and continues through Buffer Pointer (Page 4).

Host Controller Read/Write

Host Controller Read Only

62.5.5.1.1 Endpoint Capabilities/Characteristics

This DWord specifies static information about the endpoint, in other words, this information does not change over the lifetime of the endpoint. Device Controller software should not attempt to modify this information while the corresponding endpoint is enabled.

Table 62-59 describes the endpoint capabilities.

Table 62-59. Endpoint Capabilities/Characteristics

Bit	Description
31-30	Mult. This field is used to indicate the number of packets executed per transaction description as given by the following: 00 - Execute N Transactions as demonstrated by the USB variable length packet protocol where N is computed using the Maximum Packet Length (dQH) and the Total Bytes field (dTD) 01 Execute 1 Transaction. 10 Execute 2 Transactions. 11 Execute 3 Transactions. **NOTE:** Non-ISO endpoints must set Mult="00". ISO endpoints must set Mult="01", "10", or "11" as needed.
29	Zero Length Termination Select. This bit is used to indicate when a zero length packet is used to terminate transfers where to total transfer length is a multiple . This bit is not relevant for Isochronous 0 - Enable zero length packet to terminate transfers equal to a multiple of the Maximum Packet Length. (default). 1 - Disable the zero length packet on transfers that are equal in length to a multiple Maximum Packet Length.
28-27	Reserved. These bit reserved for future use and should be set to zero.
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field may contain is 0x400 (1024).
15	Interrupt On Setup (IOS). This bit is used on control type endpoints to indicate if USBINT is set in response to a setup being received.
14-0	Reserved. Bits reserved for future use and should be set to zero.

62.5.5.1.2 Transfer Overlay-Endpoint Queue Head

The seven DWords in the overlay area represent a transaction working space for the device controller.

The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it will not read the associated endpoint.

After an endpoint is readied, the dTD will be copied into this queue head overlay area by the device controller. Until a transfer is expired, software must not write the queue head overlay area or the associated transfer descriptor. When the transfer is complete, the device controller will write the results back to the original transfer descriptor and advance the queue.

See dTD for a description of the overlay fields.

62.5.5.1.3 Current dTD Pointer

The current dTD pointer is used by the device controller to locate the transfer in progress. This word is for Device Controller (hardware) use only and should not be modified by DCD software.

The following table describes the dTD Pointer.

Table 62-60. Next dTD Pointer

Bit	Description
31-5	Current dTD. This field is a pointer to the dTD that is represented in the transfer overlay area. This field will be modified by the Device Controller to next dTD pointer during endpoint priming or queue advance.
4-0	Reserved. Bit reserved for future use and should be set to zero.

62.5.5.1.4 Set-up Buffer

The set-up buffer is dedicated storage for the 8-byte data that follows a set-up PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it, and only the RX queue head is used for receiving setup data packets.

The following table describes the Multiple Mode Control.

Table 62-61. Multiple Mode Control (HCCPARAMS)

DWord	Bits	Description
1	31-0	Setup Buffer 0. This buffer contains bytes 3 to 0 of an incoming setup buffer packet and is written by the device controller to be read by software.
2	31-0	Setup Buffer 1. This buffer contains bytes 7 to 4 of an incoming setup buffer packet and is written by the device controller to be read by software.

62.5.5.2 Endpoint Transfer Descriptor (dTD)

The dTD describes to the device controller the location and quantity of data to be sent/ received for a given transfer.

The DCD should not attempt to modify any field in an active dTD except the Next Like Pointer, which should only be modified as described in section Managing Transfers with Transfer Descriptors.

Table below shows the Endpoint Transfer Descriptor (dTD).

Table 62-62. Endpoint Transfer Descriptor (dTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Link Pointer																											0				T
0	Total Bytes															ioc	0			MultO		0		Status							
Buffer Pointer (Page 0)																				Current Offset											
Buffer Pointer (Page 1)																				0	Frame Number										
Buffer Pointer (Page 2)																				Reserved											
Buffer Pointer (Page 3)																				Reserved											
Buffer Pointer (Page 4)																				Reserved											

Host Controller Read/Write

Host Controller Read Only

The following table describes the dTD Pointer.

Table 62-63. Next dTD Pointer

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next dTD to be processed. The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved. Bits reserved for future use and should be set to zero.
0	Terminate (T). 1=pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Device Controller that there are no more valid entries in the queue.

The following table describes the dTD Token.

Table 62-64. dTD Token

Bit	Description
31	Reserved. Bit reserved for future use and should be set to zero.
30-16	Total Bytes. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction and only on the successful completion of the transaction. The maximum value software may store in the field is 5*4K (5000H). This is the maximum number of bytes 5 page pointers can access. Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16K (4000H). If the value of the field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the device controller executes a zero-length transaction and retires the transfer descriptor.

Table continues on the next page...

Table 62-64. dTD Token (continued)

	It is not a requirement for IN transfers that Total Bytes To Transfer be an even multiple of *Maximum Packet Length*. If software builds such a transfer descriptor for an IN transfer, the last transaction will always be less that *Maximum Packet Length*.
15	Interrupt On Complete (IOC). This bit is used to indicate if USBINT is to be set in response to device controller being finished with this dTD.
14-12	Reserved. Bits reserved for future use and should be set to zero.
11-10	Multiplier Override (MultO). This field can be used for transmit ISO's (ie. ISO-IN) to override the multiplier in the QH. This field must be zero for all packet types that are not transmit-ISO. Example: if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 0 [default] Three packets are sent: {Data2(8); Data1(7); Data0(0)} if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 2 Two packets are sent: {Data1(8); Data0(7)} For maximal efficiency, software should compute MultO = greatest integer of (Total Bytes / Max. Packet Size) except for the case when Total Bytes = 0; then MultO should be 1. Note: Non-ISO and Non-TX endpoints must set MultO = "00".
9-8	Reserved. Bits reserved for future use and should be set to zero.
7-0	Status. This field is used by the Device Controller to communicate individual command execution states back to the Device Controller software. This field contains the status of the last transaction performed on this qTD. The bit encodings are: Bit Status Field Description 7 Active. 6 Halted. 5 Data Buffer Error. 3 Transaction Error. 4, 2, 0 Reserved.

The table below describes the dTD Buffer Page Pointer List.

Table 62-65. dTD Buffer Page Pointer List

Bit	Description
31-12	Buffer Pointer. Selects the page offset in memory for the packet buffer. Non virtual memory systems will typically set the buffer pointers to a series of incrementing integers.
0,11-0	Current Offset. Offset into the 4kb buffer where the packet is to begin.
1,10-0	Frame Number. Written by the device controller to indicate the frame number in which a packet finishes. This is typically be used to correlate relative completion times of packets on an ISO endpoint.

62.5.6 Device Operational Model

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus.

Using a set of linked list transfer descriptors, pointed to by a queue head, the device controller will perform the data transfers. The following sections explain the use of the device controller from the device controller driver (DCD) point-of-view and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

62.5.6.1 Device Controller Initialization

After hardware reset, the device is disabled until the Run/Stop bit is set to a '1'. In the disabled state, the pull-up on the USB D+ is not active which prevents an attach event from occurring. At a minimum, it is necessary to have the queue heads setup for endpoint zero before the device attach occurs.

Shortly after the device is enabled, a USB reset will occur followed by setup packet arriving at endpoint 0. A Queue head must be prepared so that the device controller can store the incoming setup packet.

In order to initialize a device, the software should perform the following steps:

- Set Controller Mode in the USB.USBMODE register to device mode.

NOTE

Transitioning from host mode to device mode requires a device controller reset before modifying USB.USBMODE.

- Allocate and Initialize device queue heads in system memory.
  - Minimum: Initialize device queue heads 0 Tx & 0 Rx.

NOTE

All device queue heads for control endpoints must be initialized before the endpoint is enabled. Non-Control device queue heads before the endpoint can be used.

  - For information on device queue heads, refer to section Device Data Structures.
- Configure USB.ENDPOINTLISTADDR Pointer.
  - For additional information on USB.ENDPOINTLISTADDR, refer to the register table.
- Enable the microprocessor interrupt associated with the USB core.
  - Recommended: enable all device interrupts including: USBINT, USBERRINT, Port Change Detect, USB Reset Received, DCSuspend.
  - For a list of available interrupts refer to the USB_nUSBINTR and the USB_nUSBSTS register tables.
- Set Run/Stop bit to Run Mode.

- After the Run bit is set and the device is connected to a host, a Bus Reset will be issued by host downstream port. The DCD must monitor the reset event and adjust the software state as described in the Bus Reset section of the Port State and Control section below.

NOTE

Endpoint 0 is designed as a control endpoint only and does not need to be configured using ENDPTCTRL0 register.

It is also not necessary to prime Endpoint 0 initially because the first packet received will always be a setup packet. The contents of the first setup packet will require a response in accordance with USB device framework (Chapter 9) command set.

62.5.6.2 Port State and Control

From a chip or system reset, the device controller enters the *powered* state. A transition from the *powered* state to the *attach* state occurs when the Run/Stop bit is set to a '1'.

After receiving a reset on the bus, the port will enter the *defaultFS or defaultHS* state in accordance with the reset protocol described in Appendix C.2 of the USB Specification Rev. 2.0.

The following state diagram depicts the state of a USB 2.0 device.

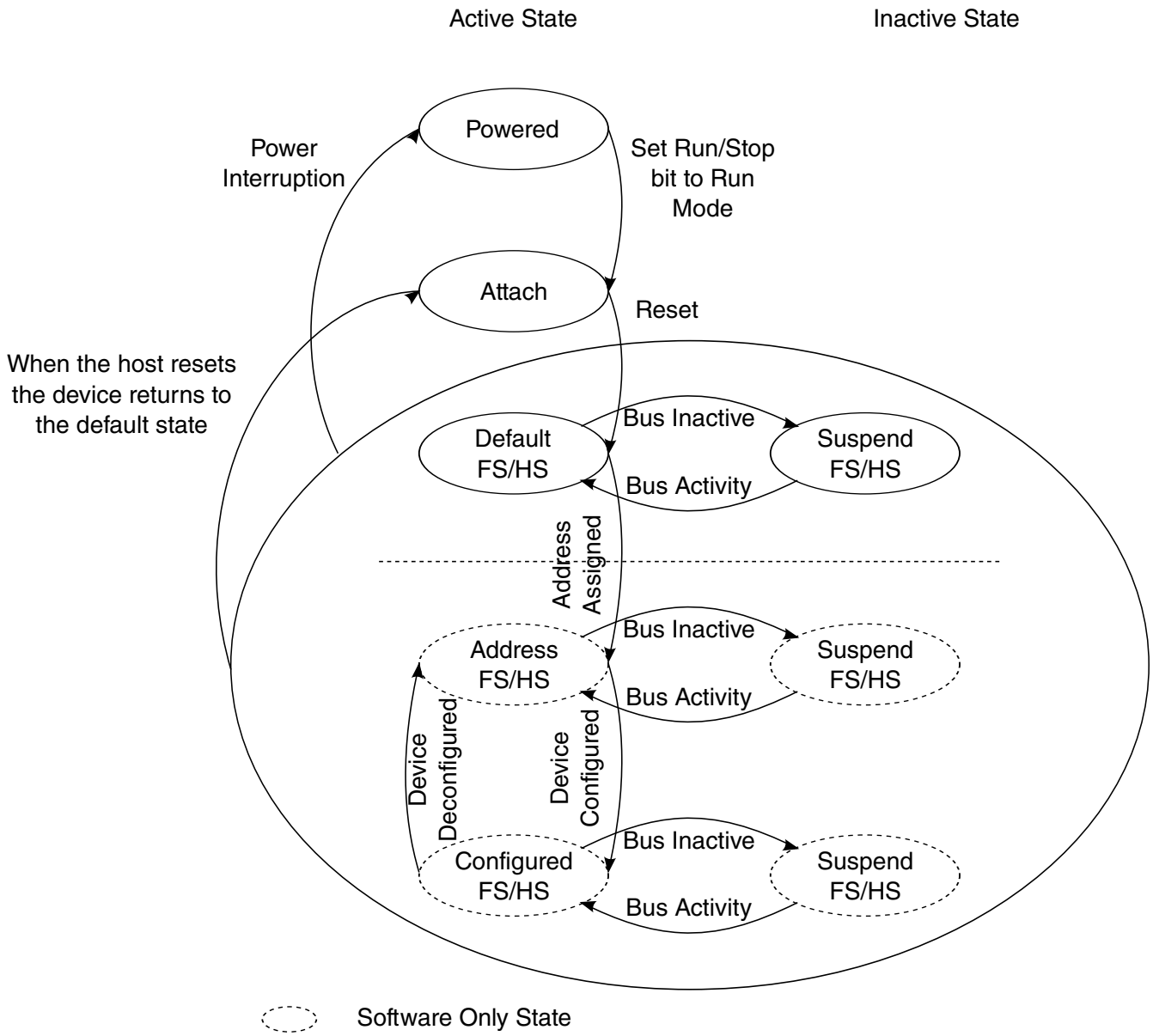


Figure 62-29. Device State Diagram

States *powered, attach, defaultFS/HS, suspendFS/HS* are implemented in the device controller and are communicated to the DCD using the following status bits:

The following table describes the Device Controller State Information Bits.

Table 62-66. Device Controller State Information Bits

Bit	Register
DCSuspend	USBSTS
USB Reset Received	USBSTS
Port Change Detect	USBSTS
High-Speed Port	PORTSC1

It is the responsibility of the DCD to maintain a state variable to differentiate between the *DefaultFS/HS* state and the *Address/Configured* states. Change of state from *Default* to *Address* and the *Configured* states is part of the enumeration process described in the device framework section of the USB 2.0 Specification.

As a result of entering the *Address* state, the device address register (DEVICEADDR) must be programmed by the DCD.

Entry into the *Configured* indicates that all endpoints to be used in the operation of the device have been properly initialized by programming the USB_UOG_ENDPTCTRLx registers and initializing the associated queue heads.

62.5.6.2.1 Bus Reset

A bus reset is used by the host to initialize downstream devices.

When a bus reset is detected, the device controller will renegotiate its attachment speed, reset the device address to 0, and notify the DCD by interrupt (assuming the USB Reset Interrupt Enable is set). After a reset is received, all endpoints (except endpoint 0) are disabled and any primed transactions will be cancelled by the device controller. The concept of priming will be clarified below, but the DCD must perform the following tasks when a reset is received:

Clear all setup token semaphores by reading the USB_nENDPTSTAT register and writing the same value back to the USB_nENDPTSTAT register.

Clear all the endpoint complete status bits by reading the USB_nPTCOMPLETE register and writing the same value back to the USB_nENDPTCOMPLETE register.

Cancel all primed status by waiting until all bits in the USB_nENDPTPRIME are 0 and then writing 0xFFFFFFFF to USB_nENDPTFLUSH.

Read the reset bit in the USB_nPORTSC1 register and make sure that it is still active. A USB reset will occur for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before end of the reset occurs, otherwise a hardware reset of the device controller is recommended (rare.)

- A hardware reset can be performed by writing a one to the device controller reset bit in the USBCMD reset. Note: a hardware reset will cause the device to detach from the bus by clearing the Run/Stop bit. Thus, the DCD must completely re-initialize the device controller after a hardware reset.

Free all allocated dTDs because they will no longer be executed by the device controller. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs have been allocated.

At this time, the DCD may release control back to the OS because no further changes to the device controller are permitted until a Port Change Detect is indicated.

After a Port Change Detect, the device has reached the default state and the DCD can read the USB_nPORTSC1 to determine if the device is operating in FS or HS mode. At this time, the device controller has reached normal operating mode and DCD can begin enumeration according to the USB Chapter 9 - Device Framework.

NOTE

The device DCD may use the FS/HS mode information to determine the bandwidth mode of the device

In some applications, it may not be possible to enable one or more pipes while in FS mode. *Beyond the data rate issue, there is no difference in DCD operation between FS and HS modes.*

62.5.6.2.2 Suspend/Resume

The detials of suspend and resume are explained in these sections.

62.5.6.2.2.1 Suspend

Suspend Description

In order to conserve power, USB devices automatically enter the suspended state when the device has observed no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. Attached devices must be prepared to suspend at any time they are powered, regardless of if they have been assigned a non-default address, are configured, or neither. Bus activity may cease due to the host entering a suspend mode of its own. In addition, a USB device shall also enter the suspended state when the hub port it is attached to is disabled.

A USB device exits suspend mode when there is bus activity. A USB device may also request the host to exit suspend mode or selective suspend by using electrical signaling to indicate remote wakeup. The ability of a device to signal remote wakeup is optional. If the USB device is capable of remote wakeup signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wakeup signaling must be disabled.

Suspend Operational Model

The device controller moves into the suspend state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the suspend state, the DCD is notified by an interrupt (assuming *DC Suspend Interrupt* is enabled). When the *DCSuspend* bit in the USB_nPORTSC1 is set to a '1', the device controller is suspended.

DCD response when the device controller is suspended is application specific and may involve switching to low power operation.

Information on the bus power limits in suspend state can be found in USB 2.0 specification.

NOTE

Review system level clocking issues defined in section (Ref: Signals-Clocking) for the clocking requirements of a suspended device controller.

62.5.6.2.2.2 Resume

If the device controller is suspended, its operation is resumed when any non-idle signaling is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port.

Resume signaling is sent upstream by writing a '1' to the Resume bit in the in the USB_nPORTSC1 while the device is in suspend state. Sending resume signal to an upstream port should cause the host to issue resume signaling and bring the suspended bus segment (one more devices) back to the active condition.

NOTE

Before resume signaling can be used, the host must enable it by using the Set Feature command defined in device framework (chapter 9) of the USB 2.0 Specification.

62.5.6.3 Managing Endpoints

The USB 2.0 specification defines an endpoint, also called a device endpoint or an address endpoint as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device.

The endpoint address is specified by the combination of the endpoint number and the endpoint direction.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a *control* type data channel used for device discovery and enumeration. Other types of endpoints support by USB include *bulk, interrupt*, and *isochronous*. Each endpoint type has specific behavior related to packet response and error handling. More detail on endpoint operation can be found in the USB 2.0 specification.

The USB OTG device controller hardware supports up to 8 endpoint numbers.

Each endpoint direction is essentially independent and can be configured with differing behavior in each direction. For example, the DCD can configure endpoint 1-IN to be a bulk endpoint and endpoint 1-OUT to be an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, is always a control endpoint and uses the pair of directions.

Each endpoint direction requires a *queue head* allocated in memory. To support the 8 endpoint numbers, 16 *queue heads* are required. The operation of an endpoint and use of *queue heads* are described later in this document.

62.5.6.3.1 Endpoint Initialization

After hardware reset, all endpoints except endpoint zero are uninitialized and disabled. The DCD must configure and enable each endpoint by writing to configuration bit in the USB_UOG_ENDPTCTRLx register.

Each 32-bit USB_UOG_ENDPTCTRLx is split into an upper and lower half. The lower half of USB_UOG_ENDPTCTRLx is used to configure the receive or OUT endpoint and the upper half is likewise used to configure the corresponding transmit or IN endpoint. Control endpoints must be configured the same in both the upper and lower half of the USB_UOG_ENDPTCTRLx register otherwise the behavior is undefined. The following table shows how to construct a configuration word for endpoint initialization. The following table shows the fields and values for the Device Controller Endpoint initialization.

Table 62-67. Device Controller Endpoint Initialization

Field	Value
Data Toggle Reset	1
Data Toggle Inhibit	0
Endpoint Type	00 Control 01 Isochronous 10 Bulk

Table continues on the next page...

Table 62-67. Device Controller Endpoint Initialization (continued)

	11 Interrupt
Endpoint Stall	0

62.5.6.3.2 Stalling

There are two occasions where the device controller may need to return to the host a STALL.

The first occasion is the functional stall, which is a condition set by the DCD as described in the USB 2.0 device framework. A functional stall is only used on non-control endpoints and can be enabled in the device controller by setting the endpoint stall bit in the USB_UOG_ENDPTCTRLx register associated with the given endpoint and the given direction. In a functional stall condition, the device controller will continue to return STALL responses to all transactions occurring on the respective endpoint and direction until the endpoint stall bit is cleared by the DCD.

A protocol stall, unlike a function stall, is used on control endpoints is automatically cleared by the device controller at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD should enable the stall bits (both directions) as a pair. A single write to the USB_UOG_ENDPTCTRLx register can ensure that both stall bits are set at the same instant.

NOTE

Any write to the USB_UOG_ENDPTCTRLx register during operational mode must preserve the endpoint type field (that is, perform a read-modify-write).

The following table shows the response matrix for the Device Controller Stall.

Table 62-68. Device Controller Stall Response Matrix

USB Packet	Endpoint Stall Bit.	Effect on STALL bit.	USB Response
SETUP packet received by a non-control endpoint.	N/A	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'1'	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'0'	None.	ACK/ NAK/ NYET
SETUP packet received by a control endpoint.	N/A	Cleared	ACK
IN/OUT/PING packet received by a control endpoint	'1'	None	STALL

Table continues on the next page...

Table 62-68. Device Controller Stall Response Matrix (continued)

IN/OUT/PING packet received by a control endpoint.	'0'	None.	ACK/ NAK/ NYET

62.5.6.3.3 Data Toggle

Data toggle is a mechanism to maintain data coherency between host and device for any given data pipe.

For more information on data toggle, refer to the USB 2.0 specification.

62.5.6.3.3.1 Data Toggle Reset

The DCD may reset the data toggle state bit and cause the data toggle sequence to reset in the device controller by writing a '1' to the data toggle reset bit in the USB_UOG_ENDPTCTRLx register.

This should only be necessary when configuring/initializing an endpoint or returning from a STALL condition.

62.5.6.3.3.2 Data Toggle Inhibit

NOTE

This feature is for test purposes only and should never be used during normal device controller operation.

Setting the *data toggle Inhibit bit* active ('1') causes the device controller to ignore the data toggle pattern that is normally sent and accept all incoming data packets regardless of the data toggle state.

In normal operation, the device controller checks the DATA0/DATA1 bit against the data toggle to determine if the packet is valid. If Data PID does not match the data toggle state bit maintained by the device controller for that endpoint, the Data toggle is considered not valid. If the data toggle is not valid, the device controller assumes the packet was already received and discards the packet (not reporting it to the DCD). To prevent the host controller from re-sending the same packet, the device controller will respond to the error packet by acknowledging it with either an ACK or NYET response.

62.5.6.3.3.3 Priming Transmit Endpoints

Priming a transmit endpoint will cause the device controller to fetch the device transfer descriptor (dTD) for the transaction pointed to by the device queue head (dQH).

After the dTD is fetched, it will be stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint up to the maximum number of endpoints configured at device synthesis time.

After a priming request is complete, an endpoint state of primed is indicated in the USB_UOG_ENDPTSTATUS register. For a primed transmit endpoint, the device controller can respond to an IN request from the host and meet the stringent bus turnaround time of High Speed USB.

Because only the leading data is stored in the device controller FIFO, it is necessary for the device controller to begin filling in behind leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that can be incurred by the system memory bus. More information about FIFO sizing is presented in section .

62.5.6.3.3.4 Priming Receive Endpoints

Priming receive endpoints is identical to priming of transmit endpoints from the point of view of the DCD. At the device controller the major difference in the operational model is that there is no data movement of the leading packet data simply because the data is to be received from the host.

Note as part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the RX FIFO does not scale with the number of endpoints.

62.5.6.4 Operational Model For Packet Transfers

All transactions on the USB bus are initiated by the host and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 Specification.

At USB 1.1 Full or Low Speed rates, this turnaround time was significant and the USB 1.1 device controllers were architected so that the device controller could access main memory or interrupt a host protocol processor in order to respond to the USB 1.1

transaction. The architecture of the USB 2.0 device controller must be different because same methods will not meet USB 2.0 High-speed turnaround time requirements by simply increasing clock rate.

A USB host will send requests to the device controller in an order that can not be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable when the endpoint number and direction is considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then we can expect the host will send IN requests to that endpoint. This device controller is architected in such a way that it can prepare packets for each endpoint/direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host initiated transaction on the bus is referred to as "priming" the endpoint. This term will be used throughout the following documentation to describe the device controller operation so the DCD can be architected properly use priming. Further, note that the term "flushing" is used to describe the action of clearing a packet that was queued for execution.

62.5.6.4.1 Interrupt/Bulk Endpoint Operational Model

The behaviors of the device controller for interrupt and bulk endpoints are identical.

All valid IN and OUT transactions to bulk pipes will handshake with a NAK unless the endpoint had been primed. Once the endpoint has been primed, data delivery will commence.

A dTD will be retired by the device controller when the packets described in the transfer descriptor have been completed. Each dTD describes N packets to be transferred according to the USB Variable Length transfer protocol. The formula and table on the following page describe how the device controller computes the number and length of the packets to be sent/received by the USB vary according to the total number of bytes and maximum packet length.

With Zero Length Termination (ZLT) = 0

N = INT(Number Of Bytes/Max. Packet Length) + 1

With Zero Length Termination (ZLT) = 1

N = MAXINT(Number Of Bytes/Max. Packet Length)

Table 62-69. Variable Length Transfer Protocol Example (ZLT = 0)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	

Table continues on the next page...

Table 62-69. Variable Length Transfer Protocol Example (ZLT = 0) (continued)

512	256	3	256	256	0
512	512	2	512	0	

Table 62-70. Variable Length Transfer Protocol Example (ZLT = 1)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	2	256	256	
512	512	1	512		

NOTE

The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.

TX-dTD is complete when:

- All packets described dTD were successfully transmitted. *** Total bytes in dTD will equal zero when this occurs.

RX-dTD is complete when:

- All packets described in dTD were successfully received. *** Total bytes in dTD will equal zero when this occurs.
- A short packet (number of bytes < maximum packet length) was received. *** This is a successful transfer completion; DCD must check Total Bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received.
- A long packet was received (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). *** This is an error condition. The device controller will discard the remaining packet, and set the Buffer Error bit in the dTD. In addition, the endpoint will be flushed and the USBERR interrupt will become active.

On the successful completion of the packet(s) described by the dTD, the active bit in the dTD will be cleared and the next pointer will be followed when the Terminate bit is clear. When the Terminate bit is set, the device controller will flush the endpoint/direction and cease operations for that endpoint/direction.

On the unsuccessful completion of a packet (see long packet above), the dQH will be left pointing to the dTD that was in error. In order to recover from this error condition, the DCD must properly reinitialize the dQH by clearing the active bit and update the nextTD pointer before attempting to re-prime the endpoint.

NOTE

All packet level errors such as a missing handshake or CRC error will be retried automatically by the device controller.

There is no required interaction with the DCD for handling such errors.

62.5.6.4.1.1 Interrupt/Bulk Endpoint Bus Response Matrix

The table below shows the response matrix for Interrput/Bulk Endpoint Bus.

Table 62-71. Interrupt/Bulk Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	BS Error	N/A
Out	STALL	NAK	Receive + NYET/ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

NOTE

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

62.5.6.4.2 Control Endpoint Operation Model

This section details the setup phase, data phase, status phase, and the control endpoint bus response matrix.

62.5.6.4.2.1 Setup Phase

All requests to a control endpoint begin with a setup phase followed by an optional data phase and a required status phase. The device controller will always accept the setup phase unless the setup lockout is engaged.

The setup lockout will engage so that future setup packets are ignored. Lockout of setup packets ensures that while software is reading the setup packet stored in the queue head, that data is not written as it is being read potentially causing an invalid setup packet.

The setup lockout mechanism can be disabled and a new tripwire type semaphore will ensure that the setup packet payload is extracted from the queue head without being corrupted be an incoming setup packet. This is the preferred behavior because ignoring repeated setup packets due to long software interrupt latency would be a compliance issue.

- Disable Setup Lockout by writing 1 to Setup Lockout Mode (SLOM) in USB_nUSBMODE. (once at initialization). Setup lockout is not necessary when using the tripwire as described below.

NOTE

Leaving the Setup Lockout Mode As 0 will result in pre-2.3 hardware behavior.

- After receiving an interrupt and inspecting UBS_nENDSETUPSTAT to determine that a setup packet was received on a particular pipe:
  a. Write 1 to clear corresponding bit UBS_nENDSETUPSTAT.
  b. Write 1 to Setup Tripwire (SUTW) in USB_nUSBCMD register.
  c. Duplicate contents of dQH.SetupBuffer into local software byte array.
  d. Read Setup TripWire (SUTW) in USB_nUSBCMD register. (if set - continue; if cleared - goto 2)
  e. Write 0 to clear Setup Tripwire (SUTW) in USB nUSBCMD register.
  f. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

62.5.6.4.2.2 Data Phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must verify a new setup packet has not been received by reading the USB.ENDPTSETUPSTAT register immediately verifying that the prime had completed. A prime will complete when the associated bit in the USB_nENDPTPRIME register is zero and the associated bit in the USB_nENDPTSTAT register is a one. If a prime fails, ie. The USB_nENDPTPRIME bit goes to zero and the USB_nENDPTSTAT bit is not set, then the prime has failed. This can only be due to

improper setup of the dQH, dTD or a setup arriving during the prime operation. If a new setup packet is indicated after the ENDPTPRIME bit is cleared, then the transfer descriptor can be freed and the DCD must reinterpret the setup packet.

Should a setup arrive after the data stage is primed, the device controller will automatically clear the prime status (USB_nENDPTSTAT) to enforce data coherency with the setup packet.

NOTE

- The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

62.5.6.4.2.3 Status Phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal zero) and prime the endpoint for the status phase.

The DCD must also perform the same checks of the USB.ENDPTSETUPSTAT as described above in the data phase.

NOTE

- The MULT field in the dQH must be set to 00 for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

62.5.6.4.2.4 Control Endpoint Bus Response Matrix

Shown in the following table is the device controller response to packets on a control endpoint according to the device controller state.

The table below shows the response matrix for the Control Endpoint Bus.

Table 62-72. Control Endpoint Bus Response Matrix

Token Type	Endpoint State					Setup Lockout
	Stall	Not Primed	Primed	Underflow	Overflow	
Setup	ACK	ACK	ACK	N/A	SYSERR	
In	STALL	NAK	Transmit	BS Error	N/A	N/A
Out	STALL	NAK	Receive + NYET/ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

62.5.6.4.3 Isochronous Endpoint Operational Model

Isochronous endpoints are used for real-time scheduled delivery of data and their operational model is significantly different than the host throttled Bulk, Interrupt, and Control data pipes.

Real time delivery by the device controller will be accomplished by the following:

- Exactly MULT Packets per (micro) Frame are transmitted/received. Note: MULT is a two-bit field in the device Queue Head. The variable length packet protocol is not used on isochronous endpoints.
- NAK responses are not used. Instead, zero length packets are sent in response to an IN request to an unprimed endpoints. For unprimed RX endpoints, the response to an OUT transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next (micro) frame. If the ISO-dTD is still active after that frame, then the ISO-dTD will be held ready until executed or canceled by the DCD.

An EHCI compatible host controller uses the periodic frame list to schedule data exchanges to Isochronous endpoints. The operational model for device mode does not use such a data structure. Instead, the same dTD used for Control/Bulk/Interrupt endpoints is also used for isochronous endpoints. The difference is in the handling of the dTD.

The first difference between bulk and ISO-endpoints is that priming an ISO-endpoint is a delayed operation such that an endpoint will become primed only after a SOF is received. After the DCD writes the prime bit, the prime bit will be cleared as usual to indicate to software that the device controller completed a priming the dTD for transfer. Internal to the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific (micro)frame.

Another difference with isochronous endpoints is that the transaction must wholly complete in a (micro)frame. Once an ISO transaction is started in a (micro)frame it will retire the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

The transaction error bit set in the status field indicates a fulfillment error condition. When a fulfillment error occurs, the frame after the transfer failed to complete wholly, the device controller will force retire the ISO-dTD and move to the next ISO-dTD.

It is important to note that fulfillment errors are only caused due to partially completed packets. If no activity occurs to a primed ISO-dTD, the transaction will stay primed indefinitely. This means it is up to software discard transmit ISO-dTDs that pile up from a failure of the host to move the data.

Finally, the last difference with ISO packets is in the data level error handling. When a CRC error occurs on a received packet, the packet is not retried similar to bulk and control endpoints. Instead, the CRC is noted by setting the *Transaction Error* bit and the data is stored as usual for the application software to sort out.

- TX Packet Retired
  - MULT counter reaches zero.
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT

NOTE

For TX-ISO, MULT Counter can be loaded with a lesser value in the dTD Multiplier Override field in hardware versions 2.3 and later. If the Multiplier Override is zero, the MULT Counter is initialized to the Multiplier in the QH.

- RX Packet Retired:
  - MULT counter reaches zero.
  - Non-MDATA Data PID is received**
    - ** Exit criteria only valid in hardware version 2.3 or later. Previous to hardware version 2.3, any PID sequence that did not match the MULT field exactly would be flagged as a transaction error due to PID mismatch or fulfillment error.
  - Overflow Error:
    - Packet received is > maximum packet length. [*Buffer Error* bit is set]
    - Packet received exceeds total bytes allocated in dTD. [*Buffer Error* bit is set]
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT
  - CRC Error [*Transaction Error* bit is set]

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For continuous (micro)frame to (micro)frame

operation the DCD should ensure that the dTD linked-list is out ahead of the device controller by at least two (micro)frames.

62.5.6.4.3.1 Isochronous Pipe Synchronization

When it is necessary to synchronize an isochronous data pipe to the host, the (micro) frame number (USB_UOG_FRINDEX register) can be used as a marker.

To cause a packet transfer to occur at a specific (micro) frame number [N], the DCD should interrupt on SOF during frame N-1. When the USB_UOG_FRINDEX=N-1, the DCD must write the prime bit. The device controller will prime the isochronous endpoint in (micro) frame N-1 so that the device controller will execute delivery during (micro) frame N.

NOTE

Priming an endpoint towards the end of (micro) frame N-1 will not guarantee delivery in (micro) frame N. The delivery may actually occur in (micro) frame N+1 if device controller does not have enough time to complete the prime before the SOF for packet N is received.

62.5.6.4.3.2 Isochronous Endpoint Bus Response Matrix

The following table shows the response matrix for the Isochronous Endpoint Bus.

Table 62-73. Isochronous Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	STALL	STALL	STALL	N/A	N/A
In	NULL Packet	NULL Packet	Transmit	BS Error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop Packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

1. BS Error = Force Bit Stuff Error

   NULL Packet = Zero Length Packet

62.5.6.5 Managing Queue Heads

The following figure shows the End Point Queue Head.

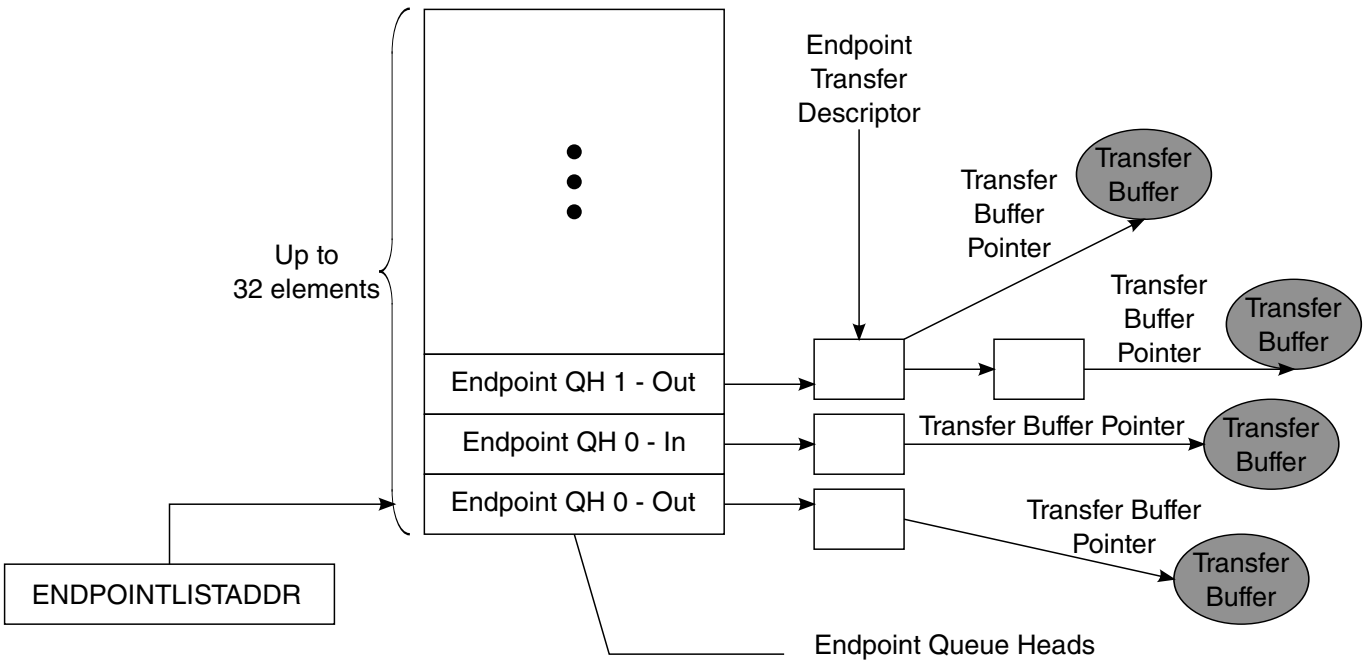


Figure 62-30. End Point Queue Head Diagram

The device queue head (dQH) points to the linked list of transfer tasks, each depicted by the device Transfer Descriptor (dTD). An area of memory pointed to by USB.ENDPOINTLISTADDR contains a group of all dQH's in a sequential list as shown in Figure 62-30. The even elements in the list of dQH's are used for receive endpoints (OUT/SETUP) and the odd elements are used for transmit endpoints (IN/INTERRUPT). Device transfer descriptors are linked head to tail starting at the queue head and ending at a terminate bit. Once the dTD has been retired, it will no longer be part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers will no longer exist within the queue head once the dTD is retired (see section Software Link Pointers).

In addition to the current and next pointers and the dTD overlay examined in section Operational Model For Packet Transfers, the dQH also contains the following parameters for the associated endpoint: Multipler, Maximum Packet Length, Interrupt On Setup. The complete initialization of the dQH including these fields is demonstrated in the next section.

62.5.6.5.1 Queue Head Initialization

One device queue head must be initialized for each active endpoint.

To initialize a device queue head:

- Write the wMaxPacketSize field as required by the USB Chapter 9 or application specific protocol.
- Write the multiplier field to 0 for control, bulk, and interrupt endpoints. For ISO endpoints, set the multiplier to 1,2, or 3 as required bandwidth and in conjunction with the USB Chapter 9 protocol.

NOTE

In FS mode, the multiplier field can only be 1 for ISO endpoints.

- Write the next dTD Terminate bit field to 1.
- Write the Active bit in the status field to 0.
- Write the Halt bit in the status field to 0.

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTD's.

62.5.6.5.2 Operational Model For Setup Transfers

As discussed in section Control Endpoint Operation Model, setup transfer requires special treatment by the DCD. A setup transfer does not use a dTD but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

Upon receiving notification of the setup packet, the DCD should handle the setup transfer as demonstrated here:

1. Copy setup buffer contents from dQH - RX to software buffer.
2. Acknowledge setup backup by writing a "1" to the corresponding bit in ENDPTSETUPSTAT.

NOTE

- The acknowledge must occur before continuing to process the setup packet.
- After the acknowledge has occurred, the DCD must not attempt to access the setup buffer in the dQH - RX. Only the local software copy should be examined.

3. Check for pending data or status dTD's from previous control transfers and flush if any exist as discussed in section Flushing/De-priming an Endpoint.
4. Decode setup packet and prepare data phase [optional] and status phase transfer as required by the USB Chapter 9 or application specific protocol.

NOTE

It is possible for the device controller to receive setup packets before previous control transfers complete. Existing control packets in progress must be flushed and the new control packet completed.

62.5.6.6 Managing Transfers with Transfer Descriptors

These sections detail how to build, execute, and complete transfers with transfer descriptors.

62.5.6.6.1 Software Link Pointers

It is necessary for the DCD software to maintain head and tail pointers to the for the linked list of dTDs for each respective queue head.

This is necessary because the dQH only maintains pointers to the current working dTD and the next dTD to be executed. The operations described in next section for managing dTD will assume the DCD can use reference the head and tail of the dTD linked list. The following figure shows the Software Link Pointers.

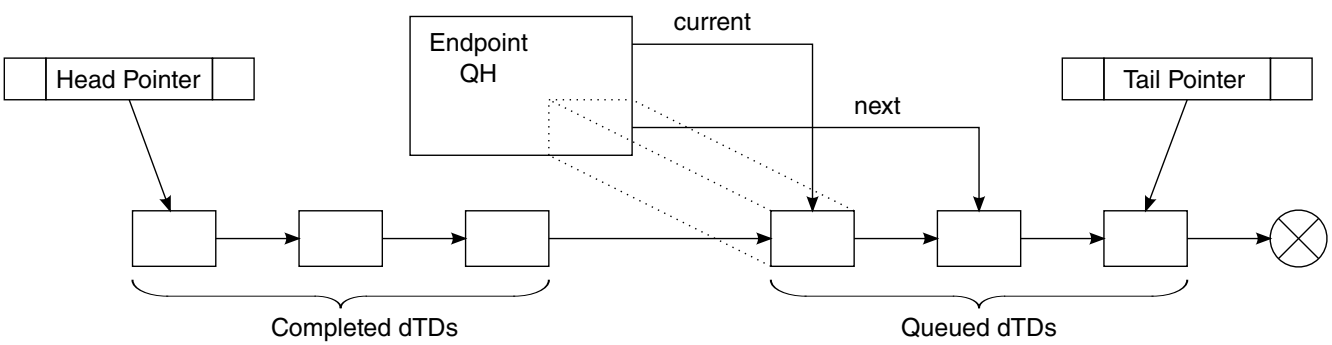


Figure 62-31. Software Link Pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the Head & Tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

62.5.6.6.2 Building a Transfer Descriptor

Before a transfer can be executed from the linked list, a dTD must be built to describe the transfer.

Use the following procedure for building dTDs.

Allocate 8-DWord dTD block of memory aligned to 8-DWord boundaries. Example: bit address 4:0 would be equal to "00000"

Write the following fields:

1. Initialize first 7 DWords to 0.
2. Set the terminate bit to 1.
3. Fill in total bytes with transfer size.
4. Set the interrupt on complete if desired.
5. Initialize the status field with the active bit set to 1 and all remaining status bits set to 0.
6. Fill in buffer pointer page 0 and the current offset to point to the start of the data buffer.
7. Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointer.

62.5.6.6.3 Executing A Transfer Descriptor

To safely add a dTD, the DCD must be follow this procedure which will handle the event where the device controller reaches the end of the dTD list at the same time a new dTD is being added to the end of the list.

Determine whether the link list is empty: Check DCD driver to see if pipe is empty (internal representation of linked-list should indicate if any packets are outstanding).

- Case 1: Link list is empty
  a. Write dQH next pointer AND dQH terminate bit to 0 as a single DWord operation.
  b. Clear active & halt bit in dQH (in case set from a previous error).
  c. Prime endpoint by writing 1 to correct bit position in ENDPTPRIME. Software should check that the dTD/dQH content in external memory is updated before setting the bit in ENDPTPRIME register.
- Case 2: Link list is not empty
  a. Add dTD to end of linked list.
  b. Read correct prime bit in ENDPTPRIME - if 1 DONE.
  c. Set ATDTW bit in USBCMD register to 1.
  d. Read correct status bit in ENDPTSTAT. (store in tmp. variable for later)
  e. Read ATDTW bit in USBCMD register.

- If 0 goto 3.
- If 1 continue to 6.

f. Write ATDTW bit in USBCMD register to 0.
g. If status bit read in (3) is 1 DONE.
h. If status bit read in (3) is 0 then Goto Case 1: Step 1.

62.5.6.6.4 Transfer Completion

After a dTD has been initialized and the associated endpoint primed the device controller will execute the transfer upon the host-initiated request. The DCD will be notified with a USB interrupt if the Interrupt On Complete bit was set or alternately, the DCD can poll the endpoint complete register to find when the dTD had been executed. After a dTD has been executed, DCD can check the status bits to determine success or failure.

NOTE

Multiple dTD can be completed in a single endpoint complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that have finished (Active bit cleared).

By reading the status fields of the completed dTDs, the DCD can determine if the transfers completed successfully. Success is determined with the following combination of status bits:

- Active = 0
- Halted = 0
- Transaction Error = 0
- Data Buffer Error = 0

Should any combination other than the one shown above exist, the DCD must take proper action. Transfer failure mechanisms are indicated in the Device Error Matrix.

In addition to checking the status bit the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the Total Bytes transferred is by decremented by the actual bytes transferred. For Transmit packets, a packet is only complete after the actual bytes reaches zero, but for receive packets, the host may send fewer bytes in the transfer according the USB variable length packet protocol.

62.5.6.6.5 Flushing/De-priming an Endpoint

It is necessary for the DCD to flush to de-prime one more endpoints on a USB device reset or during a broken control transfer.

There may also be application specific requirements to stop transfers in progress. The following procedure can be used by the DCD to stop a transfer in progress:

1. Write a '1' to the corresponding bit(s) in ENDPTFLUSH.
2. Wait until all bits in ENDPTFLUSH are '0'.
   - Software note: this operation may take a large amount of time depending on the USB bus activity. It is not desirable to have this wait loop within an interrupt service routine.
3. Read ENDPTSTAT register to ensure that for all endpoints commanded to be flushed, that the corresponding bits are now '0'. If the corresponding bits are '1' after step #2 has finished, then the flush failed as described in the following:
   - Explanation: In very rare cases, a packet is in progress to the particular endpoint when commanded flush using ENDPTFLUSH register. A safeguard is in place to refuse the flush to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush be repeating steps 1-3 until each endpoint is successfully flushed.

62.5.6.6.6 Device Error Matrix

The following table summarizes packet errors that are not automatically handled by the Device Controller.

Table 62-74. Device Error Matrix

Error	Direction	Packet Type	Data Buffer Error Bit	Transaction Error Bit
Overflow **	RX	Any	1	0
ISO Packet Error	RX	ISO	0	1
ISO Fulfillment Error	Both	ISO	0	1

Notice that the device controller handles all errors on Bulk/Control/Interrupt Endpoints except for a data buffer overflow. However, for ISO endpoints, errors packets are not retried and errors are tagged as indicated. The table below describes the errors.

Table 62-75. Error Descriptions

Error	Description
Overflow	Number of bytes received exceeded max. packet size or total buffer length. ** This error will also set the Halt bit in the dQH and if there are dTDs remaining in the linked list for the endpoint, then those will not be executed.
ISO Packet Error	CRC Error on received ISO packet. Contents not guaranteed to be correct.
ISO Fulfillment Error	Host failed to complete the number of packets defined in the dQH mult field within the given (micro)frame. For scheduled data delivery the DCD may need to readjust the data queue because a fulfillment error will cause Device Controller to cease data transfers on the pipe for one (micro)frame. During the "dead" (micro)frame, the Device Controller reports error on the pipe and primes for the following frame.

62.5.6.7 Servicing Interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency operations, and error operations and order accordingly.

62.5.6.7.1 High-Frequency Interrupts

High frequency interrupts in particular should be handled in the order below. The most important of these is listed first because the DCD must acknowledge a setup buffer in the timeliest manner possible.

The table below describes the High frequency interrupt events.

Table 62-76. High Frequency Interrupt Events

Execution Order	Interrupt	Action
1a	USB Interrupt - USB.ENDPTSETUPSTATUS	Copy contents of setup buffer and acknowledge setup packet (as indicated in Figure 62-30 shows the End Point Queue Head). Process setup packet according to USB 2.0 Chapter 9 or application specific protocol.
1b	USB Interrupt[1] - USB.ENDPTCOMPLETE	Handle completion of dTD as indicated in Figure 62-30 shows the End Point Queue Head.
2	SOF Interrupt	Action as deemed necessary by application. This interrupt may not have a use in all applications.

1. It is likely that multiple interrupts to stack up on any call to the Interrupt Service Routine AND during the Interrupt Service Routine.

62.5.6.7.2 Low-Frequency Interrupts

The low frequency interrupts can be handled in any order because they do not occur often in comparison to the high-frequency interrupts.

The table below shows the Low frequency interrupt events.

Table 62-77. Low Frequency Interrupt Events

Interrupt	Action
Port Change	Change software state information.
Sleep Enable (Suspend)	Change software state information. Low power handling as necessary.
Reset Received	Change software state information. Abort pending transfers.

62.5.6.7.3 Error Interrupts

Error interrupts will be least frequent and should be placed last in the interrupt service routine.

The following table shows the error interrput events.

Table 62-78. Error Interrupt Events

Interrupt	Action
USB Error Interrupt	This error is redundant because it combines USB Interrupt and an error status in the dTD. The DCD will more aptly handle packet-level errors by checking dTD status field upon receipt of USB Interrupt (w/ USB.ENDPTCOMPLETE).
System Error	Unrecoverable error. Immediate Reset of core; free transfers buffers in progress and restart the DCD.

62.6 USB Non-Core Memory Map/Register Definition

62.6.1 USBNC register descriptions

There are two kinds of registers in the USB module: USB core registers and USB non-core registers. USB core registers are used to control USB core functions, and more independent of USB features. Each USB controller core has its own core registers. USB non-core registers are additional to USB core registers, and more dependent on USB features. i.MX series products vary in non-core registers.

This section describes only the USB non-core registers. For detailed descriptions of USB core registers, please refer to Register Interface .

NOTE

- For reserved bits, please preserve the value when writing (read its reset value, then write this value back)

62.6.1.1 USBNC memory map

USBNC_OTG1 base address: 4043_0200h

USBNC_OTG2 base address: 4042_C200h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	USB OTG Control 1 Register (CTRL1)	32	RW	3000_1000
4	USB OTG Control 2 Register (CTRL2)	32	RW	5F00_0000
10	USB Host HSIC Control Register (HSIC_CTRL)	32	RW	1000_4084

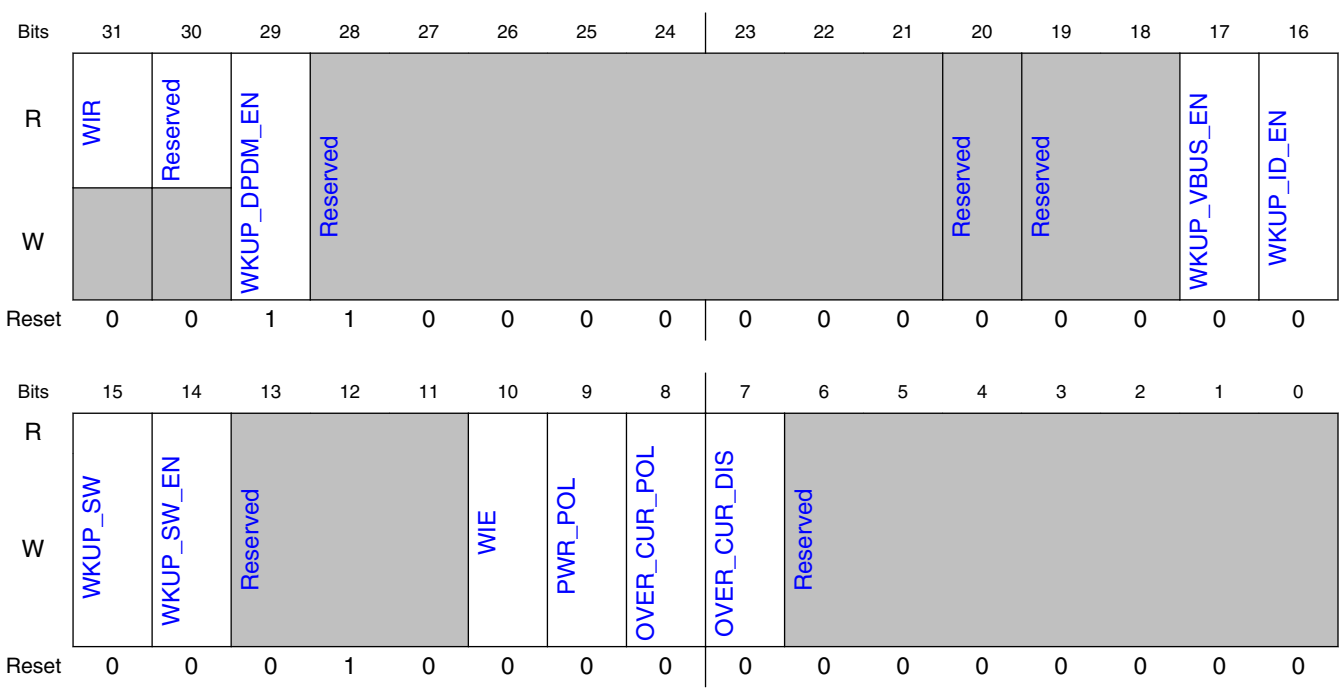
62.6.1.2 USB OTG Control 1 Register (CTRL1)

The USB Control 1 register controls the integration specific features of the USB OTG1 and OTG2 module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

62.6.1.2.1 Offset

Register	Offset
CTRL1	0h

62.6.1.2.2 Diagram



62.6.1.2.3 Fields

Field	Description
31 WIR	WIR Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the OTG1 port. This bit is cleared by disabling the wake-up interrupt (clearing bit "OWIE"). 0 - No wake-up interrupt request received 1 - Wake-up Interrupt Request received
30 —	Reserved
29 WKUP_DPDM_EN	Wake-up on DPDM change enable 0 - DPDM changes wake-up to be disabled only when VBUS is 0. 1 - (Default) DPDM changes wake-up to be enabled, it is for device only.
28-21 —	Reserved
20 —	- Reserved
19-18 —	- Reserved
17 WKUP_VBUS_EN	WKUP_VBUS_EN Wake-up on VBUS change enable 0 - Disable 1 - Enable
16 WKUP_ID_EN	WKUP_ID_EN Wake-up on ID change enable 0 - Disable 1 - Enable
15 WKUP_SW	WKUP_SW Software Wake-up 0 - Inactive 1 - Force wake-up
14 WKUP_SW_EN	WKUP_SW_EN Software Wake-up Enable 0 - Disable 1 - Enable
13-11 —	- Reserved
10 WIE	WIE Wake-up Interrupt Enable

Table continues on the next page...

Field	Description
	This bit enables or disables the OTG1 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 0 - Interrupt Disabled 1 - Interrupt Enabled
9 PWR_POL	PWR_POL Power Polarity This bit should be set according to PMIC Power Pin polarity. 0 - PMIC Power Pin is Low active. 1 - PMIC Power Pin is High active.
8 OVER_CUR_POL	OVER_CUR_POL Polarity of Overcurrent The polarity of OTG1/OTG2 port overcurrent event 0 - High active (high on this signal represents an overcurrent condition) 1 - Low active (low on this signal represents an overcurrent condition)
7 OVER_CUR_DIS	OVER_CUR_DIS Disable Overcurrent Detection 0 - Enables overcurrent detection 1 - Disables overcurrent detection
6-0 —	- Reserved

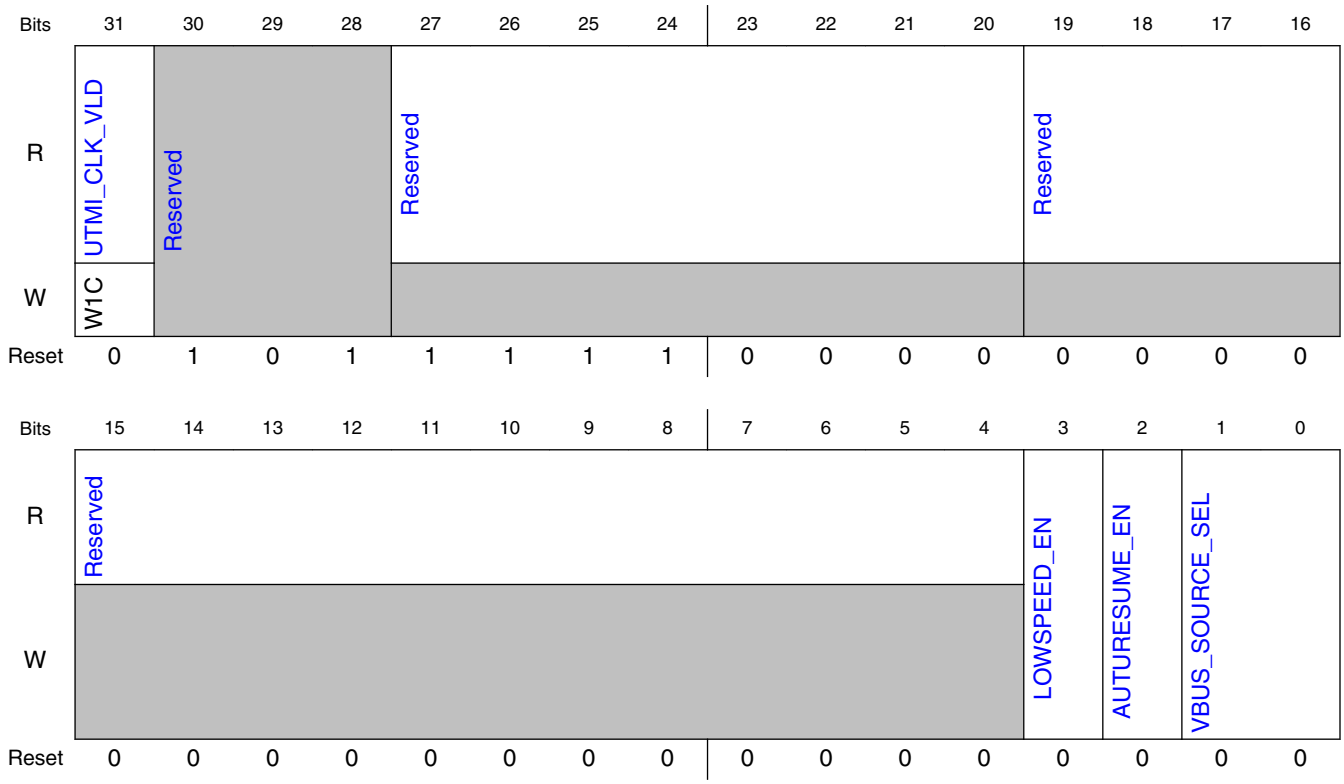
62.6.1.3 USB OTG Control 2 Register (CTRL2)

The USB Control 2 register controls the integration specific features of the USB module. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wakeup functionality.

62.6.1.3.1 Offset

Register	Offset
CTRL2	4h

62.6.1.3.2 Diagram



62.6.1.3.3 Fields

Field	Description
31 UTMI_CLK_VLD	UTMI_CLK_VLD Indicate whether the UTMI clock to the USB PHY is valid. Write 1 to clear 0 - Default
30-28 —	Reserved
27-20 —	Reserved 00011110 - Default
19-4 —	Reserved
3 LOWSPEED_EN	LOWSPEED_EN Set if AUTURESUME_EN is set and works on low speed. 0 - Default
2 AUTURESUME_EN	Auto Resume Enable

Table continues on the next page...

Field	Description
	It is for UTMI PHY host mode only (UH core has no this feature since HSIC PHY does not support auto resume). To set USB, it will send resume with 32 KHz clock when it detects remote wakeup from device. This feature is useful if device drive a very short remote wakeup (minimul value is 1 ms for USB2 spec) and system 24 MHz is off during suspend mode. 0 - Default
1-0 VBUS_SOURCE_SEL	VBUS_SOURCE_SEL VBUS source select when detect VBUS wakeup event, it is for UTMI PHY only (UH core has no such feature). 00 - vbus_valid 01 - sess_valid 10 - sess_valid 11 - sess_valid

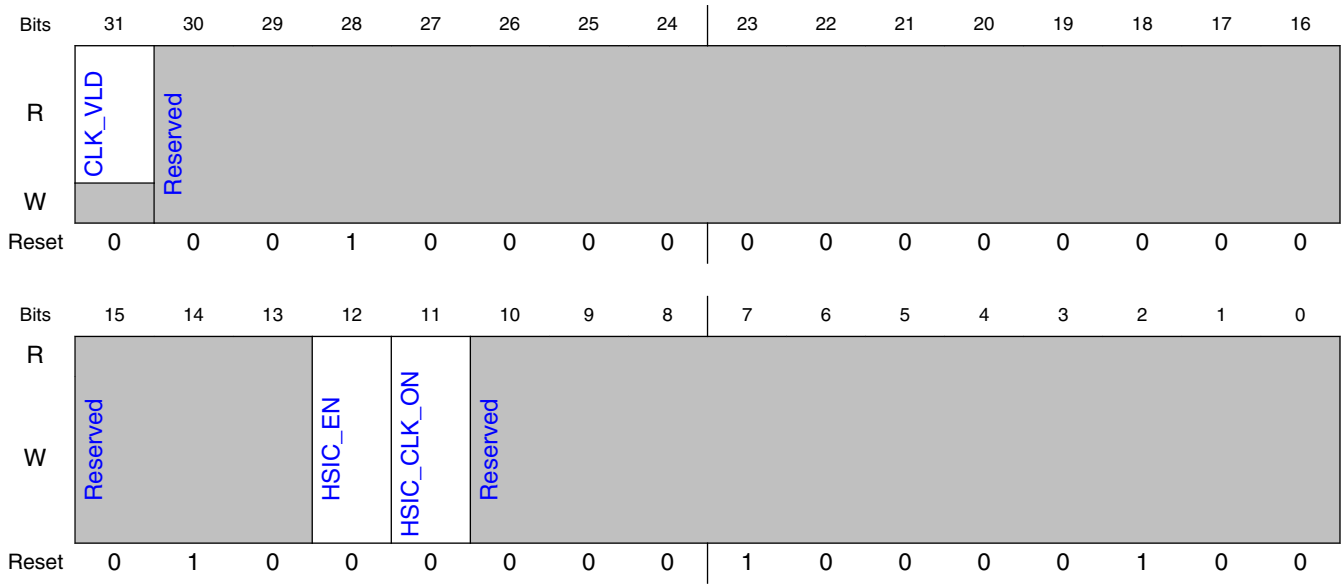
62.6.1.4 USB Host HSIC Control Register (HSIC_CTRL)

The USB Host HSIC control register controls Host high speed IC configuration. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control.

62.6.1.4.1 Offset

Register	Offset
HSIC_CTRL	10h

62.6.1.4.2 Diagram



62.6.1.4.3 Fields

Field	Description
31 CLK_VLD	CLK_VLD Indicating whether Host HSIC clock is valid. 0 - Invalid 1 - Valid
30-13 —	- Reserved
12 HSIC_EN	HSIC_EN Host HSIC enable 0 - Disabled 1 - Enabled
11 HSIC_CLK_ON	HSIC_CLK_ON Force Host HSIC module 480M clock on, even when in Host is in suspend mode. 0 - Inactive 1 - Active
10-0 —	- Reserved

62.7 USB Core Memory Map/Register Definition

62.7.1 USB register descriptions

62.7.1.1 USB memory map

USB_OTG1 base address: 4043_0000h

USB_OTG2 base address: 4042_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Identification register (ID)	32	RO	E4A1_FA05
4	Hardware General (HWGENERAL)	32	RO	0000_0015
8	Host Hardware Parameters (HWHOST)	32	RO	1002_0001
C	Device Hardware Parameters (HWDEVICE)	32	RO	0000_0011
10	TX Buffer Hardware Parameters (HWTXBUF)	32	RO	8008_0B08
14	RX Buffer Hardware Parameters (HWRXBUF)	32	RO	0000_0808
80	General Purpose Timer #0 Load (GPTIMER0LD)	32	RW	0000_0000
84	General Purpose Timer #0 Controller (GPTIMER0CTRL)	32	RW	0000_0000
88	General Purpose Timer #1 Load (GPTIMER1LD)	32	RW	0000_0000
8C	General Purpose Timer #1 Controller (GPTIMER1CTRL)	32	RW	0000_0000
90	System Bus Config (SBUSCFG)	32	RW	0000_0002
100	Capability Registers Length (CAPLENGTH)	8	RO	40
102	Host Controller Interface Version (HCIVERSION)	16	RO	0100
104	Host Controller Structural Parameters (HCSPARAMS)	32	RO	0001_0011
108	Host Controller Capability Parameters (HCCPARAMS)	32	RO	0000_0006
120	Device Controller Interface Version (DCIVERSION)	16	RO	0001
124	Device Controller Capability Parameters (DCCPARAMS)	32	RO	0000_0188
140	USB Command Register (USBCMD)	32	RW	0008_0000
144	USB Status Register (USBSTS)	32	RW	0000_0080
148	Interrupt Enable Register (USBINTR)	32	RW	0000_0000
14C	USB Frame Index (FRINDEX)	32	RW	0000_0000
154	Device Address (DEVICEADDR)	32	RW	0000_0000
154	Frame List Base Address (PERIODICLISTBASE)	32	RW	0000_0000
158	Next Asynch. Address (ASYNCLISTADDR)	32	RW	0000_0000
158	Endpoint List Address (ENDPTLISTADDR)	32	RW	0000_0000
160	Programmable Burst Size (BURSTSIZE)	32	RW	0000_0808
164	TX FIFO Fill Tuning (TXFILLTUNING)	32	RW	0000_0000
178	Endpoint NAK (ENDPTNAK)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
17C	Endpoint NAK Enable (ENDPTNAKEN)	32	RW	0000_0000
180	Configure Flag Register (CONFIGFLAG)	32	RO	0000_0001
184	Port Status & Control (PORTSC1)	32	RW	1C00_0004
1A4	On-The-Go Status & control (OTGSC)	32	RW	0020_2F20
1A8	USB Device Mode (USBMODE)	32	RW	0000_5000
1AC	Endpoint Setup Status (ENDPTSETUPSTAT)	32	RW	0000_0000
1B0	Endpoint Prime (ENDPTPRIME)	32	RW	0000_0000
1B4	Endpoint Flush (ENDPTFLUSH)	32	RW	0000_0000
1B8	Endpoint Status (ENDPTSTAT)	32	RO	0000_0000
1BC	Endpoint Complete (ENDPTCOMPLETE)	32	RW	0000_0000
1C0	Endpoint Control0 (ENDPTCTRL0)	32	RW	0080_0080
1C4	Endpoint Control 1 (ENDPTCTRL1)	32	RW	0000_0000
1C8	Endpoint Control 2 (ENDPTCTRL2)	32	RW	0000_0000
1CC	Endpoint Control 3 (ENDPTCTRL3)	32	RW	0000_0000
1D0	Endpoint Control 4 (ENDPTCTRL4)	32	RW	0000_0000
1D4	Endpoint Control 5 (ENDPTCTRL5)	32	RW	0000_0000
1D8	Endpoint Control 6 (ENDPTCTRL6)	32	RW	0000_0000
1DC	Endpoint Control 7 (ENDPTCTRL7)	32	RW	0000_0000

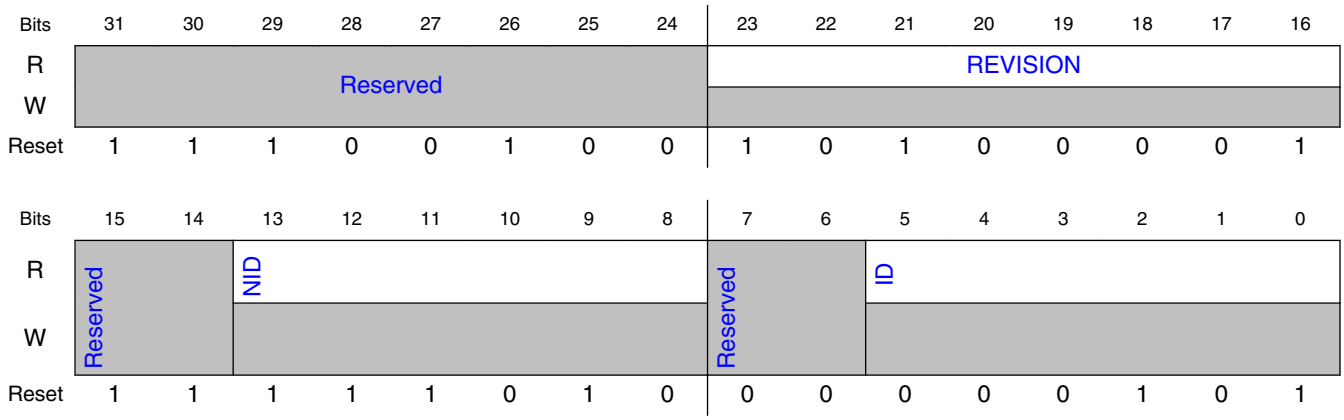
62.7.1.2 Identification register (ID)

The ID register identifies the USB 2.0 High-Speed core and its revision.

62.7.1.2.1 Offset

Register	Offset
ID	0h

62.7.1.2.2 Diagram



62.7.1.2.3 Fields

Field	Description
31-24 —	- Reserved
23-16 REVISION	REVISION Revision number of the controller core.
15-14 —	- Reserved
13-8 NID	NID Complement version of ID
7-6 —	- Reserved
5-0 ID	ID Configuration number. This number is set to 0x05 and indicates that the peripheral is USB 2.0 High-Speed core.

62.7.1.3 Hardware General (HWGENERAL)

General hardware parameters as defined in System Level Issues and Core Configuration.

NOTE

The reset value could vary from instance to instance. Please see the detail in bit field description and ignore reset value in summary table in this case!

62.7.1.3.1 Offset

Register	Offset
HWGENERAL	4h

62.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					SM		PHYM			PHYW		Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

62.7.1.3.3 Fields

Field	Description
31-11 —	- Reserved
10-9 SM	SM Serial interface mode capability 00 - No Serial Engine, always use parallel signalling. 01 - Serial Engine present, always use serial signalling for FS/LS. 10 - Software programmable - Reset to use parallel signalling for FS/LS 11 - Software programmable - Reset to use serial signalling for FS/LS
8-6 PHYM	PHYM Transciever type 000 - UTMI/UMTI+ 001 - ULPI DDR 010 - ULPI 011 - Serial Only 100 - Software programmable - reset to UTMI/UTMI+ 101 - Software programmable - reset to ULPI DDR 110 - Software programmable - reset to ULPI 111 - Software programmable - reset to Serial

Table continues on the next page...

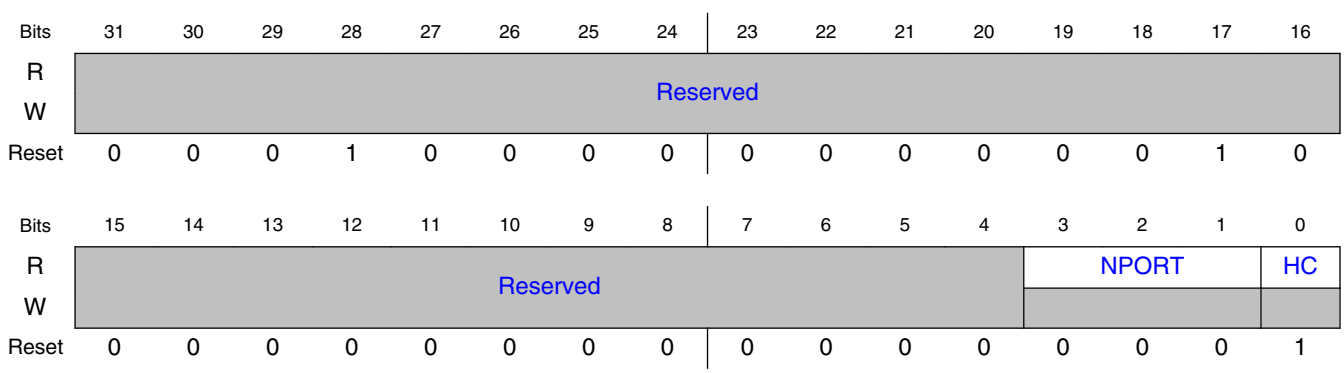
Field	Description
5-4 PHYW	PHYW Data width of the transciever connected to the controller core. 00 - 8 bit wide data bus (Software non-programmable) 01 - 16 bit wide data bus (Software non-programmable) 10 - Reset to 8 bit wide data bus (Software programmable) 11 - Reset to 16 bit wide data bus (Software programmable)
3-0 —	- Reserved

62.7.1.4 Host Hardware Parameters (HWHOST)

62.7.1.4.1 Offset

Register	Offset
HWHOST	8h

62.7.1.4.2 Diagram



62.7.1.4.3 Fields

Field	Description
31-4 —	- Reserved

Table continues on the next page...

Field	Description
3-1 NPORT	NPORT The Nmber of downstream ports supported by the host controller is (NPORT+1). **NOTE:** When these bits value is '000', it indicates a single-port host controller.
0 HC	HC Host Capable. Indicating whether host operation mode is supported or not. 0 - Not supported 1 - Supported

62.7.1.5 Device Hardware Parameters (HWDEVICE)

NOTE

This register is only available in OTG core.

62.7.1.5.1 Offset

Register	Offset
HWDEVICE	Ch

62.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										DEVEP					DC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

62.7.1.5.3 Fields

Field	Description
31-6 —	- Reserved
5-1 DEVEP	DEVEP Device Endpoint Number
0 DC	DC Device Capable. Indicating whether device operation mode is supported or not. 0 - Not supported 1 - Supported

62.7.1.6 TX Buffer Hardware Parameters (HWTXBUF)

62.7.1.6.1 Offset

Register	Offset
HWTXBUF	10h

62.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TXCHANADD							
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TXBURST							
W																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0

62.7.1.6.3 Fields

Field	Description
31-24 —	- Reserved
23-16 TXCHANADD	TXCHANADD TX FIFO Buffer size is: (2^TXCHANADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For the OTG controller operating in device mode, this is the FIFO buffer size per endpoint. As the OTG controller has 8 TX endpoint, there are 8 of these buffers. For the OTG controller operating in host mode, or for Host-only controller, the entire buffer memory is used as a single TX buffer. Therefore, there is only 1 of this buffer
15-8 —	- Reserved
7-0 TXBURST	TXBURST Default burst size for memory to TX buffer transfer. This is reset value of TXPBURST bits in USB core register USB_n_BURSTSIZE. Please see BURSTSIZE .

62.7.1.7 RX Buffer Hardware Parameters (HWRXBUF)

62.7.1.7.1 Offset

Register	Offset
HWRXBUF	14h

62.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXADD								RXBURST							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

62.7.1.7.3 Fields

Field	Description
31-16 —	- Reserved
15-8 RXADD	RXADD Buffer total size for all receive endpoints is (2^RXADD). RX Buffer size is: (2^RXADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. There is a single Receive FIFO buffer in the USB controller. The buffer is shared for all endpoints for the OTG controller in device mode.
7-0 RXBURST	RXBURST Default burst size for memory to RX buffer transfer. This is reset value of RXPBURST bits in USB core register USB_n_BURSTSIZE. Please see BURSTSIZE .

62.7.1.8 General Purpose Timer #0 Load (GPTIMER0LD)

This register controls load value of the count timer in register n_GPTIMER0CTRL. Please see GPTIMER0CTRL .

62.7.1.8.1 Offset

Register	Offset
GPTIMER0LD	80h

62.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								GPTLD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPTLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.8.3 Fields

Field	Description
31-24 —	- Reserved
23-0 GPTLD	GPTLD General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

62.7.1.9 General Purpose Timer #0 Controller (GPTIMER0CTRL)

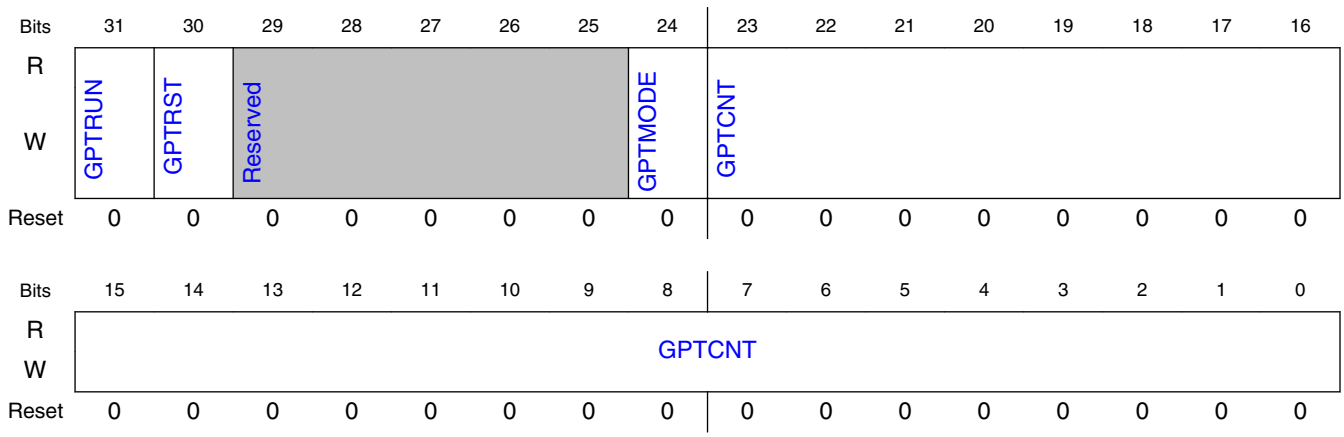
This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI0 bit in n_USBSTS register (See USBSTS), interrupt enable bit is TIE0 bit in n_USBINTR register. (See USBINTR .)

62.7.1.9.1 Offset

Register	Offset
GPTIMER0CTRL	84h

62.7.1.9.2 Diagram



62.7.1.9.3 Fields

Field	Description
31 GPTRUN	GPTRUN General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 - Stop counting 1 - Run
30 GPTRST	GPTRST General Purpose Timer Reset 0 - No action 1 - Load counter value from GPTLD bits in n_GPTIMER0LD
29-25 —	- Reserved
24 GPTMODE	GPTMODE General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software; In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 - One Shot Mode

Table continues on the next page...

Field	Description
	1 - Repeat Mode
23-0 GPTCNT	GPTCNT General Purpose Timer Counter. This field is the count value of the countdown timer.

62.7.1.10 General Purpose Timer #1 Load (GPTIMER1LD)

This register controls load value of the count timer in register n_GPTIMER1CTRL. Please see GPTIMER1CTRL .

62.7.1.10.1 Offset

Register	Offset
GPTIMER1LD	88h

62.7.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								GPTLD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPTLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.10.3 Fields

Field	Description
31-24 —	- Reserved
23-0 GPTLD	GPTLD General Purpose Timer Load Value

Field	Description
	These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

62.7.1.11 General Purpose Timer #1 Controller (GPTIMER1CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI1 bit in USB_n_USBSTS register (See USBSTS), interrupt enable bit is TIE1 bit in n_USBINTR register (See USBINTR).

62.7.1.11.1 Offset

Register	Offset
GPTIMER1CTRL	8Ch

62.7.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.11.3 Fields

Field	Description
31 GPTRUN	GPTRUN General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 - Stop counting 1 - Run
30 GPTRST	GPTRST General Purpose Timer Reset 0 - No action 1 - Load counter value from GPTLD bits in USB_n_GPTIMER0LD
29-25 —	- Reserved
24 GPTMODE	GPTMODE General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software. In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 - One Shot Mode 1 - Repeat Mode
23-0 GPTCNT	GPTCNT General Purpose Timer Counter. This field is the count value of the countdown timer.

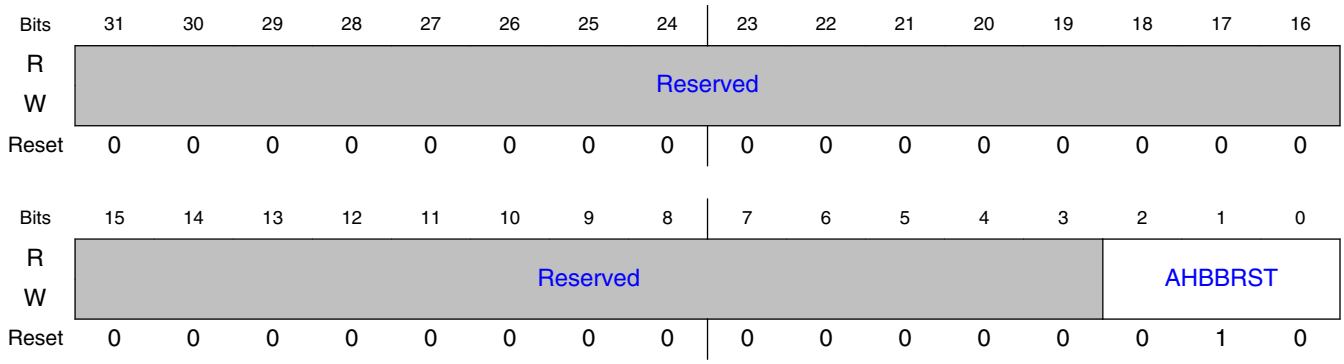
62.7.1.12 System Bus Config (SBUSCFG)

This register defines the system bus configuration.

62.7.1.12.1 Offset

Register	Offset
SBUSCFG	90h

62.7.1.12.2 Diagram



62.7.1.12.3 Fields

Field	Description
31-3 —	- Reserved
2-0 AHBBRST	AHBBRST AHB master interface Burst configuration These bits control AHB master transfer type sequence (or priority). **NOTE:** This register overrides n_BURSTSIZE register when its value is not zero. 000 - Incremental burst of unspecified length only 001 - INCR4 burst, then single transfer 010 - INCR8 burst, INCR4 burst, then single transfer 011 - INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 - Reserved, don't use 101 - INCR4 burst, then incremental burst of unspecified length 110 - INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 - INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

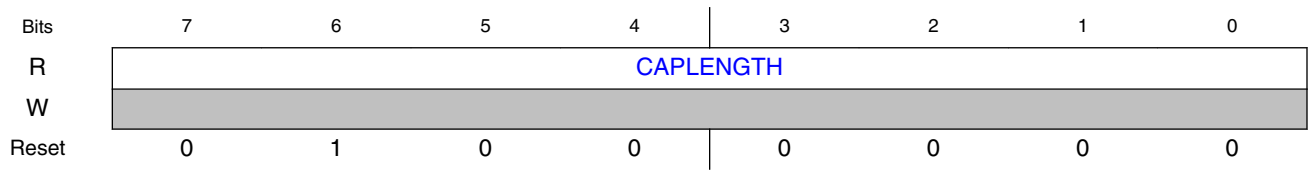
62.7.1.13 Capability Registers Length (CAPLENGTH)

The Capability Registers Length register contains the address offset to the Operational registers relative to the CAPLENGTH register.

62.7.1.13.1 Offset

Register	Offset
CAPLENGTH	100h

62.7.1.13.2 Diagram



62.7.1.13.3 Fields

Field	Description
7-0 CAPLENGTH	CAPLENGTH These bits are used as an offset to add to register base to find the beginning of the Operational Register. Default value is '40h'.

62.7.1.14 Host Controller Interface Version (HCIVERSION)

This is a 2-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

62.7.1.14.1 Offset

Register	Offset
HCIVERSION	102h

62.7.1.14.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HCIVERSION															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

62.7.1.14.3 Fields

Field	Description
15-0 HCIVERSION	HCIVERSION Host Controller Interface Version Number Default value is '10h', which means EHCI rev1.0.

62.7.1.15 Host Controller Structural Parameters (HCSPARAMS)

The following figure shows the port steering logic capabilities of Host Control Structural Parameters (n_HCSPARAMS).

62.7.1.15.1 Offset

Register	Offset
HCSPARAMS	104h

62.7.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				N_TT				N_PTT				Reserved			PI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N_CC				N_PCC				Reserved			PPC	N_PORTS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

62.7.1.15.3 Fields

Field	Description
31-28	-
—	Reserved
27-24	N_TT
N_TT	Number of Transaction Translators (N_TT). Default value '0000b'
	This field indicates the number of embedded transaction translators associated with the USB2.0 host controller.
	These bits would be set to '0001b' for Multi-Port Host, and '0000b' for Single-Port Host.
23-20	N_PTT
N_PTT	Number of Ports per Transaction Translator (N_PTT). Default value '0000b'
	This field indicates the number of ports assigned to each transaction translator within the USB2.0 host controller.
	These bits would be set to equal N_PORTS for Multi-Port Host, and '0000b' for Single-Port Host.
19-17	-
—	Reserved
16	PI
PI	Port Indicators (P INDICATOR)
	This bit indicates whether the ports support port indicator control. When set to one, the port status and control registers include a read/writeable field for controlling the state of the port indicator
	This bit is "1b" in all controller core.
15-12	N_CC
N_CC	Number of Companion Controller (N_CC).
	This field indicates the number of companion controllers associated with this USB2.0 host controller.
	These bits are '0000b' in all controller core.
	0000 - There is no internal Companion Controller and port-ownership hand-off is not supported.
	0001 - There are internal companion controller(s) and port-ownership hand-offs is supported.
11-8	N_PCC
N_PCC	Number of Ports per Companion Controller
	This field indicates the number of ports supported per internal Companion Controller. It is used to indicate the port routing configuration to the system software.
	For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC.
	These bits are '0000b' in all controller core.
7-5	-
—	Reserved
4	PPC
PPC	Port Power Control

Table continues on the next page...

Field	Description
	This field indicates whether the host controller implementation includes port power control. A one indicates the ports have port power switches. A zero indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register
3-0 N_PORTS	N_PORTS Number of downstream ports. This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register. Valid values are in the range of 1h to Fh. A zero in this field is undefined. These bits are always set to '0001b' because all controller cores are Single-Port Host.

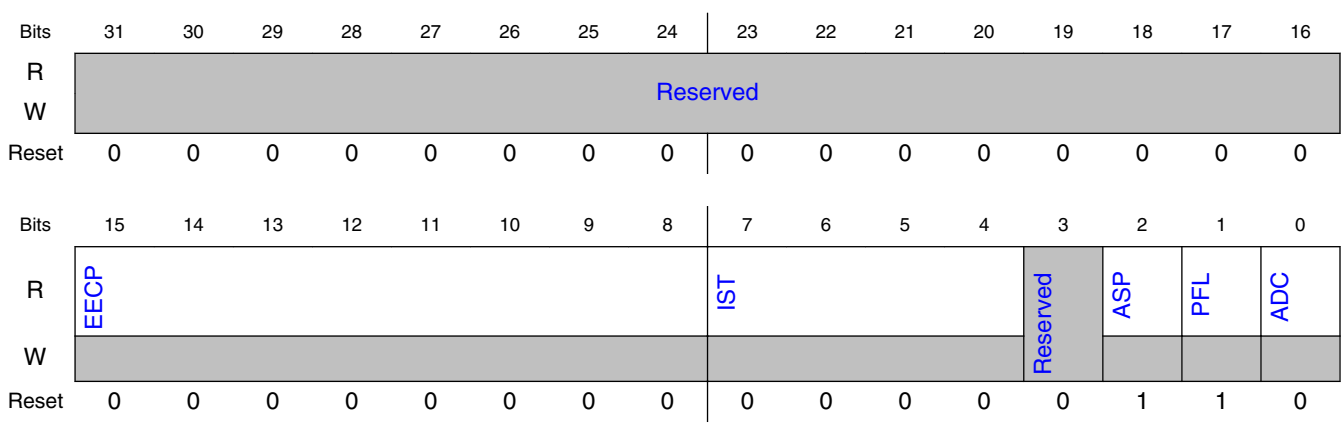
62.7.1.16 Host Controller Capability Parameters (HCCPARAMS)

This register identifies multiple mode control (time-base bit functionality), addressing capability.

62.7.1.16.1 Offset

Register	Offset
HCCPARAMS	108h

62.7.1.16.2 Diagram



62.7.1.16.3 Fields

Field	Description
31-16 —	- Reserved
15-8 EECP	EECP EHCI Extended Capabilities Pointer. This field indicates the existence of a capabilities list. A value of 00h indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capability. The pointer value must be 40h or greater if implemented to maintain the consistency of the PCI header defined for this class of device. **NOTE:** These bits are set '00h' in all controller core.
7-4 IST	IST Isochronous Scheduling Threshold. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame. These bits are set '00h' in all controller core.
3 —	- Reserved
2 ASP	ASP Asynchronous Schedule Park Capability If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USBCMD register. **NOTE:** ASP bit reset value: '00b' for OTG controller core, '11b' for Host-only controller core.
1 PFL	PFL Programmable Frame List Flag If this bit is set to zero, then the system software must use a frame list length of 1024 elements with this host controller. The USBCMD register Frame List Size field is a read-only register and must be set to zero. If set to a one, then the system software can specify and use a smaller frame list and configure the host controller via the USBCMD register Frame List Size field. The frame list must always be aligned on a 4K-page boundary. This requirement ensures that the frame list is always physically contiguous. This bit is set '1b' in all controller core.
0 ADC	ADC 64-bit Addressing Capability This bit is set '0b' in all controller core, no 64-bit addressing capability is supported.

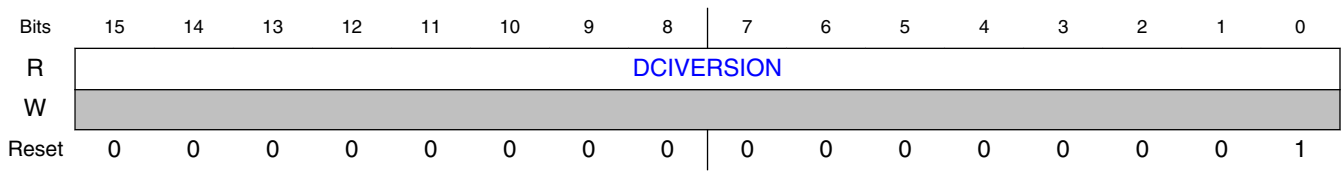
62.7.1.17 Device Controller Interface Version (DCIVERSION)

This register indicates the two-byte BCD encoding of the device controller interface version number.

62.7.1.17.1 Offset

Register	Offset
DCIVERSION	120h

62.7.1.17.2 Diagram



62.7.1.17.3 Fields

Field	Description
15-0 DCIVERSION	DCIVERSION Device Controller Interface Version Number Default value is '01h', which means rev0.1.

62.7.1.18 Device Controller Capability Parameters (DCCPARAMS)

These fields describe the overall device capability of the controller.

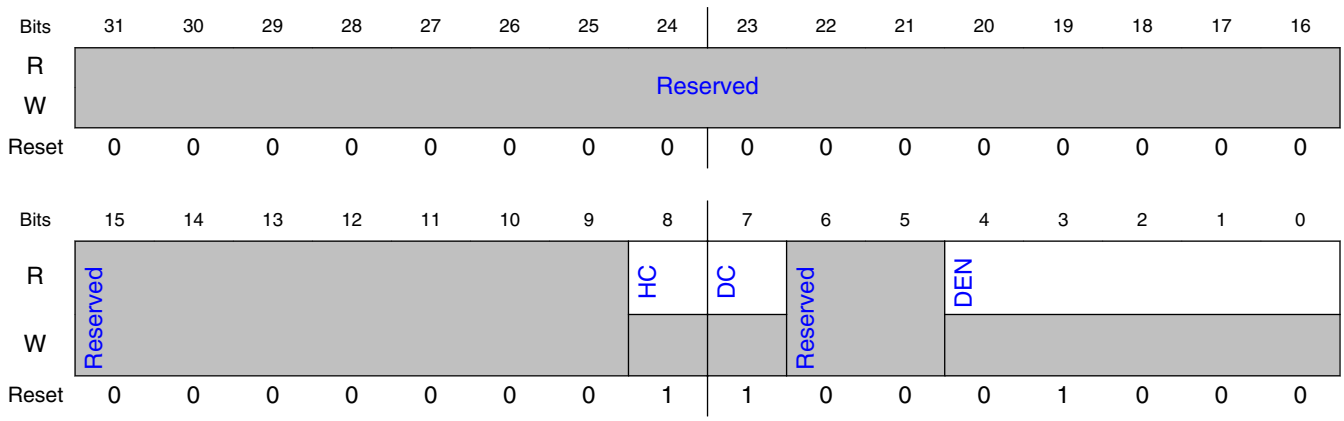
NOTE

This register is only available in OTG controller core.

62.7.1.18.1 Offset

Register	Offset
DCCPARAMS	124h

62.7.1.18.2 Diagram



62.7.1.18.3 Fields

Field	Description
31-9 —	- Reserved
8 HC	HC Host Capable When this bit is 1, this controller is capable of operating as an EHCI compatible USB 2.0 host controller.
7 DC	DC Device Capable When this bit is 1, this controller is capable of operating as a USB 2.0 device.
6-5 —	- Reserved
4-0 DEN	DEN Device Endpoint Number This field indicates the number of endpoints built into the device controller. If this controller is not device capable, then this field will be zero. Valid values are 0 - 15.

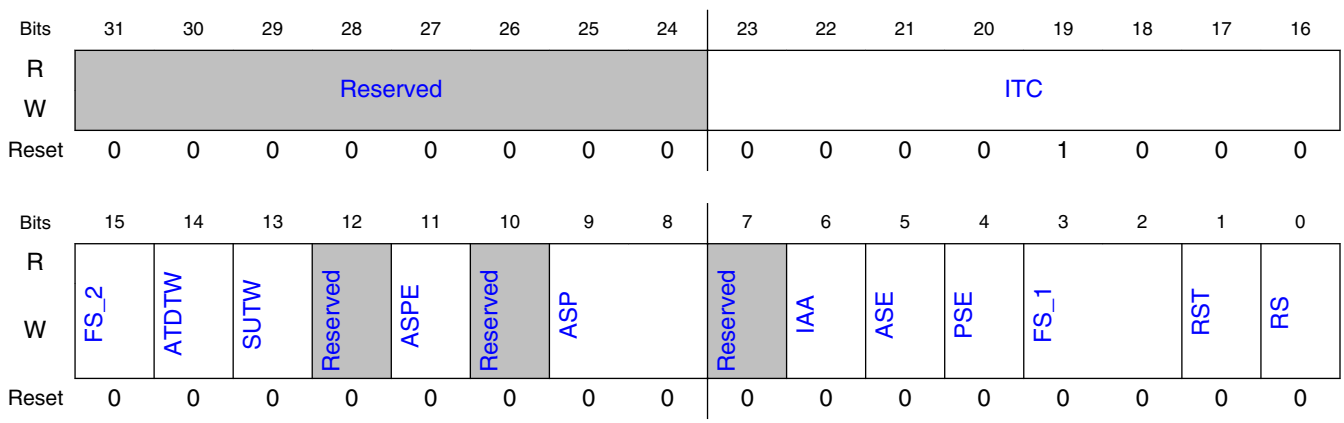
62.7.1.19 USB Command Register (USBCMD)

The Command Register indicates the command to be executed by the serial bus host/device controller. Writing to the register causes a command to be executed.

62.7.1.19.1 Offset

Register	Offset
USBCMD	140h

62.7.1.19.2 Diagram



62.7.1.19.3 Fields

Field	Description
31-24 —	- Reserved
23-16 ITC	ITC Interrupt Threshold Control -Read/Write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Valid values are shown below. Value Maximum Interrupt Interval 00000000 - Immediate (no threshold) 00000001 - 1 micro-frame 00000010 - 2 micro-frames 00000100 - 4 micro-frames

Table continues on the next page...

Field	Description
	00001000 - 8 micro-frames 00010000 - 16 micro-frames 00100000 - 32 micro-frames 01000000 - 64 micro-frames
15 FS_2	FS_2 Frame List Size - (Read/Write or Read Only). [host mode only] This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index. **NOTE:** This field is made up from USBCMD bits 15, 3 and 2. Value Meaning 0b000 - 1024 elements (4096 bytes) Default value 0b001 - 512 elements (2048 bytes) 0b010 - 256 elements (1024 bytes) 0b011 - 128 elements (512 bytes) 0b100 - 64 elements (256 bytes) 0b101 - 32 elements (128 bytes) 0b110 - 16 elements (64 bytes) 0b111 - 8 elements (32 bytes)
14 ATDTW	ATDTW Add dTD TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit would also be cleared by hardware when state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 SUTW	SUTW Setup TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (SLOM bit in USB core register n_USBMODE, see USBMODE) then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software. This bit would also be cleared by hardware when a hazard detected.
12 —	- Reserved
11 ASPE	ASPE Asynchronous Schedule Park Mode Enable - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. **NOTE:** ASPE bit reset value: '0b' for OTG controller .

Table continues on the next page...

Field	Description
10 —	- Reserved
9-8 ASP	ASP Asynchronous Schedule Park Mode Count - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is Read-Only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in undefined behavior. This field is set to 3h in all controller core.
7 —	- Reserved
6 IAA	IAA Interrupt on Async Advance Doorbell - Read/Write. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
5 ASE	ASE Asynchronous Schedule Enable - Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Only the host controller uses this bit. Values Meaning 0 - Do not process the Asynchronous Schedule. 1 - Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
4 PSE	PSE Periodic Schedule Enable- Read/Write. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Only the host controller uses this bit. Values Meaning 0 - Do not process the Periodic Schedule 1 - Use the PERIODICLISTBASE register to access the Periodic Schedule.
3-2 FS_1	FS_1 See description at bit 15
1	RST

Table continues on the next page...

Field	Description
RST	Controller Reset (RESET) - Read/Write. Software uses this bit to reset the controller. This bit is set to zero by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Host operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. Device operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, because the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.
0 RS	RS Run/Stop (RS) - Read/Write. Default 0b. 1=Run. 0=Stop. Host operation mode: When set to '1b', the Controller proceeds with the execution of the schedule. The Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the controller is in the Halted state (that is, HCHalted in the USBSTS register is a one). Device operation mode: Writing a one to this bit will cause the controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the controller has been properly initialized. Writing a 0 to this will cause a detach event.

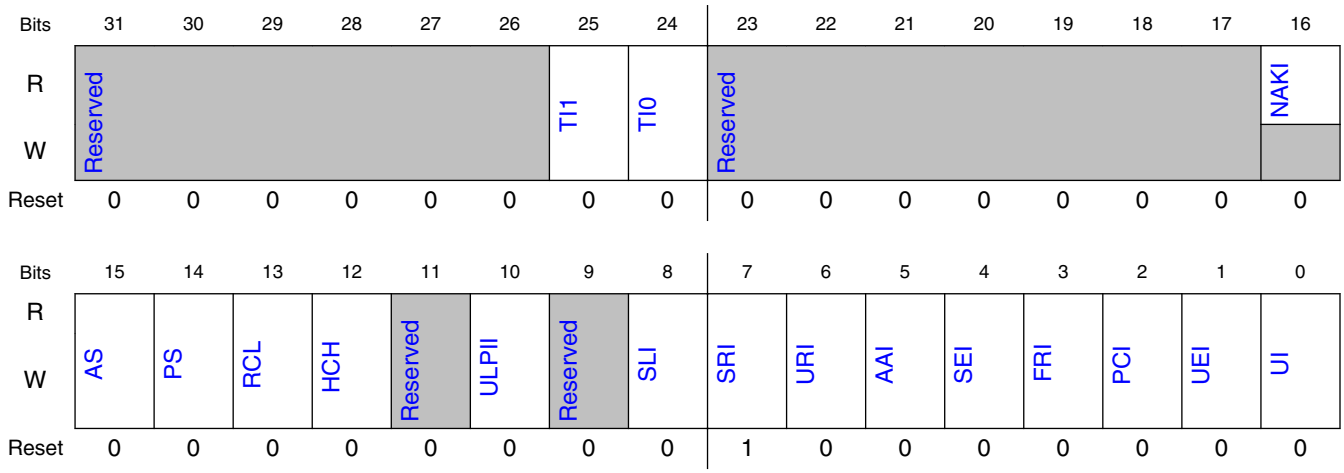
62.7.1.20 USB Status Register (USBSTS)

This register indicates various states of the Host/Device Controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

62.7.1.20.1 Offset

Register	Offset
USBSTS	144h

62.7.1.20.2 Diagram



62.7.1.20.3 Fields

Field	Description
31-26 —	- Reserved
25 TI1	TI1 General Purpose Timer Interrupt 1(GPTINT1)--R/WC. This bit is set when the counter in the GPTIMER1CTRL register transitions to zero, writing a one to this bit will clear it.
24 TI0	TI0 General Purpose Timer Interrupt 0(GPTINT0)--R/WC. This bit is set when the counter in the GPTIMER0CTRL register transitions to zero, writing a one to this bit clears it.
23-17 —	- Reserved
16 NAKI	NAKI NAK Interrupt Bit--RO. This bit is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and corresponding TX/RX Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all Enabled TX/RX Endpoint NAK bits are cleared.
15 AS	AS Asynchronous Schedule Status - Read Only. This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.

Table continues on the next page...

Field	Description
14 PS	PS Periodic Schedule Status - Read Only. This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
13 RCL	RCL Reclamation - Read Only. This is a read-only status bit used to detect an empty asynchronous schedule. Only used in the host operation mode.
12 HCH	HCH HCHalted - Read Only. This bit is a zero whenever the Run/Stop bit is a one. The Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Controller hardware (for example, an internal error). Only used in the host operation mode. Default value is '0b' for OTG core . This is because OTG core is not operating as host in default. Please see CM bit in USB_n_USBMODE register. **NOTE:** HCH bit reset value: '0b' for OTG controller core .
11 —	- Reserved
10 ULPII	ULPII ULPI Interrupt - R/WC. This bit will be set '1b' by hardware when there is an event completion in ULPI viewport. This bit is usable only if the controller support UPLI interface mode.
9 —	- Reserved
8 SLI	SLI DCSuspend - R/WC. When a controller enters a suspend state from an active state, this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used in device operation mode.
7 SRI	SRI SOF Received - R/WC. When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp.

Table continues on the next page...

Field	Description
	In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it.
6 URI	URI USB Reset Received - R/WC. When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used in device operation mode.
5 AAI	AAI Interrupt on Async Advance - R/WC. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4 SEI	SEI System Error- R/WC. This bit is will be set to '1b' when an Error response is seen to a read on the system interface.
3 FRI	FRI Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the USB_n_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2 PCI	PCI Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/ Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively.
1 UEI	UEI USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set. The device controller detects resume signaling only.
0 UI	UI USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set.

Field	Description
	This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

62.7.1.21 Interrupt Enable Register (USBINTR)

The interrupts to software are enabled with this register. An interrupt is generated when a bit is set and the corresponding interrupt source is active. The USB Status register (n_USBSTS) still shows interrupt sources even if they are disabled by the n_USBINTR register, allowing polling of interrupt events by the software.

62.7.1.21.1 Offset

Register	Offset
USBINTR	148h

62.7.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						TIE1	TIE0	Reserved				UPIE	UAIE	Reserved	NAKE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					ULPIE	Reserved	SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.21.3 Fields

Field	Description
31-26 —	- Reserved

Table continues on the next page...

Field	Description
25 TIE1	TIE1 General Purpose Timer #1 Interrupt Enable When this bit is one and the TI1 bit in n_USBSTS register is a one the controller will issue an interrupt.
24 TIE0	TIE0 General Purpose Timer #0 Interrupt Enable When this bit is one and the TI0 bit in n_USBSTS register is a one the controller will issue an interrupt.
23-20 —	- Reserved
19 UPIE	UPIE USB Host Periodic Interrupt Enable When this bit is one, and the UPI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
18 UAIE	UAIE USB Host Asynchronous Interrupt Enable When this bit is one, and the UAI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
17 —	- Reserved
16 NAKE	NAKE NAK Interrupt Enable When this bit is one and the NAKI bit in n_USBSTS register is a one the controller will issue an interrupt.
15-11 —	- These bits are reserved and should be set to zero.
10 ULPIE	ULPIE ULPI Interrupt Enable When this bit is one and the UPLII bit in n_USBSTS register is a one the controller will issue an interrupt. This bit is usable only if the controller support UPLI interface mode.
9 —	- Reserved
8 SLE	SLE Sleep Interrupt Enable When this bit is one and the SLI bit in n_n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
7 SRE	SRE SOF Received Interrupt Enable When this bit is one and the SRI bit in n_USBSTS register is a one the controller will issue an interrupt.
6 URE	URE USB Reset Interrupt Enable When this bit is one and the URI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.

Table continues on the next page...

Field	Description
5 AAE	AAE Async Advance Interrupt Enable When this bit is one and the AAI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
4 SEE	SEE System Error Interrupt Enable When this bit is one and the SEI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
3 FRE	FRE Frame List Rollover Interrupt Enable When this bit is one and the FRI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
2 PCE	PCE Port Change Detect Interrupt Enable When this bit is one and the PCI bit in n_USBSTS register is a one the controller will issue an interrupt.
1 UEE	UEE USB Error Interrupt Enable When this bit is one and the UEI bit in n_USBSTS register is a one the controller will issue an interrupt.
0 UE	UE USB Interrupt Enable When this bit is one and the UI bit in n_USBSTS register is a one the controller will issue an interrupt.

62.7.1.22 USB Frame Index (FRINDEX)

This register is used by the host controller to index the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [N: 3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the n_USBCMD register.

This register must be written as a DWord. Byte writes produce undefined results. This register cannot be written unless the Host Controller is in the 'Halted' state as indicated by the HCHalted bit. A write to this register while the Run/Stop hit is set to a one produces undefined results. Writes to this register also affect the SOF value.

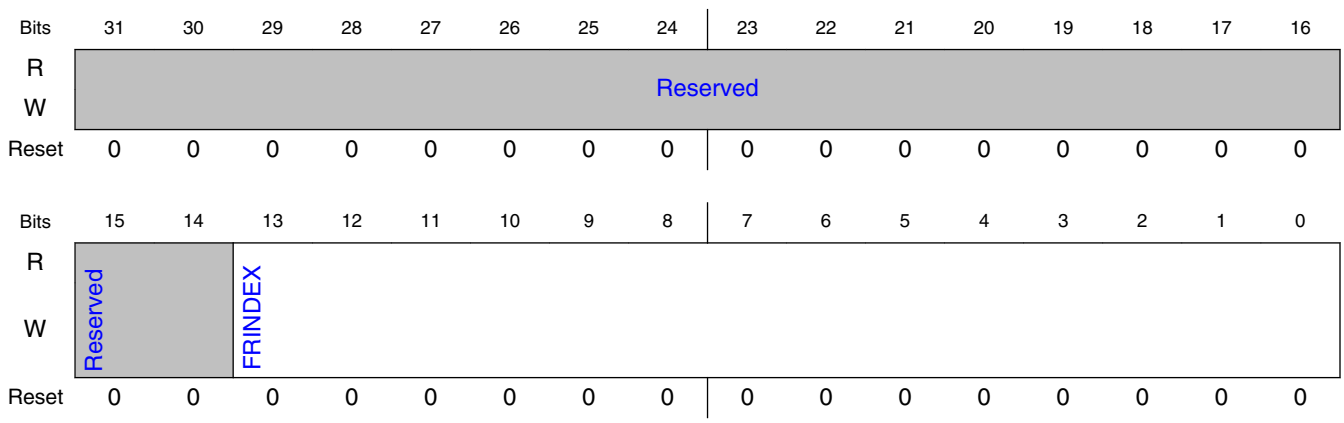
In device mode this register is read only and, the device controller updates the FRINDEX [13:3] register from the frame number indicated by the SOF marker. Whenever a SOF is received by the USB bus, FRINDEX [13:3] will be checked against the SOF marker. If

FRINDEX [13:3] is different from the SOF marker, FRINDEX [13:3] will be set to the SOF value and FRINDEX [2:0] will be set to zero (that is, SOF for 1 ms frame). If FRINDEX [13:3] is equal to the SOF value, FRINDEX [2:0] will be increment (that is, SOF for 125 us micro-frame.).

62.7.1.22.1 Offset

Register	Offset
FRINDEX	14Ch

62.7.1.22.2 Diagram



62.7.1.22.3 Fields

Field	Description
31-14 —	- Reserved
13-0 FRINDEX	FRINDEX Frame Index. The value, in this register, increments at the end of each time frame (micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe.

Field	Description
	The bit field values description below is represented as (Frame List Size) Number Elements N. 00000000000000 - (1024) 12 00000000000001 - (512) 11 00000000000010 - (256) 10 00000000000011 - (128) 9 00000000000100 - (64) 8 00000000000101 - (32) 7 00000000000110 - (16) 6 00000000000111 - (8) 5

62.7.1.23 Device Address (DEVICEADDR)

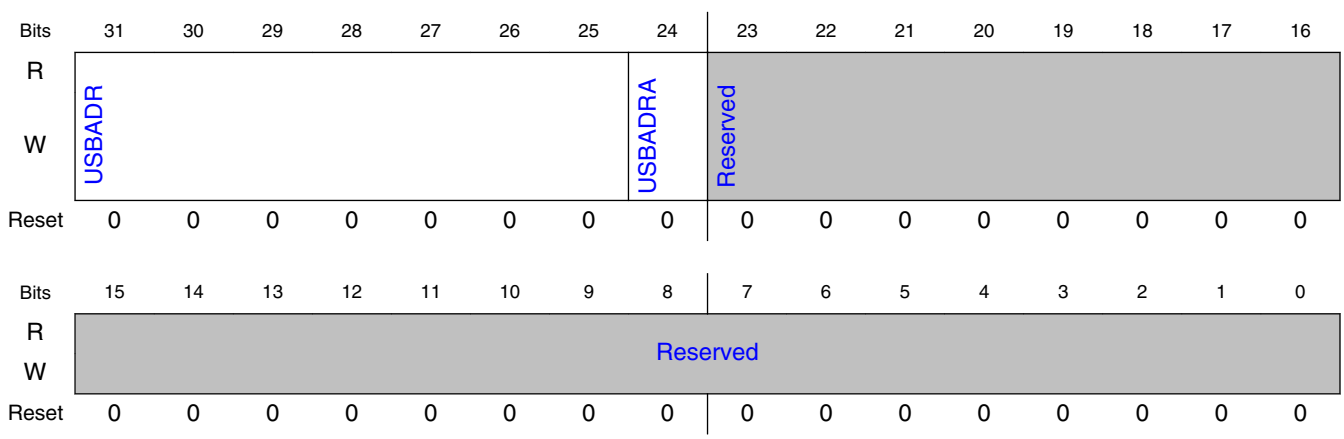
Device Controller only

The upper seven bits of this register represent the device address. After any controller reset or a USB reset, the device address is set to the default address (0). The default address will match all incoming addresses. Software shall reprogram the address after receiving a SET_ADDRESS descriptor.

62.7.1.23.1 Offset

Register	Offset
DEVICEADDR	154h

62.7.1.23.2 Diagram



62.7.1.23.3 Fields

Field	Description
31-25 USBADR	USBADR Device Address. These bits correspond to the USB device address
24 USBADRA	USBADRA Device Address Advance. Default=0. When this bit is '0', any writes to USBADR are instantaneous. When this bit is written to a '1' at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register. Hardware will automatically clear this bit on the following conditions: 1) IN is ACKed to endpoint 0. (USBADR is updated from staging register). 2) OUT/SETUP occur to endpoint 0. (USBADR is not updated). 3) Device Reset occurs (USBADR is reset to 0). **NOTE:** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes the USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.
23-0 —	- Reserved

62.7.1.24 Frame List Base Address (PERIODICLISTBASE)

Host Controller only

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (USB_n_FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.

62.7.1.24.1 Offset

Register	Offset
PERIODICLISTBASE	154h

62.7.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASEADR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASEADR				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.24.3 Fields

Field	Description
31-12 BASEADR	BASEADR Base Address (Low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.
11-0 —	- Reserved

62.7.1.25 Next Asynch. Address (ASYNCLISTADDR)

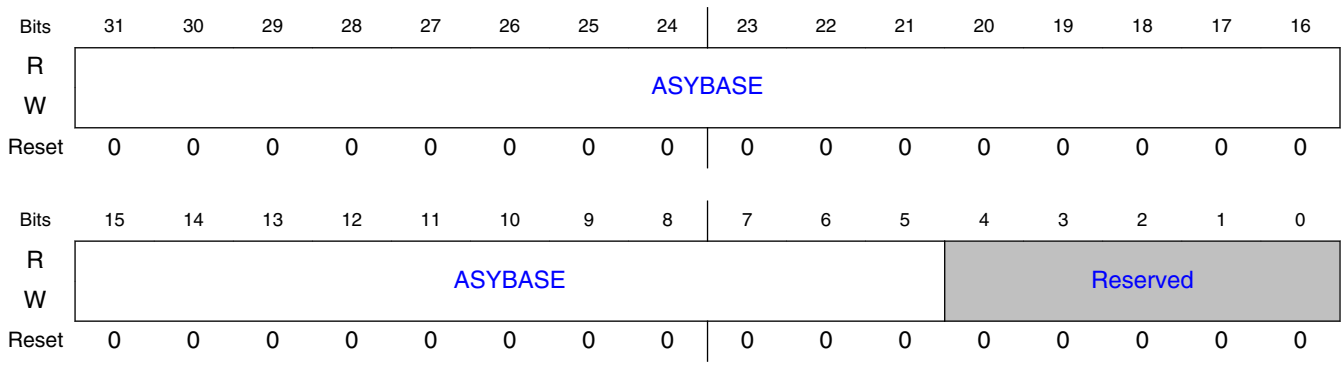
Host Controller only

This 32-bit register contains the address of the next asynchronous queue head to be executed by the host. Bits [4:0] of this register cannot be modified by the system software and will always return a zero when read.

62.7.1.25.1 Offset

Register	Offset
ASYNCLISTADDR	158h

62.7.1.25.2 Diagram



62.7.1.25.3 Fields

Field	Description
31-5 ASYBASE	ASYBASE Link Pointer Low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.
4-0 —	- Reserved

62.7.1.26 Endpoint List Address (ENDPTLISTADDR)

Device Controller only

In device mode, this register contains the address of the top of the endpoint list in system memory. Bits [10:0] of this register cannot be modified by the system software and will always return a zero when read.

The memory structure referenced by this physical memory pointer is assumed 64-byte.

62.7.1.26.1 Offset

Register	Offset
ENDPTLISTADDR	158h

62.7.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EPBASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPBASE					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.26.3 Fields

Field	Description
31-11 EPBASE	EPBASE Endpoint List Pointer(Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Head (QH) (that is, one queue head per endpoint & direction).
10-0 —	- Reserved

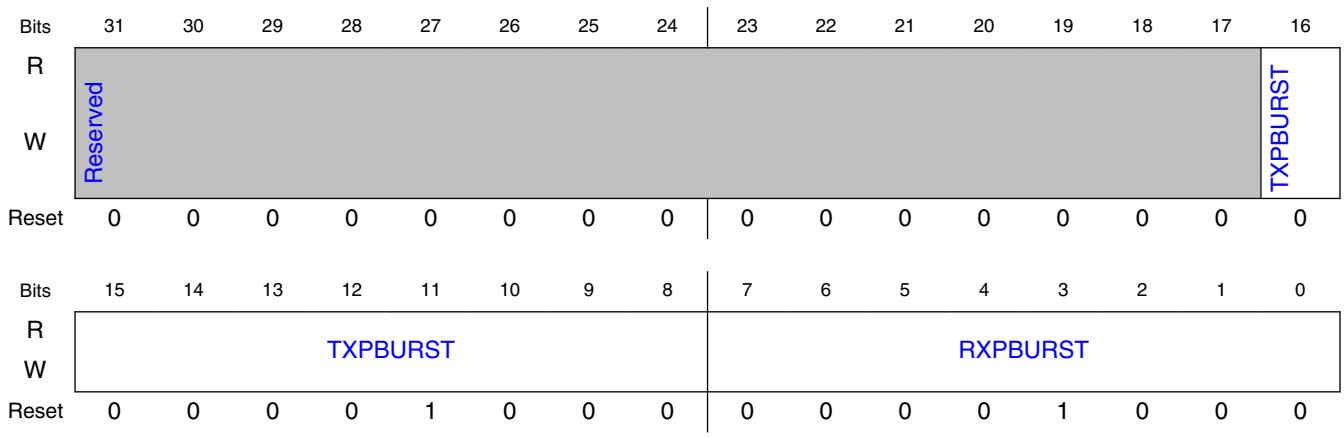
62.7.1.27 Programmable Burst Size (BURSTSIZE)

This register is used to control the burst size used during data movement on the AHB master interface. This register is ignored if AHBBRST bits in SBUSCFG register is non-zero value.

62.7.1.27.1 Offset

Register	Offset
BURSTSIZE	160h

62.7.1.27.2 Diagram



62.7.1.27.3 Fields

Field	Description
31-17 —	- Reserved
16-8 TXPBURST	TXPBURST Programmable TX Burst Size. Default value is determined by TXBURST bits in n_HWTXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from system memory to the USB bus.
7-0 RXPBURST	RXPBURST Programmable RX Burst Size. Default value is determined by TXBURST bits in n_HWRXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from the USB bus to system memory.

62.7.1.28 TX FIFO Fill Tuning (TXFILLTUNING)

The fields in this register control performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus. The specific areas of performance include the how much data to post into the FIFO and an estimate for how long that operation should take in the target system.

Definitions:

T_0 = Standard packet overhead

T_1 = Time to send data payload

T_{ff} = Time to fetch packet into TX FIFO up to specified level.

T_s = Total Packet Flight Time (send-only) packet

$T_s = T_0 + T_1$

T_p = Total Packet Time (fetch and send) packet

$T_p = T_{ff} + T_0 + T_1$

Upon discovery of a transmit (OUT/SETUP) packet in the data structures, host controller checks to ensure T_p remains before the end of the [micro]frame. If so it proceeds to pre-fill the TX FIFO. If at anytime during the pre-fill operation the time remaining the [micro]frame is < T_s then the packet attempt ceases and the packet is tried at a later time. Although this is not an error condition and the host controller will eventually recover, a mark will be made the scheduler health counter to note the occurrence of a "back-off" event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that will begin after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and thus should be minimized (not necessarily eliminated). Back-offs can be minimized with use of the n_TSCHHEALTH (T_{ff}) described below.

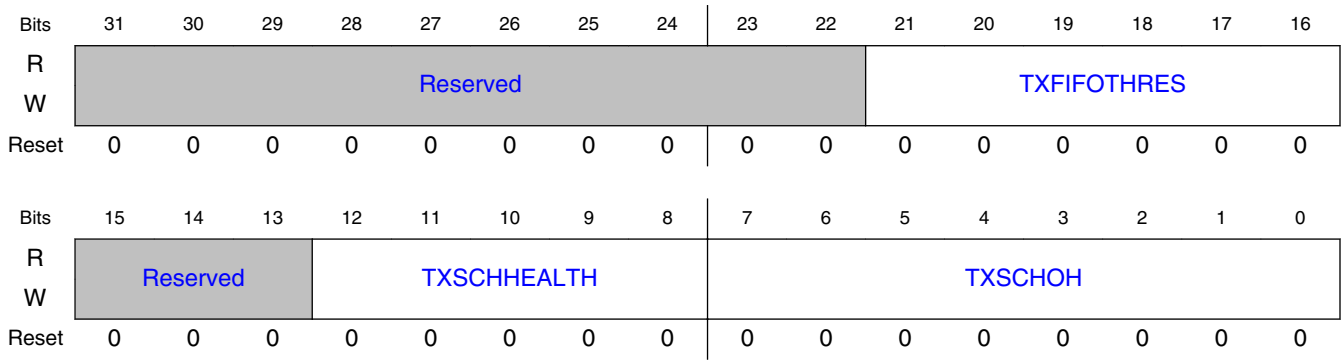
NOTE

The reset value could vary from instance to instance. Please see the detail in bit field description and ignore reset value in summary table in this case!

62.7.1.28.1 Offset

Register	Offset
TXFILLTUNING	164h

62.7.1.28.2 Diagram



62.7.1.28.3 Fields

Field	Description
31-22 —	- Reserved
21-16 TXFIFOTHRES	TXFIFOTHRES FIFO Burst Threshold. (Read/Write) This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be a low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in USB_n_USBMODE register is set.
15-13 —	- Reserved
12-8 TXSCHHEALTH	TXSCHHEALTH Scheduler Health Counter. (Read/Write To Clear) This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame. This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter and this counter will max. at 31.
7-0 TXSCHOH	TXSCHOH Scheduler Overhead. (Read/Write) [Default = 0] This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode. Default value is '08h' for OTG controller core .

62.7.1.29 Endpoint NAK (ENDPTNAK)

62.7.1.29.1 Offset

Register	Offset
ENDPTNAK	178h

62.7.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								EPTN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EPRN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.29.3 Fields

Field	Description
31-24 —	- Reserved
23-16 EPTN	EPTN TX Endpoint NAK - R/WC. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7
15-8 —	- Reserved
7-0 EPRN	EPRN RX Endpoint NAK - R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7

62.7.1.30 Endpoint NAK Enable (ENDPTNAKEN)

62.7.1.30.1 Offset

Register	Offset
ENDPTNAKEN	17Ch

62.7.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								EPTNE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								EPRNE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.30.3 Fields

Field	Description
31-24 —	- Reserved
23-16 EPTNE	EPTNE TX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7
15-8 —	- Reserved
7-0 EPRNE	EPRNE RX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7

62.7.1.31 Configure Flag Register (CONFIGFLAG)

62.7.1.31.1 Offset

Register	Offset
CONFIGFLAG	180h

62.7.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															CF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

62.7.1.31.3 Fields

Field	Description
31-1 —	- Reserved
0 CF	CF Configure Flag Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. 0 - Port routing control logic default-routes each port to an implementation dependent classic host controller. 1 - Port routing control logic default-routes all ports to this host controller.

62.7.1.32 Port Status & Control (PORTSC1)

Host Controller

A host controller could implement one to eight port status and control registers. The number is determined by N_PORTs bits in HWSPARAMs register (please see HCSPARAMS). Software could read this parameter register to determine how many ports need service.

All controller cores on this product are Single-Port, so there is only one port status and control register for each controller core.

This register is only reset by power on reset or controller core reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port supports power control, this state remains until port power is supplied (by software).

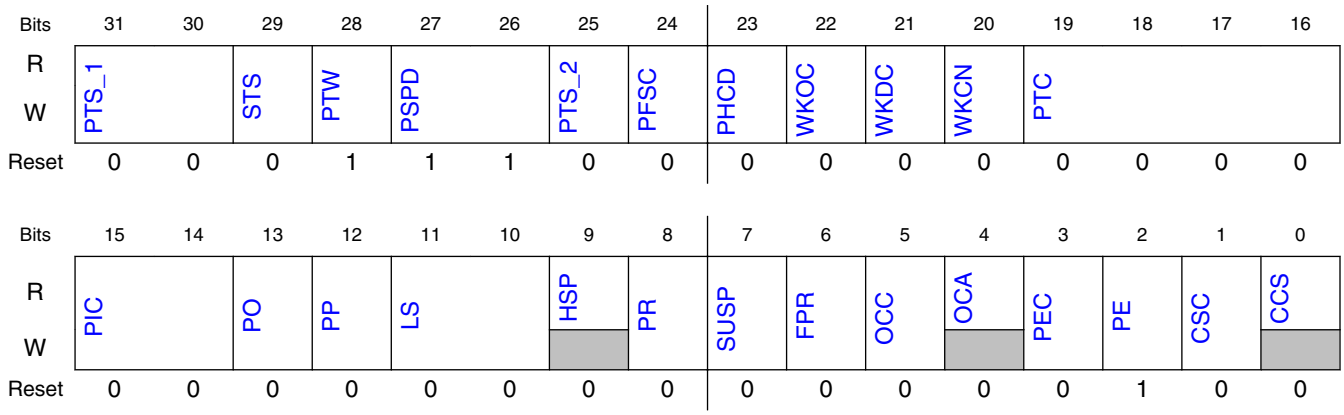
Device Controller

A controller operating in device mode has only port register one (PORTSC1) and it does not support power control in that mode. Port control in device mode is only used for status port reset, suspend, and current connect status. It is also used to initiate test mode or force signaling and allows software to put the PHY into low power suspend mode and disable the PHY clock.

62.7.1.32.1 Offset

Register	Offset
PORTSC1	184h

62.7.1.32.2 Diagram



62.7.1.32.3 Fields

Field	Description
31-30 PTS_1	PTS_1 Bit field {bit25, bit31, bit30}: "000b" UTMI/UTMI+ "001b" Reserved "010b" ULPI "011b" Serial/USB 1.1 PHY/IC-USB (FS Only) "100b" HSIC Parallel Transceiver Select (bit25, bit31, bit30). For OTG core, it's Read-Only. For Host1/Host2/Host3 core, it's Read/Write. **NOTE:** All USB port interface modes are listed in this field description, but not all are supported. For detail feature of each controller core, please see Features . The behaviour is unknown when unsupported interface mode is selected.
29 STS	STS Serial Transceiver Select 1 Serial Interface Engine is selected 0 Parallel Interface signals is selected Serial Interface Engine can be used in combination with UTMI+/ULPI physical interface to provide FS/LS signaling instead of the parallel interface signals. When this bit is set '1b', serial interface engine will be used instead of parallel interface signals. This bit has no effect unless PTS bits is set to select UTMI+/ULPI interface. The Serial/USB1.1 PHY/IC-USB will use the serial interface engine for FS/LS signaling regardless of this bit value.
28 PTW	PTW Parallel Transceiver Width

Table continues on the next page...

Field	Description
	This bit has no effect if serial interface engine is used. 0 - Select the 8-bit UTMI interface [60MHz] 1 - Select the 16-bit UTMI interface [30MHz]
27-26 PSPD	PSPD Port Speed - Read Only. This register field indicates the speed at which the port is operating. 00 - Full Speed 01 - Low Speed 10 - High Speed 11 - Undefined
25 PTS_2	PTS_2 See description at bits 31-30
24 PFSC	PFSC Port Force Full Speed Connect - Read/Write. Default = 0b. When this bit is set to '1b', the port will be forced to only connect at Full Speed, It disables the chirp sequence that allows the port to identify itself as High Speed. 0 - Normal operation 1 - Forced to full speed
23 PHCD	PHCD PHY Low Power Suspend - Clock Disable (PLPSCD) - Read/Write. Default = 0b. When this bit is set to '1b', the PHY clock is disabled. Reading this bit will indicate the status of the PHY clock. **NOTE:** The PHY clock cannot be disabled if it is being used as the system clock. In device mode, The PHY can be put into Low Power Suspend when the device is not running (USBCMD Run/Stop=0b) or the host has signalled suspend (PORTSC1 SUSPEND=1b). PHY Low power suspend will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. 0 - Enable PHY clock 1 - Disable PHY clock
22 WKOC	WKOC Wake on Over-current Enable (WKOC_E) - Read/Write. Default = 0b. Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if *Port Power(*PORTSC1*)* is zero.
21 WKDC	WKDC Wake on Disconnect Enable (WKDSCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if *Port Power(*PORTSC1*)* is zero or in device mode.
20 WKCN	WKCN Wake on Connect Enable (WKCNNT_E) - Read/Write. Default=0b.

Table continues on the next page...

Field	Description
	Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if *Port Power(*PORTSC1*)* is zero or in device mode.
19-16 PTC	PTC Port Test Control - Read/Write. Default = 0000b. Refer to Port Test Mode for the operational model for using these test modes and the USB Specification Revision 2.0, Chapter 7 for details on each test mode. The FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. **NOTE:** *Low speed operations are not supported as a peripheral device.* Any other value than zero indicates that the port is operating in test mode. Value Specific Test 0000 - TEST_MODE_DISABLE 0001 - J_STATE 0010 - K_STATE 0011 - SE0 (host) / NAK (device) 0100 - Packet 0101 - FORCE_ENABLE_HS 0110 - FORCE_ENABLE_FS 0111 - FORCE_ENABLE_LS 1000-1111 - Reserved
15-14 PIC	PIC Port Indicator Control - Read/Write. Default = Ob. Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is a zero. Refer to the USB Specification Revision 2.0 for a description on how these bits are to be used. This field is zero if *Port Power* is zero. Bit Value Meaning 00 - Port indicators are off 01 - Amber 10 - Green 11 - Undefined
13 PO	PO Port Owner-Read/Write. Default = 0. This bit unconditionally goes to a 0 when the configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that an internal companion controller owns and controls the port. Port owner handoff is not supported in all controller cores, therefore this bit will always be 0.
12	PP

Table continues on the next page...

Field	Description
PP	Port Power (PP)-Read/Write or Read Only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC PP Operation 0 1b *Read Only - Host controller does not have port power control switches. Each port is hard-wired to power.* 1 1b/0b - *Read/Write. OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (that is, PP equals a 0), the port is non-functional and will not report attaches, detaches, etc.* When an over-current condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitional by the host controller driver from a one to a zero (removing power from the port). This feature is implemented in all controller cores (PPC = 1).
11-10 LS	LS Line Status-Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary. The encoding of the bits are: Bits [11:10] Meaning 00 - SE0 01 - K-state 10 - J-state 11 - Undefined
9 HSP	HSP High-Speed Port - Read Only. Default = 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to zero, the host/device connected to the port is not in a high-speed mode. **NOTE:** HSP is redundant with PSPD(bit 27, 26) but remained for compatibility.
8 PR	PR Port Reset - Read/Write or Read Only. Default = 0b. In Host Mode: Read/Write. 1=Port is in Reset. 0=Port is not in Reset. Default 0. When software writes a one to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. *This bit will automatically change to zero after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the reset duration is timed in the driver.* In Device Mode: This bit is a read only status bit. Device reset from the USB bus is also indicated in the USBSTS register. This field is zero if *Port Power(*PORTSC1*)* is zero.

Table continues on the next page...

Field	Description
7 SUSP	SUSP Suspend - Read/Write or Read Only. Default = 0b. 1=Port in suspend state. 0=Port not in suspend state. In Host Mode: Read/Write. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. The host controller will unconditionally set this bit to zero when software sets the *Force Port Resume* bit to zero. The host controller ignores a write of zero to this bit. If host software sets this bit to a one when the port is not enabled (that is, *Port enabled* bit is a zero) the results are undefined. This field is zero if *Port Power(*PORTSC1*)* is zero in host mode. In Device Mode: Read Only. In device mode this bit is a read only status bit.
6 FPR	FPR Force Port Resume -Read/Write. 1= Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. Default = 0. In Host Mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to one. *This bit will automatically change to zero after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the resume duration is timed in the driver.* Note that when the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. This bit will remain a one until the port has switched to the high-speed idle. Writing a zero has no effect because the port controller will time the resume operation, clear the bit the port control state switches to HS or FS idle. This field is zero if *Port Power(*PORTSC1*)* is zero in host mode. This bit is not-EHCI compatible. In Device mode: After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit wil be cleared because a K-to-J transition detected, the *Port Change Detect* bit in the USBSTS register is also set to one.
5	OCC

Table continues on the next page...

Field	Description
OCC	Over-current Change-R/WC. Default=0. This bit is set '1b' by hardware when there is a change to Over-current Active. Software can clear this bit by writing a one to this bit position.
4 OCA	OCA Over-current Active-Read Only. Default 0. This bit will automatically transition from one to zero when the over current condition is removed. 0 - This port does not have an over-current condition. 1 - This port currently has an over-current condition
3 PEC	PEC Port Enable/Disable Change-R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode: For the root hub, this bit is set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this by writing a one to it. This field is zero if *Port Power(*PORTSC1*)* is zero. In Device mode: The device port is always enabled, so this bit is always '0b'.
2 PE	PE Port Enabled/Disabled-Read/Write. 1=Enable. 0=Disable. Default 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is zero if *Port Power(*PORTSC1*)* is zero in host mode. In Device Mode: The device port is always enabled, so this bit is always '1b'.
1 CSC	CSC Connect Status Change-R/WC. 1 =Change in Current Connect Status. 0=No change. Default 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be 'setting' an already-set bit (that is, the bit will remain set). Software clears this bit by writing a one to it. This field is zero if *Port Power(*PORTSC1*)* is zero in host mode. In Device Mode: This bit is undefined in device controller mode.
0	CCS

Field	Description
CCS	Current Connect Status-Read Only. In Host Mode: 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the *Connect Status Change* bit (Bit 1) to be set. This field is zero if *Port Power(*PORTSC1*)* is zero in host mode. In Device Mode: 1=Attached. 0=Not Attached. Default=0. A one indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. A zero indicates that the device did not attach successfully or was forcibly disconnected by the software writing a zero to the Run bit in the USBCMD register. It does not state the device being disconnected or suspended.

62.7.1.33 On-The-Go Status & control (OTGSC)

This register is availabe only in OTG controller core. It has four sections:

- OTG Interrupt enables (Read/Write)
- OTG Interrupt status (Read/Write to Clear)
- OTG Status inputs (Read Only)
- OTG Controls (Read/Write)

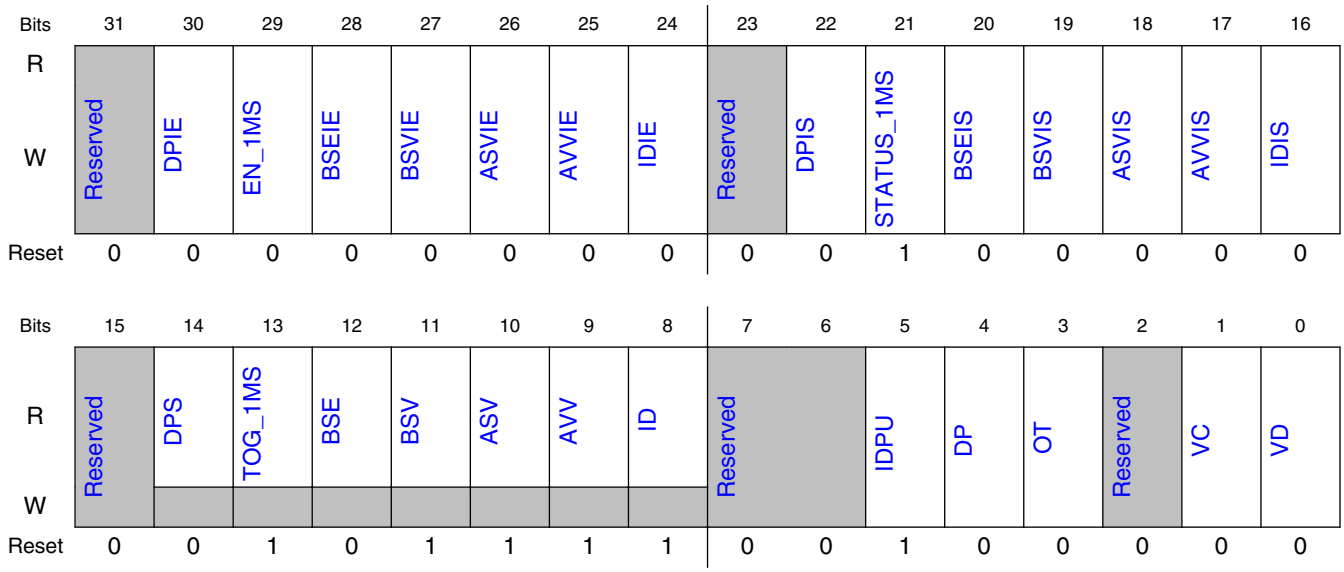
The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms does not cause an update of the status input register, or cause an OTG interrupt.

See also USBMODE register.

62.7.1.33.1 Offset

Register	Offset
OTGSC	1A4h

62.7.1.33.2 Diagram



62.7.1.33.3 Fields

Field	Description
31 —	- Reserved
30 DPIE	DPIE Data Pulse Interrupt Enable
29 EN_1MS	EN_1MS 1 millisecond timer Interrupt Enable - Read/Write
28 BSEIE	BSEIE B Session End Interrupt Enable - Read/Write. Setting this bit enables the B session end interrupt.
27 BSVIE	BSVIE B Session Valid Interrupt Enable - Read/Write. Setting this bit enables the B session valid interrupt.
26 ASVIE	ASVIE A Session Valid Interrupt Enable - Read/Write. Setting this bit enables the A session valid interrupt.
25 AVVIE	AVVIE A VBus Valid Interrupt Enable - Read/Write. Setting this bit enables the A VBus valid interrupt.
24 IDIE	IDIE USB ID Interrupt Enable - Read/Write.

Table continues on the next page...

Field	Description
	Setting this bit enables the USB ID interrupt.
23 —	- Reserved
22 DPIS	DPIS Data Pulse Interrupt Status - Read/Write to Clear. This bit is set when data bus pulsing occurs on DP or DM. Data bus pulsing is only detected when USBMODE.CM = Host (11) and PORTSC1(0)[PP] = 0. Software must write a one to clear this bit.
21 STATUS_1MS	STATUS_1MS 1 millisecond timer Interrupt Status - Read/Write to Clear. This bit is set once every millisecond. Software must write a one to clear this bit.
20 BSEIS	BSEIS B Session End Interrupt Status - Read/Write to Clear. This bit is set when VBus has fallen below the B session end threshold. Software must write a one to clear this bit.
19 BSVIS	BSVIS B Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the B session valid threshold. Software must write a one to clear this bit.
18 ASVIS	ASVIS A Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the A session valid threshold. Software must write a one to clear this bit.
17 AVVIS	AVVIS A VBus Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold on an A device. Software must write a one to clear this bit.
16 IDIS	IDIS USB ID Interrupt Status - Read/Write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.
15 —	- Reserved
14 DPS	DPS Data Bus Pulsing Status - Read Only. A '1' indicates data bus pulsing is being detected on the port.
13 TOG_1MS	TOG_1MS 1 millisecond timer toggle - Read Only. This bit toggles once per millisecond.

Table continues on the next page...

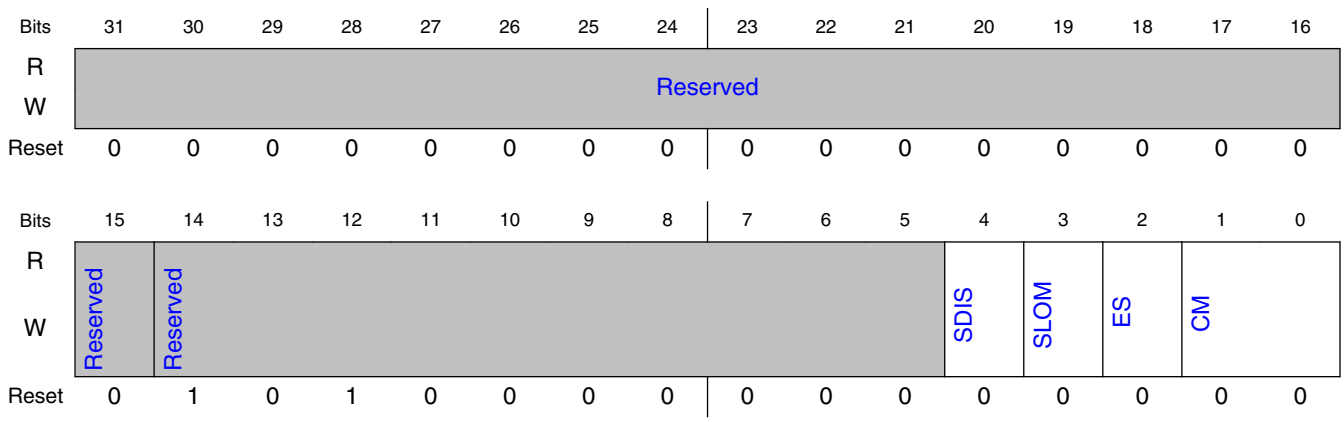
Field	Description
12 BSE	BSE B Session End - Read Only. Indicates VBus is below the B session end threshold.
11 BSV	BSV B Session Valid - Read Only. Indicates VBus is above the B session valid threshold.
10 ASV	ASV A Session Valid - Read Only. Indicates VBus is above the A session valid threshold.
9 AVV	AVV A VBus Valid - Read Only. Indicates VBus is above the A VBus valid threshold.
8 ID	ID USB ID - Read Only. 0 = A device, 1 = B device
7-6 —	- Reserved
5 IDPU	IDPU ID Pullup - Read/Write This bit provide control over the ID pull-up resistor; 0 = off, 1 = on [default]. When this bit is 0, the ID input will not be sampled.
4 DP	DP Data Pulsing - Read/Write. Setting this bit causes the pullup on DP to be asserted for data pulsing during SRP.
3 OT	OT OTG Termination - Read/Write. This bit must be set when the OTG device is in device mode, this controls the pulldown on DM.
2 —	- Reserved
1 VC	VC VBUS Charge - Read/Write. Setting this bit causes the VBus line to be charged. This is used for VBus pulsing during SRP.
0 VD	VD VBUS_Discharge - Read/Write. Setting this bit causes VBus to discharge through a resistor.

62.7.1.34 USB Device Mode (USBMODE)

62.7.1.34.1 Offset

Register	Offset
USBMODE	1A8h

62.7.1.34.2 Diagram



62.7.1.34.3 Fields

Field	Description
31-16 —	- Reserved
15 —	- Reserved
14-5 —	- Reserved
4 SDIS	SDIS Stream Disable Mode. (0 - Inactive [default]; 1 - Active) Device Mode: Setting to a '1' disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standard double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. Note: In High Speed Mode, all packets received are responded to with a NYET handshake when stream disable is active.

Table continues on the next page...

Field	Description
	Host Mode: Setting to a '1' ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the TX latency is filled to capacity before the packet is launched onto the USB. **NOTE:** Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TXFILLTUNING and TXTTFILLTUNING [MPH Only] to characterize the adjustments needed for the scheduler when using this feature. **NOTE:** The use of this feature substantially limits of the overall USB performance that can be achieved.
3 SLOM	SLOM Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. See Control Endpoint Operation Model . 0 - Setup Lockouts On (default); 1 - Setup Lockouts Off. DCD requires use of Setup Data Buffer Tripwire in USBCMD.
2 ES	ES Endian Select - Read/Write. This bit can change the byte alignment of the transfer buffers to match the host microprocessor. The bit fields in the microprocessor interface and the data structures are unaffected by the value of this bit because they are based upon the 32-bit word. Bit Meaning 0 - Little Endian [Default] 1 - Big Endian
1-0 CM	CM Controller Mode - R/WO. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host/device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the *RESET* bit in the USBCMD register before reprogramming this register. For OTG controller core, reset value is '00b'. 00 - Idle [Default for combination host/device] 01 - Reserved 10 - Device Controller [Default for device only controller] 11 - Host Controller [Default for host only controller]

62.7.1.35 Endpoint Setup Status (ENDPTSETUPSTAT)

62.7.1.35.1 Offset

Register	Offset
ENDPTSETUPSTAT	1ACh

62.7.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENDPTSETUPSTAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.35.3 Fields

Field	Description
31-16 —	- Reserved
15-0 ENDPTSETUPSTAT	ENDPTSETUPSTAT Setup Endpoint Status. For every setup transaction that is received, a corresponding bit in this register is set to one. Software must clear or acknowledge the setup transfer by writing a one to a respective bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock our mechanism is engaged. See Managing Endpoints in the Device Operational Model. This register is only used in device mode.

62.7.1.36 Endpoint Prime (ENDPTPRIME)

This register is only used in device mode.

When software sets the prime bit for a given endpoint, the device controller loads the transfer descriptor, pointed to by the queue head, such that the endpoint is ready to transmit or receive when the host sends a request (IN/OUT token). The endpoint will NAK all requests from the host until the endpoint is primed. The controller will automatically re-prime the endpoint with a new transfer descriptor when one is found via the next_dtd pointer of the current transfer descriptor. Hence, the prime bit must only be set by software when a descriptor is added to the queue head.

62.7.1.36.1 Offset

Register	Offset
ENDPTPRIME	1B0h

62.7.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								PETB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PERB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.36.3 Fields

Field	Description
31-24 —	- Reserved
23-16 PETB	PETB Prime Endpoint Transmit Buffer - R/WS. For each endpoint a corresponding bit is used to request that a buffer is prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[N] - Endpoint #N, N is in 0..7
15-8 —	- Reserved
7-0 PERB	PERB Prime Endpoint Receive Buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint queue head. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated.

Field	Description
	PERB[N] - Endpoint #N, N is in 0..7

62.7.1.37 Endpoint Flush (ENDPTFLUSH)

This register is only used in device mode.

62.7.1.37.1 Offset

Register	Offset
ENDPTFLUSH	1B4h

62.7.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								FETB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								FERB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.37.3 Fields

Field	Description
31-24 —	- Reserved
23-16 FETB	FETB Flush Endpoint Transmit Buffer - R/WS. Writing one to a bit(s) in this register causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[N] - Endpoint #N, N is in 0..7
15-8	-

Table continues on the next page...

Field	Description
—	Reserved
7-0 FERB	FERB Flush Endpoint Receive Buffer - R/WS. Writing one to a bit(s) causes the assocUOGiated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[N] - Endpoint #N, N is in 0..7\

62.7.1.38 Endpoint Status (ENDPTSTAT)

This register is only used in device mode.

62.7.1.38.1 Offset

Register	Offset
ENDPTSTAT	1B8h

62.7.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								ETBR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERBR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.38.3 Fields

Field	Description
31-24 —	- Reserved
23-16 ETBR	ETBR

Table continues on the next page...

Field	Description
	Endpoint Transmit Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to one by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[N] - Endpoint #N, N is in 0..7
15-8 ——	- Reserved
7-0 ERBR	ERBR Endpoint Receive Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to a one by the hardware as a response to receiving a command from a corresponding bit in the ENDPRIME register. There is always a delay between setting a bit in the ENDPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[N] - Endpoint #N, N is in 0..7

62.7.1.39 Endpoint Complete (ENDPTCOMPLETE)

This register is only used in device mode.

62.7.1.39.1 Offset

Register	Offset
ENDPTCOMPLETE	1BCh

62.7.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ETCE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ERCE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.39.3 Fields

Field	Description
31-24 —	- Reserved
23-16 ETCE	ETCE Endpoint Transmit Complete Event - R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred and software should read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ETCE[N] - Endpoint #N, N is in 0..7
15-8 —	- Reserved
7-0 ERCE	ERCE Endpoint Receive Complete Event - RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ERCE[N] - Endpoint #N, N is in 0..7

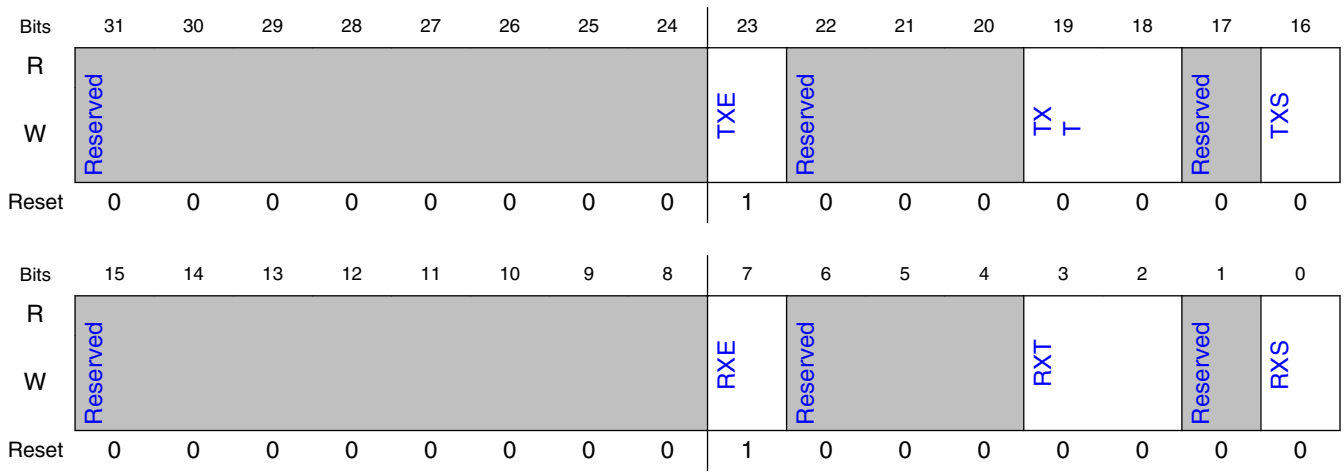
62.7.1.40 Endpoint Control0 (ENDPTCTRL0)

Every Device implements Endpoint 0 as a control endpoint.

62.7.1.40.1 Offset

Register	Offset
ENDPTCTRL0	1C0h

62.7.1.40.2 Diagram



62.7.1.40.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
22-20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 - Control Endpoint0 is fixed as a Control End Point.
17 —	- Reserved
16 TXS	TXS TX Endpoint Stall - Read/Write

Table continues on the next page...

Field	Description
	0 End Point OK [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request. After receiving a SETUP request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. **NOTE:** There is a slight delay (50 clocks max.) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the DCD software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a newsetup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
6-4 —	- Reserved
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control Endpoint0 is fixed as a Control End Point.
1 —	- Reserved
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request. After receiving a SETUP request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. **NOTE:** There is a slight delay (50 clocks max.) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a newsetup has been received by checking the associated endptsetupstat bit.

62.7.1.41 Endpoint Control 1 (ENDPTCTRL1)

This is endpoint control register for endpoint 1 in device operation mode.

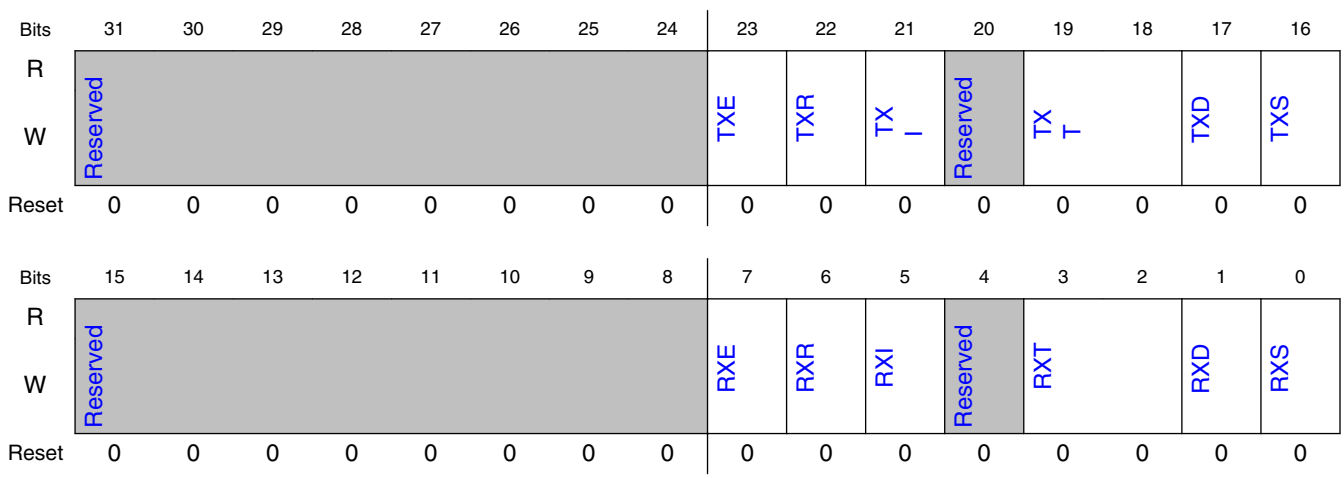
NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.41.1 Offset

Register	Offset
ENDPTCTRL1	1C4h

62.7.1.41.2 Diagram



62.7.1.41.3 Fields

Field	Description
31-24 —	- Reserved
23	TXE

Table continues on the next page...

Field	Description
TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit

Table continues on the next page...

Field	Description
	until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RXI RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared.

Field	Description
	Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.42 Endpoint Control 2 (ENDPTCTRL2)

This is endpoint control register for endpoint 2 in device operation mode.

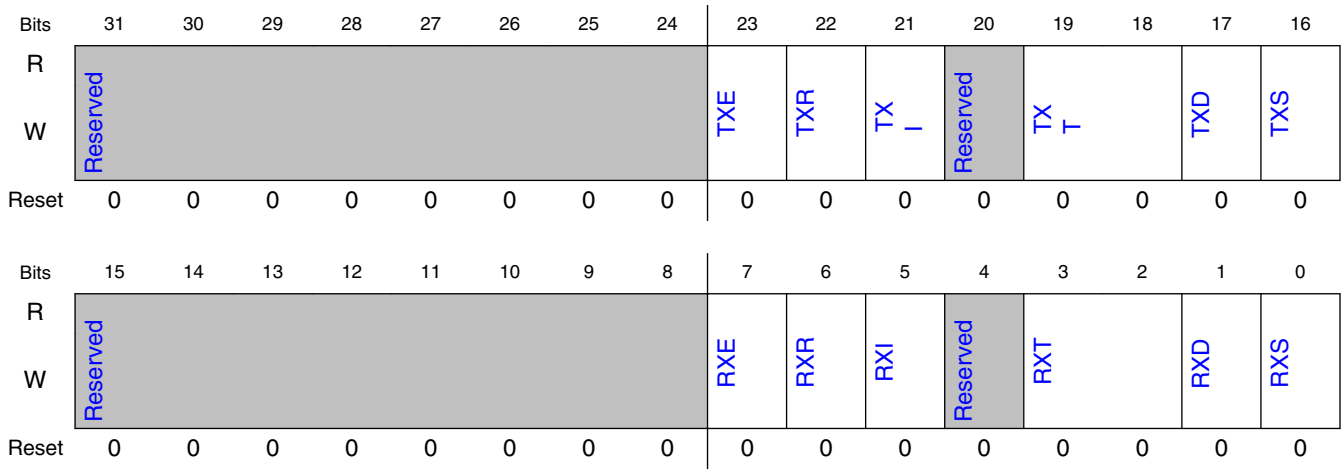
NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.42.1 Offset

Register	Offset
ENDPTCTRL2	1C8h

62.7.1.42.2 Diagram



62.7.1.42.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control

Table continues on the next page...

Field	Description
	01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RXI RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.

Table continues on the next page...

Field	Description
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.43 Endpoint Control 3 (ENDPTCTRL3)

This is endpoint control register for endpoint 3 in device operation mode.

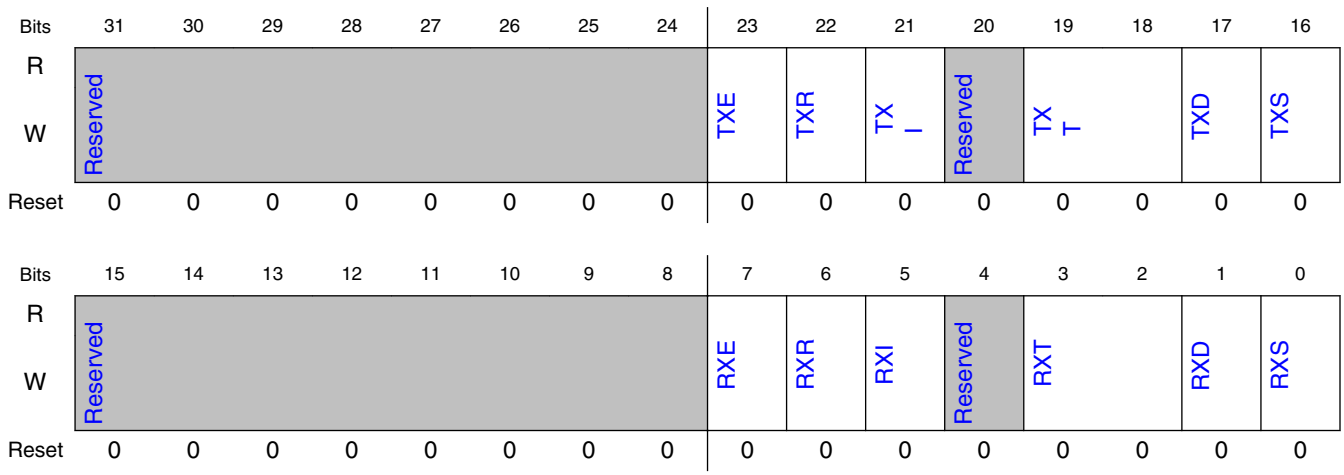
NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.43.1 Offset

Register	Offset
ENDPTCTRL3	1CCh

62.7.1.43.2 Diagram



62.7.1.43.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled.

Table continues on the next page...

Field	Description
	This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5	RXI

Table continues on the next page...

Field	Description
RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.44 Endpoint Control 4 (ENDPTCTRL4)

This is endpoint control register for endpoint 4 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.44.1 Offset

Register	Offset
ENDPTCTRL4	1D0h

62.7.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	TXR	TX I	Reserved	TX T		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.44.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22	TXR

Table continues on the next page...

Field	Description
TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default]

Table continues on the next page...

Field	Description
	1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RXI RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.45 Endpoint Control 5 (ENDPTCTRL5)

This is endpoint control register for endpoint 5 in device operation mode.

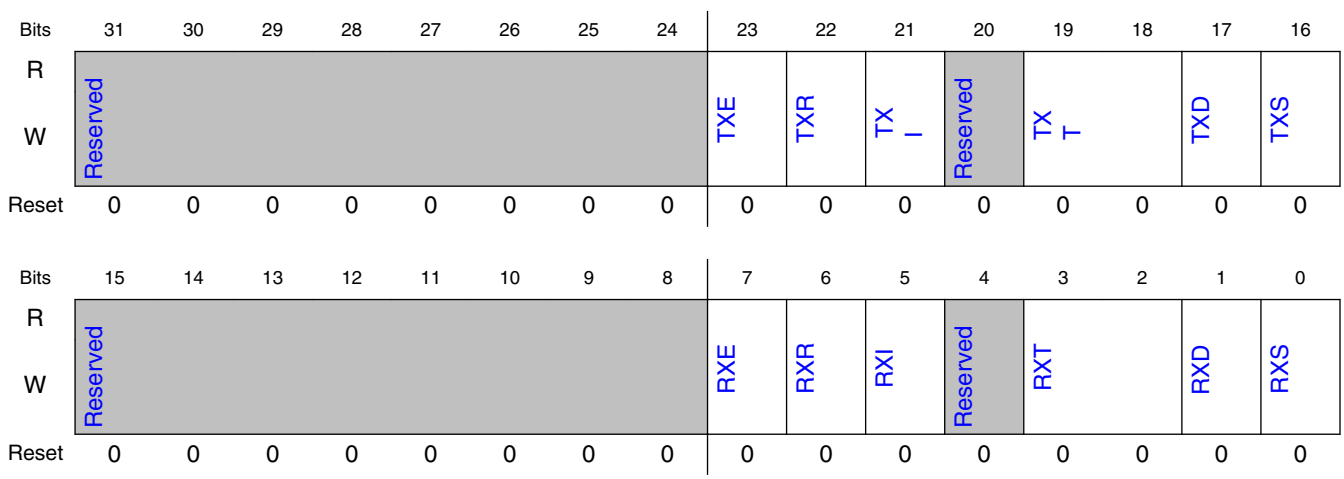
NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.45.1 Offset

Register	Offset
ENDPTCTRL5	1D4h

62.7.1.45.2 Diagram



62.7.1.45.3 Fields

Field	Description
31-24	-

Table continues on the next page...

Field	Description
—	Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the

Table continues on the next page...

Field	Description
	stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RXI RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared.

Field	Description
	Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.46 Endpoint Control 6 (ENDPTCTRL6)

This is endpoint control register for endpoint 6 in device operation mode.

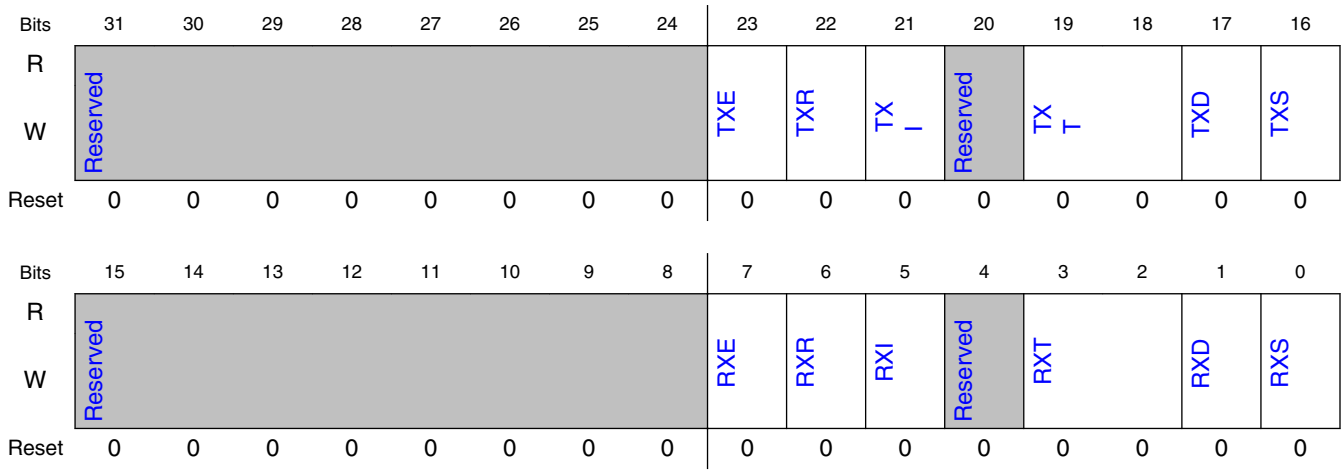
NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.46.1 Offset

Register	Offset
ENDPTCTRL6	1D8h

62.7.1.46.2 Diagram



62.7.1.46.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control

Table continues on the next page...

Field	Description
	01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RXI RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.

Table continues on the next page...

Field	Description
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpointand this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

62.7.1.47 Endpoint Control 7 (ENDPTCTRL7)

This is endpoint control register for endpoint 7 in device operation mode.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

62.7.1.47.1 Offset

Register	Offset
ENDPTCTRL7	1DCh

62.7.1.47.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								TXE	TXR	TX I	Reserved	TX T		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

62.7.1.47.3 Fields

Field	Description
31-24 —	- Reserved
23 TXE	TXE TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TXR TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TXI TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled.

Table continues on the next page...

Field	Description
	This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 —	- Reserved
19-18 TXT	TXT TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TXD TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TXS TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit will be cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
15-8 —	- Reserved
7 RXE	RXE RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RXR RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5	RXI

Table continues on the next page...

Field	Description
RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 —	- Reserved.
3-2 RXT	RXT RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
1 RXD	RXD RX Endpoint Data Sink - Read/Write 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RXS RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. This control will continue to STALL until this bit is either cleared by software or automatically cleared as above for control endpoints. **NOTE:** [CONTROL ENDPOINT TYPES ONLY]: there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. However, should the DCD observe that the stall bit is not set after writing a one to it then follow this procedure: continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.

Chapter 63
Universal Serial Bus 2.0 Integrated PHY (USB-PHY)

63.1 Chip-specific USB-PHY information

Table 63-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

63.2 Overview

The integrated USB 2.0 PHY interfaces to USB host/device systems at the USB low-speed (LS) rate of 1.5 Mbits/s, full-speed (FS) rate of 12 Mbits/s, or at the USB 2.0 high-speed (HS) rate of 480 Mbits/s. The following subsections describe the external interfaces, internal interfaces, major blocks, and programmable registers that comprise the integrated USB 2.0 PHY

63.2.1 Block Diagram

The block diagram of the USB PHY. The integrated PHY provides a standard UTM interface. The DP and DN pins connect directly to a USB connector.

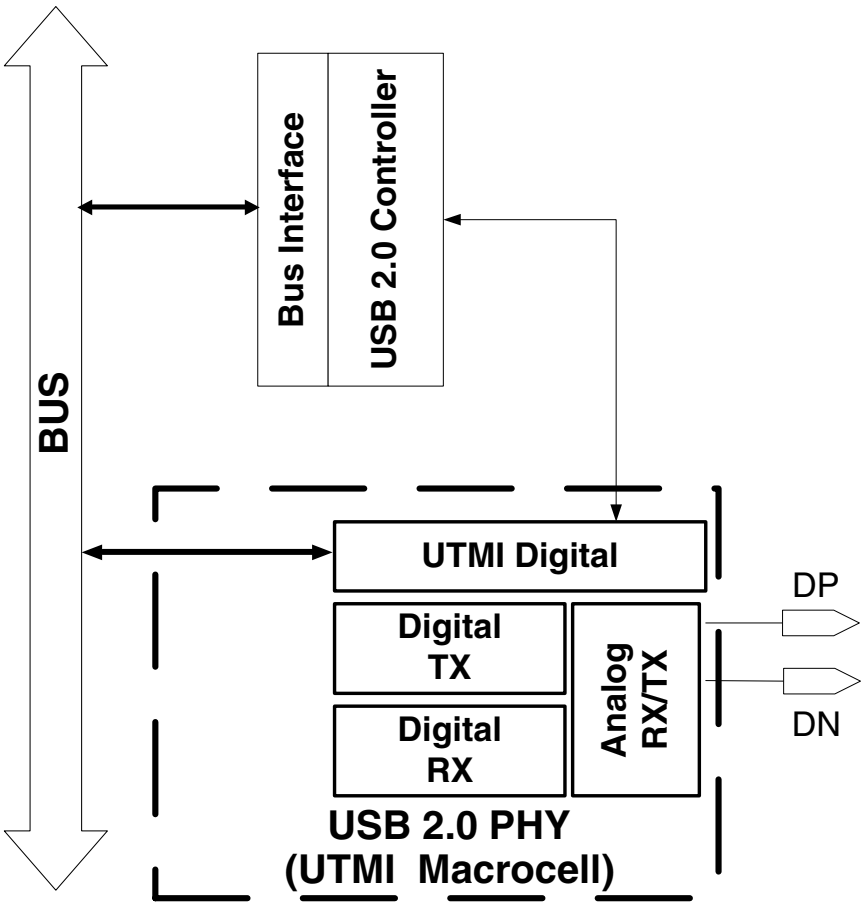


Figure 63-1. USB 2.0 PHY Block Diagram

63.3 Functional Description

This section covers the core components of the USB PHY.

- The digital portions of the USBPHY block include the UTMI, digital transmitter, digital receiver, and the programmable registers.
- The analog transceiver section comprises an analog receiver and an analog transmitter, as shown in Figure 63-2.

63.3.1 UTMI

The UTMI block handles the line_state bits, reset buffering, suspend distribution, transceiver speed selection, transceiver termination selection, and clock generation logic for the clocks derived from the PHY's USB PLL.

The PLL in the USB PHY supplies a 480MHz clock to UTMI digital. The UTMI block will divide the clock to generate the 30MHz clock used in the interface.

63.3.2 Digital Transmitter

The digital transmitter receives the 16-bit transmit data from the USB controller and handles the tx_valid, tx_validh and tx_ready handshake. In addition, it contains the transmit serializer that converts the 16-bit parallel words at 30 MHz to a single bitstream at 480 Mbit for high-speed or 12 Mbit for full-speed or 1.5 Mbit for low-speed. It does this while implementing the bit-stuffing algorithm and the NRZI encoder that are used to remove the DC component from the serial bitstream. The output of this encoder is sent to the low-speed (LS), full-speed (FS) or high-speed (HS) drivers in the analog transceiver section's transmitter block.

63.3.3 Digital Receiver

The digital receiver receives the raw serial bitstream from the low speed (LS) differential transceiver, full speed (FS) differential transceiver, and 480 MHz sampled data from the high speed (HS) differential transceiver. As the phase of the USB host transmitter shifts relative to the local PLL, the receiver section's HS DLL tracks these changes to give a reliable sample of the incoming 480 Mbit/s bitstream. Since this sample point shifts relative to the PLL phase used by the digital logic, a rate-matching elastic buffer is provided to cross this clock domain boundary.

Once the bitstream is in the local clock domain, an NRZI decoder and bit unstuffer restore the original payload data bitstream and pass it to a deserializer and holding register. The receive state machine handles the rx_valid, rx_validh, and handshake with the USB controller. The handshake is not interlocked, in that there is no rx_ready signal coming from the controller. The controller must take each 16-bit value as presented by the PHY. The receive state machine provides an rx_active signal to the controller that indicates when it is inside a valid packet (SYNC detected, and so on).

63.3.4 Analog Receiver

The analog receiver is comprised of five differential receivers, two single-ended receivers, a 9X, and a 480 MHz HS data sampling module that are illustrated in the figure below and described further in this section.

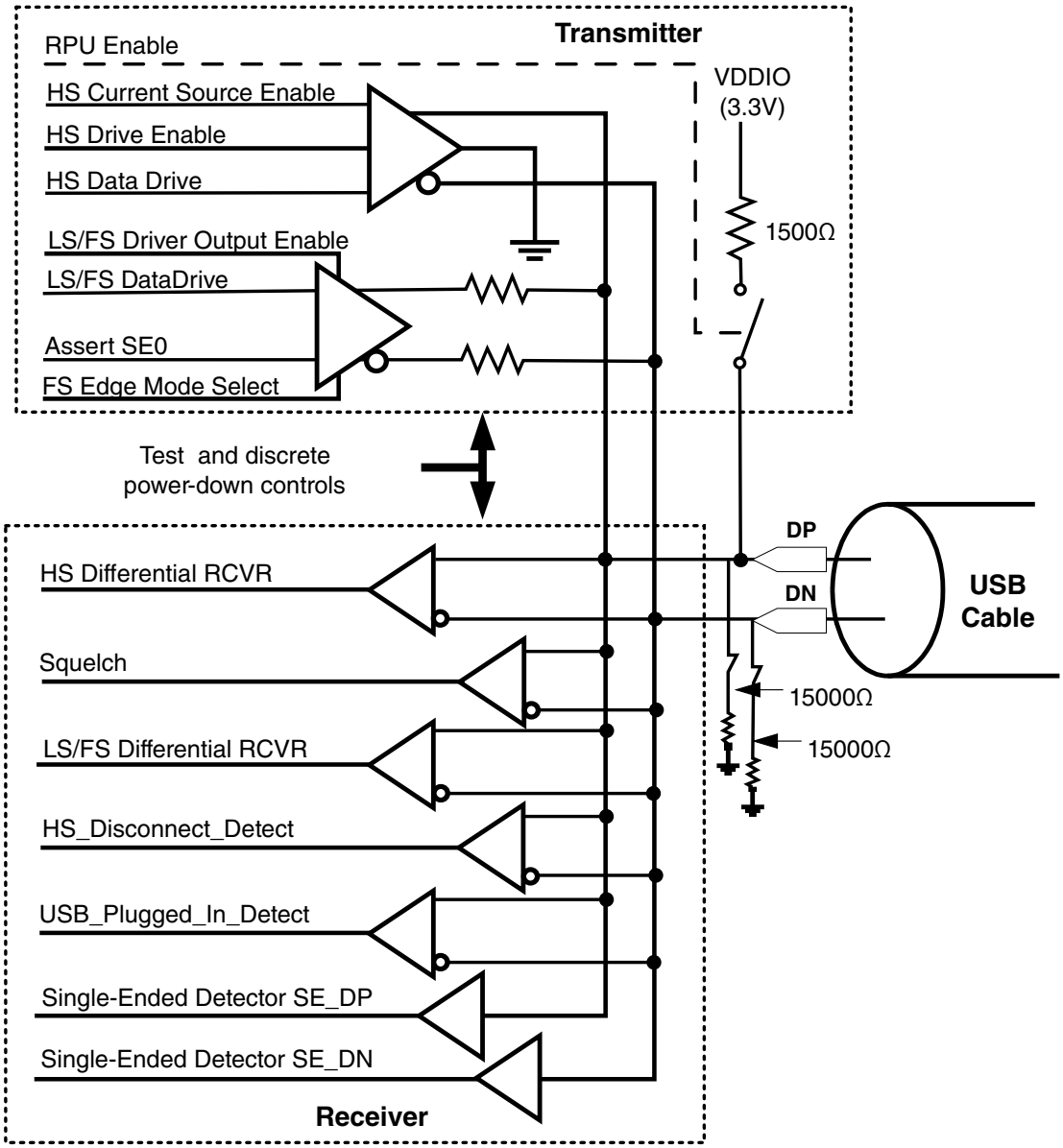


Figure 63-2. USB 2.0 PHY Analog Transceiver Block Diagram

63.3.4.1 HS Differential Receiver

The high-speed differential receiver is both a differential analog receiver and threshold comparator. Its output is high (1'b1) if the differential signal is greater than a 0-V threshold. Otherwise, its output is low (1'b0). Its purpose is to discriminate the ± 400-mV differential voltage resulting from the high-speed drivers current flow into the dual 45Ω terminations found on each pin of the differential pair. The envelope or squelch detector, described below, ensures that the differential signal has sufficient magnitude to be valid. The HS differential receiver tolerates up to 500 mV of common mode offset.

63.3.4.2 Squelch Detector

The squelch detector is a differential analog receiver and threshold comparator. Its output is high (1'b1) if the differential magnitude is less than a nominal 100 mV threshold. Otherwise, its output is low (1'b0). Its purpose is to invalidate the HS differential receiver when the incoming signal is too low to receive reliably.

63.3.4.3 LS/FS Differential Receiver

The low-speed/full-speed differential receiver is both a differential analog receiver and threshold comparator. The crossover voltage falls between 1.3 V and 2.0 V. Its output is 1, when the DP line is above the crossover point and the DN line is below the crossover point. The digital receiver section decodes the receiver data into J or K state according to the speed.

63.3.4.4 HS Disconnect Detector

It is a differential analog receiver and threshold comparator. It outputs high when differential magnitude is greater than a nominal 575-mV threshold. Otherwise, it outputs low.

63.3.4.5 USB Plugged-In Detector

When operating in device mode, USB PHY provides the following methods for the local USB device to detect the attachment of a cable to a remote USB host or hub. Only one method should be enabled at a time when waiting to identify the cable attachment event, and all of methods must be disabled during USB data communication.

- USB plugged-in detector method looks for both DP and DN to be high. There is a pair of large on-chip pullup resistors (200 KΩ) that hold both DP and DN high when the USB cable is not attached. The USB plugged-in detector signals a 0 in this case.When operating in device mode, the upstream port in host/hub interface contains a 15 KΩ pulldown resistor which could easily override the 200 KΩ pullup resistor. When plugged in, at least one signal in the pair will be low, which will force the plugged-in detector's output high. The USB plugged-in detector is controlled through the CTRL[ENDEVPLUGINDET] bit field and results are observable through the STATUS[DEVPLUGIN_STATUS] bit field.

- The Data Contact Detect method functions per the *USB Battery Charging Specification, Revision 1.2*. The use of this method is described in the USB Device Charger Detection (USBDCD) Chapter.
- Resistive detection method described in the *USB Battery Charging Specification, Revision 1.0*. This method enables a 125 KΩ pullup resistor on the DP pin and a 375 KΩ pulldown resistor on the DN pin. When no cable is plugged in, the DP pin will be high and the DN pin will be low. If the local device has a cable attached to a remote host or hub, due to the host 15 KΩ pulldowns both the DP and DN pins will be low. If the local device has a cable attached to a dedicated charger with no pullup/ pulldown resistors, both the DP and DN pins will be high. This method is enabled through the VBUS_DETECT[EN_CHARGER_RESISTOR] bit field. Results can be observed using the single ended receiver status in the CHRG_DET_STAT[DP_STATE, DM_STATE] bit fields.

63.3.4.6 Single-Ended DP Receiver

The single-ended DP receiver output is high whenever the DP input is above its nominal 1.8 V threshold.

63.3.4.7 Single-Ended DN Receiver

The single-ended DN receiver output is high whenever the DN input is above its nominal 1.8 V threshold.

63.3.5 Analog Transmitter

The analog transmitter comprises two differential drivers: one for high-speed signaling and one for full-speed signaling. It also contains the switchable 1.5 KΩ pullup resistor. See Figure 63-3.

63.3.5.1 Switchable High-Speed 45Ω Termination Resistors

High-speed current mode differential signaling requires good 90Ω differential termination at each end of the USB cable. This results from switching in 45Ω terminating resistors from each signal line to ground at each end of the cable.

Because each signal is parallel terminated with 45Ω at each end, each driver sees a 22.5Ω load. This load impedance is much too low for full-speed signaling levels, hence the need for switchable high-speed terminating resistors. Switchable trimming resistors are provided to tune the actual termination resistance of each device, as shown in USB 2.0 PHY Transmitter Block Diagram. The TX[TXCAL45DP] bit field, for example, allows one of 16 trimming resistor values to be placed in parallel with the 45Ω terminator on the DP signal.

63.3.5.2 Low-Speed/Full-Speed Differential Driver

The low-speed/full-speed differential drivers are essentially a pair of low-impedance pullup/pulldown devices that are switched in a differential mode for most low-speed or full-speed signaling. One output is driven high while the other output is driven low to signal the "J" state or the "K" state. Both outputs are driven low for the low-speed/full-speed "SE0" state.

63.3.5.3 High-Speed Differential Driver

The high-speed differential driver receives a 17.78 mA current from the constant current source (Iref) and essentially steers it down either the DP signal or the DN signal or alternatively to ground. This current will produce approximately a 400 mV drop across the 22.5 Ω termination seen by the driver when it is steered onto one of the signal lines. The approximately 17.78 mA current source is referenced back to the integrated voltage-band-gap (Vbg) circuit.

63.3.5.4 Switchable 1.5KΩ DP Pullup Resistor

This product contains a switchable 1.5 KΩ pullup resistor on the DP signal. This resistor is switched on to indicate to the host/hub controller that a full-speed-capable device is on the USB cable, powered on, and ready. This resistor is switched off at power-on reset so the host does not recognize a USB device until the processor software enables the announcement of a full-speed device.

63.3.5.5 Switchable 15KΩ DP and DN Pulldown Resistors

The chip contains switchable 15 KΩ pulldown resistors on both DP and DN signals. They are enabled in host mode to indicate to the device controller that a host is present.

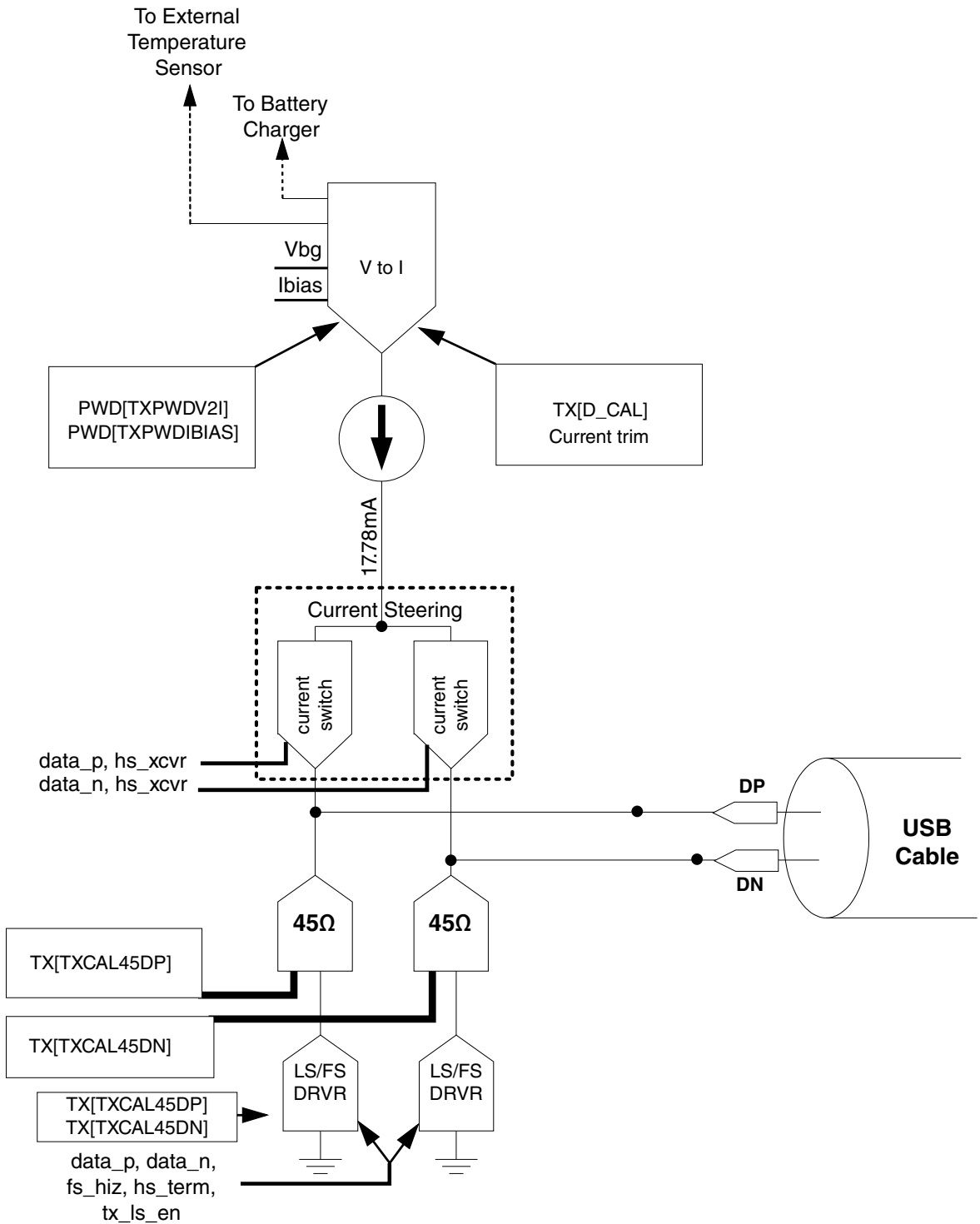


Figure 63-3. USB 2.0 PHY Transmitter Block Diagram

63.3.6 Clocks

The following table describes the clock sources for the USB PHY module. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 63-2. USB PHY Clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_s	Peripheral access clock

63.4 External Signals

There are no external signals pinned out for the USB PHY.

63.5 Initialization and Application Information

In order to bring up the internal 480MHz USB PLL clock, several conditions should be met, as noted below:

1. The reference clock for USB PLL must be enabled.
2. Enable the USB PHY PLL Regulator
3. Release the PHY from Reset
4. Power up the PLL
5. The system oscillator ranges from 16 MHz to 32 MHz. PLL_SIC[PLL_DIV_SEL] should be programmed properly as per the external reference selected.

63.6 Recommended Register Configuration for USB Certification

The USB HS PHY has programmability to adjust several transceiver parameters.

- The TX HS driver termination impedance, FS driver output impedance, and HS driver output currents can be adjusted through bit fields of the TX Register.
  - The ability to override the default parametric trim bit fields using the TX Register has to be enabled by setting bit fields in the TRIM_OVERRIDE_EN Register.

- The TX HS driver pre-emphasis feature can be enabled and configured through bitfields of the ANACTRL Register.
- The HS RX thresholds for the Envelope Detector/Squelch comparator and the Host Disconnect comparator can be adjusted through bit fields of the RX Register.

The default values of the bit fields for the transceiver parametric trims are centered with no changes needed for USB Certification testing when the chip is used with boards optimized for signal integrity on the HS USB Port.

In cases where external components are inserted between the USB DP/DN pins, or compromises are made on USB DP/DN signal routing, changes to the parametric trim bit fields may be useful.

Depending on the board layout, to pass the certification, trim the register repeatedly until the eye diagram meets the requirements of the template.

63.7 Register Macro Usage

A common operation is to update one field without disturbing the contents of the remaining fields in the register. Normally, this requires a read-modify-write (RMW) operation, where the CPU reads the register, modifies the target field, then writes the results back to the register. This is an expensive operation in terms of CPU cycles, because of the initial register read.

To address this issue, some hardware registers are implemented as a group, including registers that can be used to either set, clear, or toggle (SCT) individual bits of the primary register. When writing to an SCT register, all bits set to 1 perform the associated operation on the primary register, while all bits set to 0 are not affected. The SCT registers always read back 0, and should be considered write-only. The SCT registers are not implemented if the primary register is read-only.

With this architecture, it is possible to update one or more fields using only register writes. First, all bits of the target fields are cleared by a write to the associated clear register, then the desired value of the target fields is written to the set register. This sequence of two writes is referred to as a clear-set (CS) operation.

A CS operation does have one potential drawback. Whenever a field is modified, the hardware sees a value of 0 before the final value is written. For most fields, passing through the 0 state is not a problem. Nonetheless, this behavior is something to consider when using a CS operation.

Also, a CS operation is not required for fields that are one bit wide. While the CS operation works in this case, it is more efficient to simply set or clear the target bit (that is, one write instead of two). A simple set or clear operation is also atomic, while a CS operation is not.

Note that not all macros for set, clear, or toggle (SCT) are atomic. For registers that do not provide hardware support for this functionality, these macros are implemented as a sequence of read/modify/write operations. When atomic operation is required, the developer should pay attention to this detail, because unexpected behavior might result if an interrupt occurs in the middle of the critical section comprising the update sequence.

63.8 USB PHY Memory Map/Register Definition

The USB PHY Memory Map/Register Definition can be found here.

63.8.1 USBPHY register descriptions

USBPHY Hardware Register Format Summary

Information for using the Set, Clear, Toggle register features is located in the **Register Macro Usage** section.

63.8.1.1 USBPHY memory map

USBPHY1 base address: 4043_4000h

USBPHY2 base address: 4043_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	USB PHY Power-Down Register (PWD)	32	RW	001E_1C00
4	USB PHY Power-Down Register (PWD_SET)	32	RW	001E_1C00
8	USB PHY Power-Down Register (PWD_CLR)	32	RW	001E_1C00
C	USB PHY Power-Down Register (PWD_TOG)	32	RW	001E_1C00
10	USB PHY Transmitter Control Register (TX)	32	RW	1006_0607
14	USB PHY Transmitter Control Register (TX_SET)	32	RW	1006_0607
18	USB PHY Transmitter Control Register (TX_CLR)	32	RW	1006_0607
1C	USB PHY Transmitter Control Register (TX_TOG)	32	RW	1006_0607

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
20	USB PHY Receiver Control Register (RX)	32	RW	0000_0000
24	USB PHY Receiver Control Register (RX_SET)	32	RW	0000_0000
28	USB PHY Receiver Control Register (RX_CLR)	32	RW	0000_0000
2C	USB PHY Receiver Control Register (RX_TOG)	32	RW	0000_0000
30	USB PHY General Control Register (CTRL)	32	RW	8800_0000
34	USB PHY General Control Register (CTRL_SET)	32	RW	8800_0000
38	USB PHY General Control Register (CTRL_CLR)	32	RW	8800_0000
3C	USB PHY General Control Register (CTRL_TOG)	32	RW	8800_0000
40	USB PHY Status Register (STATUS)	32	RW	0000_0000
50	USB PHY Debug Register (DEBUG)	32	RW	7F18_0000
54	USB PHY Debug Register (DEBUG_SET)	32	RW	7F18_0000
58	USB PHY Debug Register (DEBUG_CLR)	32	RW	7F18_0000
5C	USB PHY Debug Register (DEBUG_TOG)	32	RW	7F18_0000
60	UTMI Debug Status Register 0 (DEBUG0_STATUS)	32	RO	0000_0000
70	UTMI Debug Status Register 1 (DEBUG1)	32	RW	0000_1000
74	UTMI Debug Status Register 1 (DEBUG1_SET)	32	RW	0000_1000
78	UTMI Debug Status Register 1 (DEBUG1_CLR)	32	RW	0000_1000
7C	UTMI Debug Status Register 1 (DEBUG1_TOG)	32	RW	0000_1000
80	UTMI RTL Version (VERSION)	32	RO	0403_0000
A0	USB PHY PLL Control/Status Register (PLL_SIC)	32	RW	00D1_2000
A4	USB PHY PLL Control/Status Register (PLL_SIC_SET)	32	RW	00D1_2000
A8	USB PHY PLL Control/Status Register (PLL_SIC_CLR)	32	RW	00D1_2000
AC	USB PHY PLL Control/Status Register (PLL_SIC_TOG)	32	RW	00D1_2000
C0	USB PHY VBUS Detect Control Register (USB1_VBUS_DETECT)	32	RW	0070_0004
C4	USB PHY VBUS Detect Control Register (USB1_VBUS_DETECT_SET)	32	RW	0070_0004
C8	USB PHY VBUS Detect Control Register (USB1_VBUS_DETECT_CLR)	32	RW	0070_0004
CC	USB PHY VBUS Detect Control Register (USB1_VBUS_DETECT_TOG)	32	RW	0070_0004
D0	USB PHY VBUS Detector Status Register (USB1_VBUS_DET_STAT)	32	RO	0000_0000
E0	USB PHY Charger Detect Control Register (USB1_CHRG_DETECT)	32	RW	8018_0000
E4	USB PHY Charger Detect Control Register (USB1_CHRG_DETECT_SET)	32	RW	8018_0000
E8	USB PHY Charger Detect Control Register (USB1_CHRG_DETECT_CLR)	32	RW	8018_0000
EC	USB PHY Charger Detect Control Register (USB1_CHRG_DETECT_TOG)	32	RW	8018_0000
F0	USB PHY Charger Detect Status Register (USB1_CHRG_DET_STAT)	32	RO	0000_0000
100	USB PHY Analog Control Register (ANACTRL)	32	RW	0000_0402

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
104	USB PHY Analog Control Register (ANACTRL_SET)	32	RW	0000_0402
108	USB PHY Analog Control Register (ANACTRL_CLR)	32	RW	0000_0402
10C	USB PHY Analog Control Register (ANACTRL_TOG)	32	RW	0000_0402
110	USB PHY Loopback Control/Status Register (USB1_LOOPBACK)	32	RW	0055_0000
114	USB PHY Loopback Control/Status Register (USB1_LOOPBACK_SET)	32	RW	0055_0000
118	USB PHY Loopback Control/Status Register (USB1_LOOPBACK_CLR)	32	RW	0055_0000
11C	USB PHY Loopback Control/Status Register (USB1_LOOPBACK_TOG)	32	RW	0055_0000
120	USB PHY Loopback Packet Number Select Register (USB1_LOOPBACK_HSFSCNT)	32	RW	0004_0010
124	USB PHY Loopback Packet Number Select Register (USB1_LOOPBACK_HSFSCNT_SET)	32	RW	0004_0010
128	USB PHY Loopback Packet Number Select Register (USB1_LOOPBACK_HSFSCNT_CLR)	32	RW	0004_0010
12C	USB PHY Loopback Packet Number Select Register (USB1_LOOPBACK_HSFSCNT_TOG)	32	RW	0004_0010
130	USB PHY Trim Override Enable Register (TRIM_OVERRIDE_EN)	32	RW	0000_007F
134	USB PHY Trim Override Enable Register (TRIM_OVERRIDE_EN_SET)	32	RW	0000_007F
138	USB PHY Trim Override Enable Register (TRIM_OVERRIDE_EN_CLR)	32	RW	0000_007F
13C	USB PHY Trim Override Enable Register (TRIM_OVERRIDE_EN_TOG)	32	RW	0000_007F

63.8.1.2 USB PHY Power-Down Register (PWD)

The USB PHY Power-Down Register provides overall control of the PHY power state. Before programming this register, the PHY clocks must be enabled.

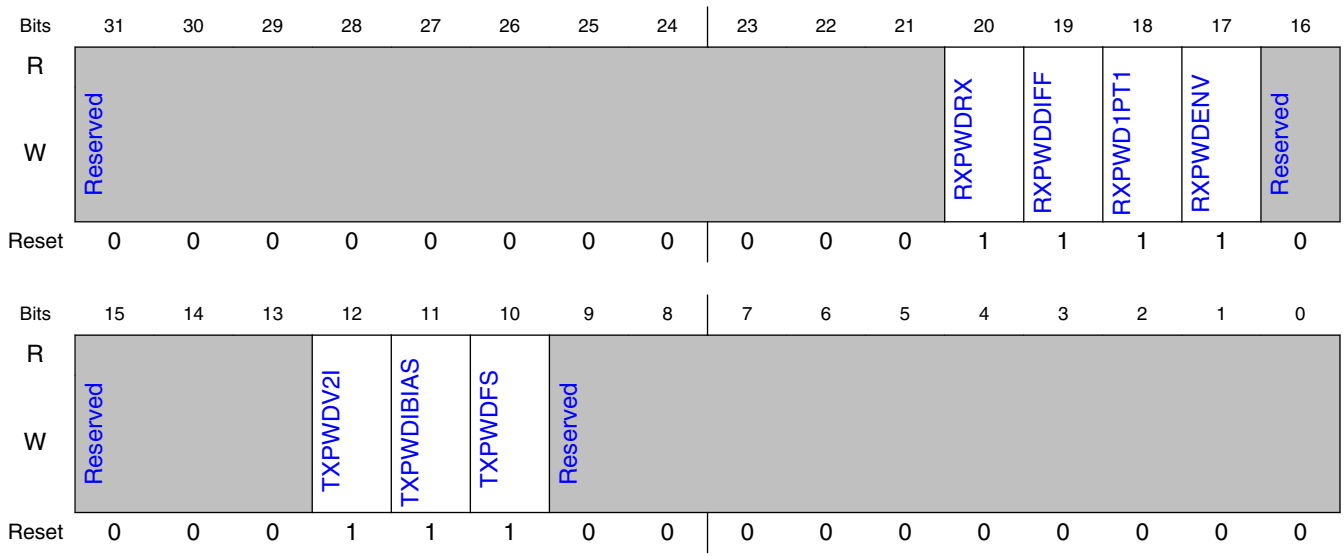
63.8.1.2.1 Offset

Register	Offset	Description
PWD	0h	USB PHY Power-Down Register
PWD_SET	4h	Writing a 1 to a bit in this register sets the corresponding bit in PWD
PWD_CLR	8h	Writing a 1 to a bit in this register clears the corresponding bit in PWD

Table continues on the next page...

Register	Offset	Description
PWD_TOG	Ch	Writing a 1 to a bit in this register toggles the corresponding bit in PWD

63.8.1.2.2 Diagram



63.8.1.2.3 Fields

Field	Description
31-21 —	- Reserved.
20 RXPWDRX	RXPWDRX **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. 0 - Normal operation 1 - Power-down the entire USB PHY receiver block except for the full-speed differential receiver
19 RXPWDDIFF	RXPWDDIFF **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. 0 - Normal operation. 1 - Power-down the USB high-speed differential receiver
18 RXPWD1PT1	RXPWD1PT1 **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. 0 - Normal operation

Table continues on the next page...

Field	Description
	1 - Power-down the USB full-speed differential receiver.
17 RXPWDENV	RXPWDENV **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. 0 - Normal operation. 1 - Power-down the USB high-speed receiver envelope detector (squelch signal)
16-13 —	- Reserved.
12 TXPWDV2I	TXPWDV2I **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. **NOTE:** These circuits are shared with the battery charge circuit. Setting this to 1 does not power-down these circuits, unless the corresponding bit in the battery charger is also set for power-down. 0 - Normal operation. 1 - Power-down the USB PHY transmit V-to-I converter and the current mirror
11 TXPWDIBIAS	TXPWDIBIAS **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. **NOTE:** These circuits are shared with the battery charge circuit. Setting this bit to 1 does not power-down these circuits, unless the corresponding bit in the battery charger is also set for power-down. 0 - Normal operation 1 - Power-down the USB PHY current bias block for the transmitter. This bit should be set only when the USB is in suspend mode. This effectively powers down the entire USB transmit path
10 TXPWDFS	TXPWDFS **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of CTRL is enabled. 0 - Normal operation. 1 - Power-down the USB full-speed drivers. This turns off the current starvation sources and puts the drivers into high-impedance output
9-0 —	- Reserved.

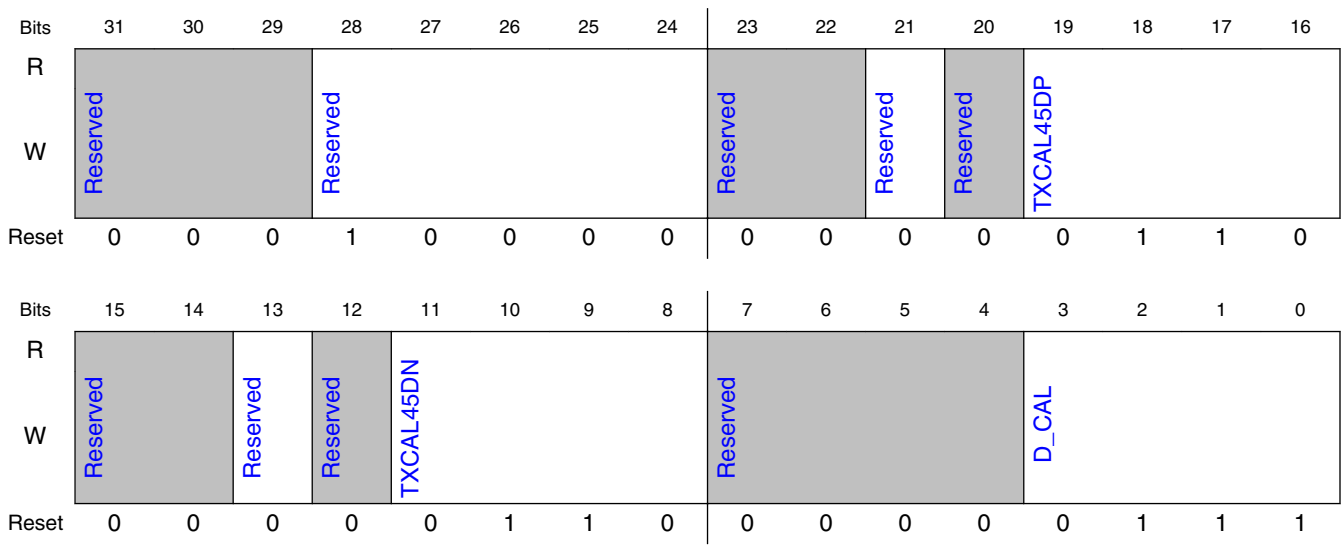
63.8.1.3 USB PHY Transmitter Control Register (TX)

The USB PHY Transmitter Control Register allows adjustment of transmit parametric values.

63.8.1.3.1 Offset

Register	Offset	Description
TX	10h	USB PHY Transmitter Control Register
TX_SET	14h	Writing a 1 to a bit in this register sets the corresponding bit in TX
TX_CLR	18h	Writing a 1 to a bit in this register clears the corresponding bit in TX
TX_TOG	1Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TX

63.8.1.3.2 Diagram



63.8.1.3.3 Fields

Field	Description
31-29 —	- Reserved.
28-24 —	Reserved Reserved. This bit field must remain at value 0x0.
23-22 —	- Reserved.
21 —	- Reserved. This bit field must remain at value 1'b0.

Table continues on the next page...

Field	Description
20 —	- Reserved.
19-16 TXCAL45DP	TXCAL45DP Decode to trim the nominal 45Ω series termination resistance to the DP output pin. Maximum resistance = 0000. Resistance is centered by design at 0110.
15-14 —	- Reserved.
13 —	- Reserved. This bit field must remain at value 1'b0.
12 —	- Reserved.
11-8 TXCAL45DN	TXCAL45DN Decode to trim the nominal 45Ω series termination resistance to the DN output pin. Maximum resistance = 0000. Resistance is centered by design at 0110.
7-4 —	- Reserved. **Note:** This bit should remain clear.
3-0 D_CAL	D_CAL Decode to trim the nominal 17.78mA current source for the High Speed TX drivers on DP and DN. This current is directly proportional to the amplitude of the High Speed TX eye diagram. 0000 - Maximum current, approximately 19% above nominal. 0111 - Nominal 1111 - Minimum current, approximately 19% below nominal.

63.8.1.4 USB PHY Receiver Control Register (RX)

The USB PHY Receiver Control Register allows adjustment of receive parametric values.

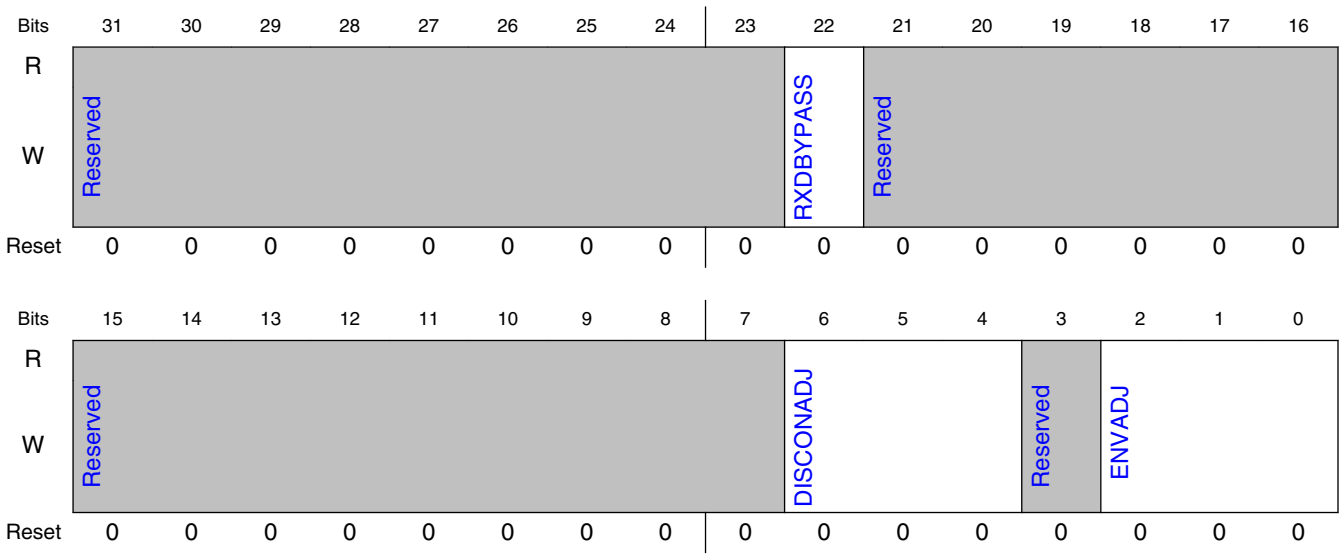
63.8.1.4.1 Offset

Register	Offset	Description
RX	20h	USB PHY Receiver Control Register
RX_SET	24h	Writing a 1 to a bit in this register sets the corresponding bit in RX
RX_CLR	28h	Writing a 1 to a bit in this register clears the corresponding bit in RX

Table continues on the next page...

Register	Offset	Description
RX_TOG	2Ch	Writing a 1 to a bit in this register toggles the corresponding bit in RX

63.8.1.4.2 Diagram



63.8.1.4.3 Fields

Field	Description
31-23 —	- Reserved.
22 RXDBYPASS	RXDBYPASS This test mode is intended for lab use only, replace FS differential receiver with DP single ended receiver. 0 - Normal operation. 1 - Use the output of the USB_DP single-ended receiver in place of the full-speed differential receiver
21-7 —	- Reserved.
6-4 DISCONADJ	DISCONADJ The DISCONADJ field adjusts the trip point for the disconnect detector. 000 - Trip-Level Voltage is 0.56875 V 001 - Trip-Level Voltage is 0.55000 V 010 - Trip-Level Voltage is 0.58125 V 011 - Trip-Level Voltage is 0.60000 V 1xx - Reserved

Table continues on the next page...

Field	Description
3 —	- Reserved.
2-0 ENVADJ	ENVADJ The ENVADJ field adjusts the trip point for the envelope detector. Values shown below are nominal DC settings to indicate effect of changing these bits. AC values measured during compliance testing will be somewhat higher. 000 - Trip-Level Voltage is 0.1000 V 001 - Trip-Level Voltage is 0.1125 V 010 - Trip-Level Voltage is 0.1250 V 011 - Trip-Level Voltage is 0.0875 V 1xx - Reserved

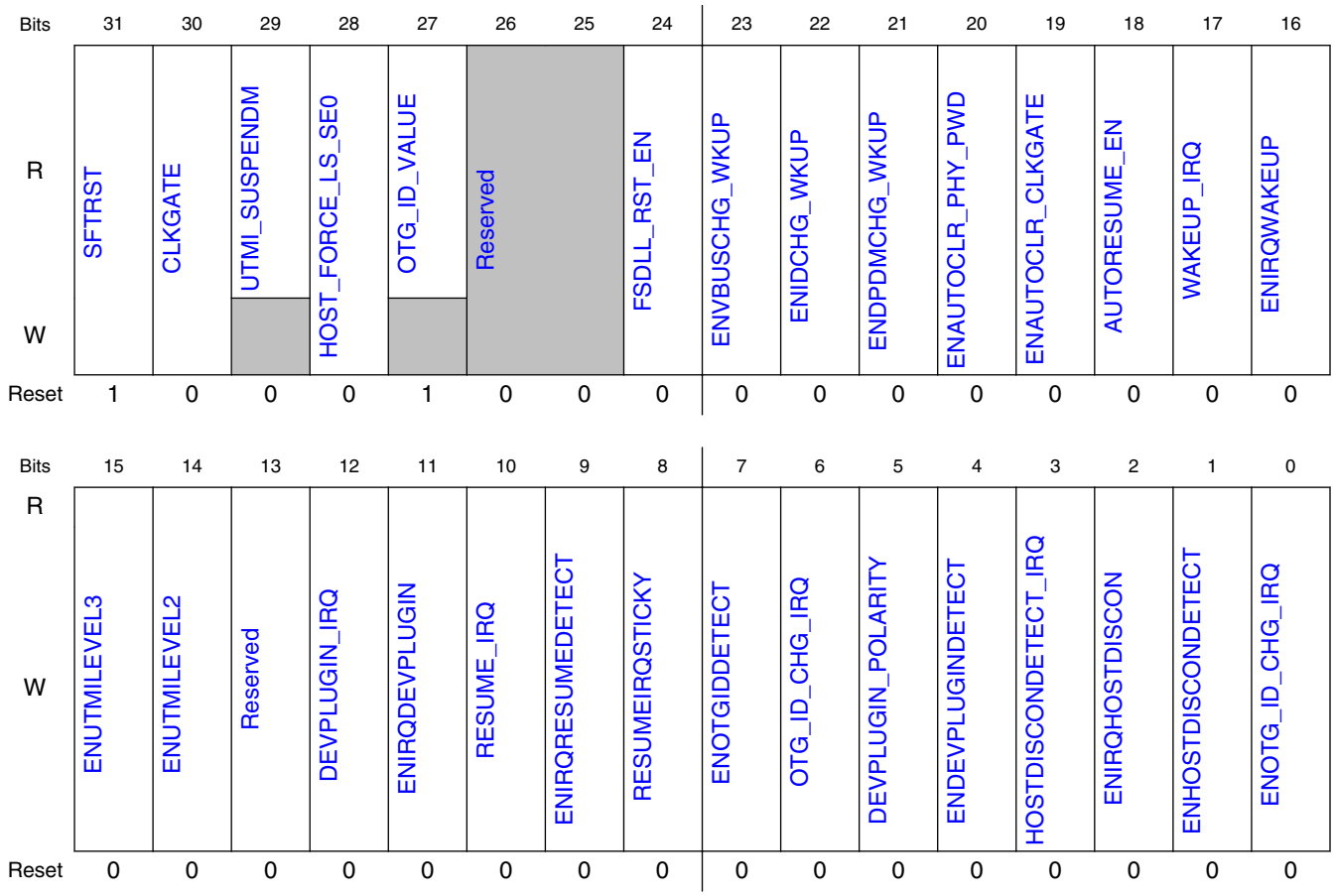
63.8.1.5 USB PHY General Control Register (CTRL)

The USB PHY General Control Register handles OTG and Host controls. This register also includes interrupt enables and connectivity detect enables and results.

63.8.1.5.1 Offset

Register	Offset	Description
CTRL	30h	USB PHY General Control Register
CTRL_SET	34h	Writing a 1 to a bit in this register sets the corresponding bit in CTRL
CTRL_CLR	38h	Writing a 1 to a bit in this register clears the corresponding bit in CTRL
CTRL_TOG	3Ch	Writing a 1 to a bit in this register toggles the corresponding bit in CTRL

63.8.1.5.2 Diagram



63.8.1.5.3 Fields

Field	Description
31 SFTRST	SFTRST Writing a 1 to this bit will soft-reset the PWD, TX, RX, and CTRL Registers. Set to 0 to release the PHY from reset.
30 CLKGATE	CLKGATE Gate UTMI Clocks. Clear to 0 to run clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated. **NOTE:** This bit can be auto-cleared if there is any wakeup event when USB is suspended while CTRL[ENAUTOCLR_CLKGATE] bit is enabled.
29 UTMI_SUSPENDM	UTMI_SUSPENDM Used by the PHY to indicate a powered-down state. If all the power-down bits in the PWD are enabled, UTMI_SUSPENDM will be 0, otherwise 1 when USB controller entering into Suspend mode. UTMI_SUSPENDM is negative logic, as required by the UTMI specification.
28 HOST_FORCE_LS_SE0	HOST_FORCE_LS_SE0

Table continues on the next page...

Field	Description
	Forces the next FS packet that is transmitted to have a EOP with low-speed timing. This bit is used in host mode for the resume sequence. After the packet is transferred, this bit is cleared. The design can use this function to force the LS SE0 or use the CTRL_UTMI_SUSPENDM to trigger this event when leaving suspend. This bit is used in conjunction with DEBUG_HOST_RESUME_DEBUG.
27 OTG_ID_VALUE	OTG_ID_VALUE Indicates the results of USB_ID pin while monitoring the cable plugged into the Micro- or Mini-AB receptacle. False (0) is when ID resistance is less than Ra_Plug_ID, indicating host (A) side. True (1) is when ID resistance is greater than Rb_Plug_ID, indicating device (B) side. Similar to the function of the OTGID_STATUS bit in the STATUS Register, but OTG_ID_VALUE has debounce and system clock synchronization logic to filter the glitches on the USB ID pad.
26-25 —	- Reserved.
24 FSDLL_RST_EN	FSDLL_RST_EN Enables the feature to reset the FSDLL lock detection logic at the end of each TX packet.
23 ENVBUSCHG_WKUP	ENVBUSCHG_WKUP Enables the feature to wakeup USB if VBUS is toggled when USB is suspended.
22 ENIDCHG_WKUP	ENIDCHG_WKUP Enables the feature to wakeup USB if ID is toggled when USB is suspended.
21 ENDPDMCHG_WKUP	ENDPDMCHG_WKUP Enables the feature to wakeup USB if DP/DN is toggled when USB is suspended.
20 ENAUTOCLR_PHY_PWD	ENAUTOCLR_PHY_PWD Enables the feature to auto-clear the PWD register bits in PWD if there is wakeup event while USB is suspended. This should be enabled if needed to support auto wakeup without software interaction.
19 ENAUTOCLR_CLKGATE	ENAUTOCLR_CLKGATE Enables the feature to auto-clear the CLKGATE bit if there is wakeup event while USB is suspended. This should be enabled if needed to support auto wakeup without software interaction.
18 AUTORESUME_EN	AUTORESUME_EN Enable the auto resume feature, when set, HW will use 32KHz clock to send Resume to respond to the device remote wakeup(for host mode only). It's useful when PLL is off and reference clock is also powered down.
17 WAKEUP_IRQ	WAKEUP_IRQ Indicates that there is a wakeup event. Reset this bit by writing a 1 to the clear address space and not by a general write.
16 ENIRQWAKEUP	ENIRQWAKEUP Enables interrupt for the wakeup events.
15 ENUTMILEVEL3	ENUTMILEVEL3 Enables UTMI+ Level 3 operation for the USB HS PHY. This should be enabled if an Embedded Host use case needs to support an external FS Hub with a LS device connected.
14	ENUTMILEVEL2

Table continues on the next page...

Field	Description
ENUTMILEVEL2	Enables UTMI+ Level 2 operation for the USB HS PHY. This should be enabled if an Embedded Host use case needs to support a LS device.
13 — 	- Reserved This bit field must remain at value 1'b0.
12 DEVPLUGIN_IRQ	DEVPLUGIN_IRQ Indicates that the device is connected. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
11 ENIRQDEVPLUGIN	ENIRQDEVPLUGIN Enables interrupt for the detection of connectivity to the USB line.
10 RESUME_IRQ	RESUME_IRQ Indicates that the host is sending a wake-up after suspend. This bit is also set on a reset during suspend. Use this bit to wake up from suspend for either the resume or the reset case. Reset this bit by writing a 1 to the clear address space and not by a general write.
9 ENIRQRESUMEDETECT	ENIRQRESUMEDETECT Enables interrupt for detection of a non-J state on the USB line. This should only be enabled after the device has entered suspend mode.
8 RESUMEIRQSTICKY	RESUMEIRQSTICKY Set to 1 will make RESUME_IRQ bit a sticky bit until software clear it. Set to 0, RESUME_IRQ only set during the wake-up period.
7 ENOTGIDDETECT	ENOTGIDDETECT Enables circuit to detect resistance of MiniAB ID pin.
6 OTG_ID_CHG_IRQ	OTG_ID_CHG_IRQ OTG ID change interrupt. Indicates the value of ID pin changed.
5 DEVPLUGIN_POLARITY	DEVPLUGIN_POLARITY For device mode, if this bit is cleared to 0, then it trips the interrupt if the device is plugged in. If set to 1, then it trips the interrupt if the device is unplugged.
4 ENDEVPLUGINDETECT	Enables non-standard resistive plugged-in detection This bit field controls connection of nominal 200kΩ resistors to both the DP and DN pins as one method of detecting when a USB cable is attached in device mode. This bit field must remain at a value of 1'b0 for normal USB data communication, or when using the USBHSDCD module for battery charger detection per the *USB Battery Charger Specification Revision 1.2* or any other detection mechanism for USB cable plugin. The results of this detection method are reported in STATUS[DEVPLUGIN_STATUS]. 0 - Disables 200kΩ pullup resistors on DP and DN pins 1 - Enables 200kΩ pullup resistors on DP and DN pins
3 HOSTDISCONDETECT_IRQ	HOSTDISCONDETECT_IRQ Indicates that the device has disconnected in High-Speed mode. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
2	ENIRQHOSTDISCON

Table continues on the next page...

Field	Description
ENIRQHOSTDISCON	Enables interrupt for detection of disconnection to Device when in high-speed host mode. This should be enabled after ENDEVPLUGINDETECT is enabled.
1 ENHOSTDISCONDETECT	ENHOSTDISCONDETECT For host mode, enables high-speed disconnect detector. This signal allows the override of enabling the detection that is normally done in the UTMI controller. The UTMI controller enables this circuit whenever the host sends a start-of-frame packet. Software shall set this bit when a high-speed device is connected, which is suggested during bus reset. Software shall ensure this bit is not set at the end of Resume, otherwise a wrong disconnect status may be detected.
0 ENOTG_ID_CHG_IRQ	ENOTG_ID_CHG_IRQ Enables the interrupt for ID change.

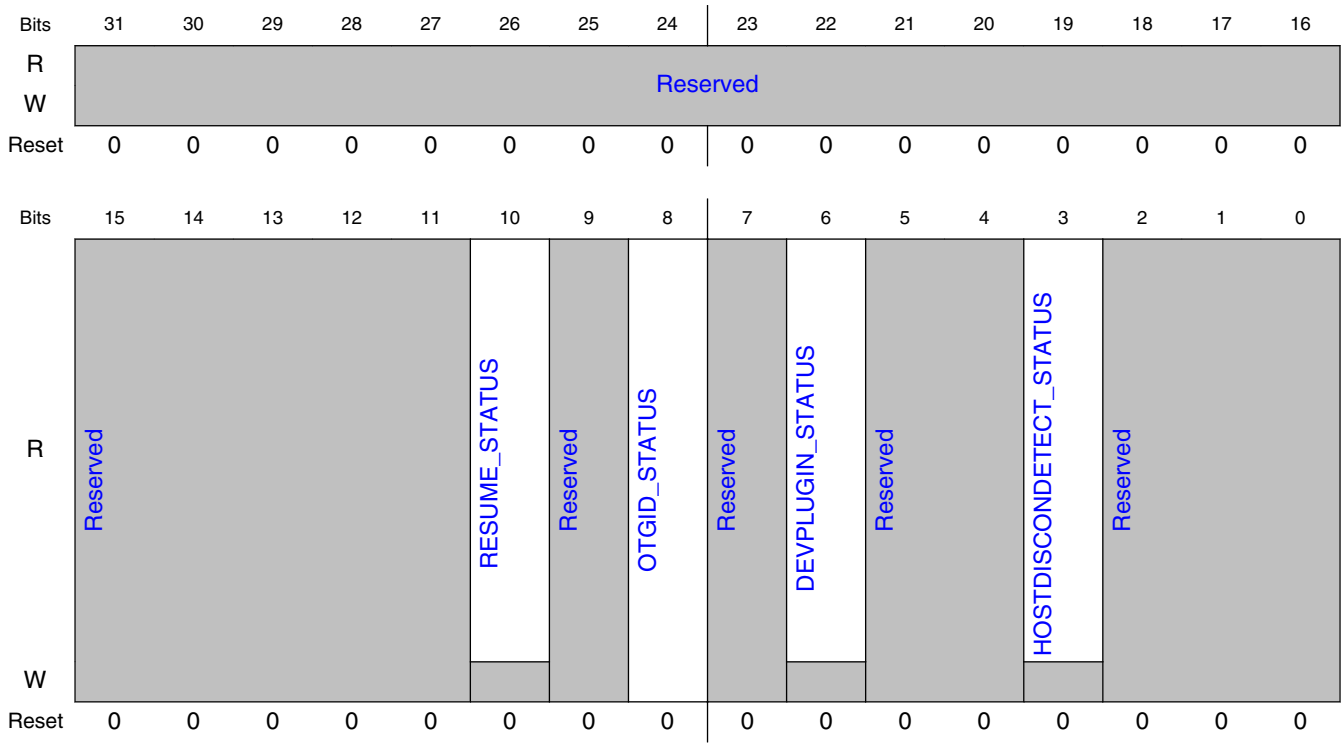
63.8.1.6 USB PHY Status Register (STATUS)

The USB PHY Status Register holds results of IRQ and other detects.

63.8.1.6.1 Offset

Register	Offset
STATUS	40h

63.8.1.6.2 Diagram



63.8.1.6.3 Fields

Field	Description
31-11 —	- Reserved.
10 RESUME_STATUS	RESUME_STATUS Indicates that the host is sending a wake-up after Suspend and has triggered an interrupt.
9 —	- Reserved.
8 OTGID_STATUS	OTGID_STATUS Indicates the results of USB_ID pin on the USB cable plugged into the local Micro- or Mini-AB receptacle. False (0) is when ID resistance to ground is less than Ra_Plug_ID, indicating host (A) side. True (1) is when ID resistance is greater than Rb_Plug_ID, indicating device (B) side.
7 —	- Reserved.
6 DEVPLUGIN_STATUS	Status indicator for non-standard resistive plugged-in detection Indicates that the device has been connected on the DP and DN lines using the non-standard resistive plugged-in detection method controlled by CTRL[ENDEVPLUGINDETECT].

Table continues on the next page...

Field	Description
	When a USB cable attached to a remote host is attached to the local device, the 15kΩ host pulldowns will override the high value resistors used in this detection method. 0 - No attachment to a USB host is detected 1 - Cable attachment to a USB host is detected
5-4 —	- Reserved.
3 HOSTDISCONDETECT_STATUS	HOSTDISCONDETECT_STATUS Indicates at the local host (downstream) port that the remote device has disconnected while in High-Speed mode. 0 - USB cable disconnect has not been detected at the local host 1 - USB cable disconnect has been detected at the local host
2-0 —	- Reserved.

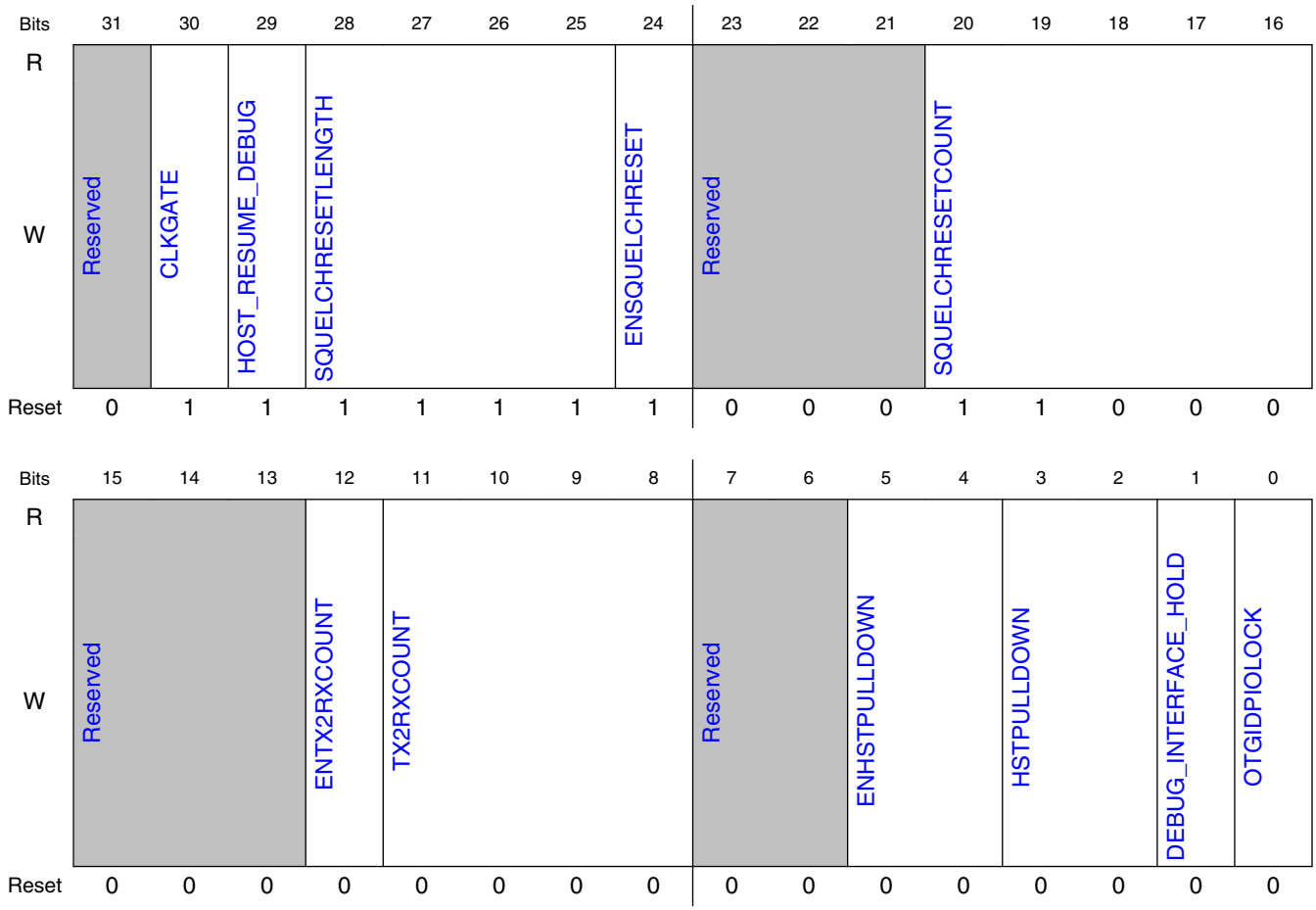
63.8.1.7 USB PHY Debug Register (DEBUG)

This register is used to debug the USB PHY.

63.8.1.7.1 Offset

Register	Offset	Description
DEBUG	50h	USB PHY Debug Register
DEBUG_SET	54h	Writing a 1 to a bit in this register sets the corresponding bit in DEBUG
DEBUG_CLR	58h	Writing a 1 to a bit in this register clears the corresponding bit in DEBUG
DEBUG_TOG	5Ch	Writing a 1 to a bit in this register toggles the corresponding bit in DEBUG

63.8.1.7.2 Diagram



63.8.1.7.3 Fields

Field	Description
31 —	- Reserved.
30 CLKGATE	CLKGATE Gate Test Clocks. Clear to 0 for running clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated.
29 HOST_RESUME_DEBUG	HOST_RESUME_DEBUG Choose to trigger the host resume SE0 with HOST_FORCE_LS_SE0 = 0 or UTMI_SUSPEND = 1.
28-25 SQUELCHRESETLENGTH	SQUELCHRESETLENGTH Duration of RESET in terms of the number of 480-MHz cycles.

Table continues on the next page...

Field	Description
24 ENSQUELCHRESET	ENSQUELCHRESET Set bit to allow squelch to reset high-speed receive.
23-21 —	- Reserved.
20-16 SQUELCHRESETCOUNT	SQUELCHRESETCOUNT Delay in between the detection of squelch to the reset of high-speed RX.
15-13 —	- Reserved.
12 ENTX2RXCOUNT	ENTX2RXCOUNT Set this bit to allow a countdown to transition in between TX and RX.
11-8 TX2RXCOUNT	TX2RXCOUNT Delay in between the end of transmit to the beginning of receive. This is a Johnson count value and thus will count to 8.
7-6 —	- Reserved.
5-4 ENHSTPULLDOWN	ENHSTPULLDOWN This bit field selects host pulldown overdrive mode. Set bit 5 to value 1'b1 to override the control of the DP 15kΩ pulldown. Set bit 4 to value 1'b1 to override the control of the DN 15Ω pulldown. Clear both bits to value 2'b00 to disable the host pulldown overdrive mode. When in host pulldown overdrive mode, the connection of the individual pulldown resistors is further controlled by the DEBUG[HSTPULLDOWN] bit field.
3-2 HSTPULLDOWN	HSTPULLDOWN This bit field selects whether to connect pulldown resistors on the DP/DN pins if the corresponding pulldown overdrive mode is enabled through DEBUG[ENHSTPULLDOWN} Set bit 3 to value 1'b1 to connect the 15Ω pulldown on DP line. Set bit 2 to value 1'b1 to connect the 15Ω pulldown on the DN line. Clear both bits to value 2'b00 to disconnect the pulldowns in override mode.
1 DEBUG_INTERFACE_HOLD	DEBUG_INTERFACE_HOLD Use holding registers to assist in timing for external UTMI interface.
0 OTGIDPIOLOCK	OTGIDPIOLOCK Once OTG ID from STATUS_OTGID_STATUS, use this to hold the value. This is to save power for the comparators that are used to determine the ID status.

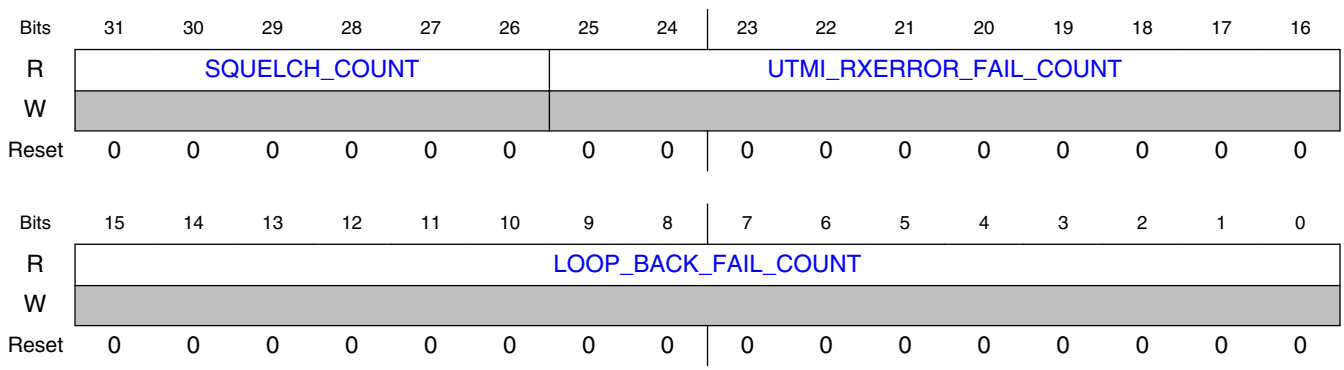
63.8.1.8 UTMI Debug Status Register 0 (DEBUG0_STATUS)

The UTMI Debug Status Register 0 holds multiple views for counters and status of state machines. This is used in conjunction with the DEBUG1[DBG_ADDRESS] field to choose which function to view. The default is described in the bit fields below and is used to count errors.

63.8.1.8.1 Offset

Register	Offset
DEBUG0_STATUS	60h

63.8.1.8.2 Diagram



63.8.1.8.3 Fields

Field	Description
31-26 SQUELCH_COUNT	SQUELCH_COUNT Running count of the squelch reset instead of normal end for HS RX.
25-16 UTMI_RXERROR_FAIL_COUNT	UTMI_RXERROR_FAIL_COUNT Running count of the UTMI_RXERROR.
15-0 LOOP_BACK_FAIL_COUNT	LOOP_BACK_FAIL_COUNT Running count of the failed pseudo-random generator loopback. Each time entering testmode, counter goes to 900D and will count up for every detected packet failure in digital/analog loopback tests.

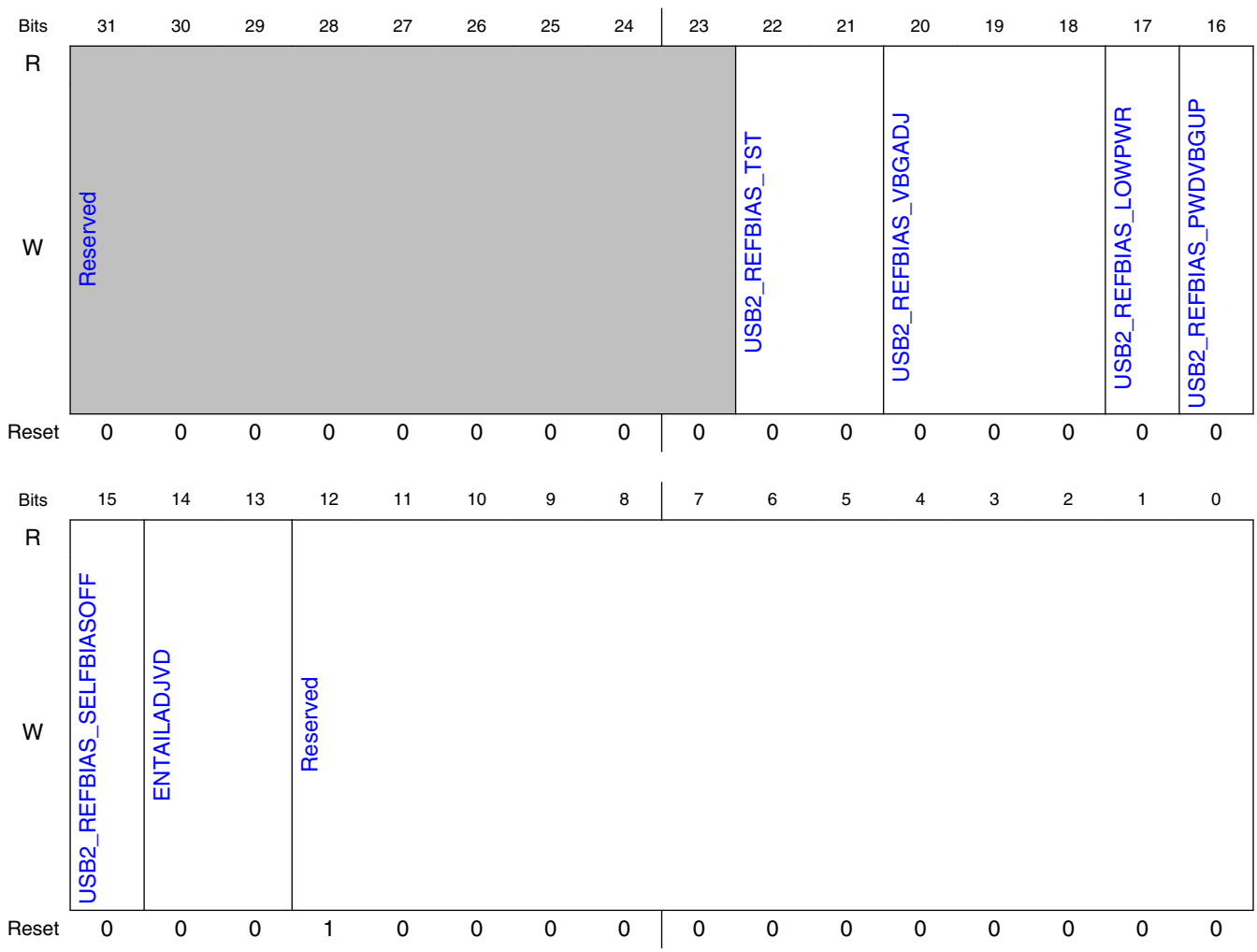
63.8.1.9 UTMI Debug Status Register 1 (DEBUG1)

Chooses the muxing of the debug register to be shown in USBPHYx_DEBUG0_STATUS.

63.8.1.9.1 Offset

Register	Offset	Description
DEBUG1	70h	UTMI Debug Status Register 1
DEBUG1_SET	74h	Writing a 1 to a bit in this register sets the corresponding bit in DEBUG1
DEBUG1_CLR	78h	Writing a 1 to a bit in this register clears the corresponding bit in DEBUG1
DEBUG1_TOG	7Ch	Writing a 1 to a bit in this register toggles the corresponding bit in DEBUG1

63.8.1.9.2 Diagram



63.8.1.9.3 Fields

Field	Description
31-23 —	- Reserved.
22-21 USB2_REFBIAS_TST	Bias current control for usb2_phy
20-18 USB2_REFBIAS_VBGADJ	Adjustment bits on bandgap
17 USB2_REFBIAS_LOWPWR	to be added
16 USB2_REFBIAS_PWDVBGUP	Powers down the bandgap detect logic, will affect vbgup on misc1 register.
15 USB2_REFBIAS_SELFBIASOFF	Set to 1 to disable self bias, 100 us after power up refbias(usb2_refbias_pwd).This can reduce noise on power. It has to be cleared before clear refbias_pwd, that is, must use self bias before power up refbias, otherwise no bias.
14-13 ENTAILADJVD	ENTAILADJVD Delay increment of the rise of squelch. This bit field only has effect if it is enabled by the TRIM_OVERRIDE_EN[TRIM_ENV_TAIL_ADJ_VD_OVERRIDE] bit. 00 - Delay is nominal 01 - Delay is +20% 10 - Delay is -20% 11 - Delay is -40%
12-0 —	- Reserved. **Note:** This bit should remain clear.

63.8.1.10 UTMI RTL Version (VERSION)

Fields for RTL Version.

63.8.1.10.1 Offset

Register	Offset
VERSION	80h

63.8.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STEP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

63.8.1.10.3 Fields

Field	Description
31-24	MAJOR
MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23-16	MINOR
MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
15-0	STEP
STEP	Fixed read-only value reflecting the stepping of the RTL version.

63.8.1.11 USB PHY PLL Control/Status Register (PLL_SIC)

This register configures the 480MHz USB PHY PLL.

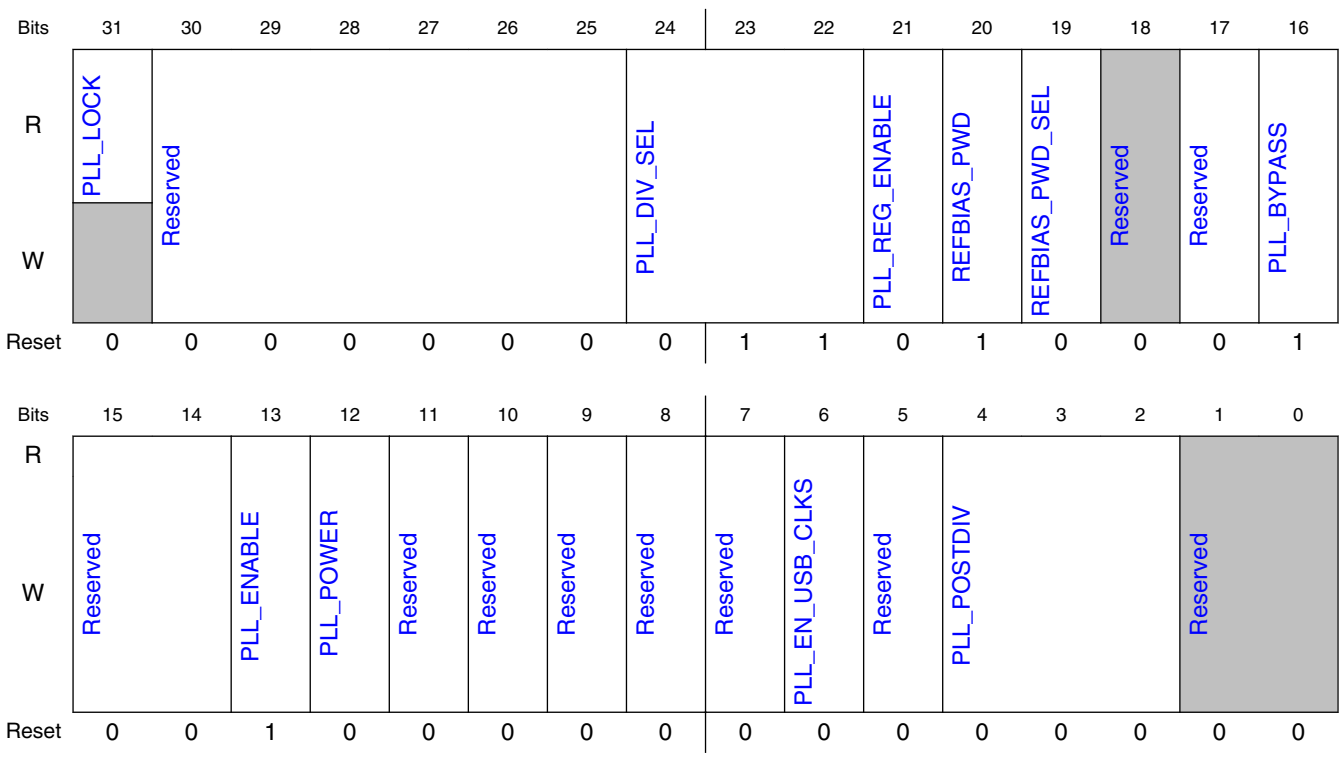
63.8.1.11.1 Offset

Register	Offset	Description
PLL_SIC	A0h	USB PHY PLL Control/Status Register

Table continues on the next page...

Register	Offset	Description
PLL_SIC_SET	A4h	Writing a 1 to a bit in this register sets the corresponding bit in PLL_SIC
PLL_SIC_CLR	A8h	Writing a 1 to a bit in this register clears the corresponding bit in PLL_SIC
PLL_SIC_TOG	ACh	Writing a 1 to a bit in this register toggles the corresponding bit in PLL_SIC

63.8.1.11.2 Diagram



63.8.1.11.3 Fields

Field	Description
31 PLL_LOCK	PLL_LOCK USB PLL lock status indicator 0 - PLL is not currently locked 1 - PLL is currently locked
30-25 —	Reserved
24-22	PLL_DIV_SEL

Table continues on the next page...

Field	Description
PLL_DIV_SEL	This field controls the USB PLL feedback loop divider. Valid range for divider values: 13-240. Fout = Fin * div_select. The USB PLL is designed to produce a 480 MHz output clock. This bit field allows use of different frequency signals for the PLL reference clock input connected to the OSCCLK signal from the system oscillator. When override is enabled through TRIM_OVERRIDE_EN[0], the USB PLL will use this register value. 000 - Divide by 13 001 - Divide by 15 010 - Divide by 16 011 - Divide by 20 100 - Divide by 22 101 - Divide by 25 110 - Divide by 30 111 - Divide by 240
21 PLL_REG_ENABLE	PLL_REG_ENABLE This field controls the USB PLL regulator, set to enable the regulator. SW must set this bit 15 us before settiing PLL_POWER to avoid glitches on PLL output clock.
20 REFBIAS_PWD	Power down the reference bias This bit is only used when REFBIAS_PWD_SEL is set to 1.
19 REFBIAS_PWD_SEL	REFBIAS_PWD_SEL Reference bias power down select. 0 - Selects PLL_POWER to control the reference bias 1 - Selects REFBIAS_PWD to control the reference bias.
18 —	- Reserved
17 —	- Reserved. This bit field must remain at value 1'b0.
16 PLL_BYPASS	PLL_BYPASS Bypass the USB PLL.
15-14 —	Reserved This bit field has no effect for this module.
13 PLL_ENABLE	PLL_ENABLE Enable the clock output from the USB PLL. PLL output enable signal is also controlled by PLL power up control. Hardware will disable the PLL output before PLL power down, and enable the PLL output after PLL power up. The software only needs to set it when initializing the PLL.
12 PLL_POWER	PLL_POWER Power up the USB PLL. PLL power up is also controlled by hardware. Hardware will power down PLL when USB is suspended and not in use.
11 —	- Reserved. This bit field must remain at value 1'b0 for normal operation.
10	Reserved

Table continues on the next page...

Field	Description
—	Not for customer use. This bit field must remain at value 1'b0.
9 —	Reserved Not for customer use. This bit field must remain at value 1'b0.
8 —	Reserved Not for customer use. This bit field must remain at value 1'b0.
7 —	Reserved Not for customer use. This bit field must remain at value 1'b0.
6 PLL_EN_USB_CLKS	PLL_EN_USB_CLKS Enable the USB clock output from the USB PHY PLL.
5 —	Reserved Not for customer use. This bit field must remain at value 1'b0.
4-2 PLL_POSTDIV	PLL_POSTDIV pll_postdiv to analog.
1-0 —	- Reserved

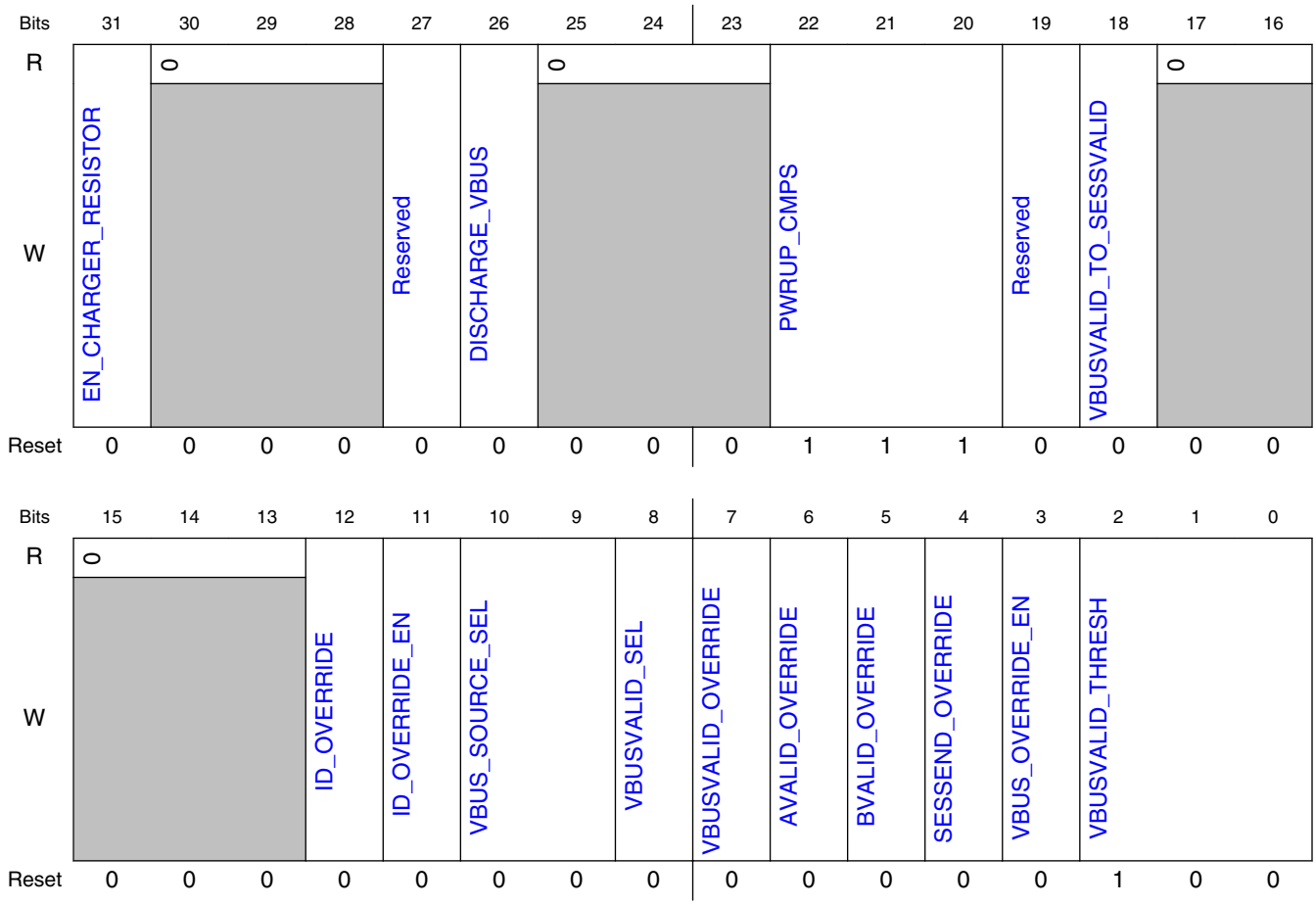
63.8.1.12 USB PHY VBUS Detect Control Register (USB1_VBUS_DETECT)

This register defines controls for USB VBUS detect and some additional out of band signaling functions.

63.8.1.12.1 Offset

Register	Offset	Description
USB1_VBUS_DETECT	C0h	USB PHY VBUS Detect Control Register
USB1_VBUS_DETECT_SET	C4h	Writing a 1 to a bit in this register sets the corresponding bit in USB1_VBUS_DETECT
USB1_VBUS_DETECT_CLR	C8h	Writing a 1 to a bit in this register clears the corresponding bit in USB1_VBUS_DETECT
USB1_VBUS_DETECT_TOG	CCh	Writing a 1 to a bit in this register toggles the corresponding bit in USB1_VBUS_DETECT

63.8.1.12.2 Diagram



63.8.1.12.3 Fields

Field	Description
31 EN_CHARGER_RESISTOR	Enables resistors used for an older method of resistive battery charger detection This bit field enables a nominal 125kΩ pullup on DP and a nominal 375kΩ pulldown on DN. This allows resistive battery charger detection per the *USB Battery Charging Specification Revision 1.0* which is now obsolete. Detection of DP and DN pin states using this method is done with the CHRG_DET_STAT[3:2] bit fields. This bit field must remain at a value of 1'b0 for normal USB data communication, or when using the USBHSDCD module for battery charger detection per the *USB Battery Charger Specification Revision 1.2* or any other detection mechanism for USB cable plugin. 0 - Disable resistive charger detection resistors on DP and DP 1 - Enable resistive charger detection resistors on DP and DP
30-28 —	Reserved

Table continues on the next page...

Field	Description
27 —	Reserved This bit field has no effect for this module.
26 DISCHARGE_VBUS	Controls VBUS discharge resistor This bit field controls a nominal 22kΩ resistor between the USB1_VBUS pin and ground. It can be used to accelerate the fall of the VBUS signal at the end of a session. 0 - VBUS discharge resistor is disabled (Default) 1 - VBUS discharge resistor is enabled
25-23 —	Reserved
22-20 PWRUP_CMPS	Enables the VBUS_VALID comparator Powers up the comparator used for the VBUS_VALID detector. This bit field can be reset to value 1'b0 to save power if the internal VBUS_VALID comparator is not used. Default value is 3'h7 000 - Powers down the VBUS_VALID comparator 001 - Enables the SESS_VALID comparator (default) 010 - Enables the 3Vdetect (default)
19 —	Reserved This bit field has no effect for this module.
18 VBUSVALID_TO_SESSVALID	Selects the comparator used for VBUS_VALID This bit field controls the comparator used to report the VBUS_VALID results in USB1_VBUS_DETECT[3] between the VBUS_VALID comparator and the Session Valid comparator. The VBUS_VALID comparator is the most accurate and has a programmable threshold set by USB_VBUS_DETECT[2:0]. The Session Valid comparator may be useful in systems using non-standard VBUS voltages. The mux selection in this bit field happens before any VBUS_VALID selection controlled by the USB1_VBUS_DETECT[10:8] bits. 0 - Use the VBUS_VALID comparator for VBUS_VALID results 1 - Use the Session End comparator for VBUS_VALID results. The Session End threshold is >0.8V and <4.0V.
17-13 —	Reserved
12 ID_OVERRIDE	TBA
11 ID_OVERRIDE_EN	TBA
10-9 VBUS_SOURCE_SEL	Selects the source of the VBUS_VALID signal reported to the USB controller This is one of the bit fields that selects the source of the VBUS_VALID signal reported to the USB controller. The VBUS_VALID source selections in this bit field only take effect if both USB1_VBUS_DETECT[8] and USB1_VBUS_DETECT[3] each have the value 1'b0. This bit field does not impact the VBUS_VALID value reported in USB1_VBUS_DET_STAT[3].

Table continues on the next page...

Field	Description
	00 - Use the VBUS_VALID comparator results for signal reported to the USB controller (Default) 01 - Use the Session Valid comparator results for signal reported to the USB controller 10 - Use the Session Valid comparator results for signal reported to the USB controller 11 - Reserved, do not use
8 VBUSVALID_SEL	Selects the source of the VBUS_VALID signal reported to the USB controller This is one of the bit fields that selects the source of the VBUS_VALID signal reported to the USB controller. The VBUS_VALID source selection in this bit field only takes effect if USB1_VBUS_DETECT[3] has the value 1'b0. This bit field does not impact the VBUS_VALID value reported in USB1_VBUS_DET_STAT[3]. 0 - Use the VBUS_VALID comparator results for signal reported to the USB controller (Default) 1 - Use the VBUS_VALID_3V detector results for signal reported to the USB controller
7 VBUSVALID_OVERRIDE	Override value for VBUS_VALID signal sent to USB controller The bit field provides the value for VBUS_VALID reported to the USB controller if the value of USB1_VBUS_DETECT[3] is set to 1'b1. The value of this bit field does not affect the value of USB1_VBUS_DET_STAT[3].
6 AVALID_OVERRIDE	Override value for A-Device Session Valid The bit field provides the value for USB1_VBUS_DET_STAT[2] if USB_VBUS_DETECT[3] is set to value 1'b1.
5 BVALID_OVERRIDE	Override value for B-Device Session Valid The bit field provides the value for USB1_VBUS_DET_STAT[1] if USB_VBUS_DETECT[3] is set to value 1'b1.
4 SESSEND_OVERRIDE	Override value for SESSEND The bit field provides the value for USB1_VBUS_DET_STAT[0] if USB_VBUS_DETECT[3] is set to value 1'b1.
3 VBUS_OVERRIDE_EN	VBUS detect signal override enable This bit field allows SW to override the results from the VBUS_VALID and Session Valid comparators using the values in USB1_VBUS_DETECT[7:4]. The VBUS_VALID, AVALID, BVALID, and SESSEND signals sent to the USB controller are each affected by these bit selections. The values reported for AVALID, BVALID, and SESSEND in USB1_VBUS_DET_STAT[2:0] are also affected but the value reported for VBUS_VALID in USB1_VBUS_DET_STAT[3] is not affected. This override method may be useful if VBUS detection is not done with the internal VBUS_VALID or Session End comparators. 0 - Use the results of the internal VBUS_VALID and Session Valid comparators for VBUS_VALID, AVALID, BVALID, and SESSEND (Default) 1 - Use the override values for VBUS_VALID, AVALID, BVALID, and SESSEND
2-0 VBUSVALID_THRESH	VBUSVALID_THRESH Sets the threshold for the VBUSVALID comparator. This comparator is the most accurate method to determine the presence of 5v, and includes hystersis to minimize the need for software debounce of the detection. This comparator has ~50mV of hystersis to prevent chattering at the comparator trip point. 000 - 4.0 V 001 - 4.1 V

Field	Description
	010 - 4.2 V 011 - 4.3 V 100 - 4.4 V (Default) 101 - 4.5 V 110 - 4.6 V 111 - 4.7 V

63.8.1.13 USB PHY VBUS Detector Status Register (USB1_VBUS_DET_STAT)

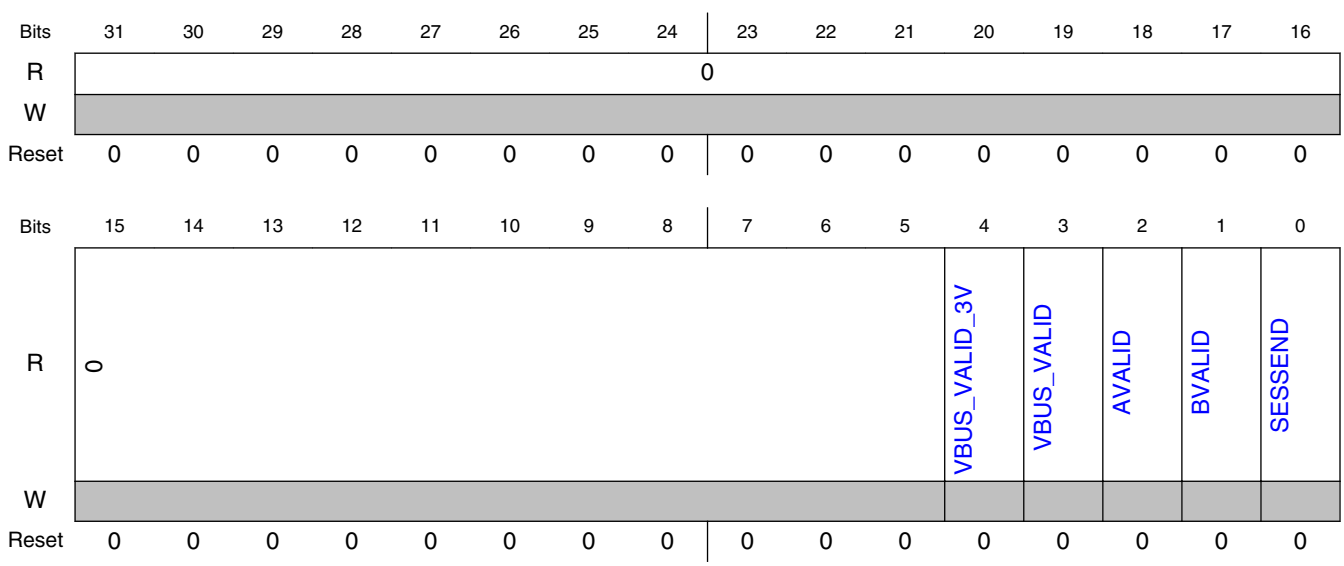
This register allows observation of status for USB VBUS detect functions.

The values reported in this register are synchronized by the apb_clk.

63.8.1.13.1 Offset

Register	Offset
USB1_VBUS_DET_STAT	D0h

63.8.1.13.2 Diagram



63.8.1.13.3 Fields

Field	Description
31-5 —	Reserved
4 VBUS_VALID_3V	VBUS_VALID_3V detector status The VBUS_VALID_3V detector has a lower threshold for the voltage on the USB1_VBUS pin than either the Session Valid or VBUS_VALID comparators. This signal may be useful for applications using a non-standard VBUS voltage. 0 - VBUS voltage is below VBUS_VALID_3V threshold 1 - VBUS voltage is above VBUS_VALID_3V threshold
3 VBUS_VALID	VBUS voltage status This bit field shows the result of VBUS_VALID detection for the USB1_VBUS pin. The VBUS_VALID comparator used is selected by USB1_VBUS_DETECT[18]. The VBUS_VALID source is not affected by the values of USB1_VBUS_DETECT[7, 3] and cannot be overwritten by SW. 0 - VBUS is below the comparator threshold 1 - VBUS is above the comparator threshold
2 AVALID	A-Device Session Valid status A-Device Session Valid status, determined by the Session Valid comparator. The default value of this bit is determined by the voltage on the USB1_VBUS pin. This bit can be overwritten by SW using the VBUS_DETECT[6, 3] bit fields. 0 - The VBUS voltage is below the Session Valid threshold 1 - The VBUS voltage is above the Session Valid threshold
1 BVALID	B-Device Session Valid status B-Device Session Valid status, determined by the Session Valid comparator. The default value of this bit is determined by the voltage on the USB1_VBUS pin. This bit can be overwritten by SW using the VBUS_DETECT[5, 3] bit fields. 0 - The VBUS voltage is below the Session Valid threshold 1 - The VBUS voltage is above the Session Valid threshold
0 SESSEND	Session End indicator Session End status, value inverted from Session Valid comparator. The default value of this bit is determined by the voltage on the USB1_VBUS pin. This bit can be overwritten by SW using the VBUS_DETECT[4, 3] bit fields. 0 - The VBUS voltage is above the Session Valid threshold 1 - The VBUS voltage is below the Session Valid threshold

63.8.1.14 USB PHY Charger Detect Control Register (USB1_CHRG_DETECT)

This register defines controls for USB Battery Charging detection functions.

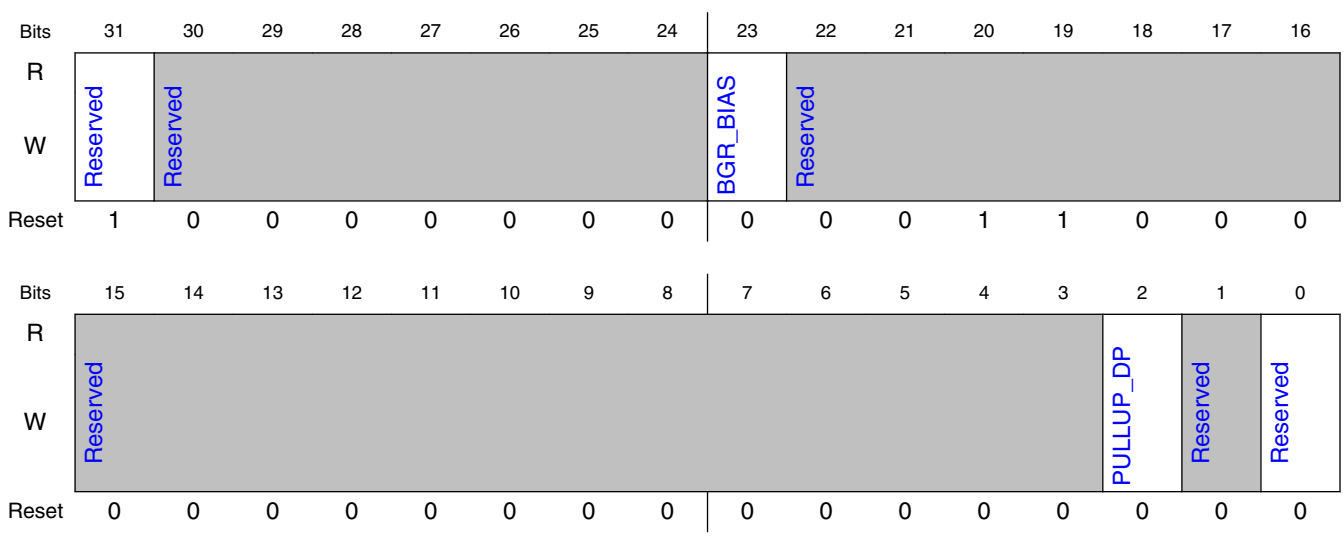
The control of the USB Battery Charging detection functions has been moved to the USBDCD module.

Bit fields are still implemented, but their values should not be changed except for characterization tasks.

63.8.1.14.1 Offset

Register	Offset	Description
USB1_CHRG_DETECT	E0h	USB PHY Charger Detect Control Register
USB1_CHRG_DETECT_SET	E4h	Writing a 1 to a bit in this register sets the corresponding bit in USB1_CHRG_DETECT
USB1_CHRG_DETECT_CLR	E8h	Writing a 1 to a bit in this register clears the corresponding bit in USB1_CHRG_DETECT
USB1_CHRG_DETECT_TOG	ECh	Writing a 1 to a bit in this register toggles the corresponding bit in USB1_CHRG_DETECT

63.8.1.14.2 Diagram



63.8.1.14.3 Fields

Field	Description
31 —	- Reserved The bit field must remain unchanged
30-24 —	- Reserved
23 BGR_BIAS	BGR_BIAS selects the bias method used for the voltage and current sources used for the analog circuits associated with USB Battery Charging detection in the USBHSDCD instantiation of the USBDCD module. The V_{DP_SRC} and V_{DN_SRC} bias is generated from the USB1_VBUS voltage independelt of the value of this bit field. 0 - Use local bias powered from USB1_VBUS for 10uA reference (Default) 1 - Use bandgap bias powered from VREGIN0/VREGIN1 for 10uA reference
22-3 —	- Reserved
2 PULLUP_DP	PULLUP_DP This bit is used to pull up DP, for digital charge detect.
1 —	- Reserved
0 —	- The bit field must remain at value 1'b0 to avoid interfering with USB Battery Charging Detection.

63.8.1.15 USB PHY Charger Detect Status Register (USB1_CHRG_DET_STAT)

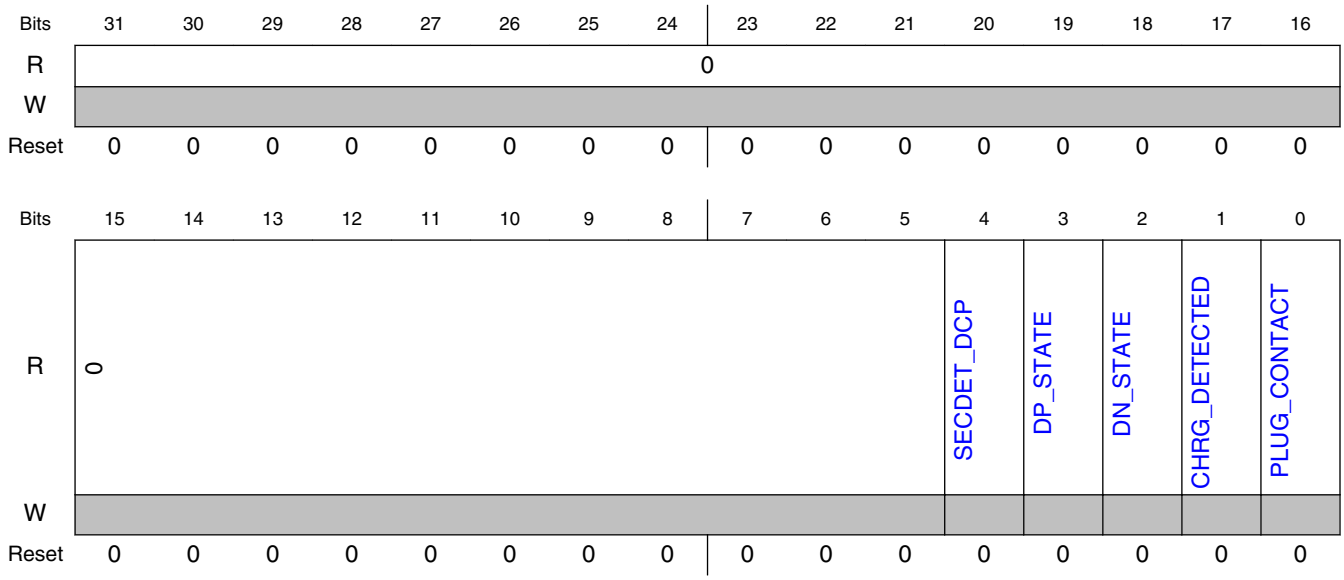
The control of the USB Battery Charging detection functions has been moved to the USBHSDCD instantiation of the USBDCD module for this product.

For standards-based charger detection purposes, the USBHSDCD registers should be used rather than this register. However, the status values in this register may be useful for debugging or in case other detection methods are used.

63.8.1.15.1 Offset

Register	Offset
USB1_CHRG_DET_STAT	F0h

63.8.1.15.2 Diagram



63.8.1.15.3 Fields

Field	Description
31-5 —	Reserved
4 SECDET_DCP	Battery Charging Secondary Detection phase output During the USB Battery Charging Secondary Detection phase using the USBHSDCD module, this bit field indicates which kind of Charging Port was detected. 0 - Charging Downstream Port (CDP) has been detected 1 - Downstream Charging Port (DCP) has been detected
3 DP_STATE	DP_STATE Single ended receiver output for the DP pin, from charger detection circuits. 0 - DP pin voltage is < 0.8V 1 - DP pin voltage is > 2.0V
2 DN_STATE	DN_STATE Single ended receiver output for the DN pin, from charger detection circuits. 0 - DN pin voltage is < 0.8V 1 - DN pin voltage is > 2.0V
1 CHRG_DETECTED	Battery Charging Primary Detection phase output During the USB Battery Charging Primary Detection phase using the USBHSDCD module, this bit field indicates whether a Standard Downstream Port or Charging Port was detected. 0 - Standard Downstream Port (SDP) has been detected

Table continues on the next page...

Field	Description
	1 - Charging Port has been detected
0 PLUG_CONTACT	Battery Charging Data Contact Detection phase output During the Data Contact Detection phase per the *USB Battery Charging Specification Revision 1.2* using the USBHSDCD module, this bit field indicates whether a USB cable has been attached between the remote host and the local device. 0 - No USB cable attachment has been detected 1 - A USB cable attachment between the device and host has been detected

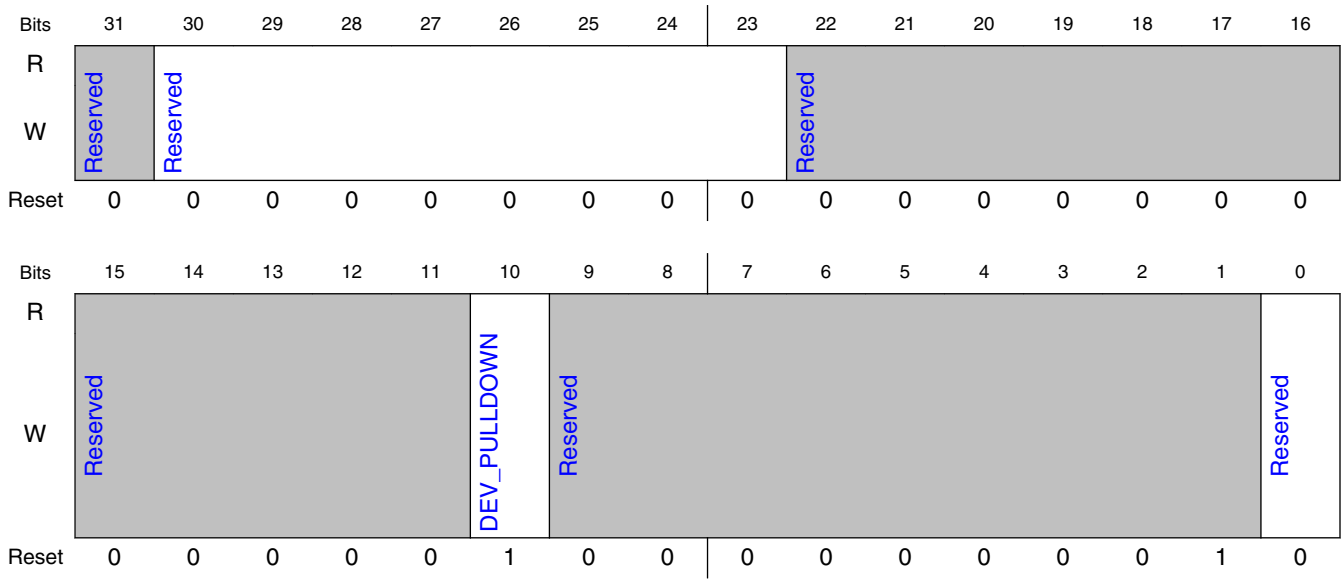
63.8.1.16 USB PHY Analog Control Register (ANACTRL)

The ANACTRL register has bit fields for adding pre-emphasis to the USB High-Speed TX output drivers.

63.8.1.16.1 Offset

Register	Offset	Description
ANACTRL	100h	USB PHY Analog Control Register
ANACTRL_SET	104h	Writing a 1 to a bit in this register sets the corresponding bit in ANACTRL
ANACTRL_CLR	108h	Writing a 1 to a bit in this register clears the corresponding bit in ANACTRL
ANACTRL_TOG	10Ch	Writing a 1 to a bit in this register toggles the corresponding bit in ANACTRL

63.8.1.16.2 Diagram



63.8.1.16.3 Fields

Field	Description
31 —	- Reserved
30-23 —	Reserved Not for customer use. For normal USB operation this bit field must remain cleared at value 8'h00.
22-11 —	- Reserved
10 DEV_PULLDOWN	DEV_PULLDOWN Setting this field to 1'b1 will enable the 15kΩ pulldown resistors on both DP and DN pins. This feature can be used in device mode while the USB cable is disconnected to keep the data pins at known values, avoiding unnecessary interrupts from the single ended receivers. This bit must be reset to 1'b0 during normal USB data communication in device mode, or while battery charger detection using the USBHSDCD module is used. 0 - The 15kΩ nominal pulldowns on the DP and DN pinsare disabled in device mode. 1 - The 15kΩ nominal pulldowns on the DP and DN pinsare enabled in device mode.
9-1 —	- Reserved
0 —	- Reserved The value of this bit field must remain at 1'b0 for normal operation.

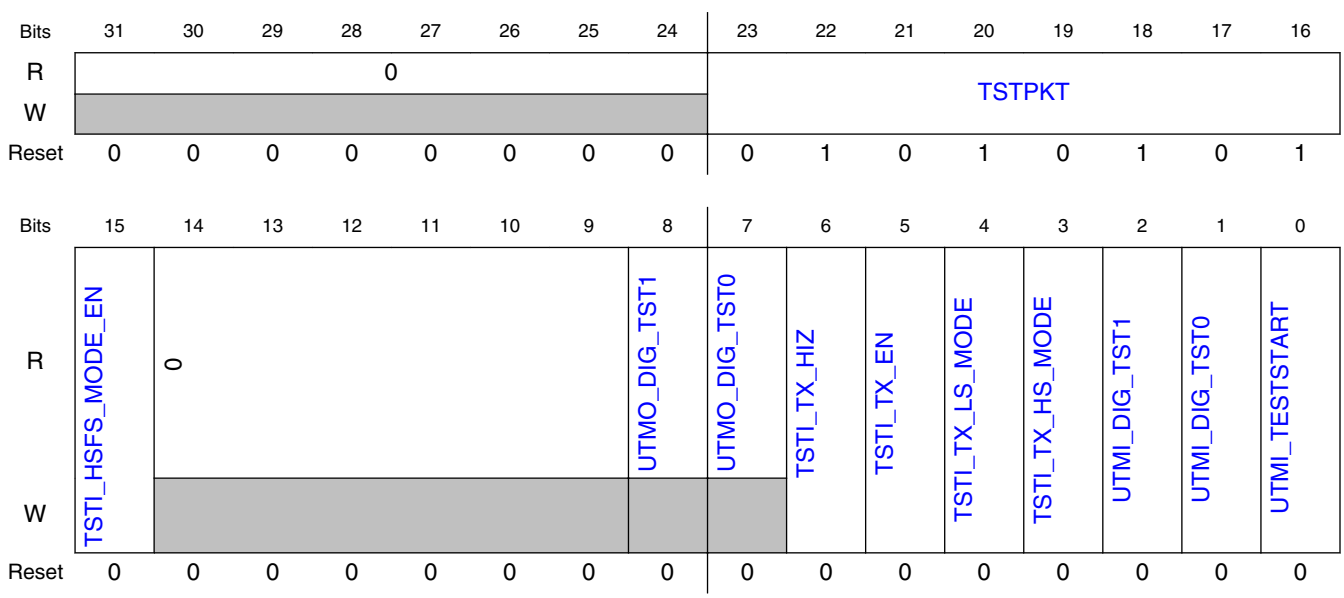
63.8.1.17 USB PHY Loopback Control/Status Register (USB1_LOOPBACK)

This register controls loopback testing of the USB PHY. Loopback mode is for test purposes only; it cannot be used during normal USB data communication.

63.8.1.17.1 Offset

Register	Offset	Description
USB1_LOOPBACK	110h	USB PHY Loopback Control/Status Register
USB1_LOOPBACK_SET	114h	Writing a 1 to a bit in this register sets the corresponding bit in USB1_LOOPBACK
USB1_LOOPBACK_CLR	118h	Writing a 1 to a bit in this register clears the corresponding bit in USB1_LOOPBACK
USB1_LOOPBACK_TOG	11Ch	Writing a 1 to a bit in this register toggles the corresponding bit in USB1_LOOPBACK

63.8.1.17.2 Diagram



63.8.1.17.3 Fields

Field	Description
31-24 —	Reserved
23-16 TSTPKT	TSTPKT Selects the packet data byte used for USB loopback testing in Pulse mode. Pulse mode is selected by setting USB1_LOOPBACK[2:1] to a value of 2'b01. Default value produces a data inversion at each unit interval.
15 TSTI_HSFS_MODE_EN	TSTI_HSFS_MODE_EN Setting this bit field to value 1'b1 will enable the loopback test to dynamically change the packet speed. It will send a number of High Speed packets determined by USB1_LOOPBACK_HSFSCNT[TSTI_HS_NUMBER], followed by a number of Full Speed packets determined by USB1_LOOPBACK_HSFSCNT[TSTI_FS_NUMBER].
14-9 —	Reserved
8 UTMO_DIG_TST1	UTMO_DIG_TST1 This read-only bit is a status bit for USB loopback test. Value = 1'b1 indicates a passing result. Test results are only valid while UTMI_TESTSTART is set to value 1'b1.
7 UTMO_DIG_TST0	UTMO_DIG_TST0 This read-only bit is a status bit for USB loopback test results. Value = 1'b0 indicates a passing result. Test results are only valid while UTMI_TESTSTART is set to value 1'b1.
6 TSTI_TX_HIZ	TSTI_TX_HIZ Sets TX Hi-Z for USB loopback test.
5 TSTI_TX_EN	TSTI_TX_EN Enable TX for USB loopback test.
4 TSTI_TX_LS_MODE	TSTI_TX_LS_MODE Set to value 1'b1 to choose LS for USB loopback testing, set to value 1'b0 to choose HS or FS mode which is defined by TSTI1_TX_HS.
3 TSTI_TX_HS_MODE	TSTI_TX_HS_MODE Select HS or FS mode for USB loopback testing. Set to value 1'b1 to choose HS for USB loopback testing, set to value 1'b0 to choose FS mode.
2 UTMI_DIG_TST1	UTMI_DIG_TST1 Mode control for USB loopback test. Setting this bit to a value of 1'b1 while UTMI_DIG_TST0 remains at a value of 1'b0 selects Psuedorandom modes for the loopback test.
1 UTMI_DIG_TST0	UTMI_DIG_TST0 Mode control for USB loopback test. Setting this bit to a value of 1'b1 while UTMI_DIG_TST1 remains at a value of 1'b0 selects Pulse mode for the loopback test.
0	UTMI_TESTSTART

Field	Description
UTMI_TESTSTART	This bit enables the USB loopback test.

63.8.1.18 USB PHY Loopback Packet Number Select Register (USB1_LOOPBACK_HSFSCNT)

63.8.1.18.1 Offset

Register	Offset	Description
USB1_LOOPBACK_HSFSCNT	120h	USB PHY Loopback Packet Number Select Register
USB1_LOOPBACK_HSFSCNT_SET	124h	Writing a 1 to a bit in this register sets the corresponding bit in USB1_LOOPBACK_HSFSCNT
USB1_LOOPBACK_HSFSCNT_CLR	128h	Writing a 1 to a bit in this register clears the corresponding bit in USB1_LOOPBACK_HSFSCNT
USB1_LOOPBACK_HSFSCNT_TOG	12Ch	Writing a 1 to a bit in this register toggles the corresponding bit in USB1_LOOPBACK_HSFSCNT

63.8.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSTI_FS_NUMBER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSTI_HS_NUMBER															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

63.8.1.18.3 Fields

Field	Description
31-16	TSTI_FS_NUMBER Full speed packet number, used when USB1_LOOPBACK[TSTI_HSFS_MODE_EN] is set to value 1'b1.

Table continues on the next page...

Field	Description
TSTI_FS_NUMBER	
15-0 TSTI_HS_NUMBER	TSTI_HS_NUMBER High speed packet number, used when USB1_LOOPBACK[TSTI_HSFS_MODE_EN] is set to value 1'b1.

63.8.1.19 USB PHY Trim Override Enable Register (TRIM_OVERRIDE_EN)

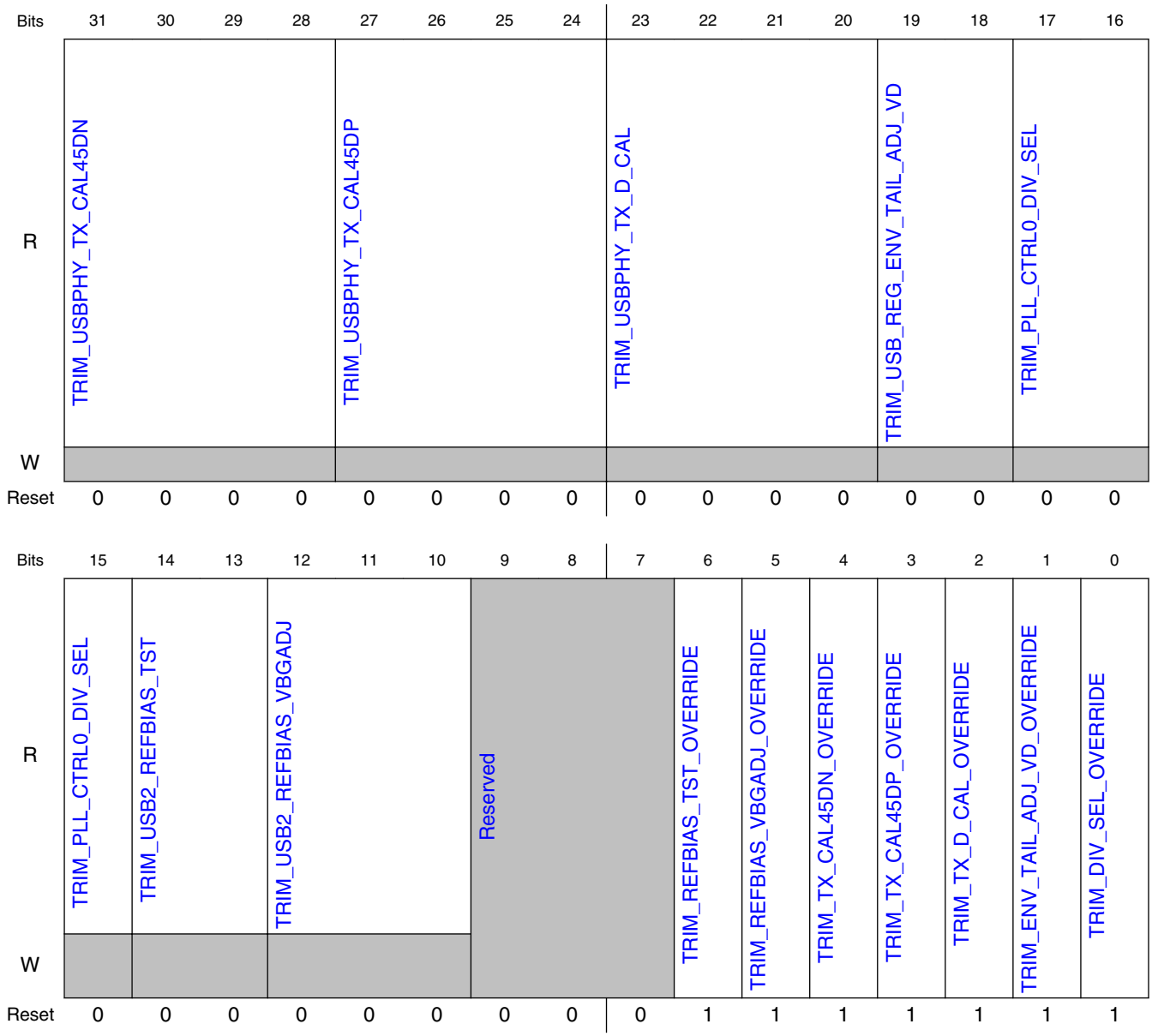
The bit fields in this register allow observation of the default IFR settings of Phy parameters for TX_CAL45DN, TX_CAL45DP, TX_D_CAL, ENV_TAIL_ADJ, and PLL_DIV_SEL.

Additional bit fields also determine whether those values can be overridden by settings in the TX, DEBUG1, and PLL_SIC registers.

63.8.1.19.1 Offset

Register	Offset	Description
TRIM_OVERRIDE_EN	130h	USB PHY Trim Override Enable Register
TRIM_OVERRIDE_EN_SET	134h	Writing a 1 to a bit in this register sets the corresponding bit in TRIM_OVERRIDE_EN
TRIM_OVERRIDE_EN_CLR	138h	Writing a 1 to a bit in this register clears the corresponding bit in TRIM_OVERRIDE_EN
TRIM_OVERRIDE_EN_TOG	13Ch	Writing a 1 to a bit in this register toggles the corresponding bit in TRIM_OVERRIDE_EN

63.8.1.19.2 Diagram



63.8.1.19.3 Fields

Field	Description
31-28 TRIM_USBPHY_TX_CAL45DN	TRIM_USBPHY_TX_CAL45DN IFR value of TX_CAL45DN.
27-24 TRIM_USBPHY_TX_CAL45DP	TRIM_USBPHY_TX_CAL45DP IFR value of TX_CAL45DP.

Table continues on the next page...

Field	Description
23-20 TRIM_USBPHY_TX_D_CAL	TRIM_USBPHY_TX_D_CAL IFR value of TX_D_CAL.
19-18 TRIM_USB_REG_ENV_TAIL_ADJ_VD	TRIM_USB_REG_ENV_TAIL_ADJ_VD IFR value of ENV_TAIL_ADJ.
17-15 TRIM_PLL_CTRL0_DIV_SEL	TRIM_PLL_CTRL0_DIV_SEL IFR value of PLL_DIV_SEL.
14-13 TRIM_USB2_REFBIAS_TST	TRIM_USB2_REFBIAS_TST Bias current control for usb2_phy
12-10 TRIM_USB2_REFBIAS_VBGADJ	TRIM_USB2_REFBIAS_VBGADJ Adjustment bits for bandgap
9-7 —	Reserved Reserved. This bit field must remain at value 1'b0.
6 TRIM_REFBIAS_TST_OVERRIDE	Override enable for bias current control When this field is set, the register value in DEBUG1[USB2_REFBIAS_TST] will be used.
5 TRIM_REFBIAS_VBGADJ_OVERRIDE	Override enable for bandgap adjustment. When this field is set, the register value in DEBUG1[USB2_REFBIAS_VBGADJ] will be used.
4 TRIM_TX_CAL45DN_OVERRIDE	TRIM_TX_CAL45DN_OVERRIDE Override enable for TX_CAL45DN, when set, the register value in TX[TXCAL45DN] will be used.
3 TRIM_TX_CAL45DP_OVERRIDE	TRIM_TX_CAL45DP_OVERRIDE Override enable for TX_CAL45DP, when set, the register value in TX[TXCAL45DP] will be used.
2 TRIM_TX_D_CAL_OVERRIDE	TRIM_TX_D_CAL_OVERRIDE Override enable for TX_D_CAL, when set, the register value in TX[D_CAL] will be used.
1 TRIM_ENV_TAIL_ADJ_VD_OVERRIDE	TRIM_ENV_TAIL_ADJ_VD_OVERRIDE Override enable for ENV_TAIL_ADJ, when set, the register value in DEBUG1[ENTAILADJVD] will be used.
0 TRIM_DIV_SEL_OVERRIDE	TRIM_DIV_SEL_OVERRIDE Override enable for PLL_DIV_SEL, when set, the register value in PLL_SIC[PLL_DIV_SEL] will be used.

Chapter 64
USB Device Charger Detection Module (USBDCD)

64.1 Chip-specific USBDCD information

Table 64-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

64.2 Overview

The USBDCD module works with the USB transceiver to detect whether the USB device is attached to a Charging Port, either a Dedicated Charging Port (DCP) or a Charging Downstream Port (CDP). System software coordinates the detection activites of the module and controls an off-chip integrated circuit that performs the battery charging.

There can be separate instantiations of the USBDCD module for each USB port. The instantiation for the High-Speed capable USB port is named USBHSDCD, so that it can be differentiated from the instantiation for Full-Speed/Low-Speed USB port which retains the name USBDCD. See the chip-specific information for the specific instantiation of your device.

The use of the term "USB transceiver" in this module documentation applies to the USB physical layer instance on the chip, whether it is a FS/LS only transceiver or a HS/FS/LS capable PHY.

64.2.1 Block diagram

The following figure is a high level block diagram of the module.

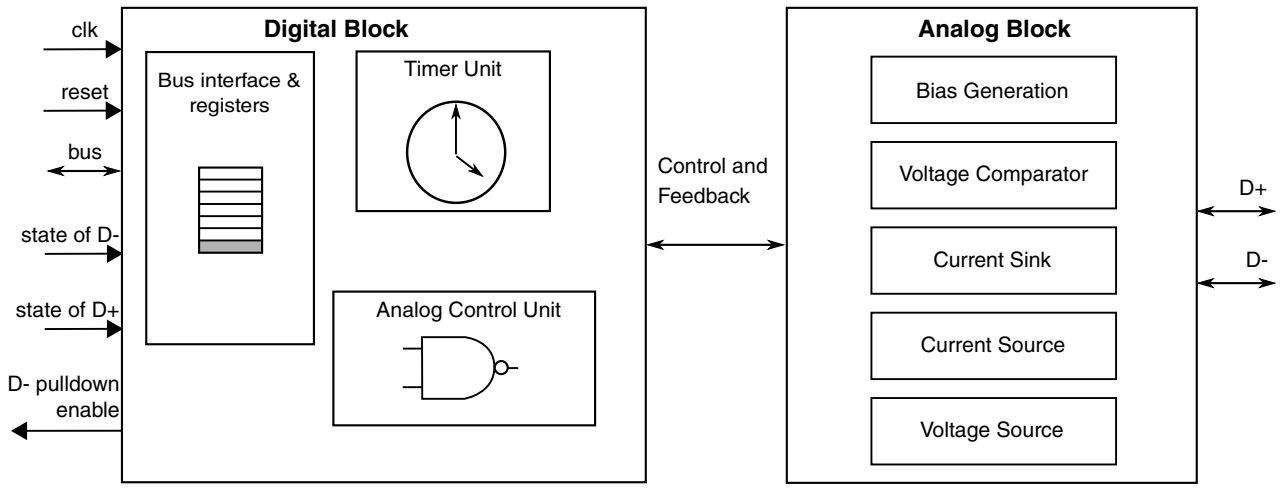


Figure 64-1. Block diagram

The USBDCD module consists of two main blocks:

- A digital block provides the programming interface (memory-mapped registers) and includes the timer unit and the analog control unit.
- An analog block provides the circuitry for the physical detection of the charger, including the bias generation, voltage source, current source, current sink, and voltage comparator circuitry. This block also contains necessary muxes between the source/sink circuits and the D- and D+ pins.

64.2.2 Features

The USBDCD module offers the following features:

- Compliant with the latest industry standard specifications: *USB Battery Charging Specification, Revisions 1.1 and 1.2*
- Programmable timing parameters default to values required by the industry standards:

- Having standard default values allows for easy configuration - simply set the clock frequency before enabling the module.
- Programmability allows the flexibility to meet future updates of the standards.

64.3 Functional description

The sequence of detecting the presence of charging port and type of charging port involves several hardware components, coordinated by system software. This collection of interacting hardware and software is called the USB Battery Charging Subsystem. The following figure shows the USBDCD module as a component of the subsystem. The following table describes the components.

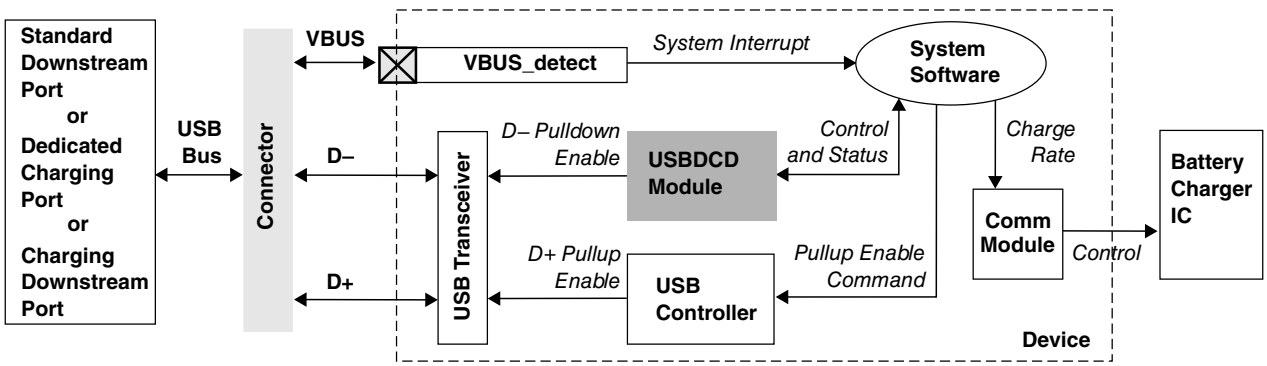


Figure 64-2. USB battery charging subsystem

Table 64-2. USB battery charger subsystem components

Component	Description
Battery Charger IC	The external battery charger IC regulates the charge rate to the rechargable battery. System software is responsible for communicating the appropriate charge rates. **Charger** / **Maximum current drawn** Standard Downstream Port (SDP) / up to 500 mA (I_{CFG_MAX}) Charging Downstream Port (CDP) / up to 1500 mA (I_{DEV_CHG}) Dedicated Charging Port (DCP) / up to 1500 mA (I_{DEV_CHG}) 1. If a USB SDP host has suspended the USB device, current drawn by the device has to follow the rules of the Dead Battery Provision in the *USB Battery Charging Specification, Rev. 1.2*.
Comm Module	A communications module on the device can be used to control the charge rate of the battery charger IC.
System software	Coordinates the detection activities of the subsystem.

Table continues on the next page...

Table 64-2. USB battery charger subsystem components (continued)

Component	Description
USB Controller	The D+ pullup enable control signal plays a role during the charger type detection phase. System software must issue a command to the USB controller to assert this signal. After this pullup is enabled, the device is considered to be connected to the USB bus. The host then attempts to enumerate it. **NOTE:** The USB controller must be used only for USB device applications when using the USBDCD module. For USB host applications, the USBDCD module must be disabled.
USB Transceiver	The USB transceiver contains the pullup resistor for the USB D+ signal and the pulldown resistors for the USB D+ and D– signals. The D+ pullup and the D– pulldown are both used during the charger detection sequence in BC1.1, but it is not used during charger detection in BC1.2. The USB transceiver also outputs the digital state of the D+ and D– signals from the USB bus. The pullup and pulldown enable signals are controlled by other modules during the charger detection sequence in BC1.1: The D+ pullup enable is output from the USB controller and is under software control. The USBDCD module controls the D– pulldown enable.
USBDCD Module	Detects whether the device has been plugged into either an SDP, a CDP, or a DCP.
VBUS_detect	This interrupt pin connected to the USB VBUS signal detects when the device has been plugged into or unplugged from the USB bus. If the system requires waking up from a low power mode on being plugged into the USB port, this interrupt should also be a low power wake up source. If this pin multiplexes other functions, such as GPIO, the pin can be configured as an interrupt so that the USB plug or unplug event can be detected.

1. If a USB SDP host has suspended the USB device, current drawn by the device has to follow the rules of the Dead Battery Provision in the *USB Battery Charging Specification, Rev. 1.2*.

64.3.1 Modes of operation

The operating modes of the USBDCD module are shown in the following table.

Table 64-3. Module modes and their conditions

Module mode	Description	Conditions when used
Enabled for Charger Detection	The module performs the charger detection sequence.	System software should enable the module only when *all* of the following conditions are true: • The system uses a rechargeable battery or is otherwise capable of being powered up by an SDP or a Charging Port. • The device is being used in a USB device application. • The device has detected that it is attached to the USB cable.
Enabled for Signal Overrides	Module bias circuits are enabled for special signaling on DP/DM pins.	See description for SIGNAL_OVERRIDE[PS] bit field.
Disabled	The module is not active and is held in a low power state.	System software should disable the module when *either* of the following conditions is true:

Table continues on the next page...

Table 64-3. Module modes and their conditions (continued)

Module mode	Description	Conditions when used
		• The charger detect sequence is complete. • The conditions for being enabled are not met.
Powered Off	The digital supply voltage dvdd is removed.	Low system performance requirements allow putting the device into a very low-power stop mode.

Operating mode transitions are shown in the following table.

Table 64-4. Entering and exiting module modes

Module mode	Entering	Exiting	Mode after exiting
Enabled	Set CONTROL[START].	Set CONTROL[SR].[1]	Disabled
Disabled	Take *either* of the following actions: • Set CONTROL[SR].[1] • Reset the module. By default, the module is disabled.	Set CONTROL[START].	Enabled
Powered Off	Perform the following actions: 1. Put the device into very low-power stop mode. 2. Adjust the supply voltages.	Perform the following actions: 1. Restore the supply voltages. 2. Take the device out of very low-power stop mode.	Disabled

1. The effect of setting the SR bit is immediate; that is, the module is disabled even if the sequence has not completed.

64.3.2 The charger detection sequence

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the *USB Battery Charging Specification, Rev. 1.2*.

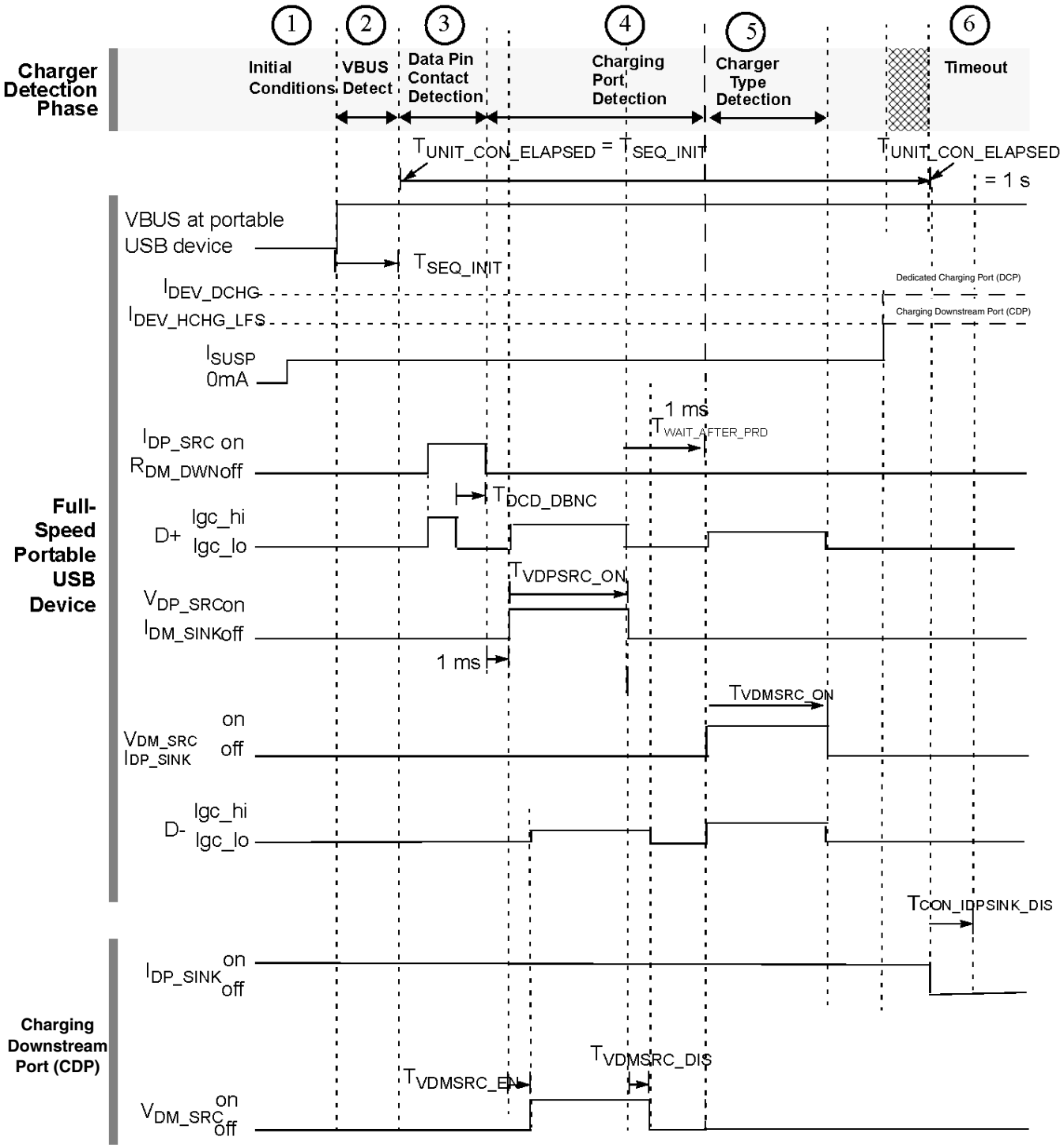


Figure 64-3. Full speed charger detection timing for BC1.2

Timing parameter values used in this module for BC1.2 are listed in the following table.

Table 64-5. Timing parameters for the charger detection sequence for BC1.2

Parameter	USB Battery Charging Spec	Module default	Module programmable range
$T_{DCD_DBNC'}$ [1]	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
$T_{WAIT_AFTER_PRD}$	N/A	1 ms	0– 1023 ms
T_{VDMSRC_ON}	40 ms	40 ms	0 –1023 ms
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON} [1]	1 s	N/A	N/A
T_{VDMSRC_EN} [1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS} [1]	0 – 20 ms	From the USB host	N/A
$T_{CON_IDPSINK_DIS}$ [1]	0 – 10 ms	From the USB host	N/A

1. This parameter is defined by the *USB Battery Charging Specification, Rev. 1.2*.

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the *USB Battery Charging Specification, Rev. 1.1*.

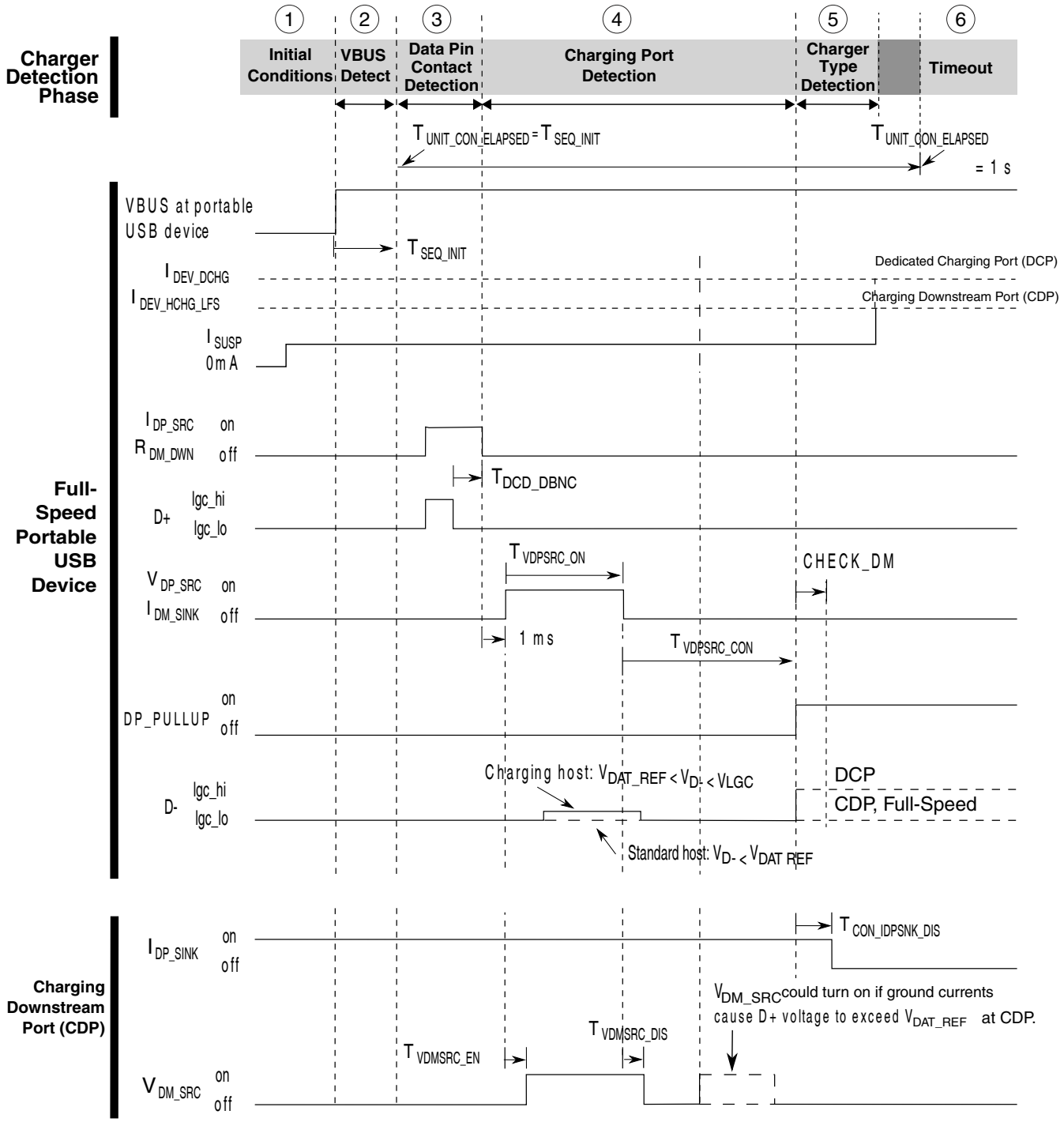


Figure 64-4. Full speed charger detection timing for BC1.1

Timing parameter values used in this module for BC1.1 are listed in the following table.

Table 64-6. Timing parameters for the charger detection sequence for BC1.1

Parameter	USB Battery Charging Spec	Module default	Module programmable range
$T_{DCD_DBNC'}$ [1]	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
T_{VDPSRC_CON} [1]	40 ms min (no max)	40 ms	0 –1023 ms
CHECK_DM	N/A	1 ms	0– 15 ms
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON} [1]	1 s	N/A	N/A
T_{VDMSRC_EN} [1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS} [1]	0 –20 ms	From the USB host	N/A
$T_{CON_IDPSINK_DIS}$ [1]	0– 20 ms	From the USB host	N/A

1. This parameter is defined by the *USB Battery Charging Specification, Rev. 1.1*.

The following table provides an overview description of the charger detection sequence shown in the preceding figure.

Table 64-7. Overview of the charger detection sequence

Phase		Overview description	Full description
1	Initial Conditions	Initial system conditions that need to be met before the detection sequence is initiated.	Initial System Conditions
2	VBUS Detection	System software detects contact of the VBUS signal with the system interrupt pin VBUS_detect.	VBUS contact detection
3	Data Pin Contact Detection	The USBDCD module detects that the USB data pins D+ and D– have made contact with the USB port.	Data pin contact detection
4	Charging Port Detection	The USBDCD module detects if the port is an SDP or either type of charging port, that is CDP or DCP.	Charging port detection
5	Charger Type Detection	The USBDCD module detects the type of charging port, if applicable.	Charger type detection
6	Sequence Timeout	The USBDCD module did not finish the detection sequence within the timeout interval. The sequence will continue until halted by software.	Charger detection sequence timeout

64.3.2.1 Initial System Conditions

The USBDCD module is intended for use with USB device applications using a rechargable battery. The module does not have support for interfacing with ACA or ACA-Dock equipment as defined in the *USB Battery Charging Specification, Revision 1.2*. It cannot be used with USB applications that are embedded host or OTG.

In addition, before the USBDCD module's charger detection sequence can be initiated, the system must be:

- Powered-up and in run mode. The USBDCD instantiation of this module for the High-Speed port does not directly depend on the VBUS voltage and can operate as long as the avdd33 supply is in a valid range.
- Recently plugged into a USB port.
- Drawing not more than 2.5 mA total system current from the USB bus, except as allowed by the *USB 2.0 Connect Timing Update ECN*.

Examples of allowable precursors to this set of initial conditions include:

- A powered-down device is subsequently powered-up upon being plugged into the USB bus.
- A device in a low power mode subsequently enters run mode upon being plugged into the USB bus.

64.3.2.2 VBUS contact detection

Once the device is plugged into a USB port, the VBUS_detect system interrupt is triggered. System software must do the following to initialize the module and start the charger detection sequence:

1. Restore power if the module is powered-off.
2. Set CONTROL[SR] to initiate a software reset.
3. Configure the USBDCD module by programming the CLOCK register and the timing parameters as needed.
4. Set CONTROL[IE] to enable interrupts, or clear the bit if software polling method is used.
5. Program CONTROL[BC12] based on which revision of the USB Battery Charging Specification is needed.
6. Set CONTROL[START] to start the charger detection sequence.

64.3.2.3 Data pin contact detection

The module must ensure that the data pins have made contact because the detection sequence depends upon the state of the USB D+ and D- signals. USB plugs and receptables are designed such that when the plug is inserted into the receptable, the power pins make contact before the data pins make contact. See the following figure.

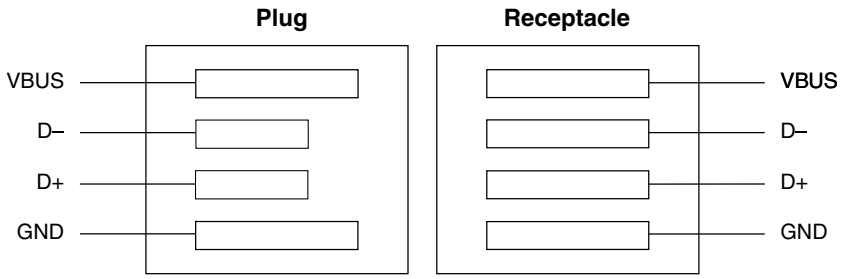


Figure 64-5. Relative pin positions in USB plugs and receptacles

As a result, when a portable USB device is attached to an upstream port, the portable USB device detects VBUS before the data pins have made contact. The time between power pins and data pins making contact depends on how fast the plug is inserted into the receptable. Delays of several hundred milliseconds are possible.

64.3.2.3.1 Debouncing the data pin contact

When system software has initiated the charger detection sequence, as described in Initial System Conditions, the USBDCD module turns on the I_{DP_SRC} current source and enables the R_{DM_DWN} pulldown resistor. If the data pins have not made contact, the D+ line remains high. After the data pins make contact, the D+ line goes low and debouncing begins.

After the D+ line goes low, the module continuously samples the D+ line over the duration of the T_{DCD_DBNC} debounce time interval.By deafult, T_{DCD_DBNC} is 10 ms, but it can be programmed in the TIMER0[TDCD_DBNC] field. See the description of the TIMER0 Register for register information.

When it has remained low for the entire interval, the debouncing is complete. However, if the D+ line returns high during the debounce interval, the module waits until the D+ line goes low again to restart the debouncing. This cycle repeats until either of the following happens:

- The data pin contact has been successfully debounced (see Success in detecting data pin contact (phase completion)).
- A timeout occurs (see Charger detection sequence timeout).

64.3.2.3.2 Success in detecting data pin contact (phase completion)

After successfully debouncing the D+ state, the module does the following:

- Updates the STATUS register to reflect phase completion (See Table 64-12 for field values.)
- Directly proceeds to the next step in the sequence: detection of a charging port (See Charging port detection.)

64.3.2.4 Charging port detection

After it detects that the data pins have made contact, the module waits for a fixed delay of 1 ms, and then attempts to detect whether it is plugged into a charging port. The module connects the following analog units to the USB D+ or D– lines during this phase:

- The voltage source V_{DP_SRC} connects to the D+ line
- The current sink I_{DM_SINK} connects to the D– line
- The voltage comparator connects to the USB D– line, comparing it to the voltage V_{DAT_REF}.

After a time of T_{VDPSRC_ON}, the module samples the D– line. The T_{VDPSRC_ON} parameter is programmable and defaults to 40 ms. After sampling the D– line, the module disconnects the voltage source, current sink, and comparator.

The next steps in the sequence depend on the voltage on the D– line as determined by the voltage comparator. See the following table.

Table 64-8. Sampling D– in the charging port detection phase

If the voltage on D- is...	Then...	See...
Below V_{DAT_REF}	The port is an SDP that does not support using charging currents above I_{CFG_MAX}.	Standard downstream port
Above V_{DAT_REF} but below V_{LGC}	The port is a charging port.	Charging port
Above V_{LGC}	This is an error condition.	Error in charging port detection

64.3.2.4.1 Standard downstream port

As part of the charger detection handshake with a standard USB host, the module does the following without waiting for the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) to elapse:

- Updates the STATUS register to reflect that a standard downstream port (SDP) has been detected with SEQ_RES = 01. See Table 64-12 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. The rest of the sequence, which detects the type of charging port, is not applicable, so software should perform the following steps:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 64-2.

64.3.2.4.2 Charging port

As part of the charger detection handshake with any type of USB host, the module waits until the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) has elapsed before it does the following:

For *USB Battery Charging Specification, Rev. 1.2*:

- Enables V_{DM_SRC}.
- Updates the STATUS register to reflect that a charging port has been detected with SEQ_RES = 10. See Table 64-12 for field values.
- At this point the detection sequence progresses to Charger type detection without needing software interaction.

For *USB Battery Charging Specification, Rev. 1.1*:

- Updates the STATUS register to reflect that a charging port has been detected with SEQ_RES = 10. See Table 64-12 for field values.
- Sets CONTROL[IF].

- Generates an interrupt if enabled in CONTROL[IE].
- At this point, control has passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Issue a command to the USB controller to pullup the USB D+ line.
4. Wait for the module to complete the final phase of the sequence. See Charger type detection.

64.3.2.4.3 Error in charging port detection

For this error condition, the module does the following:

- Updates the STATUS register to reflect the error with SEQ_RES = 00. See Table 64-12 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

Note that in this case the module does not wait for the interval ($T_{WAIT_AFTER_PRD}$ or T_{VDPSRC_CON}) to elapse.

At this point, control has been passed to system software via the interrupt. The rest of the sequence (detecting the type of charging port) is not applicable, so software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.

64.3.2.5 Charger type detection

For *USB Battery Charging Specification, Rev. 1.2*:

After the USBDCD module enables the V_{DM_SRC}, the module starts the T_{VDMSRC_ON} timer counting down the time interval programmed into the TIMER2[T_{VDMSRC_ON}] field.

Once the T_{VDMSRC_ON} timer has elapsed, the module samples the USB D+ line to determine the type of charger. See the following table.

Table 64-9. Sampling D+ in the charger type detection phase (BC1.2)

If the voltage on D+ is...	Then...	See...
High (D+>V_{DAT_REF})	The port is a DCP. [1]	Dedicated charging port
Low (D+<V_{DAT_REF})	The port is a CDP. [2]	Charging downstream port

1. In a DCP, the D+ and D– lines are shorted together through a small resistor.
2. In a CDP, the D+ and D– lines are not shorted.

For *USB Battery Charging Specification, Rev. 1.1*:

After software enables the D+ pullup resistor, the module is notified automatically (via internal signaling) to start the CHECK_DM timer counting down the time interval programmed in the TIMER2[CHECK_DM] field.

After the CHECK_DM time has elapsed, the module samples the USB D– line to determine the type of charger. See the following table.

Table 64-10. Sampling D– in the charger type detection phase (BC1.1)

If the voltage on D– is...	Then...	See...
High	The port is a DCP. [-1]	Dedicated charging port
Low	The port is a CDP. [-1]	Charging downstream port

64.3.2.5.1 Dedicated charging port

For a dedicated charging port (DCP), the module does the following:

- Updates the STATUS register to reflect that a dedicated charging port has been detected with SEQ_RES = 11. See Table 64-12 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE] bit.

The *USB Battery Charging Specification, Rev. 1.2* indicates that additionally if the detection sequence determines an attachment to a DCP, then the USB device should signal on the D+ pin by either enabling the D+ pullup resistor or enabling V_{DP_SRC}.

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Disable the USB controller to prevent transitions on the USB D+ or D– lines from causing spurious interrupt or wakeup events to the system.
3. Set CONTROL[IACK] to acknowledge the interrupt.
4. Set CONTROL[SR] to issue a software reset to the module.
5. Disable the module.
6. Enable signaling on the D+ pin. This can be done by enabling the Transceiver D+ pullup resistor through configuration of register bit fields in the USB controller instance for this USB port. On this product this signaling can alternatively be done by enabling V_{DP_SRC} through setting the SIGNAL_OVERRIDE[PS] bit field to a value of 2'b10.
7. Communicate the appropriate charge rate to the external battery charger IC; see Table 64-2.

64.3.2.5.2 Charging downstream port

For a charging downstream port (CDP), the module does the following:

- Updates the STATUS register to reflect that a CDP has been detected with SEQ_RES = 10. See Table 64-12 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 64-2.

64.3.2.6 Charger detection sequence timeout

The maximum time allowed to connect according to the *USB Battery Charging Specification* is one second. If the Unit Connection Timer reaches the one second limit and the sequence is still running as indicated by the STATUS[ACTIVE] bit, the module does the following:

- Updates the STATUS register to reflect that a timeout error has occured. See Table 64-12 for field values.
- Sets the CONTROL[IF] bit.
- Generates an interrupt if enabled in CONTROL[IE].
- The detection sequence continues until explicitly halted by software setting the CONTROL[SR] bit.
- The Unit Connection Timer continues counting. See the description of the TIMER0 Register.

At this point, control has been passed to system software via the interrupt, which has two options: ignore the interrupt and allow more time for the sequence to complete, or halt the sequence. To halt the sequence, software should:

1. Read the STATUS register.
2. Set the CONTROL[IACK] bit to acknowledge the interrupt.
3. Set the CONTROL[SR] bit to issue a software reset to the module.
4. Disable the module.

This timeout function is also useful in case software does not realize that the USB device is unplugged from USB port during the charger detection sequence. If the interrupt occurs but the V_{BUS_DETECT} input is low, software can disable and reset the module.

System software might allow the sequence to run past the timeout interrupt under these conditions:

1. The USB Battery Charging Spec is amended to allow more time. In this case, software should poll TIMER0[$T_{UNITCON}$] periodically to track elapsed time after 1s; or
2. For debug purposes.

Note that the $T_{UNITCON}$ register field will stop incrementing when it reaches its maximum value so it will not rollover to zero and start counting up again.

64.3.2.7 Dead Battery Provision signaling

If the system needs to operate under the Dead Battery Provision of the *USB Battery Charging Specification, Revision 1.2*, the USBDCD module must be configured to signal V_{DP_SRC} on the USB_DP pin.

For information on Dead Battery Provision conditions and signaling, see the Dead or weak battery later in this chapter.

64.3.3 Resets

There are two ways to reset various register contents in this module: hardware resets and a software reset.

64.3.3.1 Hardware resets

Hardware resets originate at the system or device level and propagate down to the individual module level. They include start up reset, low-voltage reset, and all other hardware reset sources.

Hardware resets cause the register contents to be restored to their default state as listed in the register descriptions.

64.3.3.2 Software reset

A software reset re-initializes the module's status information, but leaves configuration information unchanged. The software reset allows software to prepare the module without needing to reprogram the same configuration each time the USB device is plugged into a USB port.

Setting CONTROL[SR] initiates a software reset. The following table shows all register fields that are reset to their default values by a software reset.

Table 64-11. Software reset and register fields affected

Register	Fields affected	Fields not affected
CONTROL[1]	IF	IE, START
STATUS	All	None
CLOCK	None	All

Table continues on the next page...

Table 64-11. Software reset and register fields affected (continued)

Register	Fields affected	Fields not affected
TIMER*n*	TUNITCON	All other

1. CONTROL[SR] and CONTROL[IACK] are self-clearing.

A software reset also returns all internal logic, timers, and counters to their reset states. If the module is already active (STATUS[ACTIVE] = 1), a software reset stops the sequence.

Note

Software must always initiate a software reset before starting the sequence to ensure the module is in a known state.

64.3.4 Interrupts

The USBDCD module has an interrupt to alert system software of certain events, which are listed in the following table. All events except the Phase Complete event for the Data Pin Detection phase can trigger an interrupt.

Table 64-12. Events triggering an interrupt by sequence phase

Sequence phase	Event	Event description	STATUS fields [1]	Phase description
Data Pin Detection	Phase Complete	The module has detected data pin contact. *No interrupt occurs*: CONTROL[IF] = 0.	ERR = 0 SEQ_STAT = 01 SEQ_RES = 00 TO = 0	VBUS contact detection
Charging Port Detection	Phase Complete	The module has completed the process of identifying if the USB port is a charging port or not. **NOTE:** An interrupt is triggered at completion of this phase for BC 1.1 operation whether an SDP or a Charging port is detected. For BC 1.2 operation, if an SDP is detected an interrupt is triggered but if a Charging port is detected the sequence goes directly to the Charger Type Detection phase.	ERR = 0 SEQ_STAT = 10 SEQ_RES = 01 or 10 TO = 0	Charging port detection
	Error	The module cannot identify the type of port because the D– line is above the USB's VLGC threshold.	ERR = 1 SEQ_STAT = 10 SEQ_RES = 00 TO = 0	Error in charging port detection
Charger Type Detection	Phase Complete	The module has completed the process of identifying the charger type detection.	ERR = 0 SEQ_STAT = 11	Charger type detection

Table continues on the next page...

Table 64-12. Events triggering an interrupt by sequence phase (continued)

Sequence phase	Event	Event description	STATUS fields [1]	Phase description
		NOTE: The ERR flag always reads zero because no known error conditions are checked during this phase.	SEQ_RES = 11 or 10 TO = 0	
Sequence Timeout	Error	The timeout interval from the time the USB device attaches to a USB port until it connects has elapsed.	ERR = 1 SEQ_STAT = last value [2] SEQ_RES = last value [2] TO = 1	Charger detection sequence timeout

1. See the description of the Status register for register information.
2. The SEQ_STAT and SEQ_RES fields retain the values held at the time of the timeout error.

64.3.4.1 Interrupt Handling

Software can read which event caused the interrupt from the STATUS register during the interrupt service routine.

An interrupt is generated only if CONTROL[IE] is set. The CONTROL[IF] bit is always set under interrupt conditions, even if CONTROL[IE] is cleared. In this case, software can poll CONTROL[IF] to determine if an interrupt condition is pending.

Writes to CONTROL[IF] are ignored. To reset CONTROL[IF], set CONTROL[IACK] to acknowledge the interrupt. Writing to CONTROL[IACK] when CONTROL[IF] is cleared has no effect.

64.4 External Signals

This section describes the module signals. The following table shows a summary of module signals that interface with the pins of the device.

Table 64-13. Signal descriptions

Signal	Description	I/O
USB_DM	USB D– analog data signal. The analog block interfaces directly to the D– signal on the USB bus.	I/O
USB_DP	USB D+ analog data signal. The analog block interfaces directly to the D+ signal on the USB bus.	I/O
avdd33 [1]	3.3 V regulated analog supply	I
vss_bulk	Digital/Analog ground	I
dvdd	Core voltage digital supply	I

1. Voltage must be 3.3 V ± 10% for full functionality of the module. That is, the charger detection function does not work when this voltage is below 3.0 V, and the CONTROL[START] bit should not be set.

NOTE

The transceiver module also interfaces to the USB_DM and USB_DP signals. Both modules and the USB host/hub use these signals as bidirectional, tristate signals.

Information about the signal integrity aspects of the lines including shielding, isolated return paths, input or output impedance, packaging, suggested external components, ESD, and other protections can be found in the USB 2.0 Specification and in Application information.

64.5 Initialization information

This module has been designed for minimal configuration while retaining significant programmability. The CLOCK register needs to be initialized to the actual system clock frequency, unless the default value already matches the system requirements.

The proper operation of the USBHSDCD instantiation of this module requires that the USBPHY_ANACTRL[DEV_PULLDOWN] bit field be reset to value 1'b0 during the period when this module is in use.

The other registers generally do not need to be modified, because they default to values that comply with the USB Battery Charging Specification. However, several timing parameters can be changed for a great deal of flexibility if a particular system requires it.

All module configuration must occur *before* initiating the charger detection sequence. Configuration changes made *after* setting CONTROL[START] result in undefined behavior.

64.6 Application information

This section provides application information.

64.6.1 External pullups

Any external pullups applied to the USB D+ or D- data lines must be capable of being disabled to prevent incorrect pullup values or incorrect operation of the USB subsystem.

64.6.2 Dead or weak battery

According to the *USB Battery Charging Specification, Revision 1.2*, a USB device with a dead or weak battery that is attached to an SDP can draw charging current from VBUS without connecting for several minutes, until the battery is charged to the point that the USB device can connect. The conditions for this operation are referred to as the Dead Battery Provision.

The *USB Battery Charging Specification, Revision 1.2* limits this charging condition to 100mA for a duration of 45 minutes. The later *USB 2.0 Connect Timing Update ECN* modifies this charging condition to 500mA for a duration of 2 minutes. Customers designing systems to take advantage of the Dead Battery Provision should study both specifications closely.

The USBDCD module is compatible with systems that do not check the strength of the battery. Therefore, this module assumes that the battery is good, so the USB device must immediately connect to the USB bus by pulling the USB_DP pin high after the USBDCD module has determined that the device is attached to an SDP or CDP. (By definition, a DCP does not support connection but still requires a signaling event, see Dedicated charging port.)

The module is also compatible with systems that have other circuitry to check the strength of the battery. In these systems, if it is known that the battery is weak or dead, software can delay connecting to the USB while charging using the Dead Battery Provision. Once the battery is charged to the good battery threshold, software must then connect to the USB host by pulling the USB_DP pin high.

While using the Dead Battery Provision, the USB_DP pin must signal by enabling V_{DP_SRC}. On this product that signaling can be done most simply by setting the SIGNAL_OVERRIDE[PS] bit field to value 2'b10.

64.6.3 Handling unplug events

If the device is unplugged from the USB bus during the charger detection sequence, the contents of the STATUS register must be ignored and the USBDCD module must get a Software Reset, as described in Software reset.

64.7 Memory map/Register definition

This section describes the memory map and registers for the USBHSDCD module.

64.7.1 USBDCD register descriptions

64.7.1.1 USBDCD memory map

USBHSDCD1 base address: 4043_4800h

USBHSDCD2 base address: 4043_8800h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control register (CONTROL)	32	RW	0001_0000
4	Clock register (CLOCK)	32	RW	0000_00C1
8	Status register (STATUS)	32	RO	0000_0000
C	Signal Override Register (SIGNAL_OVERRIDE)	32	RW	0000_0000
10	TIMER0 register (TIMER0)	32	RW	0010_0000
14	TIMER1 register (TIMER1)	32	RW	000A_0028
18	TIMER2_BC11 register (TIMER2_BC11)	32	RW	0028_0001
18	TIMER2_BC12 register (TIMER2_BC12)	32	RW	0001_0028

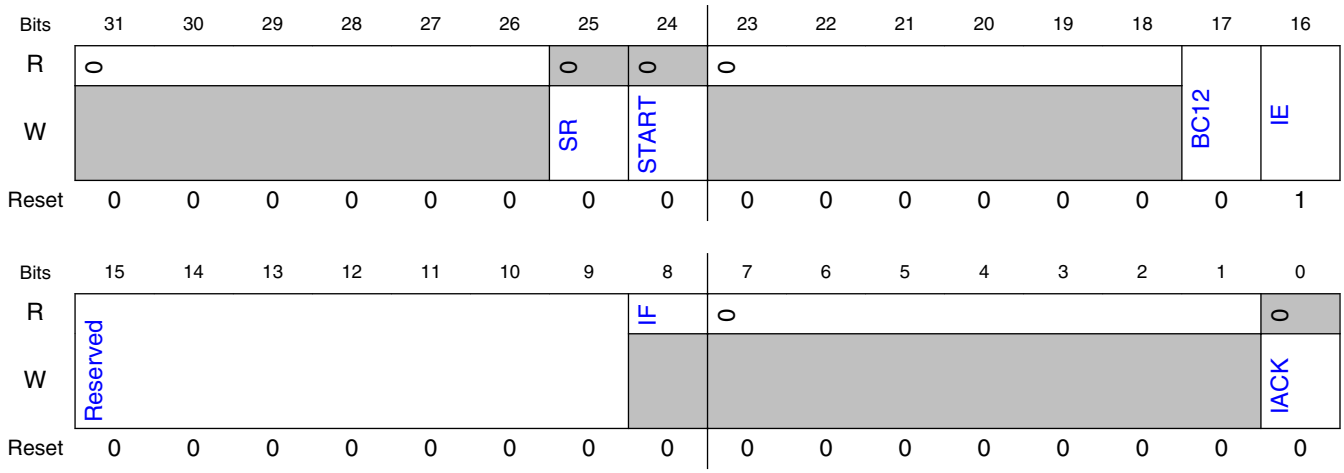
64.7.1.2 Control register (CONTROL)

This register controls various global functions, such as software reset, initiating the charger detection sequence, BC12 compatibility, and interrupt enablement.

64.7.1.2.1 Offset

Register	Offset
CONTROL	0h

64.7.1.2.2 Diagram



64.7.1.2.3 Fields

Field	Description
31-26 —	Reserved
25 SR	Software Reset Determines whether a software reset is performed. 0 - Do not perform a software reset. 1 - Perform a software reset.
24 START	Start Change Detection Sequence Determines whether the charger detection sequence is initiated. 0 - Do not start the sequence. Writes of this value have no effect. 1 - Initiate the charger detection sequence. If the sequence is already running, writes of this value have no effect.
23-18 —	Reserved
17 BC12	BC12 BC1.2 compatibility. This bit cannot be changed after start detection. 0 - Compatible with BC1.1 (default) 1 - Compatible with BC1.2
16 IE	Interrupt Enable Enables/disables interrupts to the system. 0 - Disable interrupts to the system. 1 - Enable interrupts to the system.
15-9 —	Reserved

Table continues on the next page...

Field	Description
8 IF	Interrupt Flag Determines whether an interrupt is pending. 0 - No interrupt is pending. 1 - An interrupt is pending.
7-1 —	Reserved
0 IACK	Interrupt Acknowledge Determines whether the interrupt is cleared. 0 - Do not clear the interrupt. 1 - Clear the IF bit (interrupt flag).

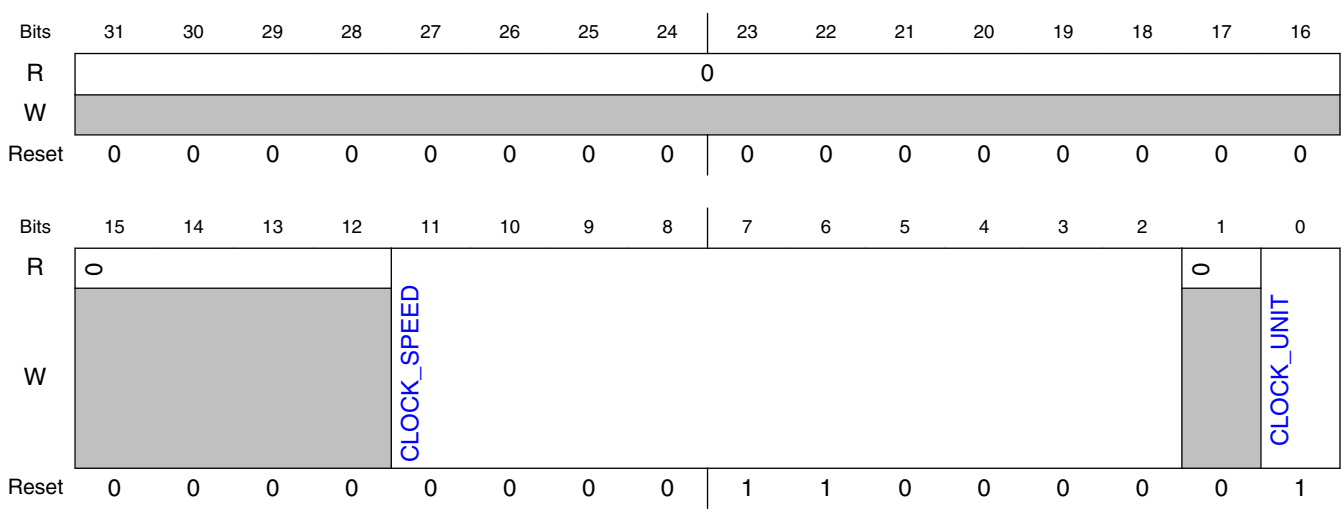
64.7.1.3 Clock register (CLOCK)

This register controls the clock speed.

64.7.1.3.1 Offset

Register	Offset
CLOCK	4h

64.7.1.3.2 Diagram



64.7.1.3.3 Fields

Field	Description
31-12 —	Reserved
11-2 CLOCK_SPEED	Numerical Value of Clock Speed in Binary The unit of measure is programmed in CLOCK_UNIT. The valid range is from 1 to 1023 when clock unit is MHz and 4 to 1023 when clock unit is kHz. Examples with CLOCK_UNIT = 1: • For 48 MHz: 0b00_0011_0000 (48) (Default) • For 24 MHz: 0b00_0001_1000 (24) Examples with CLOCK_UNIT = 0: • For 100 kHz: 0b00_0110_0100 (100) • For 500 kHz: 0b01_1111_0100 (500)
1 —	Reserved
0 CLOCK_UNIT	Unit of Measurement Encoding for Clock Speed Specifies the unit of measure for the clock speed. 0 - kHz Speed (between 1 kHz and 1023 kHz) 1 - MHz Speed (between 1 MHz and 1023 MHz)

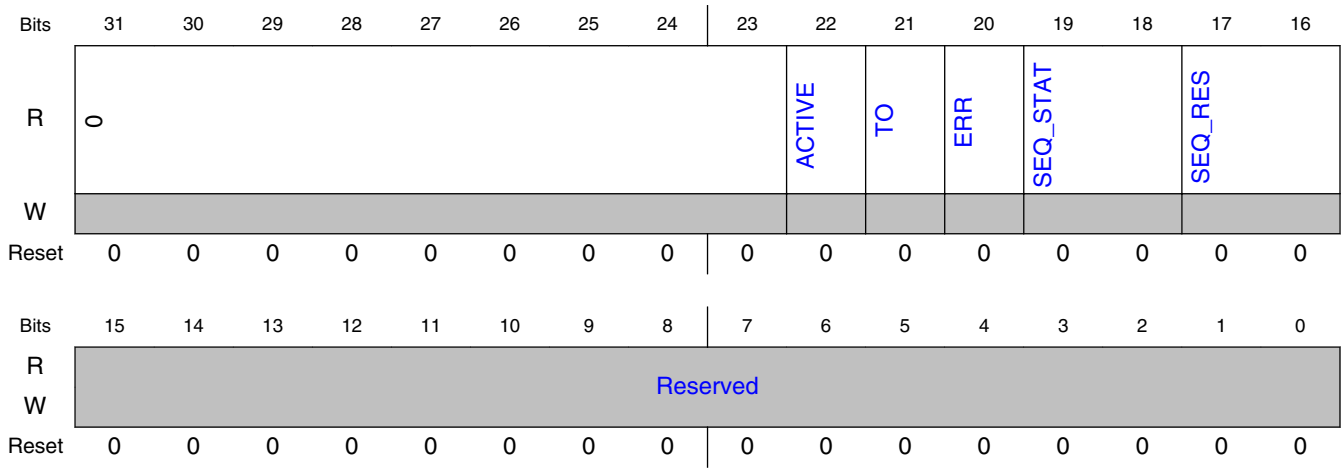
64.7.1.4 Status register (STATUS)

This register stores the current state of the module for system software monitoring.

64.7.1.4.1 Offset

Register	Offset
STATUS	8h

64.7.1.4.2 Diagram



64.7.1.4.3 Fields

Field	Description
31-23 —	Reserved
22 ACTIVE	Active Status Indicator Indicates whether the sequence is running. 0 - The sequence is not running. 1 - The sequence is running.
21 TO	Timeout Flag Indicates whether the detection sequence has passed the timeout threshhold. 0 - The detection sequence has not been running for over 1s. 1 - It has been over 1 s since the data pin contact was detected and debounced.
20 ERR	Error Flag Indicates whether there is an error in the detection sequence. 0 - No sequence errors. 1 - Error in the detection sequence. See the SEQ_STAT field to determine the phase in which the error occurred.
19-18 SEQ_STAT	Charger Detection Sequence Status Indicates the status of the charger detection sequence. 00 - The module is either not enabled, or the module is enabled but the data pins have not yet been detected. 01 - Data pin contact detection is complete. 10 - Charging port detection is complete. 11 - Charger type detection is complete.
17-16	Charger Detection Sequence Results

Table continues on the next page...

Field	Description
SEQ_RES	Reports how the charger detection is attached. 00 - No results to report. 01 - Attached to an SDP. Must comply with USB 2.0 by drawing only 2.5 mA (max) until connected. 10 - Attached to a charging port. The exact meaning depends on bit 18 (value 0: Attached to either a CDP or a DCP. The charger type detection has not completed. value 1: Attached to a CDP. The charger type detection has completed.) 11 - Attached to a DCP.
15-0 —	Reserved **NOTE:** Bits do not always read as 0.

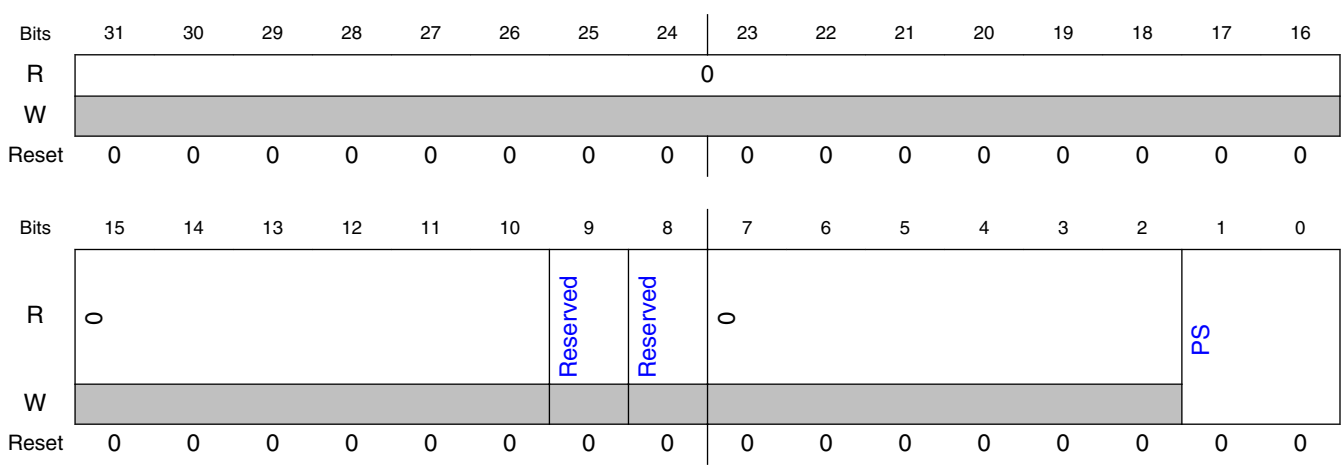
64.7.1.5 Signal Override Register (SIGNAL_OVERRIDE)

This register provides a way for the customer to enable signaling required by the USB BC Rev. 1.2 Specification after the battery charger detection sequences have completed.

64.7.1.5.1 Offset

Register	Offset
SIGNAL_OVERRIDE	Ch

64.7.1.5.2 Diagram



64.7.1.5.3 Fields

Field	Description
31-10 —	Reserved
9 —	Reserved Reserved, not for customer use. The value of this bit field may change during module operation.
8 —	Reserved Reserved, not for customer use. The value of this bit field may change during module operation.
7-2 —	Reserved
1-0 PS	Phase Selection Used to enable specified voltage and current source circuits on the USB_DP and USB_DM pins. Use of the override value below enables the charger detection bias circuits without needing to set the CONTROL[START] bit. Customers may set this bit field to 2'b10 for required signaling if attached to a Dedicated Charging Port, or during operation under the Dead Battery Provision. 00 - No overrides. Bit field must remain at this value during normal USB data communication to prevent unexpected conditions on USB_DP and USB_DM pins. (Default) 01 - Reserved, not for customer use. 10 - Enables VDP_SRC voltage source for the USB_DP pin and IDM_SINK current source for the USB_DM pin. 11 - Reserved, not for customer use.

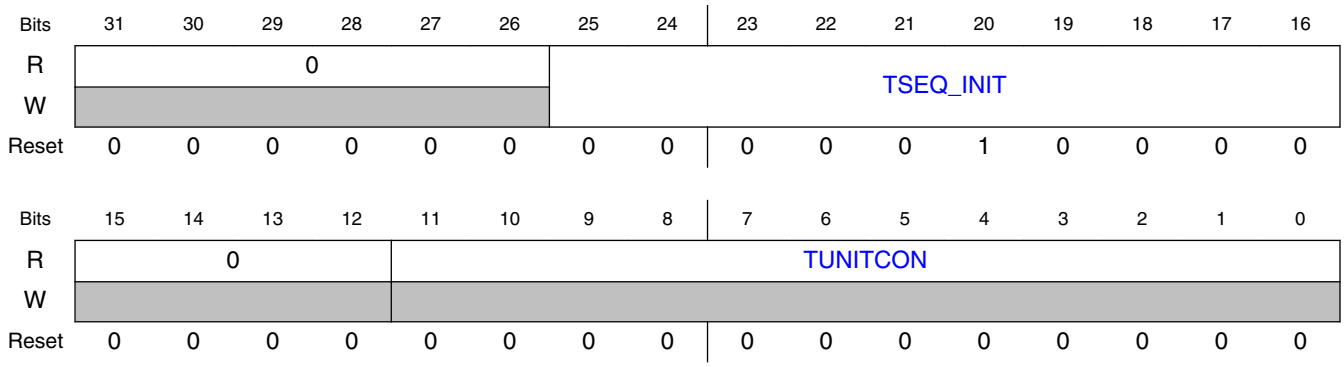
64.7.1.6 TIMER0 register (TIMER0)

This register has an TSEQ_INIT field that represents the system latency in ms. Latency is measured from the time when VBUS goes active until the time system software initiates charger detection sequence in USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT. Valid values are 0-1023, however the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less.

64.7.1.6.1 Offset

Register	Offset
TIMER0	10h

64.7.1.6.2 Diagram



64.7.1.6.3 Fields

Field	Description
31-26 —	Reserved
25-16 TSEQ_INIT	Sequence Initiation Time TSEQ_INIT represents the system latency (in ms) measured from the time VBUS goes active to the time system software initiates the charger detection sequence in the USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT. Valid values are 0-1023, but the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less. 0000000000-1111111111 - 0ms - 1023ms
15-12 —	Reserved
11-0 TUNITCON	Unit Connection Timer Elapse (in ms) Displays the amount of elapsed time since the event of setting the CONTROL[START] bit plus the value of TSEQ_INIT. The timer is automatically initialized with the value of TSEQ_INIT before starting to count. This timer enables compliance with the maximum time allowed to connect T_{UNIT_CON} under the USB Battery Charging Specification. If the timer reaches the one second limit, the module triggers an interrupt and sets the error flag STATUS[ERR]. The timer continues counting throughout the charger detection sequence, even when control has been passed to software. As long as the module is active, the timer continues to count until it reaches the maximum value of 0xFFF (4095 ms). The timer does not rollover to zero. A software reset clears the timer.

64.7.1.7 TIMER1 register (TIMER1)

This register contains timing parameters.

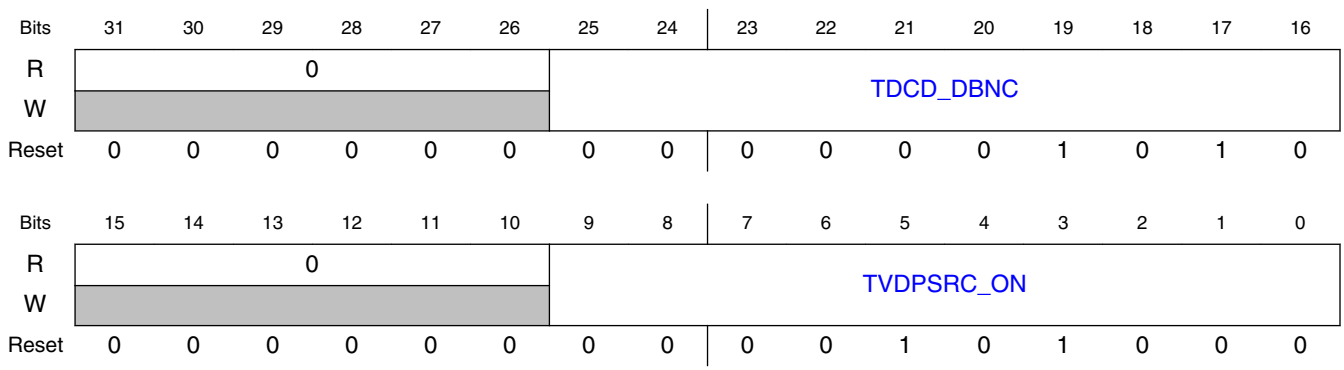
NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

64.7.1.7.1 Offset

Register	Offset
TIMER1	14h

64.7.1.7.2 Diagram



64.7.1.7.3 Fields

Field	Description
31-26 —	Reserved
25-16 TDCD_DBNC	Time Period to Debounce D+ Signal Sets the time period (ms) to debounce the D+ signal during the data pin contact detection phase. See "Debouncing the data pin contact" Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 10 ms. 0000000001-1111111111 - 1ms - 1023ms
15-10 —	Reserved
9-0 TVDPSRC_ON	Time Period Comparator Enabled This timing parameter is used after detection of the data pin. See "Charging Port Detection". Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 40 ms. 0000000001-1111111111 - 1ms - 1023ms

64.7.1.8 TIMER2_BC11 register (TIMER2_BC11)

TIMER2_BC11 contains timing parameters for *USB Battery Charging Specification, Rev 1.1*.

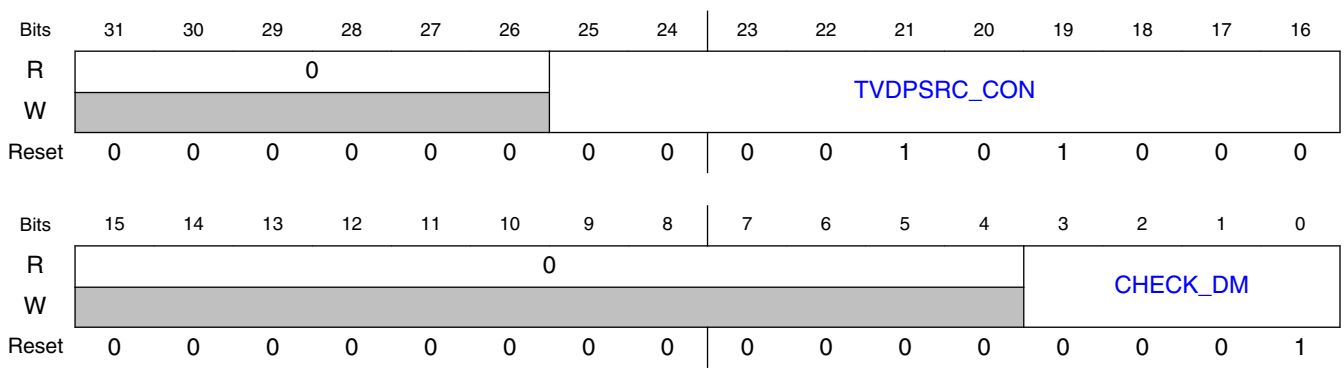
NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

64.7.1.8.1 Offset

Register	Offset
TIMER2_BC11	18h

64.7.1.8.2 Diagram



64.7.1.8.3 Fields

Field	Description
31-26 —	Reserved
25-16 TVDPSRC_CON	Time Period Before Enabling D+ Pullup Sets the time period (ms) that the module waits after charging port detection before system software must enable the D+ pullup to connect to the USB host. Valid values are 1-1023, but the USB Battery Charging Specification requires a minimum value of 40 ms. 0000000001-1111111111 - 1ms - 1023ms

Table continues on the next page...

Field	Description
15-4 —	Reserved
3-0 CHECK_DM	Time Before Check of D- Line Sets the amount of time (in ms) that the module waits after the device connects to the USB bus until checking the state of the D- line to determine the type of charging port. See "Charger Type Detection." Valid values are 1-15ms. 0001-1111 - 1ms - 15ms

64.7.1.9 TIMER2_BC12 register (TIMER2_BC12)

This register contains timing parameters for *USB Battery Charging Specification, Rev 1.2*.

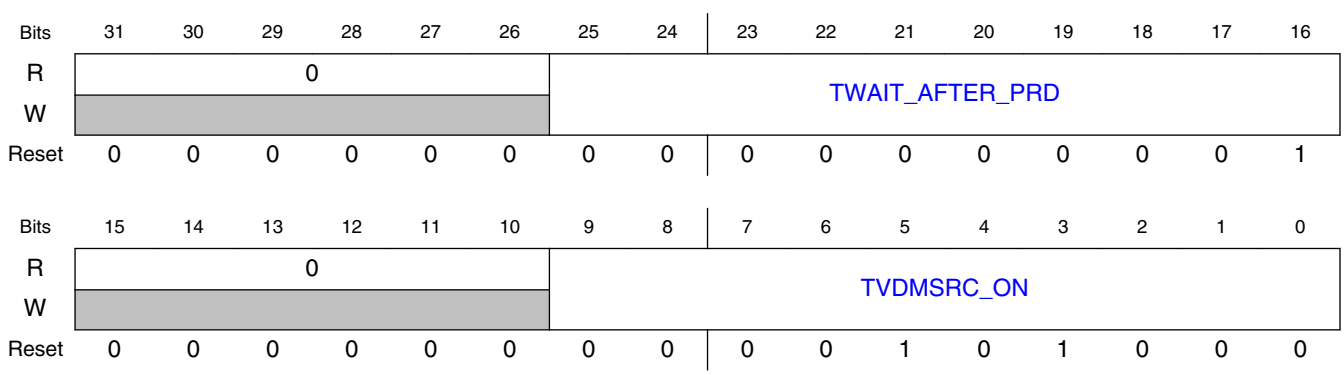
NOTE

Register values can be written that are not compliant with the USB Battery Charging Specification, so care should be taken when overwriting the default values.

64.7.1.9.1 Offset

Register	Offset
TIMER2_BC12	18h

64.7.1.9.2 Diagram



64.7.1.9.3 Fields

Field	Description
31-26 —	Reserved
25-16 TWAIT_AFTER_PRD	TWAIT_AFTER_PRD Sets the amount of time (in ms) that the module waits after primary detection before start to secondary detection. Valid values are 1-1023ms. 0000000001-1111111111 - 1ms - 1023ms
15-10 —	Reserved
9-0 TVDMSRC_ON	TVDMSRC_ON Sets the amount of time (in ms) that the module enables the V_{DM_SRC} . Valid values are 0-40ms. 0000000000-0000101000 - 0ms - 40ms

Chapter 65
Euro, MasterCard, Visa Subscriber Identification Module (EMVSIM)

65.1 Chip-specific EMVSIM information

Table 65-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

Each EMVSIM has two EVENT IRQ assignments. The first EVENT IRQ maps to the data_irq OR'd together (TC, ETC, TFE, and RDT). The second EVENT IRQ maps to all the local interrupts OR'd together, including the data_irq. See the INT_MASK register for more information.

65.2 Overview

The Euro, MasterCard, Visa Subscriber Identification Module (EMVSIM) is designed to facilitate communication to Smart Cards compatible to the EMV v4.3 standard (Book 1) and Smart Cards compatible with ISO/IEC 7816-3 Standard.

65.2.1 Block Diagram

The figure below shows the block diagram for the EMVSIM module.

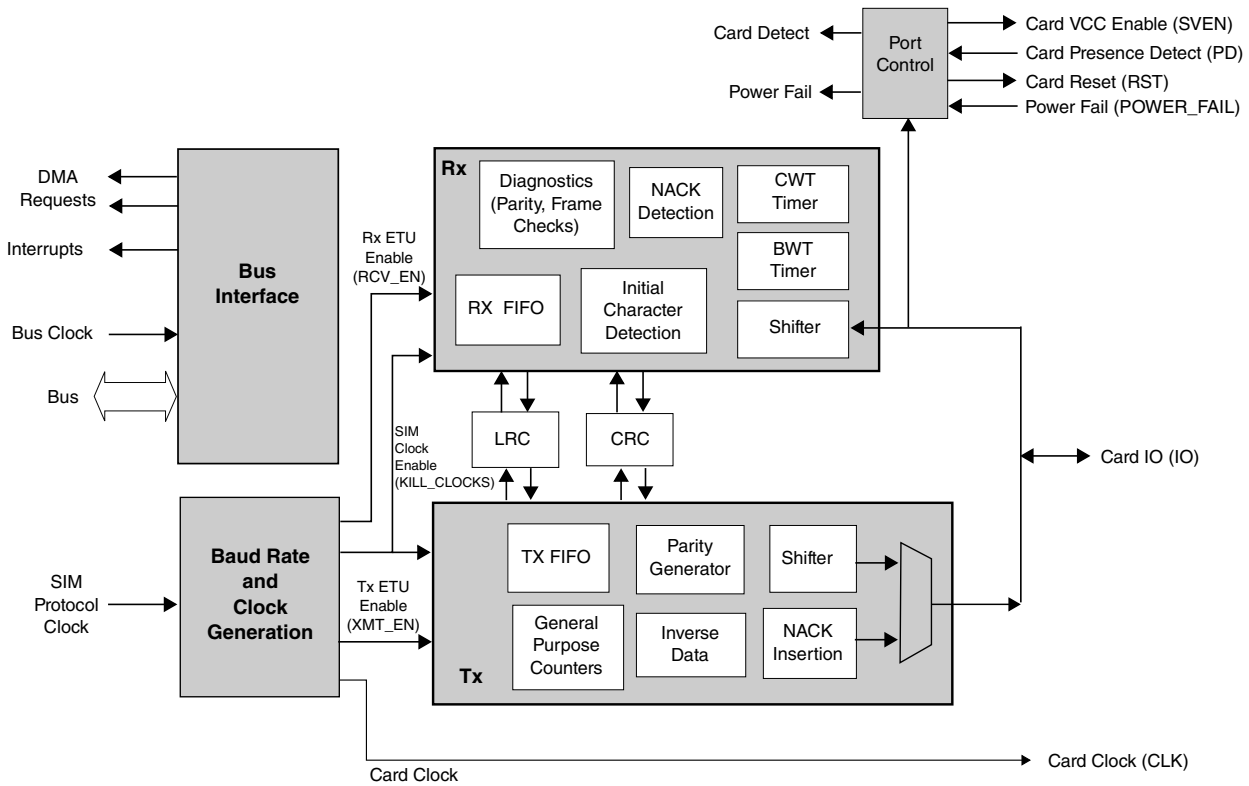


Figure 65-1. EMVSIM Block Diagram

65.2.2 Features

The EMVSIM module supports the following features:

- Independent clock for SIM logic (transmitter + receiver) and independent clock for register read-write interface
- 16 byte deep FIFO for transmitter and receiver
- Automatic NACK generation on parity error and receiver FIFO overflow error
- Support for both Inverse and Direct conventions
- Re-transmission of byte upon Smart Card NACK request with programmable threshold of re-transmissions
- Auto detection of Initial Character in receiver and setting of data format (inverse or direct)

- NACK detection in receiver
- Independent timers to measure character wait time, block wait time and block guard time
- Two general purpose counters are available for use by software application with programmable clock selection for the counters
- DMA support is available to transfer data to/from FIFOs. Programmable option is available to select interrupt or DMA feature
- Programmable Prescaler generates the desired frequency for Card Clock and Baud Rate Divisor generates the internal ETU clocks for transmitter and receiver for any F/D ratio
- Deep sleep wake-up via Smart Card presence detect interrupt
- Manual control of all Smart Card interface signals
- Automatic power down of port logic on Smart Card presence detect
- Support 8-bit LRC and 16-bit CRC generation for bytes sent out from transmitter and check incoming message checksum for receiver

65.3 Functional Description

65.3.1 Bus Interface

The EMVSIM module is a block that is configured via the device's peripheral bus interface. The module has 32-bit aligned memory mapped registers that are used to configure and control the operations of the module. The software can write the transmit bytes or read the received bytes using either interrupt operation or DMA operation. The module can generate necessary DMA requests for the operation.

65.3.2 Smart Card Interface and Control

The EMVSIM module controls all the interface signals for the Smart Card. The Smart Card clock is generated by using the prescaler (CLKCFG[CLK_PRSC]) and the ETU periods are generated by using the baud rate divisor (DIVISOR register). The Smart Card Reset and Voltage Enable are controlled by software through the PCSR register. The message bytes are transmitted and received on the Card IO device pin (IO) which is used in open drain configuration.

The Smart Card interface signals, except IO (controlled by the transmitter and receiver), are generated by the port control block which controls the assertion and de-assertion of the clock, reset and VCC signals to Smart Card. It can do so manually (software needs to explicitly write the necessary bits) or using auto power down wherein the module can

itself power down the Smart Card when configured to do so by software. The port control block can also detect the Smart Card presence and generate necessary interrupts to indicate the same.

For more information about powering up and automatic power down, see Powering Up and Automatic Power Down sections.

65.3.2.1 Smart Card Presence Detect

The Smart Card Presence Detect pin (PD) allows for detection of the insertion or removal of a Smart Card. The PCSR[SPDES] bit allows the software to configure which edge of the PD pin causes a presence detect interrupt flag to be asserted.

An internal pull-down is required on PD pin to allow a high to low transition on this pin when the Smart Card is removed.

65.3.3 EMVSIM Receiver

The EMVSIM Receiver is designed to receive all message bytes and store valid bytes into the receive FIFO. It performs checks on the incoming messages and indicates the transmitter to enable NACK insertion for each message. In addition, the receiver automatically detects NACKs received when the transmitter is sending out message bytes.

The receiver comprises a 16-byte deep FIFO to store the correctly received bytes from the Smart Card. Software must ensure that it reads out the data from FIFO to prevent any overflow when more bytes are sent from the card. The receiver can be configured to detect the initial character. When configured to do so, the data format (direct or inverse convention) is also determined by the receiver and appropriate bit is set in the register map. Internal timers in the receiver measure the various wait times (Character, Block and Guard). On reception of a byte, the receiver checks the byte for Parity Error (in T=0 mode) and Framing Error. It can then direct the transmitter to insert NACK on parity error detection. For a block of bytes received, the receiver checks for LRC or CRC errors (in T=1 mode). See LRC and CRC for more LRC and CRC information. Other errors are stored as status for software to be aware of their occurrence. The status of the receiver operation is intimated to software via maskable interrupts.

65.3.3.1 Data Sampling

The incoming data is internally synchronized and sampled by using a 16x oversampled clock, called the EMVSIM logic clock. The frequency of this clock is 16 times the ETU Clock (1/ETU Period). The whole receiver works on this EMVSIM logic clock.

65.3.3.2 Initial Character Detection

The EMVSIM module supports the detection of the Initial Character and determine the data format when it is placed in the initial character mode by setting the CTRL[ICM] bit. When placed in this mode, the EMVSIM module samples in the input line and waits for one of the initial characters to be detected. Once detected, the CTRL[ICM] bit is cleared and the data format bit, CTRL[IC] bit, is set to appropriate value depending on the data format detected using the initial character.

The two possible data formats are inverse convention and direct convention. Figures below illustrate the differences between the two formats. Essentially, inverse convention differs from direct convention in that the order of the bits is flipped (i.e. MSB for LSB) and the data bits and parity bit are logically inverted. When receiving inverse convention data, the transformation of the data back to direct convention format is done in hardware, including the inversion of the data and parity bits.

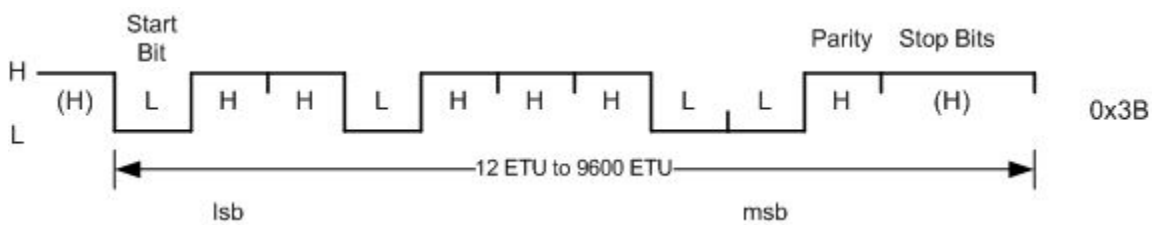


Figure 65-2. Initial Character in Direct Conversion

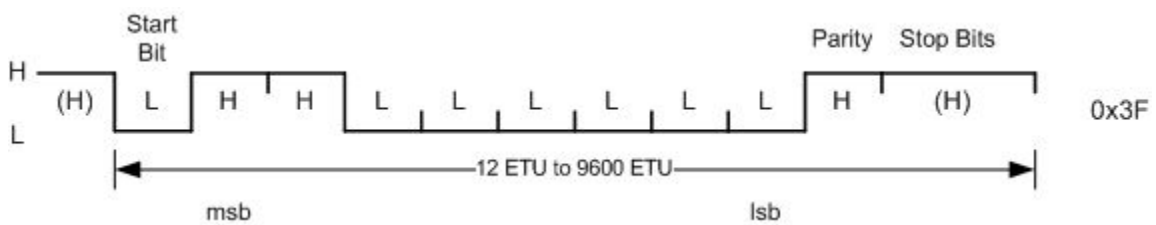


Figure 65-3. Initial Character in Inverse Conversion

To place the EMVSIM into initial character mode, set the CTRL[ICM] bit. Once a valid initial character is received, the CTRL[IC] bit is set accordingly by the hardware, and the CTRL[ICM] bit is cleared. Software can read the state of the CTRL[IC] bit to determine which mode the EMVSIM is currently using. The 0x3B (as decoded by direct

convention) with parity bit high causes direct convention to be used (IC set to logic 0), whereas a 0x3F (as decoded by inverse convention) with parity bit high causes inverse convention to be used (IC set to logic 1).

The receiver remains in initial character mode until disabled by software or a valid initial character is detected. Upon detection of valid character, the receiver enters data receive mode. It is recommended to use one of the general purpose counter as a timeout during initial character mode.

When the receiver is in initial character mode, all received bytes continues to be placed into the receive FIFO whether they are valid initial characters or not. If a valid initial character is received that causes the data format being used to change, all subsequent bytes are decoded with that format before being placed into the FIFO, including the initial character byte itself. That is, if the CTRL[IC] bit is low, and the correct initial character for setting inverse convention is received, that character and all subsequently received characters are stored in the FIFO after having been decoded using inverse convention (for example, the initial character is stored as 0x3F).

NOTE

There is a condition where a parity error in the initial byte for direct convention (0x3B) could be decoded as what appears to be a valid initial character for inverse convention (that is, 0x3F). The EMVSIM module does not recognize this as a valid initial character for inverse convention and sets the PEF flag.

65.3.3.3 Receiver Diagnostics

The EMVSIM module performs diagnostic checks on the incoming messages. These checks include parity check and message frame checks.

Parity Error Detection: The receiver is responsible for detecting parity errors in the received data. Data is always transmitted with even parity, except in inverse convention mode. In inverse convention mode, all data bits and the parity bit are complemented making the data appear to be odd parity. The parity bit is defined as the 10th bit of the received data. The parity of the 2nd through 10th received bits is calculated by the receiver parity logic. This logic determines if the parity of the 9 received bits is correct.

When a parity error is detected on a given byte, the parity error flag (PEF bit) is asserted in the RX_STATUS register if NACK generation on errors is disabled (ANACK = 0 in CTRL register). The parity error flag does not generate an interrupt to the system, however, the module can cause the transmitter to transmit a NACK (if ANACK = 1 in CTRL register) without software needed to enable transmitter for this and request re-transmission of the current byte.

Framing Error Detection: The receiver is responsible for detecting framing errors in the received data. Data is always transmitted with 2 STOP bits, except in 11 ETU mode, where there is one STOP bit. The STOP bits are defined as high state at the 11th and 12th bit positions in the received data. In either mode, the receiver expects to see a high at the 11th bit position in the received data. If a low is detected at the 11th bit position, the receiver indicates a framing error for that byte.

When a framing error is detected on a given byte, the FEF bit is set in the RX_STATUS register. A framing error can not cause an interrupt, nor create a NACK pulse to the Smart Card asking for a retransmission of the corrupted data.

LRC & CRC: The receiver also checks the LRC or CRC on the incoming block of bytes (if LRC or CRC is enabled). The result of the CRC or LRC check is updated in the RX_STATUS register once the complete message block is received. See LRC and CRC for more LRC and CRC information.

65.3.3.4 NACK Detection

During transmission of message bytes, the receiver automatically checks for NACKs sent by the Smart Card. The software does not need to enable receiver for this. The NACK is detected by sampling the IO line at 11 ETUs after the falling edge of the START bit. Once a NACK is detected, the transmitter is signaled and the transmitter retransmits the current byte after two ETU time (as per ISO 7816 specification).

65.3.3.5 Using EMVSIM Receiver with "T=1" Smart Cards

The EMVSIM module provides hardware support for “T=1” type Smart Cards. These type of cards present several requirements above and beyond the standard “T=0” cards. The features provided to meet the requirements that pertain to the EMVSIM receiver are as follows:

- 11 ETU Characters
  - “T=1” cards can transmit with character lengths of 11 ETUs (that is, one STOP bit). The EMVSIM module provides the CTRL[RCVR_11] bit in order to configure the receiver state machine to accept 11 ETU characters.
- Character Wait Time Counter
  - The character waiting time (CWT) is defined as the time between the start bits of two consecutive characters received from the Smart Card. The value of CWT can range from 12 ETU to 32779 ETU. The EMVSIM module provides a 16-bit counter with programmable comparator clocked at the ETU bit rate to identify when the CWT has been exceeded by the Smart Card.
- Block Wait Time

    - The block waiting time (BWT) is defined as the maximum time between the start bits of the last character of a transmitted block and the first character of the next received block. The block wait time must not exceed a value that is programmable. The EMVSIM module provides a 32-bit Block Wait Timer that can be used to identify when the BWT has been violated.
- Block Guard Time
    - The block guard time (BGT) is defined as the minimum delay between the start bits of the last character of a transmitted block and the first character of the next received block. The block guard must be greater than a value that is programmable. The EMVSIM module provides a 16-bit Block Guard Timer that can be used to identify when the BGT has been violated.
- Error Detection Code
    - “T=1” cards can specify LRC or CRC error detection codes to be used. The EMVSIM module provides hardware support for both the LRC and CRC operations.

65.3.4 EMVSIM Transmitter

The transmitter is responsible for reading the message bytes from the Tx FIFO, adding the LRC or CRC at the end of the message block and re-transmit any byte on reception of a NACK. It assists the receive operation by inserting NACKs when directed by the receiver.

The transmitter comprises a 16-byte deep FIFO to store the bytes sent to the Smart Card. These bytes are sent to the Smart Card serially by the transmitter. The data format (i.e. inverse or direct conversion) is supported and the byte is converted into the right data format before being shifted out serially. The transmitter also supports retransmission of the current byte on the reception of a NACK. The transmitter automatically inserts NACK when the receiver indicates a parity or FIFO overflow error, if configured to do so by software. The transmitter is not required to be enabled for this. The transmitter also includes timer to insert the programmable guard time between the transmitter bytes and general purpose timers to allow software to perform certain measurements. The status of the transmitter operation is intimated to software via maskable interrupts.

65.3.4.1 Message Transmission

The transmitter operation does not start until the Tx FIFO is empty. Software can configure the EMVSIM module and enable the transmitter, but no transmission starts unless a byte is written into the Tx FIFO. Clearing the XMT_EN bit while the transmitter is in operation halts any transmission in progress, and flushes the transmit FIFO. See the Transmit FIFO section for more details on using the Tx FIFO.

The transmitter continuously transmits the messages from the Tx FIFO. Upon NACK detection, the transmitter resends the same byte until the Transmitter NACK Threshold limit is reached. When TNTE is set, the EMVSIM transmitter is halted, all pending transfers are aborted, TC, ETC and TFE flags are set. All bytes remaining in the transmit FIFO are lost. There is no way to restart the transmission on the next byte in the FIFO. The transmitter remains frozen until TNTE is cleared by software. The only way to clear TNTE is to write a 1 to the TX_STATUS[TNTE] bit. It is possible to disable the detection of NACKs from the Smart Card by setting TNCK_THD[3:0] to 0x0. By setting TNCK_THD[3:0] to 0x1, it is possible to disable all retransmissions while still setting TNTE on the first NACK received. In general, TNTE is set on the NACK that causes the threshold set by TNCK_THD[3:0] to be reached. This final NACK does not cause a retransmission, whereas all previous NACKs causes a retransmission.

If all bytes are successfully transmitted and the Tx FIFO is empty, the transmitter send the check byte (i.e. LRC or CRC) if the XMT_CRC_LRC and LRC_EN or CRC_EN bits are set. The transmitter sends each byte to the LRC or CRC block to allow the check byte generation.

At the end of each byte, the transmitter inserts the programmed Guard ETUs. Refer to Transmit Guard Time section for more details on Guard Time Counter.

65.3.4.2 NACK Insertion

The transmitter is responsible for driving the output low during the STOP bit time to signify an error was detected in the received data from the Smart Card. This logic responds to a NACK request generated by the receiver block.

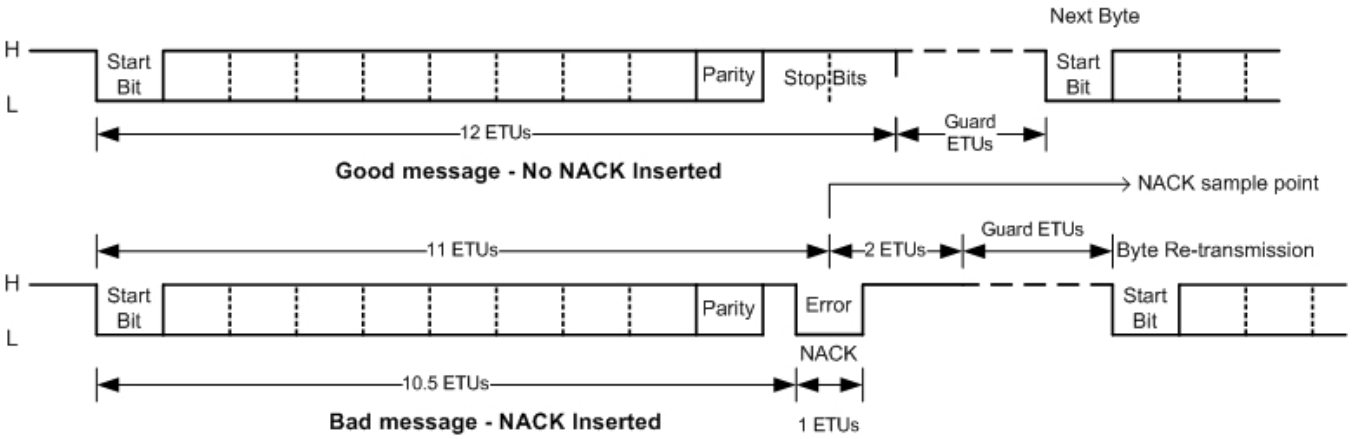


Figure 65-4. NACK Insertion

The NACK generation logic is also responsible for keeping track of the number of NACKs received during a transmit operation. The receiver detects NACKs generated by the Smart Card, and reports them to the transmit NACK logic. Once the number of detected NACKs has reached the programmed threshold (TNCK_THD[3:0]), an interrupt flag (TNTE) is generated, the transmit FIFO is flushed, and the transmitter is disabled.

65.3.4.3 Using EMVSIM Transmitter with "T=1" Smart Cards

The EMVSIM module provides hardware support for “T=1” type Smart Cards. These type of cards present several requirements above and beyond the standard “T=0” cards. The features provided to meet the requirements that pertain to the EMVSIM transmitter are as follows:

- 11 ETU Characters
  - The EMVSIM module transmitter has a programmable guard time register that allows the programmer to specify the number of ETUs between character transmissions. Programming a value of 255 (0xFF) in the TX_GETU[GETU] bits sets the number of ETUs per character transmitted to 11.
- Character Waiting Time
  - The character waiting time (CWT) is defined as the time between the start bits of two consecutive characters. The value of CWT can range from 12 ETU to 32779 ETU. The time between transmitted characters is controlled by the programmable guard time in the TX_GETU register. However, the time between the last byte in the transmit FIFO, and the next transmitted byte can be largely affected by software response time to the transmit interrupts. The EMVSIM transmitter provides a Transmit FIFO threshold (TDTF) interrupt to signal the system when the expected number of characters have been transmitted from the transmit FIFO. The minimum CWT is achieved only if the software can respond

to the TDTF interrupt and write new data to the transmit FIFO before the last character in the Transmit FIFO has been sent.

- Block Waiting Time
  - The block waiting time (BWT) is defined as the maximum time between the start bits of the last character of a transmitted block and the first character of the next received block. The value of BWT is always greater than 1800 ETU. The EMVSIM transmitter provides two General Purpose Counter(s) either of which can be used to track the BWT. The BWT is purely determined by software response time to the transmit interrupts.
- Block Guard Time
  - The block guard time (BGT) is defined as the minimum delay between the start bits of the last character of a transmitted block and the first character of the next received block. The value of BGT is 22 ETU. The EMVSIM module supports the BGT by providing the ability to generate an interrupt when the last byte is received, and transmitting within 2 ETU after the XMT_EN bit is set. The BGT is determined by the speed at which the software can react to an interrupt and enable the transmitter.
- Error Detection Code
  - “T=1” cards can specify LRC or CRC error detection codes to be used. The EMVSIM module provides hardware support for both the LRC and CRC operation.

65.3.5 LRC and CRC

The module also contains the 8-bit LRC and the 16-bit CRC blocks that are common to the transmitter and receiver. Both transmitter and receiver can use the LRC and CRC blocks and can insert the LRC or CRC into the byte stream or check the validity of the input stream using the LRC and CRC values received. Since transmit and receive operations are mutually exclusive, common LRC and CRC blocks for the transmitter and receiver can be used.

65.3.5.1 LRC Block

The EMVSIM module provides an 8-bit Linear Redundancy Check (LRC) generator / checker. The block is provided for use with T=1 Smart Cards that support LRC. This block can be enabled through the CTRL[LRC_EN] bit. This block performs an 8-bit exclusive-OR (XOR) on all received or transmitted characters. At the end of the reception of a block of characters, the result is expected to be 00. If so, the LRC_OK bit is set in the RX_STATUS register. During transmission, the LRC block Exclusive-ORs each

character that is transmitted with the current value of the LRC. If the CTRL[XMT_CRC_LRC] bit is set, the LRC value is automatically sent by the transmitter as the final character when the transmit FIFO empties.

The LRC value can be reset in multiple ways. Clearing the CTRL[LCR_EN] bit resets the LRC value. At the end of a transmission (either after the LRC byte is transmitted, or after the last character in the transmit FIFO is sent when XMT_CRC_LRC is clear), the LRC value is automatically reset by the EMVSIM hardware. Finally, when setting the XMT_EN bit, the EMVSIM hardware resets the LRC value.

Summary of configuration needed to use LRC block:

- Use the CTRL[LRC_EN] bit to enable the LRC block
- Use the XMT_CRC_LRC bit to enable the transmission of the LRC Character after the last character in the Transmit FIFO is sent.

65.3.5.2 CRC Block

The EMVSIM module provides an 16-bit Cyclic Redundancy Check (CRC) generator / checker. The block is provided for use with T=1 Smart Cards that support CRC. This block can be enabled through the CTRL[CRC_EN] bit. This block performs a polynomial based check on all received or transmitted characters. The polynomial used for calculating the CRC is $G(x) = x^{16} + x^{12} + x^5 + 1$. The CRC register is initialized to all “1” before the data is shifted in. The CRC block can be configured to compute using the ISO 7816 CRC method or the 16-bit CCITT CRC method. The software can select the following bits: CRC_IN_FLIP, CRC_OUT_FLIP, and INV_CRC_VAL of the CTRL register.

- To select CCITT CRC method, set
  - CRC_IN_FLIP = 1
  - CRC_OUT_FLIP = 1
  - INV_CRC_VAL = 1
- To select normal CRC method (i.e. ISO 7816 CRC method), set
  - CRC_IN_VAL = 0
  - CRC_OUT_VAL = 0
  - INV_CRC_VAL = 1

During transmission, the CRC block updates the current value of the CRC using each character. If the CTRL[XMT_CRC_LRC] bit is set, the CRC value is automatically formatted as per the selected CRC method (CCITT or normal) and sent by the EMVSIM transmitter as the final two characters when the transmit FIFO empties. At the end of the reception of a block of characters, the residual from the CRC calculation is compared and the CRC_OK bit is set in the RX_STATUS register.

NOTE

Setting any other value for CRC_IN_FLIP, CRC_OUT_FLIP and INV_CRC_VAL causes CRC value to be formatted accordingly; however the CRC is not compliant to the standard CRC used in Smart Cards.

The CRC value can be reset/initialized in multiple ways. Clearing the CTRL[CRC_EN] bit resets the CRC value. At the end of a transmission (either after the CRC characters are transmitted, or after the last character in the transmit FIFO is sent when XMT_CRC_LRC is clear), the CRC value is automatically reset by the EMVSIM hardware. Finally, when setting the XMT_EN bit, the EMVSIM hardware resets the CRC value.

Summary of configuration needed to use CRC block:

- Use CRC_IN_FLIP, CRC_OUT_FLIP and INV_CRC_VAL bits in CTRL register to select the desired CRC method (CCITT or normal)
- Use the CTRL[CRC_EN] bit to enable the CRC block
- Use the XMT_CRC_LRC bit to enable the transmission of the CRC Characters after the last character in the Transmit FIFO is sent.

65.3.6 Message Handling

The EMVSIM module has FIFOs on both transmit and receive side for handling all message bytes.

65.3.6.1 Transmit FIFO

A 16-byte deep FIFO is implemented in the transmitter. The transmit FIFO can be written into by the software but reads to this FIFO returns zeros. To take care of clock synchronization, each write access has wait states inserted by the module but this is transparent to the software. The Tx FIFO can be written to while the transmitter is enabled, however, the transmitter does not start transmitting while the FIFO is empty. A write to the Tx FIFO initiates the transmit operation. Software can store more bytes into the Tx FIFO until the programmed threshold is reached. Software can enable DMA operation by setting TX_DMA_EN = 1 in the CTRL register. On setting this bit, a DMA requestis asserted and remain asserted till the number of bytes in FIFO reached the programmed threshold value (TDT[3:0] in the TX_THD register).

The transmit FIFO can be flushed by setting the CTRL[FLSH_TX] bit. A transmit NACK threshold error (TNTE) halts the transmitter, and flush the transmit FIFO. The flush operation resets the transmit read and write pointers to equal values. Everything in

the transmitter block is reset by the transmit flush operation. This does not include the control registers associated with the transmitter. The Transmit data threshold flag (TDTF) does not get cleared by the FLSH_TX operation.

65.3.6.2 Receive FIFO

A 16-byte deep FIFO is implemented in the receiver. Since more than 16-bytes can be received in the FIFO, the software must ensure that the Rx FIFO is periodically read. The Rx FIFO can be read by software via normal CPU accesses on interrupt (RX_DATA interrupt flag) or via DMA by setting the DMA_RX_EN bit in CTRL register. The software can not write to this register. A write access may terminate in a transfer error.

The Rx FIFO is loaded by the receiver when it receives an error free message byte with correct parity. The FIFO stores the message byte only. When the total bytes in the Rx FIFO equals the programmed threshold value (RDT[3:0] in RX_THD register), the RDTF bit is asserted in the RX_STATUS register. An interrupt is asserted if the INT_MASK[RDT_IM] bit is cleared. If DMA access to RX_FIFO is enabled, the DMA request to read the Rx FIFO is asserted when the threshold flag is set and clears when all the bytes are read out from the Rx FIFO.

The receive FIFO can be flushed by setting the CTRL[FLSH_RX] bit. The flush operation resets the receiver except for receiver control registers.

NOTE

Since the register bits and the receiver logic are in different clock domains, the software needs to make sure to allow clock synchronization time when sampling the RDTF bit. One proposed method could be to read one byte at a time (from Rx FIFO) when RDTF is asserted, allow a time of 3 bus clock cycles after Rx FIFO read and then check the status of the RDTF bit and repeat as necessary. Alternatively, one could set the receiver threshold one more than actually needed.

Rx FIFO Overflow Detection: When a byte is received by the receiver and the Rx FIFO is not read and already contains 16 bytes, a FIFO overflow error is asserted (RFO bit set to 1 in RX_STATUS register). The received byte is discarded leaving the FIFO with the first 16 bytes received. If the ONACK bit in the CTRL register is asserted, the transmitter generates a NACK for the overflow condition every time a new byte is received that can not be stored in the Rx FIFO. The EMVSIM module continually sends NACKs till the Rx FIFO is read or NACK transmit threshold is reached.

65.3.7 Protocol Timers

The EMVSIM module has several 16-bit and 32-bit timers to allow the software to perform protocol timing checks and measurements like:

- Transmit Guard Time
- Character Wait Time
- Block Wait Time
- Block Guard Time
- Work Wait Time

In addition to the above, the EMVSIM module also contains two 32-bit general purpose counters which can be used by software for any specific time measurement application.

65.3.7.1 Transmit Guard Time

The Transmit Guard time is the number of ETUs inserted by the transmitter between each character being transmitted. The number of ETUs is controlled by programming the TX_GETU register. The Guard Time is measured as the time (in ETUs) between the STOP bits of previous character and the START bit of next character.

For a Guard Time programmed as 0xFF in TX_GETU[GETU] bits, the transmitter inserts one STOP bit instead of two.

65.3.7.2 Character Wait Time

The EMVSIM receiver block includes a 16-bit counter that counts the number of bit times (ETUs) between received characters (i.e. time between start bits in the consecutive characters received). When enabled, the Character Wait Time Counter (CWT) does not start counting until after the START bit(s) of a valid character has been received. The counter is synchronized to the receive character bit positions so that an accurate count of the number of ETUs between characters can be made. The Character Wait Time Counter has a 16-bit programmable comparator. Software can write the expected number of ETUs between characters to the comparator. If the time between characters exceeds this value, an interrupt flag is set and an interrupt generated if the mask is clear. The CWT can be configured as follows:

- Program the CWT_VAL register to the value that needs to be enforced
- Activate CWT function by setting the CTRL[CWT_EN] bit
- Enable the CWT_ERR interrupt by clearing the INT_MASK[CWT_IM] bit

- If an interrupt occurs, then the software should read the RX_STATUS register to determine if the error was caused by a CWT_ERR violation. The software should clear the CWT_ERR interrupt by writing a “1” to the RX_STATUS[CWT_ERR] bit
- Software can disable the CWT counter by clearing the CTRL[CWT_EN] bit at any time

65.3.7.3 Block Wait Time and Block Guard Time

A 32-bit timer is used to measure both Block Wait Time (BWT) and Block Guard Time (BGT). The BWT and BGT timer are used to measure the delay between the leading edge of the last character of the block received by the card and the leading edge of the first character of the next block sent by the card. If this time measured is larger than the value in BWT_VAL register, then the BWT_ERR flag is set and an interrupt generated. If the time is less than the value in the BGT_VAL register, then the BGT_ERR flag is set and an interrupt generated. The block wait timer can be configured as follows:

- Program the BWT_VAL and BGT_VAL registers to the value that need to be enforced
- Activate the block wait timer by writing '1' to CTRL[BWT_EN] bit
- Enable the BWT_ERR and BGT_ERR interrupts by clearing the INT_MASK[BWT_IM] and INT_MASK[BGT_IM] bits
- If an interrupt occurs, then the software should read the RX_STATUS register to determine if the error was caused by a BWT_ERR or BGT_ERR violation. The software should clear the interrupt status bit by writing a “1” to the BWT_ERR bit or BGT_ERR, respectively, in the RX_STATUS register
- Software can disable the BWT counter by clearing the CTRL[BWT_EN] bit at any time

NOTE

BWT should be enabled after all bytes are written to in Tx FIFO. This is necessary to avoid the condition where writes to Tx FIFO are very slow (more than 1 ETU time apart) and if BWT is enabled earlier, then BWT error can trigger falsely.

65.3.7.4 Work Wait Time

There is no separate counter to measure the work wait time (WWT). Work wait time is a combination of BWT and CWT. To measure WWT, software must configure both CWT_VAL and BWT_VAL to the value to be enforced. When measuring WWT, the BGT timer is not used so it should be masked (inactive) by the BGT_IM bit. Below is the sequence to program the EMVSIM to send and receive data while checking WWT:

- Program both the CWT_VAL and the BWT_VAL registers to the WWT (work wait time) value that needs to be enforced. Set BGT_VAL register to 0 (this is the default value).
- Activate both the CWT and BWT functions by setting the CTRL[CWT_EN] and CTRL[BWT_EN] bits
- Enable the CWT_ERR and BWT_ERR interrupts by clearing the INT_MASK[CWT_IM] and INT_MASK[BWT_IM] bits
- Program the EMVSIM module to enable Receiver or Transmitter as desired by application
- If an interrupt occurs, then the software should consider that a WWT violation if the CWT_ERR or BWT_ERR bits are asserted. The software should clear the CWT_ERR or BWT_ERR interrupt by writing a “1” to the RX_STATUS[BWT_ERR] bit or RX_STATUS[CWT_ERR] bit

65.3.7.5 General Purpose Counters

The EMVSIM module provides two 16-bit counters for use, when timing events during Smart Card communication. The clock source for the counter is selectable between three sources: Card Clock, Receiver Clock (16 times the ETU clock) or ETU Clock (for Transmitter). The CLKCFG[GPCNT0_CLK_SEL] and CLKCFG[GPCNT1_CLK_SEL] bits are used to select the clock input. The counter is enabled as soon as the input clock is selected. The starting of the counter is immediate once the input clock is running. Software can control the three input clock sources by using the SCEN, KILL_CLOCKS, RCV_EN and XMT_EN bits provided in the CTRL register.

To run the counters from the card clock source the following conditions must be met:

- CTRL[KILL_CLOCKS] = 0
- CTRL[RCV_EN] = 1 or CTRL[XMT_EN] = 1
- PCSR[SCEN] = 1
- CLKCFG[GPCNT0/1_CLK_SEL] = 01

The counter begins to count at the card clock rate as soon as these conditions are met.

To run the counters from the receive clock source the following conditions must be met:

- CTRL[KILL_CLOCKS] = 0
- CTRL[RCV_EN] = 1 or CTRL[XMT_EN] = 1
- CLKCFG[GPCNT0/1_CLK_SEL] = 10

The counter begins to count at the receive clock rate as soon as these conditions are met.

To run the counters from the ETU (transmit) clock source the following conditions must be met:

- CTRL[KILL_CLOCKS] = 0

- CTRL[RCV_EN] = 1 or CTRL[XMT_EN] = 1
- CLKCFG[GPCNT0/1_CLK_SEL] = 11

The counters begins to count at the ETU clock rate as soon as these conditions are met.

The counters can be reset by setting GPCNT0/1_CLK_SEL to 00. A 16-bit comparator value (GPCNT0/1_VAL register) is provided that allows the software to select a count value at which the interrupt flags (TX_STATUS[GPCNT0_TO] and TX_STATUS[GPCNT1_TO] bits) can be set and an interrupt generated if the mask (INT_MASK[GPCNT0/1_IM] bits) are clear.

The following sequence should be followed to configure both general purpose timers:

- Program counter comparator using the GPCNT0/1_VAL register
- Enable the conditions described above using the CTRL register
- Select desired clock source for the General Purpose Counter using the CLKCFG register
- Enable interrupts in the INT_MASK register

65.3.8 Answer To Reset (ATR) Detection

The first communication that occurs between the Smart Card and the EMVSIM module is a block of data sent from the Smart Card to the EMVSIM module after the card is powered up and the card reset is removed (As described in Smart Card Interface and Control Section). This block is called the Answer To Reset (ATR). To receive the ATR, the EMV SIM module should be configured for 12 ETU character reception. According to the ISO 7816-3 spec, both “T=0” and “T=1” cards communicate initially using 12 ETU character durations.

For more information about ATR Detect Configuration, see the Configuration for ATR Detect section.

65.3.9 Clocks

The key functional blocks in the module are clocking, transmitter, and receiver which control the interactions with the Smart Card. These blocks function on a clock that is different and asynchronous to the register read/write clock. In order for the transmitter and receiver to function properly, the respective configuration must be done before enabling the transmitter or receiver.

The EMVSIM clocking module generates the following clocks:

- Card clock (output to Smart Card)

    - The prescaler (CLKCFG[CLK_PRSC]) generates the card clock in the range of 1 to 5 MHz with near 50% duty cycle
- Internal clocks for the transmitter and receiver - the internal clocks control the ETU period and the sampling and driving of data on the IO line
    - The receive operation uses an internal over-sampled clock that oversamples the received input at 16x times the ETU rate
    - The baud divider (DIVISOR register) generates the internal ETU clocks for transmitter and receiver.

65.4 External Signals

The module interfaces to the external card (Smart Card) via card clock, card reset, card V_{CC} enable and IO pins. The transmitter output from the module and the receiver input to the module are connected to the same IO pin and used in the open drain configuration.

NOTE

A pull-up resistor must be connected to the IO pin on board to allow proper functioning.

The following table shows the user-accessible signals available on EMVSIM. See the chip-level specification to find out which signals are connected to the external pins.

Table 65-2. EMVSIM Signal Description

Signal	Description	I/O
CLK	Card Clock. Clock to Smart Card.	O
RST	Card Reset. Reset signal to Smart Card	O
SVEN	Card Power Enable. This signal controls the power to Smart Card	O
IO	Card Data Line. Bi-directional data line.	I/O
PD	Card Presence Detect. Signal indicating presence or removal of card	I
POWER_FAIL	External PMIC power fail indication.	I

65.5 Initialization

The first operation that must be done is to configure the EMVSIM interface signals to allow the EMVSIM Transmit or Receive operation to commence. The following steps describe a typical way to enable the Smart Card:

1. Configure prescaler by programming the desired value in CLKCFG register to generate the card clock. The card clock frequency is calculated as:

- f_{SIM} / (CLKCFG[CLK_PRSC]); where f_{SIM} is the EMV SIM Clock Frequency
- Prescaler value must be set such that the maximum card clock frequency specification set by EMV & ISO-7816 specifications are not violated

2. Configure the DIVISOR[DIVISOR_VALUE]. This value determines the ETU period to be used by the transmitter or receiver
3. Enable power to the Smart Card by setting the PCSR[SVCC_EN] bit
4. Enable the Smart Card clock by setting the PCSR[SCEN] bit
5. Release Smart Card reset using the PCSR[SRST] bit to bring the Smart Card out of reset
6. Program the power down functionality i.e. if Auto Power Down of Smart Card is required or not
7. Select the Data Format type for the EMVSIM
   - Either set the CTRL[IC] bit to the desired data format (inverse or direct) OR
   - Set the CTRL[ICM] bit to allow automatic detection of Initial Character. The CTRL[IC] bit is updated to desired data format upon valid Initial Character detection
8. Configure if the EMVSIM module is enabled to run in STOP and DOZE modes by setting the CTRL[STOP_EN] and CTRL[DOZE_EN] bits as desired
9. Enable DMA request generation for Transmitter or Receiver as desired

Once the above configuration is complete, the receiver or transmitter can be configured depending on the operation to be carried out on the Smart Card.

NOTE

XMT_EN and RCV_EN bits should not be changed in the same write cycle. Software should attempt separate write accesses to CTRL register for updating these bits, one by one. For example, if XMT_EN = 1 and RCV_EN = 0 and in order to enable the receiver, first write to CTRL register should make XMT_EN = 0 and second write to CTRL register should make RCV_EN = 1.

NOTE

The Transmit and Receive operations are mutually exclusive and should not be enabled together.

65.5.1 Configuring Receiver

The following is a list of configurations that need to be performed (after the "Configuring EMVSIM" step) in order to configure the EMVSIM receiver for operation:

1. Enable NACK generation capability

- To generate NACK on parity errors and invalid initial characters, set the CTRL[ANACK] bit
- To generate NACK on receiver FIFO overflow error, set the CTRL[ONACK] bit

2. Configure the desired threshold for NACK generation and data threshold in the RX_THD register. These are the thresholds at which the RTE and RDTF flags are set, respectively.
   - If auto power down is desired on RTE flag assertion, then write to SAPD bit PCSR to initiate the Smart Card power down
3. Configure the timeout values for Character Wait Time Counter, the Block Wait Timer Counter and the Block Guard Time Counter
4. Enable necessary interrupts by clearing respective bits in the INT_MASK register
5. Configure the General Purpose Counter as needed and select desired clock source (CLKCFG[GPCNT0/1_CLK_SEL])
6. Enable Receiver by setting CTRL[RCV_EN] bit to 1
7. Receiver enters Initial Character Detection mode if CTRL[ICM] bit is set or enter normal receive mode if CTRL[ICM] bit is not set
8. Enable LRC or CRC as desired

65.5.2 Configuring Transmitter

The following is a list of configurations that need to be performed (after the "Configuring EMVSIM" step) in order to configure the EMVSIM transmitter for operation:

1. Select desired re-transmission threshold for NACKed characters in TX_THD register
   - This is the threshold at which the TNTE flag is set by using the TNACK_THD bits
2. Configure the desired Guard Time between the characters transmitted by writing to the TX_GETU[GETU] bits
3. Configure the desired data threshold in TX_THD register.
4. Enable necessary interrupts by clearing respective bits in the INT_MASK register
5. Configure the General Purpose Counter as needed and select desired clock source (CLKCFG[GPCNT0/1_CLK_SEL])
6. Enable Transmitter by setting the CTRL[XMT_EN]
7. Enable LRC & CRC as desired
8. Program bytes to be transmitted into the Tx FIFO. If DMA is enabled, DMA request is asserted automatically.

NOTE

Tx FIFO can be written to after the transmitter is enabled (XMT_EN = 1). Writing to Tx FIFO while the transmitter is disabled is not successful and write access is ignored.

65.6 Application Information

65.6.1 Powering Up

The first step to communicate with a Smart Card is to provide power and a clock signal to the card. Once the card is detected as present (using the presence detect features, or some other method), the Smart Card should be powered up according to the power up sequence specified in the ISO 7816-3 specification.

1. Apply voltage to the Smart Card by setting the PCSR[SVCC_EN] bit
2. Select the appropriate clock frequency for the Smart Card by programming the CLKCFG and DIVISOR registers
3. Enable the clock to the Smart Card by setting the PCSR[SCEN] bit
4. Card exits reset state by setting the PCSR[SRST] bit

65.6.2 Automatic Power Down

Smart Cards require a proper sequence of operations to power down. The power down sequence as per ISO 7816 specifications is:

1. Smart Card Reset (RST) pin transitions from high to low
2. Smart Card Clock (CLK) is turned off to low
3. Smart Card IO (IO) is transitioned from high impedance to low
4. Smart Card Voltage Enable (SVCC_EN) is turned off (pin driven low)

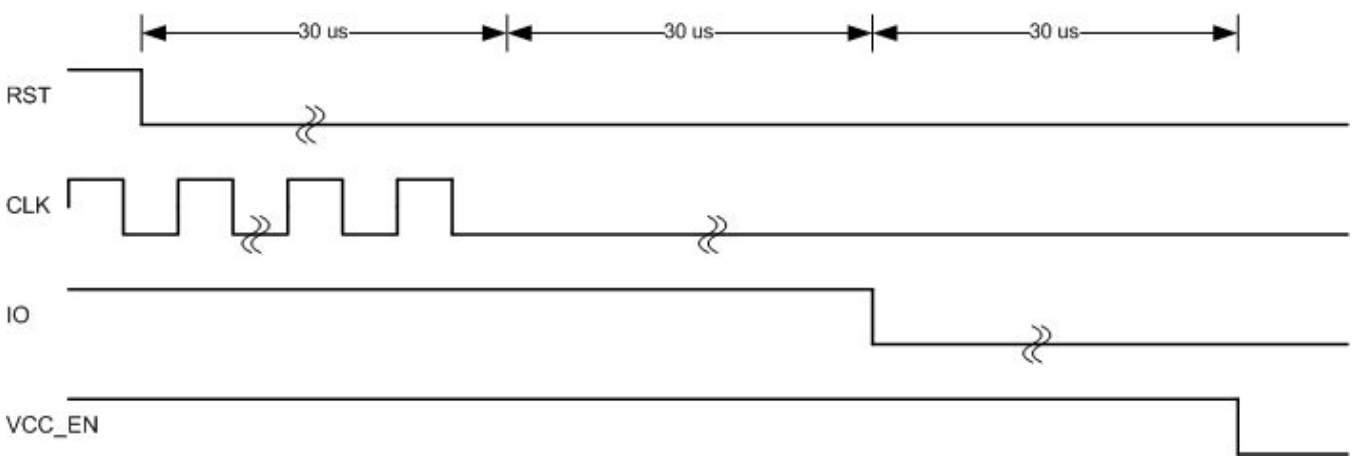


Figure 65-5. Auto Power Down Sequence

These operations can be done manually by software which writes to appropriate bits (SCEN, SRST and SVCC_EN) in the PCSR register or automatically by hardware under the following conditions.

- Setting the PSCR[SPD] bit by software
- Assertion of RX_STATUS[RTE] bit
- Assertion of PCSR[SPDIF] bit when configured to detect card removal
- Power Fail indication from off-chip power management controller

NOTE

The PCSR[SPDES] bit determines which edge transition of the Smart Card Presence Detect pin is used for Smart Card presence detection. Presence detection can be used to determine if the card has been inserted or removed. The occurrence of the edge specified by SPDES bit causes the following: SPDIF to be set; if the SPDIM mask is clear, an interrupt to the CPU; and if SAPD in the PCSR register is set, an auto power down sequence to begin. If Smart card insertion is expected, SAPD can be set low to avoid the auto power down sequence from falsely triggering. The bit SPDP can be used to determine the current state of the Smart Card Presence Detect pin.

NOTE

The auto power down operation requires a low frequency clock (usually 32 kHz clock source) to be active on the device.

The auto power down sequence clears the following register bits upon power down sequence completion:

1. PCSR[SPD] bit
2. PCSR[SCEN] bit
3. PCSR[SRST] bit
4. PCSR[SVCC_EN] bit
5. CTRL[XMT_EN] and CTRL[RCV_EN] bits

65.6.3 Configuration for ATR Detect

The following steps provide a suggested approach to configure the EMVSIM module to receive the ATR.

1. Clear the CTRL[RCVR_11] bit
2. Set the CTRL[ANACK] bit to enable NACK generation. The ISO 7816-3 spec allows the EMVSIM module to NACK any communication errors that occur during the initial communication at 12 ETU.

NOTE

The Europay MasterCard and VISA (EMV) cards are similar to “T=1”, but do not allow the EMVSIM module to NACK during the initial communication

3. Enable RDTF and RFO interrupts by setting RDT_IM and RFO_IM bits in INT_MASK register. These interrupts are enabled to notify the software when characters are received. A threshold can be set for the number of characters to receive before generating an interrupt.
4. Set desired threshold for received characters by writing RDT in RX_THD register

The EMVSIM should be setup to perform in initial character mode during ATR detection. This causes the hardware to identify the first valid character sent during the ATR as an initial character. This character automatically configures the hardware for the data convention used by the Smart Card.

- Set Initial Character Mode by setting the CTRL[ICM] bit

The ISO 7816-3 spec requires that Smart Cards meet certain timing restrictions. One of these is the time from the de-assertion of the card reset to the beginning of the ATR sequence. The General Purpose Counter of EMVSIM module can be used to verify that the Smart Card begins its ATR within the 400 to 40,000 card clock cycle range.

1. Set General Purpose Counter Timeout Value with 0x9C40 by programming GPCNT0/1_VAL register.
2. Enable General Purpose Counter Interrupt by clearing GPCNT0/1_IM bit in INT_MASK register.
3. Enable General Purpose Counter by programming the GPCNT0/1_CLK_SEL bits to 01, so that the card clock is used for counting.

The ISO7816-3 spec states that the maximum allowed time between two characters during the ATR is 9600 ETUs (Initial Waiting Time). The Character Wait Time Counter should be setup to detect any errors for this condition.

1. Set Character Wait Time Counter Comparator to 9600 using the CWT_VAL register
2. Enable the Character Wait Time Counter Interrupt by clearing CWT_ERR_IM in INT_MASK register
3. Enable the Character Wait Time Counter by setting the CTRL[CWT_EN] bit

The last step in preparing for ATR reception is to enable the receiver.

- Set RCV_EN bit in CTRL register

The EMVSIM module generates interrupts once a threshold number of characters is received. The software should react to these interrupts and read the characters from the receive FIFO (RX_BUF) until the complete ATR has been received. If a General Purpose Counter interrupt occurs before the final ATR character is received, then the card should

be deactivated according to the ISO 7816-3 spec (See Section Smart Card Interface and Control). Otherwise, once a valid ATR is received, the software knows from the ATR information the specific characteristics for this card (refer to the ISO 7816-3 spec for details).

65.6.3.1 Programming Considerations for T=0 Smart Cards

For type T=0 card, the software should adjust the following parameters according to the information in the ATR:

- Adjust the baud rate by changing the values of DIVISOR register based on the values of 'F' and 'D' received in the ATR
- Adjust the guard time between characters by changing the value of GETU in the TX_GETU register
- Adjust NACK capability by modifying the values of the ONACK and ANACK bits in the CTRL register
- Adjust the stop clock polarity by modifying the value of the PCSR[SCSP] bit
- Adjust the level of transmit NACK re-transmissions allowed by modifying the value of the TX_THD[TNCK_THD] bits
- Adjust the level for the Receive NACK threshold by modifying the RX_THD[RNCK_THD] bits

If a negotiation with the Smart Card is desired, the software sends a PPS response to the Smart Card. To send the response, the following steps should be performed:

- Set the desired transmit FIFO threshold level by writing the TDT bits in the TX_THD register
- Clear all transmit interrupt flags in the TX_STATUS register by writing a one to the bits of the register
- Enable the transmit interrupts desired by clearing the mask bits in the INT_MASK register. If more than 16 characters are to be sent, it is suggested that the TDTF interrupt be used to signify when to write more characters to the transmit FIFO. This results in the most efficient transfer times to the Smart Card
- Enable the transmitter by setting the CTRL[XMT_EN] bit
- Write the characters to be sent as response (max 16) to the transmit FIFO using the TX_BUF register

At this point, the EMVSIM module transmits the characters in the transmit FIFO. If more than 16 characters are to be sent, the transmit threshold interrupt is set when the threshold number of characters are remaining in the FIFO. The software can then write an additional number of characters to be sent without interrupting transmission to the Smart Card.

Once the transmission is complete, the EMVSIM module should be completely configured for standard operation with the T=0 Smart Card. The software can continue to service RDTF interrupts for received characters, and TDTF interrupts for transmitted characters.

65.6.3.2 Programming Considerations for T=1 Smart Cards

For type T=1 card, the software should adjust the following parameters according to the information in the ATR:

- Adjust the baud rate by changing the values of DIVISOR register based on the values of 'F' and 'D' received in the ATR
- Adjust the guard time between characters by changing the value of GETU in the TX_GETU register. Setting GETU to 0xFF configures the SIM transmitter for 11 ETU transmissions
- Disable NACK capability by clearing the CTRL[ONACK] and CTRL[ANACK] bits. T=1 cards do not allow NACKs
- Adjust the stop clock polarity by modifying the value of the PCSR[SCSP] bit
- Set Character Wait Time Counter Timeout Value to value specified in the ATR by using the CWT_VAL register
- Enable the Character Wait Time Counter Interrupt by clearing CWT_ERR_IM in INT_MASK register
- Enable the Character Wait Time Counter by setting the CTRL[CWT_EN] bit
- Enable CRC or LRC error checking according to the ATR information by setting either the CTRL[CRC_EN] bit or the CTRL[LRC_EN] bit. These bits should never be set at the same time!

For T=1 cards, the ATR is sent using a T=0 type of structure (12 ETU, no LRC or CRC). If a negotiation with the Smart Card is desired, the software sends a PPS response to the Smart Card. Otherwise, the protocol is initiated with a block transfer from the EMVSIM module. In order to send the response or the first block, the following steps should be performed:

- Set the desired transmit FIFO threshold level by writing the TDT[3:0] bits in the TX_THD register
- Clear all transmit interrupt flags in the TX_STATUS register by writing a one to the bits of the register
- Enable the transmit interrupts desired by clearing the mask bits in the INT_MASK register. If more than 16 characters are to be sent, it is suggested that the TDTF interrupt be used to signify when to write more characters to the transmit FIFO. This results in the most efficient transfer times to the Smart Card
- Enable the transmission of the error checking characters (LRC or CRC) by setting the CTRL[XMT_CRC_LRC] bit

NOTE

If the card supports PPS, the software may not be allowed to send the LRC/CRC information until the PPS exchange is completed. If so, do not set the XMT_CRC_LRC bit during the PPS exchange.

- Enable the transmitter by setting the CTRL[XMT_EN] bit
- Write the characters to be sent as response (max 16) to the transmit FIFO using the TX_BUF register

At this point, the EMVSIM module transmits the characters in the transmit FIFO. If more than 16 characters are to be sent, the transmit threshold interrupt is set when the threshold number of characters are remaining in the FIFO. The software can then write an additional number of characters to be sent without interrupting transmission to the Smart Card.

Once the transmission is complete, the EMVSIM module should be completely configured for standard operation with the T=1 Smart Card. The software can continue to service RDTF interrupts for received characters, and TDTF interrupts for transmitted characters.

65.7 Memory Map and Registers

65.7.1 EMVSIM register descriptions

The memory map comprises 32-bit aligned registers which can be accessed via 8-bit, 16-bit, or 32-bit accesses. Write access to reserved locations generate transfer error. Read access to reserved locations also generate transfer error and the read data bus show all 0s.The module not check for correctness of programmed values in the registers and software must ensure that correct values are being written.

65.7.1.1 EMVSIM memory map

EMVSIM1 base address: 4015_4000h

EMVSIM2 base address: 4015_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID Register (VER_ID)	32	RO	0000_0000
4	Parameter Register (PARAM)	32	RO	0000_1010
8	Clock Configuration Register (CLKCFG)	32	RW	0000_0000
C	Baud Rate Divisor Register (DIVISOR)	32	RW	0000_0174
10	Control Register (CTRL)	32	RW	0100_0006
14	Interrupt Mask Register (INT_MASK)	32	RW	0000_FFFF
18	Receiver Threshold Register (RX_THD)	32	RW	0000_0001
1C	Transmitter Threshold Register (TX_THD)	32	RW	0000_000F
20	Receive Status Register (RX_STATUS)	32	W1C	0000_0000
24	Transmitter Status Register (TX_STATUS)	32	W1C	0000_00B8
28	Port Control and Status Register (PCSR)	32	RW	0100_0000
2C	Receive Data Read Buffer (RX_BUF)	32	RO	0000_0000
30	Transmit Data Buffer (TX_BUF)	32	WORZ	0000_0000
34	Transmitter Guard ETU Value Register (TX_GETU)	32	RW	0000_0000
38	Character Wait Time Value Register (CWT_VAL)	32	RW	0000_FFFF
3C	Block Wait Time Value Register (BWT_VAL)	32	RW	FFFF_FFFF
40	Block Guard Time Value Register (BGT_VAL)	32	RW	0000_0000
44	General Purpose Counter 0 Timeout Value Register (GPCNT0_VAL)	32	RW	0000_FFFF
48	General Purpose Counter 1 Timeout Value (GPCNT1_VAL)	32	RW	0000_FFFF

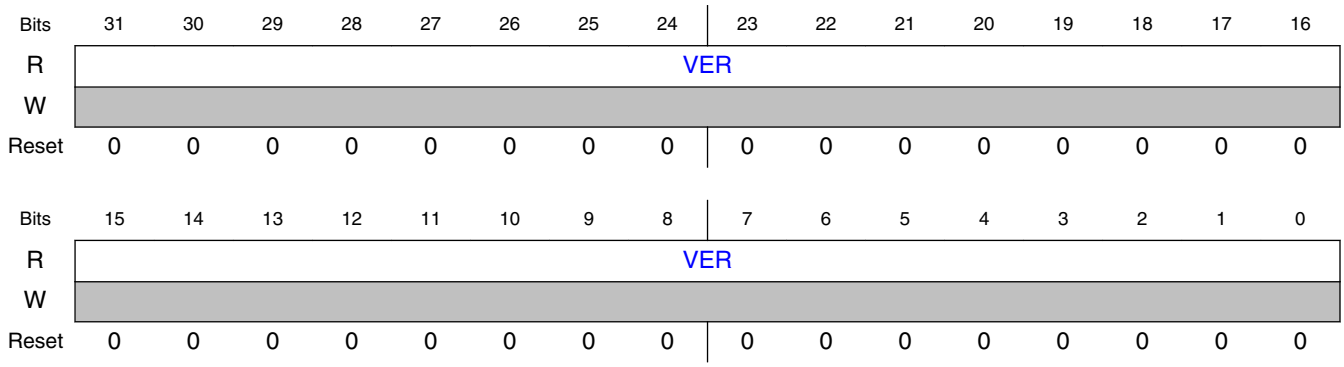
65.7.1.2 Version ID Register (VER_ID)

Version ID for the IP. It corresponds to the version of IP being used.

65.7.1.2.1 Offset

Register	Offset
VER_ID	0h

65.7.1.2.2 Diagram



65.7.1.2.3 Fields

Field	Description
31-0 VER	Version ID of the module EMVSIM Version Number used on the chip 31:16 - Major Version (example: 01.00) 15:0 - Minor Version (example: 01.00)

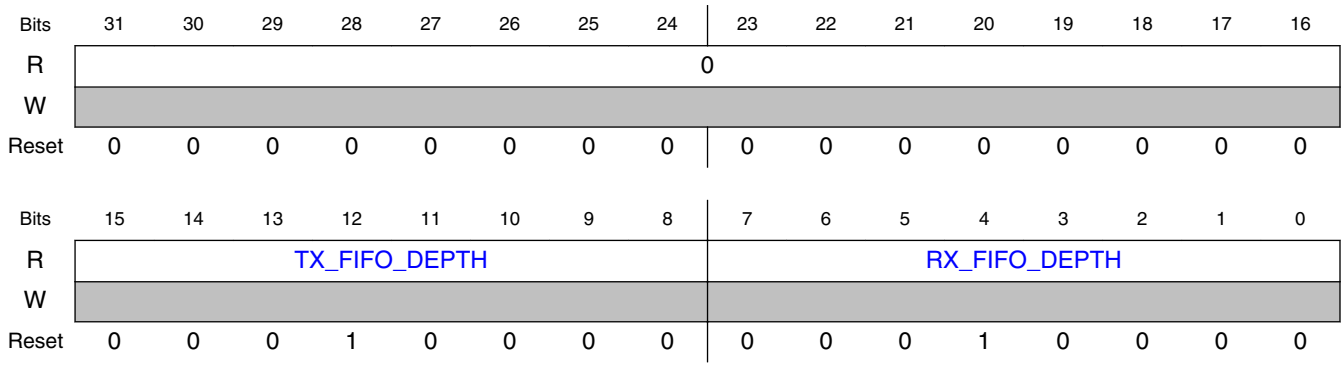
65.7.1.3 Parameter Register (PARAM)

This register provides details on the parameter settings that were used while including this module in the chip.

65.7.1.3.1 Offset

Register	Offset
PARAM	4h

65.7.1.3.2 Diagram



65.7.1.3.3 Fields

Field	Description
31-16 —	Reserved
15-8 TX_FIFO_DEPTH	Transmit FIFO Depth Value of parameter for Transmit FIFO Depth (in Bytes)
7-0 RX_FIFO_DEPTH	Receive FIFO Depth Value of parameter for Receive FIFO Depth (in Bytes)

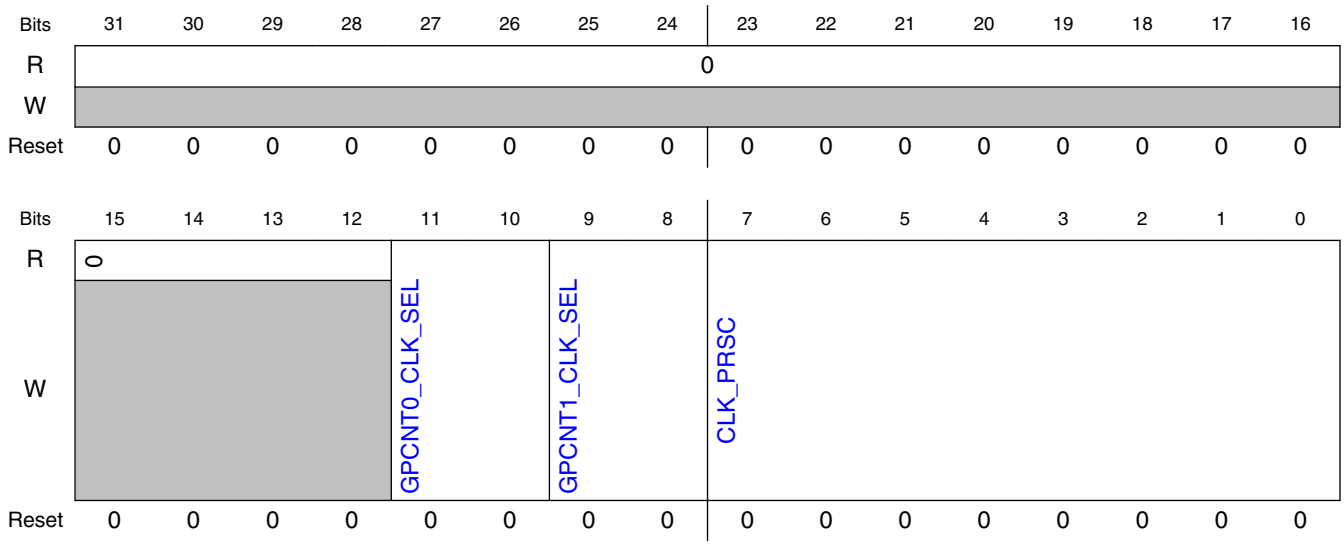
65.7.1.4 Clock Configuration Register (CLKCFG)

This register provides configuration details to enable the right clock frequency of clocks used by EMVSIM module.

65.7.1.4.1 Offset

Register	Offset
CLKCFG	8h

65.7.1.4.2 Diagram



65.7.1.4.3 Fields

Field	Description
31-12 —	Reserved
11-10 GPCNT0_CLK_SEL	General Purpose Counter 0 Clock Select Selects which clock source is used by EMVSIM Module general purpose counter 0. The only way to reset the counter is to set these bits to zero. The counter begins counting as soon as the clock input is selected and the clocks are active. These input clocks are enabled through other register bits of the EMVSIM module (KILL_CLOCK, RCV_EN, and XMT_EN). Counter is active while RCV_EN or XMT_EN are set. 00 - Disabled / Reset 01 - Card Clock 10 - Receive Clock 11 - ETU Clock (transmit clock)
9-8 GPCNT1_CLK_SEL	General Purpose Counter 1 Clock Select Selects which clock source is used by EMVSIM Module general purpose counter 1. The only way to reset the counter is to set these bits to zero. The counter begins counting as soon as the clock input is selected and the clocks are active. These input clocks are enabled through other register bits of the EMVSIM module (KILL_CLOCK, RCV_EN, and XMT_EN). Counter is active while RCV_EN or XMT_EN are set. 00 - Disabled / Reset 01 - Card Clock 10 - Receive Clock 11 - ETU Clock (transmit clock)
7-0 CLK_PRSC	Clock Prescaler Value

Field	Description
	The value written to this register determines the desired card clock frequency. The Card Clock frequency is the EMVSIM Protocol (or logic) clock divided by this prescaler value. The prescaler value should be updated only when card clock is disabled and transmitter and receiver are not operational. When CLK_PRSC = 0 or 1, the maximum prescaler value is used (0xFF). For all other values of CLK_PRSC, the prescaler value = CLK_PRSC.

65.7.1.5 Baud Rate Divisor Register (DIVISOR)

This register configures the divisor value to generate the baud clock which drives the card clock and also generate the transmit and receive clocks and respective ETUs.

65.7.1.5.1 Offset

Register	Offset
DIVISOR	Ch

65.7.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DIVISOR_VALUE								
W																
Reset	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0

65.7.1.5.3 Fields

Field	Description
31-9 —	Reserved
8-0 DIVISOR_VALUE	Divisor (F/D) Value The value written to this register is used to generate the ETU bit period that is used by the transmitter and receiver. The divisor value is the integer result of F/D value. 1 ETU = F / D * 1 / f

Field	Description
	• ETU is Elementary Time Unit, which is the nominal time to transmit a bit (0 or 1) • F is the clock rate conversion integer • D is the baud rate adjustment integer • f is the frequency of the communication clock used between the controller and the smartcard. The value in this register can be changed when no transaction is active with the smart card. When programming this field, the fractional part of F/D should be rounded off the nearest integer. 000000000-000000100 - Invalid. As per ISO 7816 specification, minimum value of F/D is 5 000000101-011111111 - For example, 101110100b = Divisor value for F = 372 and D = 1

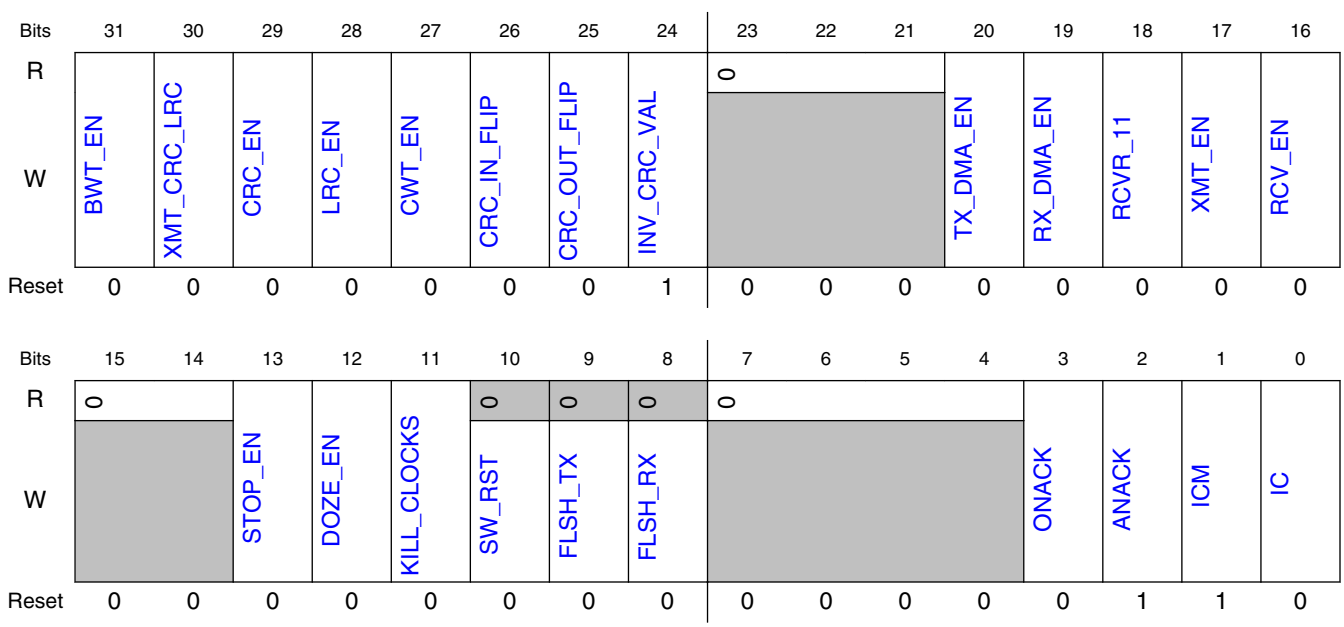
65.7.1.6 Control Register (CTRL)

This register provides EMVSIM configuration details.

65.7.1.6.1 Offset

Register	Offset
CTRL	10h

65.7.1.6.2 Diagram



65.7.1.6.3 Fields

Field	Description
31 BWT_EN	Block Wait Time Counter Enable Writing a '1' to this bit enables the BWT and BGT functions. The BWT and BGT functions can then be individually selected using the interrupt mask. 0 - Disable BWT, BGT Counters 1 - Enable BWT, BGT Counters
30 XMT_CRC_LRC	Transmit CRC or LRC Enable This bit specifies whether or not to transmit the redundancy checking data (LRC or CRC) at the end of a transmission (that is, when the FIFO becomes empty). 0 - No CRC or LRC value is transmitted 1 - Transmit LRC or CRC info when FIFO empties (whichever is enabled)
29 CRC_EN	CRC Enable This bit enables the calculation of the 16-bit CRC value for both receiver and transmitter. The result of the calculation is continuously compared to the expected remainder and reflected in the RX_STATUS[CRC_OK] bit. Clearing this bit resets the current CRC residual value in the EMVSIM hardware. 0 - 16-bit Cyclic Redundancy Checking disabled 1 - 16-bit Cyclic Redundancy Checking enabled
28 LRC_EN	LRC Enable This bit enables the calculation of the 8-bit LRC value for both receiver and transmitter. The result of the calculation is continuously compared to zero and reflected in the RX_STATUS[LRC_OK] bit. Clearing this bit resets the current LRC value in the EMVSIM hardware. 0 - 8-bit Linear Redundancy Checking disabled 1 - 8-bit Linear Redundancy Checking enabled
27 CWT_EN	Character Wait Time Counter Enable Enables the character wait time counter. Clearing this bit resets the counter to zero. 0 - Character Wait time Counter is disabled 1 - Character Wait time counter is enabled
26 CRC_IN_FLIP	CRC Input Byte's Bit Reversal or Flip Control This bit control whether the bits in the CRC input byte are reversed before CRC calculation or not. When set to '1', the bits within the input byte for CRC changes from 7:0 to 0:7. For CCITT CRC calculation, this bit should be set to '1'. For any other CRC using same polynomial, this bit can be set accordingly. 0 - Bits in the input byte are not reversed (i.e. 7:0 remain 7:0) before the CRC calculation 1 - Bits in the input byte are reversed (i.e. 7:0 becomes 0:7) before CRC calculation
25 CRC_OUT_FLIP	CRC Output Value Bit Reversal or Flip This bit control whether the bits in the CRC output bytes are reversed or not. When set to '1', the bits within the two bytes for CRC Output changes from 15:0 to {8:15,0:7}. For CCITT CRC calculation, this bit should be set to '1'. For any other CRC using same polynomial, this bit can be set accordingly. 0 - Bits within the CRC output bytes are not reversed i.e. 15:0 remains 15:0 1 - Bits within the CRC output bytes are reversed i.e. 15:0 becomes {8:15,0:7}
24 INV_CRC_VAL	Invert bits in the CRC Output Value

Table continues on the next page...

Field	Description
	This bit control whether the bits within the CRC Output value are inverted (1's complement) or not. For CCITT CRC calculation, this bit should be set to '1'. For any other CRC using same polynomial, this bit can be set accordingly. 0 - Bits in CRC Output value are not inverted. 1 - Bits in CRC Output value are inverted.
23-21 —	Reserved
20 TX_DMA_EN	Transmit DMA Enable Enables assertion of DMA write request when Transmit FIFO is empty. Request is held asserted till Transmit FIFO reaches the programmed data threshold value. Transmit Data Threshold Interrupt is not generated when this bit is asserted. 0 - No DMA Write Request asserted for Transmitter 1 - DMA Write Request asserted for Transmitter
19 RX_DMA_EN	Receive DMA Enable Enables assertion of DMA read request when Receive FIFO reaches the programmed data threshold value. Request is held asserted till all the programmed threshold bytes are read out. Receiver Data Interrupt is not be generated when this bit is asserted. 0 - No DMA Read Request asserted for Receiver 1 - DMA Read Request asserted for Receiver
18 RCVR_11	Receiver 11 ETU Mode Enable Used to configure the EMVSIM module receiver for 11 ETU operation (that is, 1 Stop bit). This bit is provided for support of T=1 cards. 0 - Receiver configured for 12 ETU operation mode 1 - Receiver configured for 11 ETU operation mode
17 XMT_EN	Transmitter Enable Used to enable/disable the EMVSIM transmitter block. It can be set to 0 during the auto power down sequence. **Note:** Setting this bit (transition from 0 to 1) resets the CRC and LRC values. 0 - EMVSIM Transmitter disabled 1 - EMVSIM Transmitter enabled
16 RCV_EN	Receiver Enable Used to enable/disable the EMVSIM receiver block. Once the transmitter has completed its operation, the software must enable the receiver using this bit. It can be set to 0 during the auto power down sequence. 0 - EMVSIM Receiver disabled 1 - EMVSIM Receiver enabled
15-14 —	Reserved
13 STOP_EN	STOP Enable Used to configure the operation of the EMVSIM module when a processor STOP instruction is executed. This bit is added to provide support for Smart Cards that do not allow the Smart Card clock to be stopped while power is applied. 0 - STOP instruction shuts down all EMVSIM clocks

Table continues on the next page...

Field	Description
	1 - STOP instruction shuts down all clocks except for the Smart Card Clock (SCK) (clock provided to Smart Card)
12 DOZE_EN	Doze Enable Used to configure the operation of the EMVSIM module when a processor DOZE instruction is executed. 0 - DOZE instruction gates all internal EMVSIM clocks as well as the Smart Card clock when the transmit FIFO is empty 1 - DOZE instruction has no effect on EMVSIM module
11 KILL_CLOCKS	Kill all internal clocks Used to enable/disable the clock input to the EMVSIM module. This bit gates all clocks including the Smart Card clock regardless of the state of the STOP bit described above. **Note:** This bit has no effect on the register read write clock. Only EMVSIM logic clock is gated. 0 - EMVSIM input clock enabled 1 - EMVSIM input clock is disabled
10 SW_RST	Software Reset Bit Used to reset the entire EMVSIM module. This acts the same as a hardware reset for the EMVSIM module. This bit is self-clearing and always reads 0. **Note:** Software should allow a minimum of 4 Protocol clock cycles before attempting to access the EMVSIM module after a software reset. 0 - EMVSIM Normal operation 1 - EMVSIM held in Reset
9 FLSH_TX	Flush Transmitter Bit This bit operates as an EMVSIM transmitter reset. The receive portion of the EMVSIM module is not affected. This bit clears automatically and always reads 0. 0 - EMVSIM Transmitter normal operation 1 - EMVSIM Transmitter held in Reset
8 FLSH_RX	Flush Receiver Bit This bit operates as an EMVSIM receiver reset. The transmit portion of the EMVSIM module is not affected. This bits clears automatically and always reads 0. 0 - EMVSIM Receiver normal operation 1 - EMVSIM Receiver held in Reset
7-4 —	Reserved
3 ONACK	Overrun NACK Enable Enables NACK generation when Rx FIFO is full and another message is ready for writing into the FIFO. 0 - NACK generation on overrun is disabled 1 - NACK generation on overrun is enabled
2 ANACK	Auto NACK Enable Enables NACK generation for parity errors in receieved messages or when invalid initial characters are received in ICM mode. 0 - NACK generation on errors disabled 1 - NACK generation on errors enabled
1	Initial Character Mode

Table continues on the next page...

Field	Description
ICM	Enables initial character mode. It is automatically cleared by hardware once a valid initial character is received. 0 - Initial Character Mode disabled 1 - Initial Character Mode enabled
0 IC	Inverse Convention Used to configure the EMVSIM to use either inverse convention or direct convention for its data format. The IC bit can be controlled by software, but it is normally set by hardware as a result of the interpretation of the initial character when in ICM mode. 0 - Direction convention transfers enabled 1 - Inverse convention transfers enabled

65.7.1.7 Interrupt Mask Register (INT_MASK)

This register provides interrupt masking configuration details.

65.7.1.7.1 Offset

Register	Offset
INT_MASK	14h

65.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PEF_IM	RX_DATA_IM	GPCNT1_IM	BGT_ERR_IM	BWT_ERR_IM	RNACK_IM	CWT_ERR_IM	GPCNT0_IM	TDT_IM	TFF_IM	TNACK_IM	TFE_IM	ETC_IM	RFO_IM	TC_IM	RDT_IM
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

65.7.1.7.3 Fields

Field	Description
31-16 —	Reserved
15 PEF_IM	Parity Error Interrupt Mask Used to enable/disable the ability of the RX_STATUS[PEF] flag to generate EMVSIM interrupts. 0 - PEF interrupt enabled 1 - PEF interrupt masked
14 RX_DATA_IM	Receive Data Interrupt Mask Used to enable/disable the ability of the RX_STATUS[RX_DATA] flag to generate EMVSIM interrupts. 0 - RX_DATA interrupt enabled 1 - RX_DATA interrupt masked
13 GPCNT1_IM	General Purpose Counter 1 Timeout Interrupt Mask Used to enable/disable the ability of the TX_STATUS[GPCNT1_TO] flag to generate EMVSIM interrupts. 0 - GPCNT1_TO interrupt enabled 1 - GPCNT1_TO interrupt masked
12 BGT_ERR_IM	Block Guard Time Error Interrupt Used to enable/disable the ability of the RX_STATUS[BGT_ERR] flag to generate EMVSIM interrupts. 0 - BGT_ERR interrupt enabled 1 - BGT_ERR interrupt masked
11 BWT_ERR_IM	Block Wait Time Error Interrupt Mask Used to enable/disable the ability of the RX_STATUS[BWT_ERR] flag to generate EMVSIM interrupts. 0 - BWT_ERR interrupt enabled 1 - BWT_ERR interrupt masked
10 RNACK_IM	Receiver NACK Threshold Interrupt Mask Used to enable/disable the ability of the RX_STATUS[RTE] flag to generate EMVSIM interrupts. 0 - RTE interrupt enabled 1 - RTE interrupt masked
9 CWT_ERR_IM	Character Wait Time Error Interrupt Mask Used to enable/disable the ability of the RX_STATUS[CWT_ERR] flag to generate EMVSIM interrupts. 0 - CWT_ERR interrupt enabled 1 - CWT_ERR interrupt masked
8 GPCNT0_IM	General Purpose Timer 0 Timeout Interrupt Mask Used to enable/disable the ability of the TX_STATUS[GPCNT0_TO] flag to generate EMVSIM interrupts. 0 - GPCNT0_TO interrupt enabled 1 - GPCNT0_TO interrupt masked
7 TDT_IM	Transmit Data Threshold Interrupt Mask Used to enable/disable the ability of the TX_STATUS[TDTF] flag to generate EMVSIM interrupts. 0 - TDTF interrupt enabled

Table continues on the next page...

Field	Description
	1 - TDTF interrupt masked
6 TFF_IM	Transmit FIFO Full Interrupt Mask Used to enable/disable the ability of the TX_STATUS[TFF] flag to generate EMVSIM interrupts. 0 - TFF interrupt enabled 1 - TFF interrupt masked
5 TNACK_IM	Transmit NACK Threshold Interrupt Mask Used to enable/disable the ability of the TX_STATUS[TNTE] flag to generate EMVSIM interrupts. 0 - TNTE interrupt enabled 1 - TNTE interrupt masked
4 TFE_IM	Transmit FIFO Empty Interrupt Mask Used to enable/disable the ability of the TX_STATUS[TFE] flag to generate EMVSIM interrupts. 0 - TFE interrupt enabled 1 - TFE interrupt masked
3 ETC_IM	Early Transmit Complete Interrupt Mask Used to enable/disable the ability of the TX_STATUS[ETC] flag to generate EMVSIM interrupts. 0 - ETC interrupt enabled 1 - ETC interrupt masked
2 RFO_IM	Receive FIFO Overflow Interrupt Mask Used to enable/disable the ability of the RX_STATUS[RFO] flag to generate EMVSIM interrupts. 0 - RFO interrupt enabled 1 - RFO interrupt masked
1 TC_IM	Transmit Complete Interrupt Mask Used to enable/disable the ability of the TX_STATUS[TCF] flag to generate EMVSIM interrupts. 0 - TCF interrupt enabled 1 - TCF interrupt masked
0 RDT_IM	Receive Data Threshold Interrupt Mask Used to enable/disable the ability of the RX_STATUS[RDTF] flag to generate EMVSIM interrupts. 0 - RDTF interrupt enabled 1 - RDTF interrupt masked

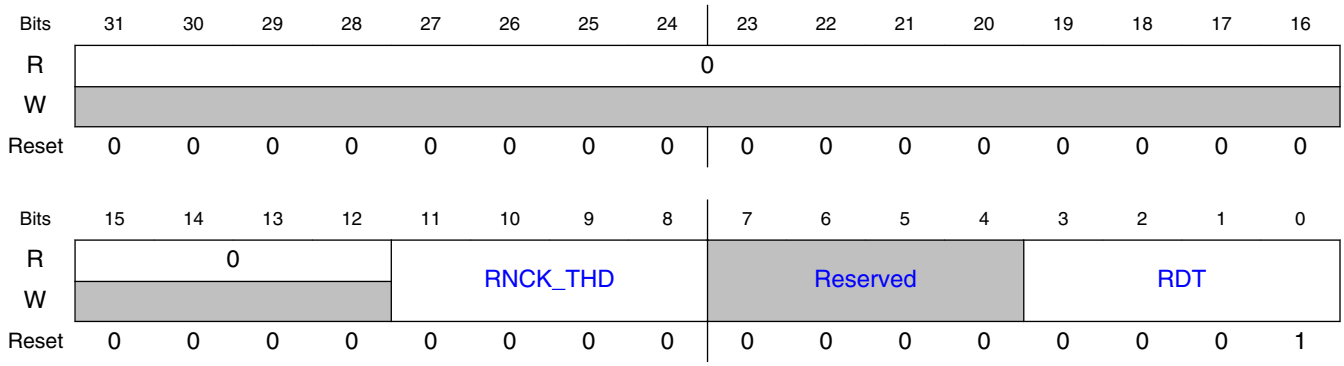
65.7.1.8 Receiver Threshold Register (RX_THD)

This register provides receiver threshold configuration details.

65.7.1.8.1 Offset

Register	Offset
RX_THD	18h

65.7.1.8.2 Diagram



65.7.1.8.3 Fields

Field	Description
31-12 —	Reserved
11-8 RNCK_THD	Receiver NACK Threshold Value Used to specify the number of consecutive NACK's transmitted by the EMVSIM module, for a given character, before the receive threshold error (RTE) flag is triggered. A value of 0 indicates that RTE is never set. When a valid character is received by the EMV SIM, the internal counter keeping track of the NACK count resets to zero for the subsequent byte being received. If the ANACK bit is clear in the CTRL register, RNCK_THD has no effect. 0000 - Zero Threshold. RTE is not set 0001 - RTE is triggered after 1 nack is transmitted 0010 - RTE is triggered after 2 nacks are transmitted 0011 - RTE is triggered after 3 nacks are transmitted ... 1111 - RTE is triggered after 15 nacks are transmitted
7-4 —	Reserved
3-0 RDT	Receiver Data Threshold Value Determines the number of bytes that must exist in the Receive FIFO to trigger the receive data threshold interrupt flag (RDTF). Value must not exceed the total bytes that can be stored in the Receive FIFO.

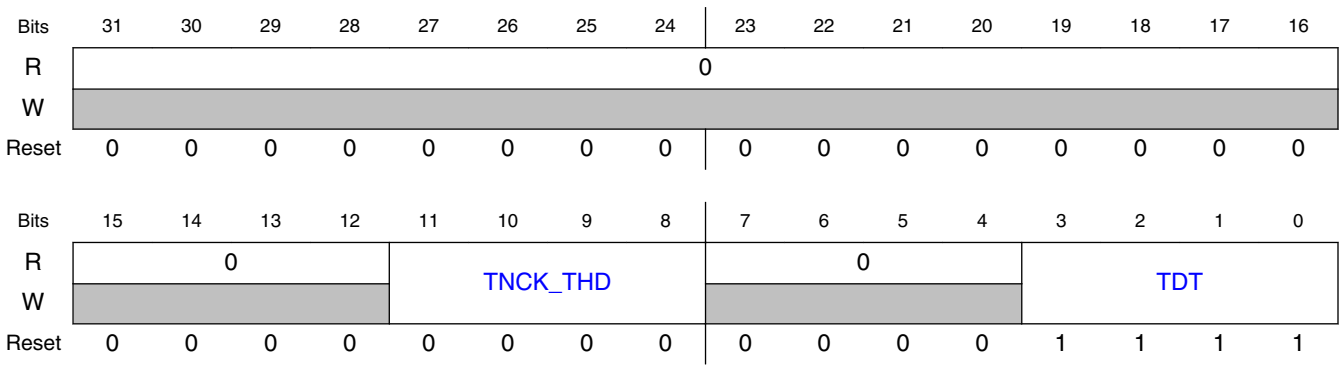
65.7.1.9 Transmitter Threshold Register (TX_THD)

This register provides transmitter threshold configuration details.

65.7.1.9.1 Offset

Register	Offset
TX_THD	1Ch

65.7.1.9.2 Diagram



65.7.1.9.3 Fields

Field	Description
31-12 —	Reserved
11-8 TNCK_THD	Transmitter NACK Threshold Value Used to set the NACK threshold for the transmitter. Once the threshold number set by TNCK_THD has been reached for the current byte being transmitted, the error flag TNTE in the TX_STATUS register is set. Setting of TNTE causes the remaining transmissions queued in the transmit FIFO to be aborted and no more transmissions to occur until software clears TNTE. To trigger TNTE, a given byte being transmitted must reach the TNCK_THD threshold itself. Transmit NACKs accumulated on one byte are not carried over to the next. 0000 - TNTE is never set; retransmission after NACK reception is disabled 0001 - TNTE is set after 1 nack is received; 0 retransmissions occurs 0010 - TNTE is set after 2 nacks are received; at most 1 retransmission occurs 0011 - TNTE is set after 3 nacks are received; at most 2 retransmissions occurs ... 1111 - TNTE is set after 15 nacks are received; at most 14 retransmissions occurs
7-4 —	Reserved
3-0 TDT	Transmitter Data Threshold Value Used to set the threshold value for the Transmit FIFO at which the TX_STATUS[TDTF] bit is set. When the number of bytes in the Transmit FIFO is less than or equal to TDT, TDTF is set.

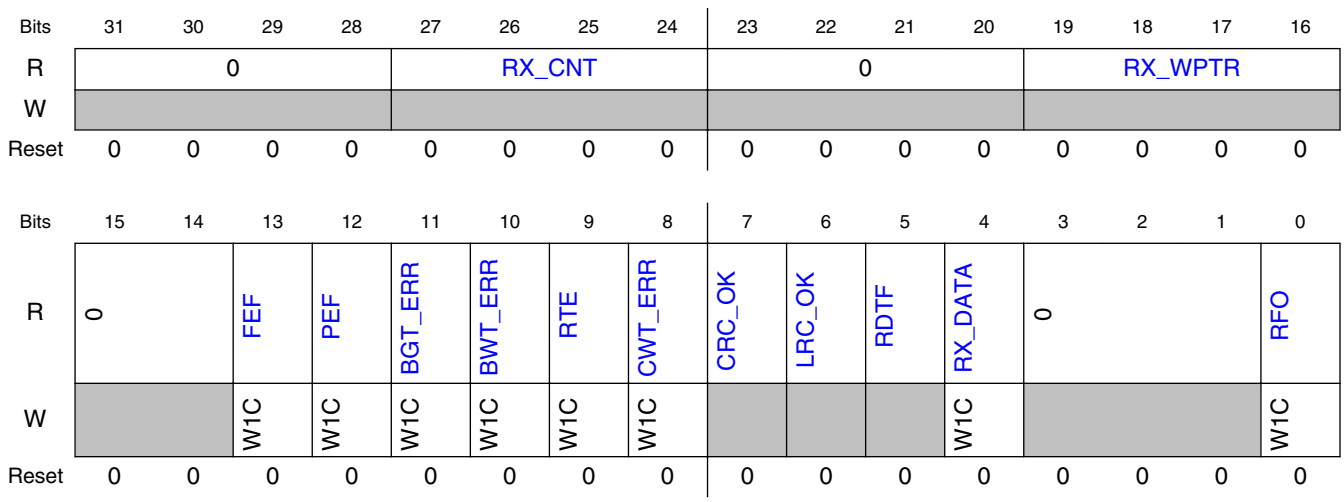
65.7.1.10 Receive Status Register (RX_STATUS)

This register provides receive status details.

65.7.1.10.1 Offset

Register	Offset
RX_STATUS	20h

65.7.1.10.2 Diagram



65.7.1.10.3 Fields

Field	Description
31-28 —	Reserved
27-24 RX_CNT	Receive FIFO Byte Count These bits indicate the number of bytes stored in the receive FIFO. 0000 - FIFO is emtpy
23-20 —	Reserved
19-16 RX_WPTR	Receive FIFO Write Pointer Value Value of write pointer of Receive FIFO
15-14 —	Reserved
13	Frame Error Flag

Table continues on the next page...

Field	Description
FEF	Used to indicate if the current received byte did not have proper STOP bits. This does not cause a NACK transmission. This bit does not cause an interrupt assertion. It is simply an indication that a framing error has occured. 0 - No frame error detected 1 - Frame error detected
12 PEF	Parity Error Flag Used to indicate if a received byte had a parity error or not. A parity error causes transmission of NACK if ANACK bit is set in CTRL register and PEF bit is not asserted. When ANACK is *not* set, PEF is asserted. This bit also does not assert when the ICM bit is set; however parity is checked in ICM = 1 mode. It is simply an indication that a parity error has occured. 0 - No parity error detected 1 - Parity error detected
11 BGT_ERR	Block Guard Time Error Flag Used to indicate if the block guard time was too small. The threshold is set by the BGT_VAL register. 0 - Block guard time was sufficient 1 - Block guard time was too small
10 BWT_ERR	Block Wait Time Error Flag Used to indicate if the block wait time has been exceeded. The threshold is set by the BWT_VAL registers. 0 - Block wait time not exceeded 1 - Block wait time was exceeded
9 RTE	Received NACK Threshold Error Flag Used to indicate whether the number of consecutive NACK's generated by the EMVSIM module in response to receive parity errors, for the byte being received, equals the value programmed in the RX_THD[RNCK_THD]. This bit is never set unless the ANACK bit is set in the CTRL register. The PCSR[SAPD] bit must be set to enable the threshold error to trigger the auto power down sequence. RTE is a write once clear bit. Clearing this bit also resets the internal counter for consecutive NACKs being transmitted for a given byte. 0 - Number of NACKs generated by the receiver is less than the value programmed in RNCK_THD 1 - Number of NACKs generated by the receiver is equal to the value programmed in RNCK_THD
8 CWT_ERR	Character Wait Time Error Flag Used to indicate when the time between received characters is equal to or greater than the value programmed in the CWT_VAL register. 0 - No CWT violation has occurred 1 - Time between two consecutive characters has exceeded the value in CWT_VAL.
7 CRC_OK	CRC Check OK Flag Used to indicate when the calculated 16-bit CRC value matches the expected value for the current input data stream. The value is calculated across all received characters from the point the CRC_EN bit is set in the CTRL register. The current CRC residual can be reset by three mechanisms: • Clear CTRL[CRC_EN] bit • Set CTRL[XMT_EN] bit • Automatically by hardware when ETC flag is set at the end of a transmission. 0 - Current CRC value does not match remainder. 1 - Current calculated CRC value matches the expected result.

Table continues on the next page...

Field	Description
6 LRC_OK	LRC Check OK Flag Used to indicate when the calculated 8-bit LRC value is correct for the current input data stream. The value is calculated across all received characters from the point the LRC_EN bit is set in the CTRL register. The current LRC residual can be reset by three mechanisms: • Clear CTRL[LRC_EN] bit • Set CTRL[XMT_EN] bit • Automatically by hardware when ETC flag is set at the end of a transmission. 0 - Current LRC value does not match remainder. 1 - Current calculated LRC value matches the expected result (i.e. zero).
5 RDTF	Receive Data Threshold Interrupt Flag Interrupt flag asserted when total bytes in the Receive FIFO equal or is greater than the programmed receive threshold RDT. The RDTF flag is set any time the number of unread bytes in the receive FIFO is equal to or greater than the value set by RDT. The flag can be cleared by reading enough bytes out of the receive FIFO so as to bring the number of bytes left in the FIFO below the RDT level. Another way to clear the flag is to set the RDT level higher than the number of unread bytes currently in the FIFO. The RDTF flag creates an interrupt if the INT_MASK[RDT_IM] bit is cleared. 0 - Number of unread bytes in receive FIFO less than the value set by RDT 1 - Number of unread bytes in receive FIFO greater or than equal to value set by RDT.
4 RX_DATA	Receive Data Interrupt Flag Interrupt asserted when a new data byte is received and entered into the Receive FIFO. 0 - No new byte is received 1 - New byte is received ans stored in Receive FIFO
3-1 —	Reserved
0 RFO	Receive FIFO Overflow Flag Used to indicate that the EMVSIM was unable to store received data due to already unread bytes in the FIFO (FIFO full or almost full). It does not necessarily indicate that data has been lost. If the CTRL[ONACK] control bit is set, there is a NACK pulse generated on bytes that would otherwise cause a loss of data due to a full FIFO. These bytes should be retransmitted by the Smart Card which implies that no data has actually been lost. In this case, the RFO flag is just an indicator that this situation has occurred which may be helpful in system debug. For the case where ONACK is not set, a set RFO flag does indicate a loss of data since all bytes received with the OEF flag set is indeed lost (including the byte that caused the bit to be set). The RFO flag causes an interrupt if the INT_MASK[RFO_IM] bit is enabled (RFO_IM=0). The RFO flag is a write-one-to-clear bit. 0 - No overrun error has occurred 1 - A byte was received when the received FIFO was already full

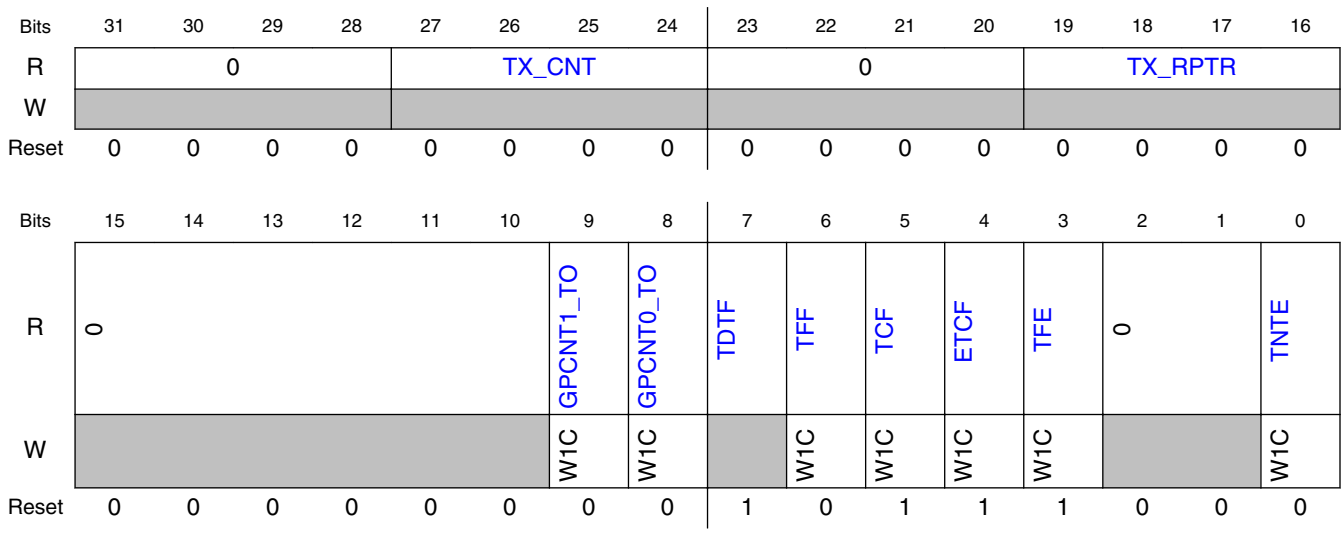
65.7.1.11 Transmitter Status Register (TX_STATUS)

This register provides transmitter status details.

65.7.1.11.1 Offset

Register	Offset
TX_STATUS	24h

65.7.1.11.2 Diagram



65.7.1.11.3 Fields

Field	Description
31-28 —	Reserved
27-24 TX_CNT	Transmit FIFO Byte Count These bits indicate the number of bytes stored in the transmit FIFO. 0000 - FIFO is emtpy
23-20 —	Reserved
19-16 TX_RPTR	Transmit FIFO Read Pointer Value of the read pointer of transmit FIFO.
15-10 —	Reserved
9 GPCNT1_TO	General Purpose Counter 1 Timeout Flag Used to indicate when the General Purpose Counter 1 has reached the GPCNT1 value in the GPCNT1_VAL register. 0 - GPCNT1 time not reached, or bit has been cleared.

Table continues on the next page...

Field	Description
	1 - General Purpose counter has reached the GPCNT1 value
8 GPCNT0_TO	General Purpose Counter 0 Timeout Flag Used to indicate when the General Purpose Counter 0 has reached the GPCNT0 value in the GPCNT0_VAL register. 0 - GPCNT0 time not reached, or bit has been cleared. 1 - General Purpose counter has reached the GPCNT0 value
7 TDTF	Transmit Data Threshold Flag Interrupt flag asserted when total bytes in the Transmit FIFO is less than or equal to the programmed transmit data threshold TDT. The TDTF flag is set any time the number of bytes in the transmit FIFO is less than or equal to the value set by TDT. The flag can be cleared by writing enough bytes into the transmit FIFO so as to take the number of bytes in the FIFO above the TDT level. Another way to clear the flag is to set the TDT level lower than the number of bytes currently in the FIFO.The TDTF flag creates an interrupt if the INT_MASK[TDT_IM] bit is cleared. 0 - Number of bytes in FIFO is greater than TDT, or bit has been cleared 1 - Number of bytes in FIFO is less than or equal to TDT
6 TFF	Transmit FIFO Full Flag Used to indicate when the Transmit FIFO has been written with maximum number of bytes that it can store. Subsequent writes may get ignored. The TFF bit can only be cleared by writing 1 to this bit or setting the CTRL[SW_RST] bit. 0 - Transmit FIFO Full condition has not occurred 1 - A Transmit FIFO Full condition has occurred
5 TCF	Transmit Complete Flag Used to indicate whether the EMVSIM transmitter is ready for a new transmission. The TC flag becomes set when the guard time has expired after the last byte in the transmit FIFO (if XMT_CRC_LRC = 0) or the LRC or CRC byte (if XMT_CRC_LRC bit = 1) has been transmitted. The TC flag creates an interrupt if TC_IM in the INT_MASK register is low. The TC bit is a write-one-to-clear bit. 0 - Transmit pending or in progress 1 - Transmit complete
4 ETCF	Early Transmit Complete Flag Used to indicate that the EMVSIM transmitter has finished sending the last byte in transmit FIFO (if XMT_CRC_LRC = 0) or finished sending the LRC or CRC byte (if XMT_CRC_LRC = 1). This bit differs from the TC bit in that it is set before the guard time of the last byte has elapsed. The ETC flag creates an interrupt if ETC_IM in the INT_MASK register is low. The ETC bit is a write-one-to-clear bit. 0 - Transmit pending or in progress 1 - Transmit complete
3 TFE	Transmit FIFO Empty Flag Used to indicate that the EMVSIM transmit FIFO has been emptied. This bit is set when the last byte in the transmit FIFO has been transferred out of the EMVSIM transmitter to the Smart Card successfully. TFE is also set when the TNTE flag is set. The TFE flag creates an interrupt if TFE_IM in the INT_MASK register is low. The TFE bit is a write-one-to-clear bit. 0 - Transmit FIFO is not empty 1 - Transmit FIFO is empty
2-1 —	Reserved
0	Transmit NACK Threshold Error Flag

Field	Description
TNTE	Used to indicate the transmit NACK threshold has been reached. When TNTE is high, no further transmissions are done until the TNTE flag is cleared. Any data transmissions still pending in the transmit FIFO are aborted, and the TC, ETC, and TFE flags are set. The TNTE flag creates an interrupt if TNACK_IM in the INT_MASK register is low. The TNTE bit is a write-one-to-clear bit. 0 - Transmit NACK threshold has not been reached 1 - Transmit NACK threshold reached; transmitter frozen

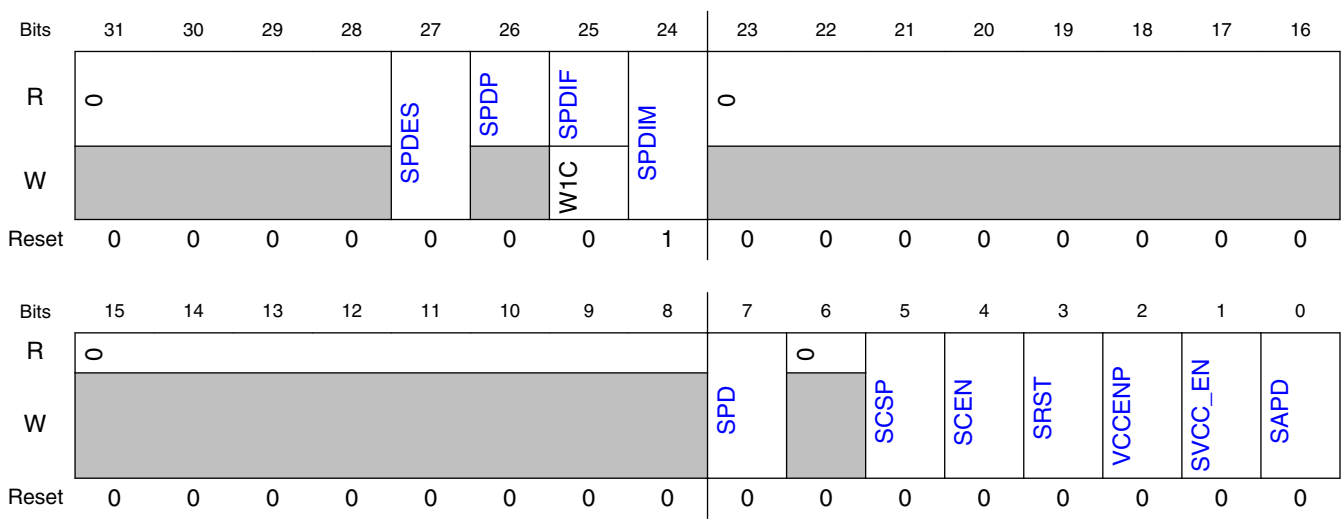
65.7.1.12 Port Control and Status Register (PCSR)

This register provides control and status details related to the smart card port.

65.7.1.12.1 Offset

Register	Offset
PCSR	28h

65.7.1.12.2 Diagram



65.7.1.12.3 Fields

Field	Description
31-28 —	Reserved

Table continues on the next page...

Field	Description
27 SPDES	SIM Presence Detect Edge Select Controls which edge of the Smart Card Presence Detect pin is used to detect the presence of the Smart Card. 0 - Falling edge on the pin 1 - Rising edge on the pin
26 SPDP	Smart Card Presence Detect Pin Status This bit reflects the state of the Smart Card Presence Detect pin. It is not a latched register bit, but instead a synchronized version of the state of the pin itself. 0 - SIM Presence Detect pin is logic low 1 - SIM Presence Detectpin is logic high
25 SPDIF	Smart Card Presence Detect Interrupt Flag Status flag to indicate that an insertion or removal of a Smart Card has been detected on port. Can create an interrupt to the MCU if SPDIM is low and/or initiate the auto power down sequence if SAPD bit is '1'. Write a '1' to this bit to clear. 0 - No insertion or removal of Smart Card detected on Port 1 - Insertion or removal of Smart Card detected on Port
24 SPDIM	Smart Card Presence Detect Interrupt Mask Interrupt mask for the presence detect interrupt flag (SPDIF). 0 - SIM presence detect interrupt is enabled 1 - SIM presence detect interrupt is masked
23-8 —	Reserved
7 SPD	Auto Power Down Control Writing a '1' to this location starts the autopower down sequence provided the auto power down feature is enabled by writing '1' to SAPD bit of this register. The SPD bit autoclears when power down is complete. 0 - No effect 1 - Start Auto Powerdown or Power Down is in progress
6 —	Reserved
5 SCSP	Smart Card Clock Stop Polarity Used to control the polarity of the idle EMVSIM clock when the clock is disabled by SCEN. 0 - Clock is logic 0 when stopped by SCEN 1 - Clock is logic 1 when stopped by SCEN
4 SCEN	Clock Enable for Smart Card Used to enable/disable the clock to the Smart Card. It can be forced low by hardware during the auto power down sequence. 0 - Smart Card Clock Disabled 1 - Smart Card Clock Enabled
3 SRST	Reset to Smart Card Used to control state of reset line to the Smart Card. It can be forced low by hardware during the auto power down sequence. Smart Card reset signals are active low.

Table continues on the next page...

Field	Description
	0 - Smart Card Reset is asserted 1 - Smart Card Reset is de-asserted
2 VCCENP	VCC Enable Polarity Control Used to control the polarity of the SVEN output pad via the SVCC_EN bit. When set to '1', this bit inverts the value of SVCC_EN bit of this register and output to SVEN output pad of device. 0 - SVCC_EN is active high. Polarity of SVCC_EN is unchanged. 1 - SVCC_EN is active low. Polarity of SVCC_EN is inverted.
1 SVCC_EN	Vcc Enable for Smart Card Used to control the state of the SVEN pin on Smart Card port. The SVEN pin controls the Smart Card Vcc enable in the power management chip. It can be forced low by hardware during the auto power down sequence. This bit is XORed with the VCCENP bit to control the polarity of the SVEN output pad. 0 - Smart Card Voltage disabled 1 - Smart Card Voltage enabled
0 SAPD	Auto Power Down Enable Used to enable/disable the auto power down feature. This bit must be set prior to the auto power conditions are met. This bit returns to 0 at the end of the auto power down sequence. This bit controls the auto power down function which can be initiated by: • Setting the SPD bit in this register by software • Assertion of the RX_STATUS[RTE] bit • Assertion of the SPDIF bit when the interrupt is used for detecting card removal Manual Power Down via software by writing to the SCEN, SRST, and SVCC_EN bits in PCSR register can still be carried out irrespective of this bit. 0 - Auto power down disabled 1 - Auto power down enabled

65.7.1.13 Receive Data Read Buffer (RX_BUF)

This register provides the byte value from the top of the receive FIFO.

65.7.1.13.1 Offset

Register	Offset
RX_BUF	2Ch

65.7.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RX_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

65.7.1.13.3 Fields

Field	Description
31-8 —	Reserved
7-0 RX_BYTE	Receive Data Byte Read Provides the byte value from the top of the receive FIFO. Each read access to this register increments the read pointer of the Receive FIFO.

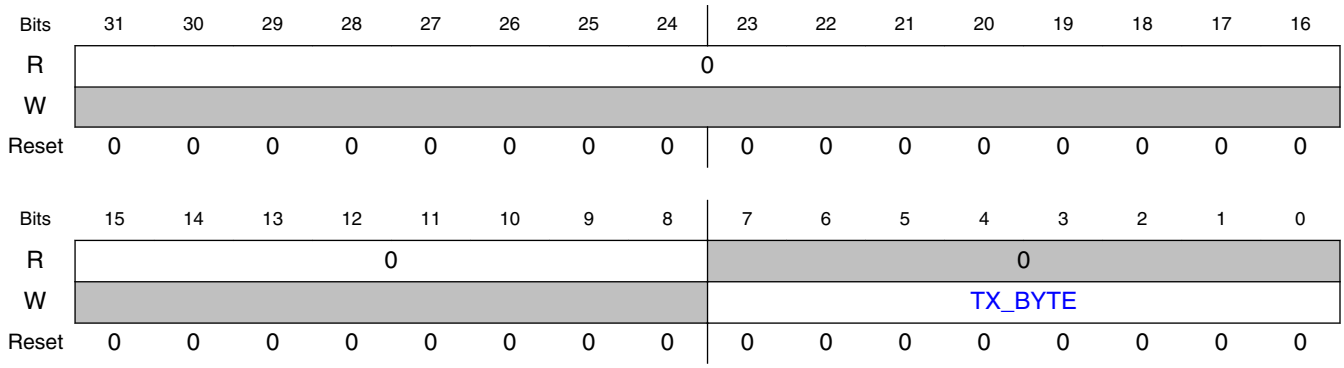
65.7.1.14 Transmit Data Buffer (TX_BUF)

This register is written to fill the transmit FIFO with the bytes to be transmitted.

65.7.1.14.1 Offset

Register	Offset
TX_BUF	30h

65.7.1.14.2 Diagram



65.7.1.14.3 Fields

Field	Description
31-8 —	Reserved
7-0 TX_BYTE	Transmit Data Byte Write to this register to fill the transmit FIFO with the bytes to be transmitted. The Tx FIFO can be written to only when the transmitter is enabled. Writing to this register while transmitter is disabled leads to ignoring of all write accesses to this register. Reads to this register returns zeros. **NOTE:** Writing more data to the transmit FIFO than it can hold, causes a Transmit FIFO Full (TFF) error.

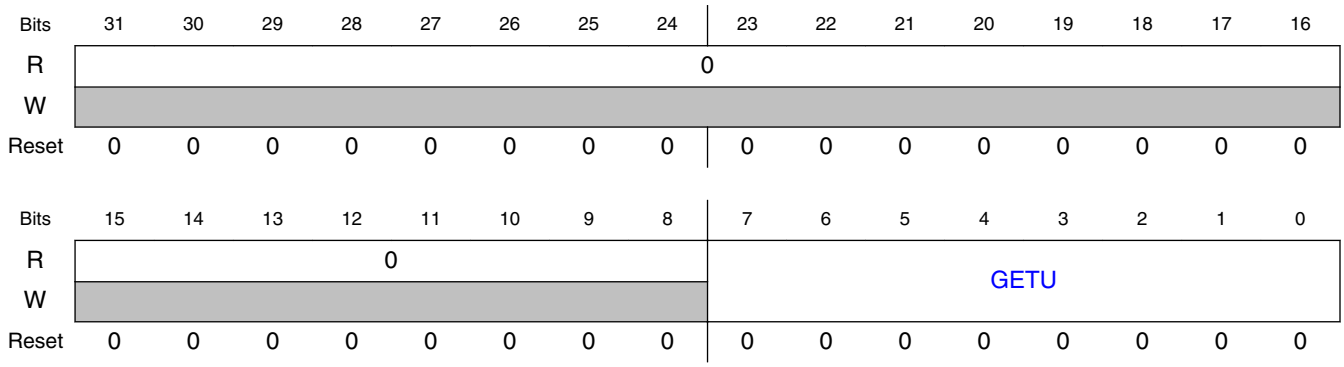
65.7.1.15 Transmitter Guard ETU Value Register (TX_GETU)

This register is used to control the number of additional Elementary Time Units (ETUs) inserted between bytes transmitted by the EMV SIM transmitter.

65.7.1.15.1 Offset

Register	Offset
TX_GETU	34h

65.7.1.15.2 Diagram



65.7.1.15.3 Fields

Field	Description
31-8 —	Reserved
7-0 GETU	Transmitter Guard Time Value in ETU Used to control the number of additional Elementary Time Units (ETUs) inserted between bytes transmitted by the EMVSIM transmitter. An ETU is equivalent to one bit time at the given baud rate (for example, the length of a START bit). The guard time has no effect on the EMVSIM receiver. A value of 0x00 inserts no additional ETUs, while a value of 0xFE inserts 254 additional ETUs. A value of 0xFF subtracts one ETU by reducing the number of STOP bits from two to one. 00000000 - No additional ETUs inserted 00000001 - 1 additional ETU inserted ... 11111110 - 254 additional ETUs inserted 11111111 - Subtracts one ETU by reducing the number of STOP bits from two to one

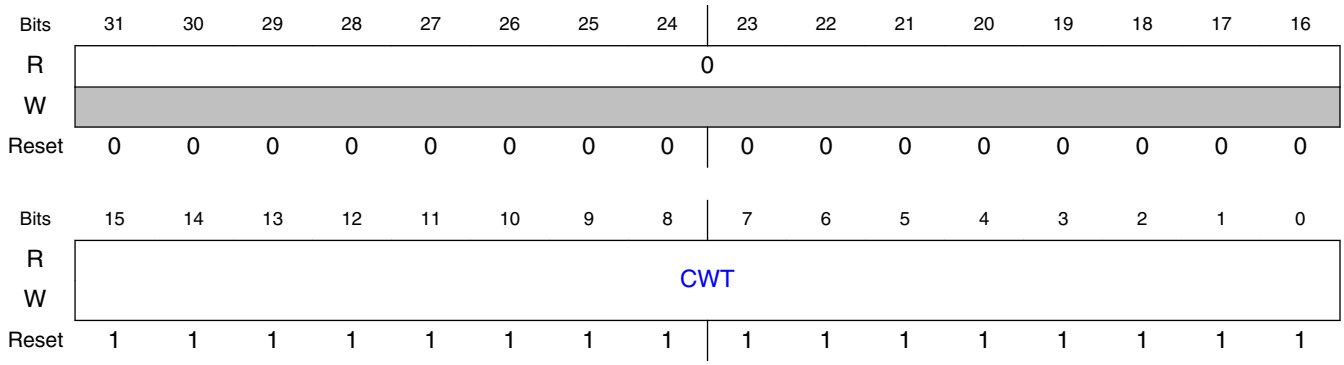
65.7.1.16 Character Wait Time Value Register (CWT_VAL)

This register is used to specify the number of ETU times allowed between characters.

65.7.1.16.1 Offset

Register	Offset
CWT_VAL	38h

65.7.1.16.2 Diagram



65.7.1.16.3 Fields

Field	Description
31-16 —	Reserved
15-0 CWT	Character Wait Time Value The value written to this register specifies the number of ETU times allowed between characters. Default is 0xFFFF

65.7.1.17 Block Wait Time Value Register (BWT_VAL)

This register is used to specify the maximum time between the time from the START bit of last byte sent from the EMVSIM module to the START bit of the first byte sent from the Smart Card.

65.7.1.17.1 Offset

Register	Offset
BWT_VAL	3Ch

65.7.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BWT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BWT															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

65.7.1.17.3 Fields

Field	Description
31-0 BWT	Block Wait Time Value The time from START bit of last byte sent from the EMVSIM module to the START bit of the first byte sent from the Smart Card must be less than the value in this register. If it is not, then the BWT_TO flag is set.

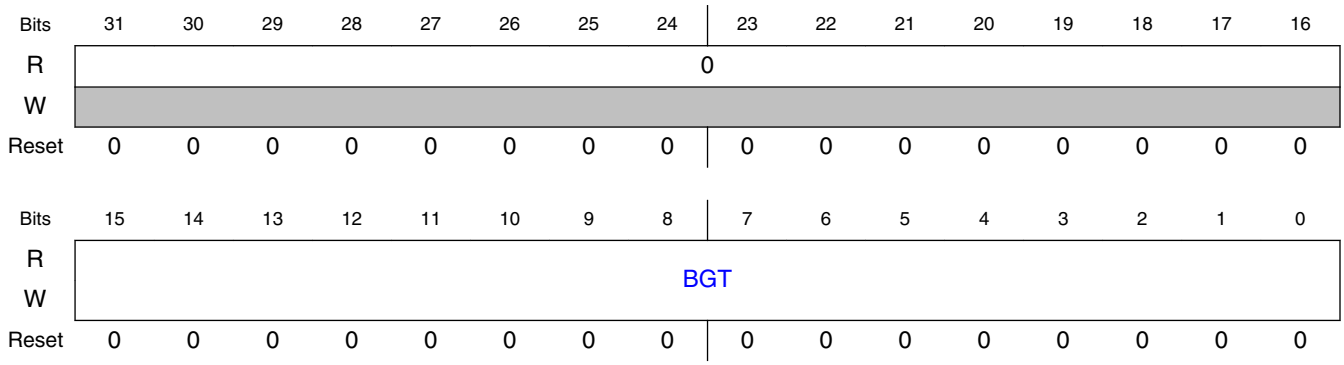
65.7.1.18 Block Guard Time Value Register (BGT_VAL)

The value in this register is the block guard time. The time from the START bit of last byte sent from the EMVSIM module to the START bit of the first byte sent from the SmartCard must be greater than this value.

65.7.1.18.1 Offset

Register	Offset
BGT_VAL	40h

65.7.1.18.2 Diagram



65.7.1.18.3 Fields

Field	Description
31-16 —	Reserved
15-0 BGT	Block Guard Time Value The value in this register is the block guard time. Time from START bit of last byte sent from the EMVSIM module to the START bit of the first byte sent from the SmartCard must be greater than this value. If it is not, then the BGT flag is set.

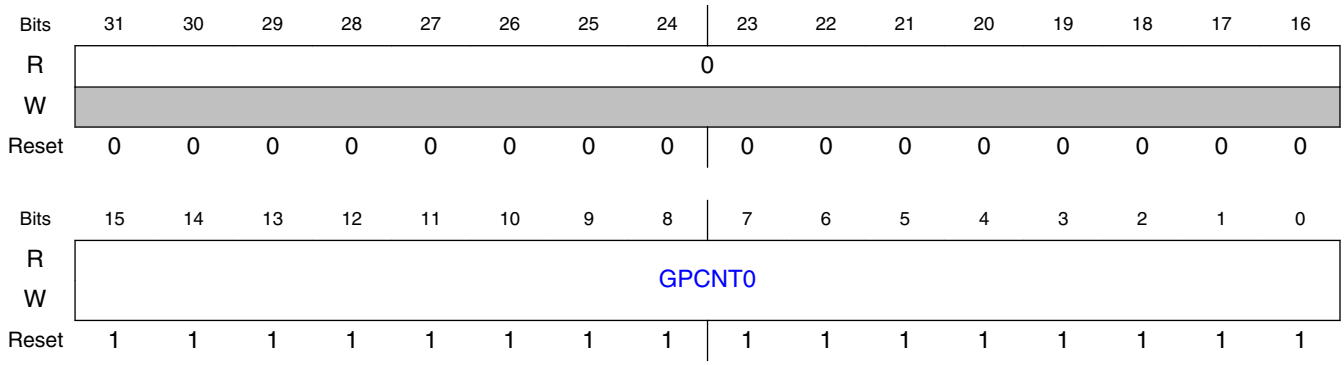
65.7.1.19 General Purpose Counter 0 Timeout Value Register (GPCNT0_VAL)

The value written to this register is used to compare to the general purpose counter 0 in the EMVSIM module.

65.7.1.19.1 Offset

Register	Offset
GPCNT0_VAL	44h

65.7.1.19.2 Diagram



65.7.1.19.3 Fields

Field	Description
31-16 —	Reserved
15-0 GPCNT0	General Purpose Counter 0 Timeout Value The value written to this register is used to compare to the general purpose counter 0 in the EMVSIM module. Once the General purpose counter reaches this value, the GPCNT0_TO flag in the TX_STATUS register is set. This counter is intended to be used for any events that must be monitored for duration based on the card clock, receiver sample rate, or ETU rate (transmit clock). Example: ATR arrival time and ATR duration.

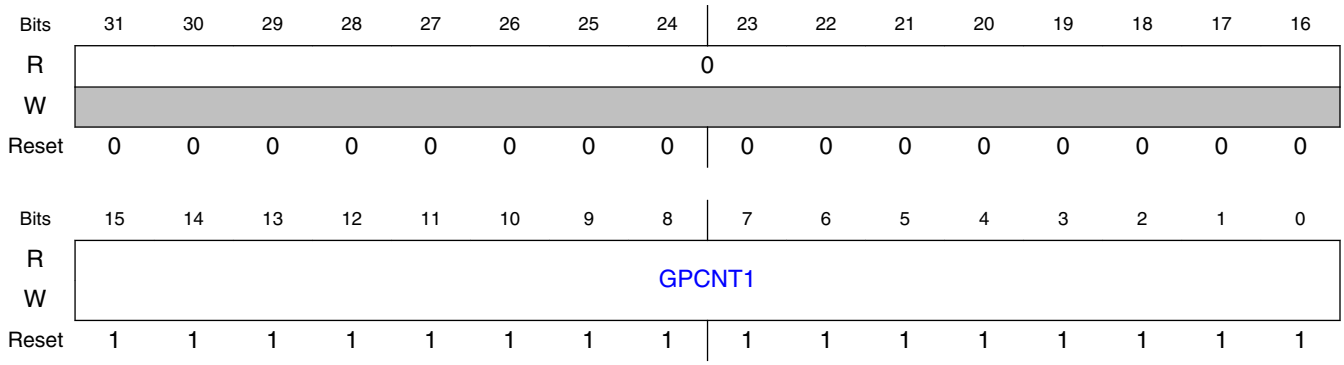
65.7.1.20 General Purpose Counter 1 Timeout Value (GPCNT1_VAL)

The value written to this register is used to compare to the general purpose counter in the EMVSIM module.

65.7.1.20.1 Offset

Register	Offset
GPCNT1_VAL	48h

65.7.1.20.2 Diagram



65.7.1.20.3 Fields

Field	Description
31-16 —	Reserved
15-0 GPCNT1	General Purpose Counter 1 Timeout Value The value written to this register is used to compare to the general purpose counter in the EMVSIM module. Once the General purpose counter reaches this value, the GPCNT1_TO flag in the TX_STATUS register is set. This counter is intended to be used for any events that must be monitored for duration based on the card clock, receiver sample rate, or ETU rate (transmit clock). Example: ATR arrival time and ATR duration.

Chapter 66
Flexible Controller Area Network (FLEXCAN)

66.1 Chip-specific FLEXCAN information

Table 66-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

- PIT1 trigger out can route through XBAR to CAN1 and CAN2, and PIT2 trigger out is connected to CAN3.
- If CTRL1[CLKSRC] = 1, CAN PE clock will be sourced from BUS_LPSR_CLK_ROOT.

  If CTRL1[CLKSRC] = 0, CAN PE clock will be sourced from CANn_CLK_ROOT.

66.2 Overview

The FlexCAN module is a communication controller implementing the CAN protocol according to the ISO 11898-1 standard and CAN 2.0 B protocol specifications. A general block diagram is shown in Figure 66-1, which describes the main subblocks implemented in the FlexCAN module, including:

- One associated memory for storing message buffers
- Receive Global Mask registers
- Receive Individual Mask registers
- Receive FIFO filters
- Receive FIFO ID filters

NOTE

Rx FIFO cannot be used in FD mode.

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific requirements of this field:

- Real-time processing
- Reliable operation in the EMI environment of a vehicle
- Cost-effectiveness
- Required bandwidth

The FlexCAN module is a full implementation of the CAN protocol specification, the CAN with Flexible Data rate (CAN FD) protocol, and the CAN 2.0 version B protocol, which supports both standard and extended message frames and long payloads up to 64 bytes, transferred at faster rates (up to 8 Mb/s—Note that this baud rate can be reached only in the Rx and Tx digital pins, and also depends on the physical layer of the application). The message buffers are stored in an embedded RAM dedicated to the FlexCAN module. See the chip-specific FlexCAN information for the actual number of message buffers configured in the chip.

The Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames
- Validating received messages
- Performing error handling
- Detecting CAN FD messages

The Controller Host Interface (CHI) submodule manages message buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms for both CAN FD and non-CAN FD message formats.

The Bus Interface Unit (BIU) submodule controls access to and from the internal interface bus, in order to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs, DMA and test signals are accessed through the BIU.

66.2.1 Block diagram

The following figure illustrates the block diagram of FLEXCAN.

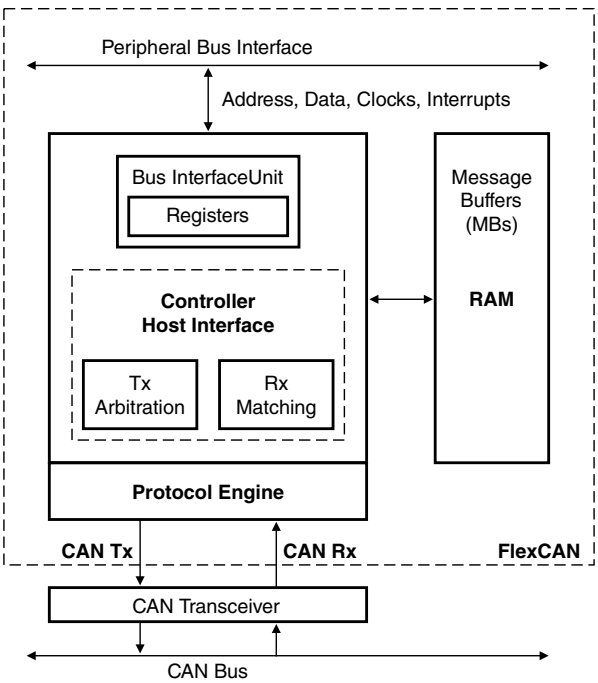


Figure 66-1. FlexCAN block diagram

66.2.2 Features

The FlexCAN module includes these distinctive features:

- Full implementation of the CAN with Flexible Data Rate (CAN FD) protocol specification and CAN protocol specification, Version 2.0 B
  - Standard data frames
  - Extended data frames
  - Zero to sixty four bytes data length
  - Programmable bit rate (see the chip-specific FlexCAN information for the specific maximum rate configuration)
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Flexible mailboxes configurable to store 0 to 8, 16, 32, or 64 bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Rx FIFO with storage capacity for up to six frames and automatic internal pointer handling with DMA support

- Transmission abort capability
- Flexible message buffers (MBs), totaling 64 message buffers of 8 bytes data length each, configurable as Rx or Tx
- Programmable clock source to the CAN Protocol Interface, either peripheral clock or oscillator clock
- RAM not used by reception or transmission structures can be used as general purpose RAM space
- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free running timer, with an optional external time tick
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes, with programmable wakeup on bus activity
- Transceiver delay compensation feature when transmitting CAN FD messages at faster data rates
- Remote request frames may be managed automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to indicate that the module is synchronous with CAN bus
- CRC status for transmitted message
- Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 individual masking capability
- 100% backward compatibility with previous FlexCAN version
- Supports detection and correction of errors in memory read accesses

  Each byte of FlexCAN memory is associated to 5 parity bits, and the error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in 2 bits can be detected but not corrected (non-correctable errors).

66.3 Functional description

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed of a set of message buffers (MB) that store configuration and control data, time stamp, message ID, and data (see Message buffer structure).

For Classical CAN frames, simultaneous reception through FIFO and mailbox is supported. For CAN FD frames, reception is supported through mailboxes only. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed in the ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A message buffer is said to be active at a given time if it can participate in both the matching and arbitration processes. An Rx MB with a 0b code is inactive (see Table 66-26). Similarly, a Tx MB with a 1000b or 1001b code is also inactive (see Table 66-27).

The FlexCAN module is also able to receive and transmit messages in CAN FD format. The message buffers are sized to adequately store the quantity of data bytes selected by the FDCTRL[MBDSRn] fields. The quantity of FD MBs available for a given quantity of data bytes is described in the FDCTRL register. See also FlexCAN memory partition for CAN FD.

66.3.1 Modes of operation

The FlexCAN module has these functional modes:

- Normal mode (User or Supervisor):

  In Normal mode, the module operates receiving and/or transmitting message frames, errors are managed normally, and all CAN Protocol functions are enabled. User and Supervisor modes differ in the access to some restricted control registers.

- Freeze mode:

Freeze mode is enabled when MCR[FRZ] is asserted. If enabled, Freeze mode is entered when MCR[HALT] is set or when is requested at chip level and MCR[FRZ_ACK] is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze mode for more information.

- Loop-Back mode:

  The module enters this mode when CTRL1[LPB] is asserted. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Listen-Only mode:

  The module enters this mode when CTRL1[LOM] is asserted. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- CAN FD Active mode:

  In this mode, FlexCAN is capable of transmitting and receiving all messages formatted according to the CAN FD Protocol and CAN 2.0 Protocol 2.0 in an interleaved fashion. The CPU can set the FlexCAN into CAN FD Active mode by configuring MCR[FDEN] in Freeze mode.

  It is important to know which features are available in CAN FD active mode. This table lists the differences between FD and classical modes.

Table 66-2. Differences between CAN classical and CAN FD

Feature	CAN classical	CAN FD
Rx FIFO DMA	Yes	No
Rx FIFO	Yes	No

For low-power operation, the FlexCAN module has:

- Module Disable mode:

This low-power mode is entered when MCR[MDIS] is asserted by the CPU and MCR[LPM_ACK] is asserted by FlexCAN. When disabled, the module issues a request to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Exit from this mode is done by negating MCR[MDIS]. See Module Disable mode for more information.

- Doze mode:

  This low power mode is entered when MCR[DOZE] is asserted and Doze mode is requested at chip level, and MCR[LPM_ACK] is asserted by FlexCAN. When in Doze mode, the module issues a request to disable the clocks to the CAN Protocol Engine and the CAN Controller-Host Interface submodules. Exit from this mode happens when MCR[DOZE] is negated, when the chip is removed from Doze mode, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Doze mode for more information.

- Stop mode:

  This low power mode is entered when Stop mode is requested at chip level and MCR[LPM_ACK] is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the CPU that the clocks can be shut down globally. Exit from this mode happens when the Stop mode request is removed, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Stop mode for more information.

66.3.1.1 Modes of operation details

The FlexCAN module has functional modes and low-power modes. See Modes of operation for an introductory description of all the modes of operation. The following sub-sections contain functional details on Freeze mode and the low-power modes.

CAUTION

"Permanent Dominant" failure on CAN Bus line is not supported by FlexCAN. If a Low-Power request or Freeze mode request is done during a "Permanent Dominant", the corresponding acknowledge can never be asserted.

66.3.1.1.1 Freeze mode

This mode is requested either by the CPU through the assertion of MCR[HALT] or when the chip is put into Debug mode. In both cases it is also necessary that MCR[FRZ] be asserted and the module not be in a low-power mode. This mode is also requested by

FlexCAN through the automatic assertion of both MCR[HALT] and MCR[FRZ] when MECR[NCEFAFRZ] is set and a non-correctable error is detected in a memory read access performed by FlexCAN internal processes (see Response to errors).

The acknowledgement is obtained through the assertion by the FlexCAN of MECR[FRZ_ACK]. The CPU must only consider the FlexCAN in Freeze mode when both request and acknowledgement conditions are satisfied.

When Freeze mode is requested, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off, or Idle state.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in does not prevent entering Freeze mode.
- Ignores the Rx input pin and drives the Tx pin as recessive.
- Stops the prescaler, thus halting all CAN protocol activities.
- Grants write access to the Error Counters register, which is read-only in other modes.
- Sets MCR[NOTRDY] and MCR[FRZACK].

After requesting Freeze mode, the user must wait for MCR[FRZ_ACK] to be asserted before executing any other action; otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory-mapped registers are accessible, except for CTRL1[CLKSRC], which can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates MCR[FRZ].
- The chip is removed from Debug mode and/or the HALT bit is negated.

MCR[FRZ_ACK] is negated after the protocol engine recognizes the negation of the freeze request. When out of Freeze mode, FlexCAN tries to resynchronize to the CAN bus by waiting for 11 consecutive recessive bits.

66.3.1.1.2 Module Disable mode

This low-power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the CPU through the assertion of MCR[MDIS], and the acknowledgement is obtained through the assertion by the FlexCAN of MCR[LPMACK]. The CPU must only consider the FlexCAN in Disable mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI submodules, sets MCR[LPMACK] and negates MCR[FRZACK].

If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules
- Sets MCR[NOTRDY] and MCR[LPMACK].

The Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except the Rx Mailboxes Global Mask registers, the Rx Buffer 14 Mask register, the Rx Buffer 15 Mask register,. The message buffers, the Rx Individual Mask registers, and the reserved words within RAM may not be accessed when the module is in Disable mode. Exiting from this mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and negate the LPMACK bit after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

66.3.1.1.3 Doze mode

This is a system low power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, MCR[DOZE] needs to have been asserted previously for Doze mode to be triggered. The acknowledgement is obtained through the assertion by the FlexCAN of MCR[LPMACK]. The CPU must only consider the FlexCAN to be in Doze mode when both request and acknowledgement conditions are satisfied.

If Doze mode is triggered during Freeze mode, FlexCAN requests to shut down the clocks to the PE and CHI submodules, sets MCR[LPMACK] and negates MCR[FRZACK]. If Doze mode is triggered during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.

- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules.
- Sets MCR[NOTRDY] and MCR[LPMACK].

The Bus Interface Unit continues to operate, enabling the CPU to access memory-mapped registers, except the Rx Mailboxes Global Mask registers, the Rx Buffer 14 Mask register, the Rx Buffer 15 Mask register. The the message buffers, the Rx Individual Mask registers, and the reserved words within RAM may not be accessed when the module is in Doze mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request
- CPU negating MCR [DOZE]
- Self Wake mechanism

In the Self Wake mechanism, if MCR[SLFWAK] was set at the time FlexCAN entered Doze mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN negates the DOZE bit, requests to resume its clocks and negates the LPMACK after the CAN protocol engine recognizes the negation of the Doze mode request. It also sets ESR [WAKINT] and, if enabled by MCR[WAKMSK], generates a Wake-Up interrupt to the CPU. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of SLFWAK and WAKMSK upon wakeup from Doze mode.

Table 66-3. Wakeup from Doze mode

SLFWAK	WAKINT	WAKMSK	FlexCAN clocks enabled	Wakeup interrupt generated
0	—	—	No	No
0	—	—	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	Yes	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line when in Doze mode. (See MCR[WAKSRC].) This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

66.3.1.1.4 Stop mode

This is a system low-power mode in which all chip clocks can be stopped for maximum power savings. The Stop mode is globally requested by the CPU and acknowledgement is obtained through the assertion by the FlexCAN of a Stop Acknowledgement signal. The CPU must only consider the FlexCAN in Stop mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it sets MCR[LPMACK], negates MCR[FRZACK], and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Stop mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Sets MCR[NOTRDY] and MCR[LPMACK]
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Stop mode is exited when the CPU resumes the clocks and removes the Stop mode request. This can be as a result of the Self Wake mechanism.

In the Self Wake mechanism, if MCR[SLFWAK] was set at the time FlexCAN entered Stop mode, then upon detection of a recessive to dominant transition on the CAN bus, FlexCAN sets ESR[WAKINT] and, if enabled by MCR[WAKMSK], generates a Wake Up interrupt to the CPU. Upon receiving the interrupt, the CPU should resume the clocks and remove the Stop mode request. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. The following table details the effect of MCR[SLFWAK] and MCR[WAKMSK] upon wakeup from Stop mode. Note that wakeup from Stop mode only works when both bits are asserted.

After the CAN protocol engine recognizes the negation of the Stop mode request, the FlexCAN negates MCR[LPMACK]. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up.

Table 66-4. Wakeup from Stop mode

SLFWAK	WAKINT	WAKMSK	Chip clocks enabled	Wakeup interrupt generated
0	—	—	No	No
0	—	—	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	No	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line when in Stop mode. (See the description in MCR[WAKSRC].) This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

66.3.2 Transmit process

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (1001b) to the CODE field of the Control and Status word to request an abort of the transmission.
3. Wait for the corresponding IFLAG bit to be asserted by polling the IFLAG register, or by the interrupt request if enabled by the respective IMASK bit.
4. Read back the CODE field to check if the transmission was aborted or transmitted (see Transmission abort mechanism).
5. Clear the corresponding interrupt flag.
6. Write the ID register (containing the local priority if enabled via MCR[LPRIO_EN]).
7. Write payload data bytes.
8. Configure the Control and Status word with the desired configuration.
   a. Set ID type via MB_CS[IDE].
   b. Set Remote Transmission Request (if needed) via MB_CS[RTR].

c. If MCR[FDEN] is enabled, also configure the MB_CS[EDL] and MB_CS[BRS] fields. For details about the relationship between the written value and transmitted value of the MB_CS[ESI] field, see Table 66-8.[1]
d. Set Data Length Code in bytes via MB_CS[DLC]. See Table 66-28 for detailed information.
e. Activate the message buffer to transmit the CAN frame by setting MB_CS[CODE] to Ch.

NOTE

It is strongly recommended that all the fields in MB_CS word be configured in only one 32-bit write operation to maximize software performance. If the fields are configured in separate writes, the MB_CS[CODE] must be the last write in the C/S word.

When the MB is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the DLC value stored in the MB selected for transmission is larger than the respective MB payload size, FlexCAN adds the necessary number of bytes with constant CCh pattern to complete the expected DLC.

At the end of the successful transmission:

- The value of the free running timer is written into the Time Stamp field.
- The CODE field in the Control and Status word is updated.
- Both CRC and FDCRC registers are updated.
- A status flag is set in the Interrupt Flag register.
- An interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The new CODE field after transmission depends on the code that was used to activate the MB (see Table 66-26 and Table 66-27 in Message buffer structure).

When the Abort feature is enabled (MCR[AEN] is asserted), after the Interrupt flag is asserted for a mailbox configured as transmit buffer, the mailbox is blocked. Therefore the CPU is not able to update it until the Interrupt Flag is negated by the CPU. This means that the CPU must clear the corresponding IFLAG bit before starting to prepare this MB for a new transmission or reception.

NOTE

If backwards compatibility is desired (MCR[AEN] is negated), write the INACTIVE code (1000b) to the CODE field to

1. The ESI field does not need to be written, and will automatically be transmitted dominant by error active nodes and recessive by error passive nodes. Note that there is an exception if the FlexCAN is operating as a network gateway: In that case, the CPU writes the MB_CS[ESI] bit according to the error status of the node which sent the message.

inactivate the MB. However, in this case the pending frame may be transmitted without notification (see Mailbox inactivation).

66.3.3 Arbitration process

The arbitration process scans the mailboxes, searching for the Tx mailbox that holds the message to be sent in the next opportunity. This mailbox is called the arbitration winner.

The scan starts from the lowest number mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the value of CTRL2[TASD].
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is a CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all mailboxes before the CAN bus has reached the first bit of the Intermission field, the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx mailboxes at the end of the scan according to the settings of both CTRL1[LBUF] and MCR[LPRIOEN].

66.3.3.1 Lowest-number mailbox first

If CTRL1[LBUF] is asserted, the first (lowest number) active Tx mailbox found is the arbitration winner. MCR[LPRIOEN] has no effect when CTRL1[LBUF] is asserted.

66.3.3.2 Highest-priority mailbox first

If CTRL1[LBUF] is negated, then the arbitration process searches the active Tx mailbox with the highest priority, which means that this mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the *arbitration value* of the mailbox. The highest-priority Tx mailbox is the one that has the lowest arbitration value among all Tx mailboxes.

If two or more mailboxes have equivalent arbitration values, the mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on the MCR[LPRIOEN] setting.

66.3.3.2.1 Local priority disabled

If MCR[LPRIOEN] is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see the following table) in such a way that the local priority is disabled.

Table 66-5. Composition of the arbitration value when local priority is disabled

Format	Mailbox arbitration value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

66.3.3.2.2 Local priority enabled

To enable local priority, MCR[LPRIOEN] must be asserted. In this case the mailbox PRIO field is included at the very left of the arbitration value (see the following table).

Table 66-6. Composition of the arbitration value when local priority is enabled

Format	Mailbox arbitration value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

Because the PRIO field is the most significant part of the arbitration value, mailboxes with low PRIO values have higher priority than mailboxes with high PRIO values regardless of the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

66.3.3.3 Arbitration process (continued)

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and after it is done, write access to the C/S word of the corresponding MB is blocked (if MCR[AEN] is asserted). Write access is restored in the following events:

- After the MB is transmitted and the corresponding IFLAG bit is cleared by the CPU.
- FlexCAN enters Freeze mode or Bus Off.
- FlexCAN loses the bus arbitration or there is an error during the transmission.

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. CTRL2[TASD] value may be changed to optimize the arbitration start point.
- During CAN Bus Off state from TX_ERR_CNT=124 to 128. CTRL2[TASD] value may be changed to optimize the arbitration start point.
- During C/S write by CPU in Bus Idle. First C/S write starts arbitration process, and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after the arbitration process has finished, then the TX arbitration machine begins a new arbitration process. If there is a pending arbitration and Bus Idle state starts, then an arbitration process is triggered. In this case the first and second C/S write in Bus Idle will not restart the arbitration process. It is possible that there is not enough time to finish arbitration in Wait For Bus Idle state and the next state is Idle. In this case the scan is not interrupted, and it is completed during Bus Idle state. During this arbitration C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon exiting Freeze mode (first bit of the Wait For Bus Idle state). If there is a re-synchronization during Wait For Bus Idle, the arbitration process is restarted.

Arbitration process stops in the following situations:

- All mailboxes were scanned.
- A Tx active mailbox is found if lowest buffer feature is enabled.
- Arbitration winner inactivation or abort during any arbitration process.
- There was not enough time to finish Tx arbitration process (for instance, when a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or overload flag in the bus.
- Low Power or Freeze mode request in Idle state.

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time.
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer.
- Any C/S write if there is no Tx arbitration process in progress.
- Rx Match has just updated a Rx code to Tx code.
- Entering Bus Off state.

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If C/S write is performed in a MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

66.3.4 Receive process

To be able to receive CAN frames into a mailbox, the CPU must prepare it for reception by executing the following steps:

1. If the mailbox is active (either Tx or Rx) inactivate the mailbox (see Mailbox inactivation), preferably with a safe inactivation (see Transmission abort mechanism).
2. Write the ID word into the mailbox.
3. Write the EMPTY code (0100b) to the CODE field of the Control and Status word to activate the mailbox. No setup is required for EDL, BRS, and ESI bits—they are overwritten by the respective bit fields in the received message.

After the MB is activated, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the mailbox is updated by the move-in process (see Move-in) as follows:

1. The received data field (8 bytes at most for Classical CAN message format and up to 64 bytes for CAN FD message format) is stored.
2. The received Identifier field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the mailbox's Time Stamp field.
4. The received SRR, IDE, RTR, EDL, BRS, ESI, and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see Table 66-26 and Table 66-27 in Section Message buffer structure).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for the CPU to service (read) the frame received in a mailbox is by the following procedure:

1. Read the Control and Status word of that mailbox.
2. Check if the BUSY bit is deasserted, indicating that the mailbox is locked. Repeat step 1) while it is asserted. See Mailbox lock mechanism.
3. Read the contents of the mailbox. After the mailbox is locked, its contents won't be modified by FlexCAN move-in processes. See Move-in.
4. Acknowledge the proper flag at IFLAG registers.
5. Read the free running timer to unlock the mailbox.

The CPU should poll for frame reception by the status flag bit for the specific mailbox in one of the IFLAG registers and not by the CODE field of that mailbox. Polling the CODE field does not work because after a frame is received and the CPU services the mailbox (by reading the C/S word followed by unlocking the mailbox), the CODE field will not return to EMPTY. It will remain FULL, as explained in Table 66-26. If the CPU tries to work around this behavior by writing to the C/S word to force an EMPTY code after reading the mailbox without a prior safe inactivation, a newly received frame matching the filter of that mailbox may be lost.

CAUTION

In summary: never do polling by reading directly the C/S word of the mailboxes. Instead, read the IFLAG registers.

Note that the received frame's Identifier field is always stored in the matching mailbox, thus the contents of the ID field in a mailbox may change if the match was due to masking. When MCR[SRXDIS] is asserted, FlexCAN will not store frames transmitted by itself in any MB, even if it contains a matching Rx mailbox, and no interrupt flag or interrupt signal will be generated. Otherwise, when MCR[SRXDIS] is deasserted, FlexCAN can receive frames transmitted by itself if a matching Rx mailbox exists.

When MCR[DMA] is asserted, upon receiving a frame in FIFO, IFLAG1[BUF5I] generates a DMA request and does not generate a CPU interrupt (see Rx FIFO under DMA operation). The IMASK1 bits in Rx FIFO region are not used.

The DMA controller must service the received frame using the following procedure:

1. Read the Control and Status word (read 80h address, optional).
2. Read the ID field (read 84h address, optional).
3. Read all data bytes (start read at 88h address, optional).

66.3.5 Matching process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. . The matching starts from the lowest number message buffer toward the higher ones.

For Rx FIFO see Rx FIFO.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame.

If a matching ID is found in one of the mailboxes, the contents of the Rx SMB are transferred to to the matched mailbox by the move-in process. If any CAN protocol error is detected then no match results are transferred to the matched mailbox at the end of reception.

The matching process scans all matching elements of the active Rx mailboxes (CODE is EMPTY, FULL, OVERRUN, or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The Rx SMB has the same structure as a mailbox. The reception structures (mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. See the following table.

Table 66-7. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Mailbox	0	—	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY or FULL or OVERRUN
Mailbox	0	—	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	—	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN

1. For mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). If SMB[IDE] is negated, the ID is only 11 bits (ID Standard). For FIFO structure, the ID depends on IDAM.
2. cmp: Compares the Rx SMB contents with the MB contents regardless the masks.
3. no_cmp: The Rx SMB contents are not compared with the MB contents.
4. cmp_msk: Compares the Rx SMB contents with MB contents taking into account the masks.

A reception structure is free-to-receive when any of the following conditions is satisfied:

- The CODE field of the mailbox is EMPTY.
- The CODE field of the mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in Mailbox lock mechanism).
- The CODE field of the mailbox is either FULL or OVERRUN and an inactivation (see Mailbox inactivation) is performed.

The scan order for mailboxes is from the matching element with lowest number to the higher ones.

The matching winner search for mailboxes is affected by MCR[IRMQ]. If it is negated, the matching winner is the first matched mailbox regardless if it is free-to-receive or not. If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched mailbox;
2. the last non free-to-receive matched mailbox.

If a non-safe mailbox inactivation (see Mailbox inactivation) occurs during matching process and the mailbox inactivated is the temporary matching winner, then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted—it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that IRMQ is enabled, there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm finds the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm finds MB number 2 again, but it is not "free-to-receive", so it keeps looking, finds MB number 5 and stores the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages are queued into the MBs. The CPU can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID acceptance masks. FlexCAN supports individual masking per MB (see the description of #d46430e8149a1310). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is a "don't care". Note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed when the module is in Freeze mode; otherwise, they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (RXMGMASK, RX14MASK, and RX15MASK) for backwards compatibility with legacy applications. This alternate masking scheme is enabled when the IRMQ bit in the MCR register is negated.

66.3.6 Move process

There are two types of move process: move-in and move-out.

66.3.6.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to an Rx mailbox that has matched it. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see Matching process) and all of the following conditions are true:

- The CAN bus has either reached or already gone past:
  - The second bit of Intermission field next to the frame that carried the message that is in the Rx SMB.
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB.
- There is no ongoing matching process.
- The destination mailbox is not locked by the CPU.
- There is no ongoing move-in process from another Rx SMB. If more than one move-in process is to be started at the same time, both are performed and the newest substitutes for the oldest.

The term pending move-in is used throughout the documentation and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished.
- There is a previous pending move-in to the same destination mailbox.
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and self-reception is disabled (MCR[SRXDIS] is asserted).
- Any CAN protocol error is detected.

Note that the pending move-in is not cancelled if the module enters Freeze or Low-Power mode. It stays on hold, only waiting for Freeze and Low-Power mode to be exited and the module to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. Read all data words from the Rx SMB in accordance with the selected payload size for the Rx storage element.
2. Write all data words to the Rx mailbox in accordance with the selected payload size for the Rx storage element. If the data size of the storage element is smaller than the original payload size described in the message's DLC field, the payload is truncated and the high order bytes that do not fit the destination size are lost.
3. Read the Control/Status and ID words from the Rx SMB.
4. Write Control/Status and ID words to the Rx mailbox, and update the CODE field.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination mailbox (see Mailbox inactivation). In this case the mailbox may remain partially updated, thus incoherent.

The BUSY Bit (least significant bit of the CODE field) of the destination message buffer is asserted while the move-in is being performed to alert the CPU that the message buffer content is temporarily incoherent.

66.3.6.2 Move-out

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Arbitration process). The move-out occurs in the following conditions:

- The first bit of Intermission field
- During Bus Off state when TX Error Counter is in the 124 to 128 range
- During Bus Idle state
- During Wait For Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

66.3.7 Data coherence

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in Transmit process and Receive process.

66.3.7.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abort of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- MCR[AEN] must be asserted.
- The first CPU action must be the writing of abort code (1001b) into the CODE field of the Control and Status word.

Active MBs configured for transmission must be aborted first before they can be updated. If the abort code is written to a Mailbox that is currently being transmitted or to a Mailbox that was already loaded into the Tx SMB for transmission, the write operation is blocked and the transmission is not disturbed. However, the abort request is captured and kept pending until one of the following conditions is satisfied:

- The module loses the bus arbitration.
- There is an error during the transmission.
- The module is put into Freeze mode.
- The module enters Bus Off state.
- There is an overload frame.

If none of the conditions above are reached:

- The MB is transmitted correctly.
- The interrupt flag is set in the IFLAG register.
- An interrupt to the CPU is generated (if enabled).

The abort request is automatically cleared when the interrupt flag is set. On the other hand, if only one of the above conditions is reached, the frame is not transmitted; therefore:

- The abort code is written into the CODE field.
- The interrupt flag is set in the IFLAG.
- An interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way the CPU only needs to read the abort code to make sure the active MB was safely inactivated. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is aborted only when the abort request is captured and kept pending until one of the previous conditions is satisfied.

66.3.7.2 Mailbox inactivation

Inactivation is a mechanism provided to protect the mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission mailboxes must be performed just when MCR[AEN] is deasserted.

If a mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See Transmit process and Receive process for more detailed instructions on how to inactivate and reactivate a mailbox.

To inactivate a mailbox, the CPU must update its CODE field to INACTIVE (either 0b or 1000b).

Because you will not be able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can have the following consequences:

- A frame in the bus that matches the filtering of the inactivated Rx mailbox may be lost without notice, even if there are other mailboxes with the same filter.
- A frame containing the message within the inactivated Tx mailbox may be transmitted without setting the respective IFLAG.

In order to perform a safe inactivation and avoid the above consequences for Tx mailboxes, the CPU must use the transmission abort mechanism (see Transmission abort mechanism).

The inactivation automatically unlocks the mailbox (see Mailbox lock mechanism).

66.3.7.3 Mailbox lock mechanism

Other than mailbox inactivation, FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation, and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the free running timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code. A CPU write into the C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending move and potentially may lose a received message. The MB locking prevents a new frame from being written into the MB while the CPU is reading it.

NOTE

The locking mechanism applies only to Rx MBs that have a code different than INACTIVE (0b) or EMPTY[1] (0100b). Also, Tx MBs cannot be locked.

Suppose, for example, that the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the Control and Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no free-to-receive MBs, so it decides to override MB number 5. However, this MB is locked, so the new message cannot be written there. It will remain in the Rx SMB waiting for the MB to be unlocked, and only then will be written to the MB.

1. In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the Rx SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved in from the Rx SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

Note

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the Rx SMB will not be transferred anymore to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register (see #d46430e1851a1310), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see Modes of operation), and it takes place only when the module returns to Normal or Freeze modes.

66.3.8 Rx FIFO

The Rx FIFO is receive-only and is enabled by asserting MCR[RFEN]. The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature.

CAUTION

Rx FIFO must not be enabled when CAN FD feature is enabled.

The FIFO is 6-message deep. The memory region occupied by the FIFO structure (both message buffers and FIFO engine) is described in Rx FIFO structure. The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a message buffer structure at the output of the FIFO.

IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message

(accessing the output of the FIFO as a message buffer) and the RXFIR register, and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and updates RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the FIFO is valid only while IFLAG1[BUF5I] is asserted.

IFLAG1[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

IFLAG1[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing workload. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read in the IDHIT field from C/S word, as shown in the Rx FIFO Structure description. Another way the CPU can obtain this information is by accessing the RXFIR register. RXFIR[IDHIT] refers to the message at the output of the FIFO and is valid while the IFLAG1[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to 32 elements of the filter table are individually affected by the Individual Mask Registers (RXIMRx), according to the setting of CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If MCR[IRMQ] is negated, then the FIFO filter table is affected by RXFGMASK.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 66-2.

66.3.8.1 Rx FIFO under DMA operation

The receive-only FIFO can support DMA. This feature is enabled by asserting both MCR[RFEN] and MCR[DMA]. The reset value of MCR[DMA] is zero to maintain backward compatibility with previous versions of the module that did not have the DMA feature.

The DMA controller can read the received message by reading a message buffer structure at the FIFO output port at the 80h–8Ch address range.

When MCR[DMA] is asserted the CPU must not access the FIFO output port address range. Before enabling MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. Otherwise, these IFLAGs may show that the FIFO has data to be serviced, and mistakenly generate a DMA request. Before disabling MCR[DMA], the CPU must perform a clear FIFO operation.

IFLAG1[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. Consequently a DMA request is generated simultaneously. Upon receiving the request, the DMA controller can read the message (accessing the output of the FIFO as a message buffer). The DMA reading process must end by reading address 8Ch, which clears IFLAG1[BUF5I] and updates both the FIFO output with the next message (if FIFO is not empty) and the RXFIR register with the attributes of the new message. If there are more messages stored in the FIFO, IFLAG1[BUF5I] will be re-asserted and another DMA request is issued. Otherwise, the flag remains negated.

IFLAG1[BUF6I] and IFLAG1[BUF7I] are not used when the DMA feature is enabled.

When FlexCAN is working with DMA, the CPU does not receive any Rx FIFO interruption and must not clear the related IFLAGs. In addition, the related IMASKs are not used to mask the generation of DMA requests.

NOTE

For more information about the difference between FD and non-FD regarding this feature, see Table 66-2.

66.3.8.2 Clear FIFO operation

When MCR[RFEN] is asserted, the clear FIFO operation is a feature used to empty FIFO contents. With MCR[RFEN] asserted the Clear FIFO occurs when the CPU writes one in IFLAG1[BUF0I]. This operation can only be performed in Freeze mode and is blocked by hardware in other modes. This operation does not clear the FIFO IFLAGs; consequently the CPU must service all FIFO IFLAGs before executing the clear FIFO task.

When Rx FIFO is working with DMA, the clear FIFO operation clears IFLAG1[BUF5I] and the DMA request is canceled.

CAUTION

Clear FIFO operation does not clear IFLAGs, except when MCR[DMA] is asserted; in this case only IFLAG1[BUF5I] is cleared.

66.3.9 CAN protocol related features

This section describes the CAN protocol related features.

66.3.9.1 CAN FD ISO compliance

The CAN FD protocol has been improved to increase the failure detection capability that was in the original CAN FD protocol, which is also called non-ISO CAN FD, by CAN in Automation (CiA). A three-bit stuff counter and a parity bit have been introduced in the improved CAN FD protocol, now called ISO CAN FD. The CRC calculation has also been modified. All these improvements make the ISO CAN FD protocol incompatible with the non-FD CAN FD protocol. The non-ISO CAN FD is still supported by FlexCAN so that it can be used mainly during an intermediate phase, for evaluation and development purposes.

Therefore, it is strongly recommended to configure FlexCAN to the ISO CAN FD protocol by setting the ISOCANFDEN field in the CTRL2 register.

66.3.9.2 CAN FD frames

The ISO 11898-1 standard specifies the Classical Frame format compliant to ISO 11898-1 (2003) and introduces the CAN Flexible Data Rate Frame format. The Classical Frame format allows bit rates up to 1 Mbit/s and payloads up to 8 bytes per frame. The Flexible Data Rate Frame format allows bit rates higher than 1 Mbit/s and payloads longer than 8 bytes per frame. FlexCAN can receive and transmit CAN FD messages interleaved with Classical CAN messages.

There are three additional control bits in the CAN FD frame. The Extended Data Length (EDL) bit enables a longer data payload with different data length coding. The Bit Rate Switch (BRS) bit decides whether the bit rate is switched inside a CAN FD format frame. The Error State Indicator (ESI) flag is transmitted dominant by error active nodes, and recessive by error passive nodes. There are no Remote Frames (see Remote frames) in the CAN FD format. A message configured to transmit a Remote Frame is always sent out in the Classical CAN format. When an FD frame is received and matches a mailbox, the RTR bit in the receiving message buffer is negated. The RTR bit must be considered in classical frames only.

CAN FD messages may be formatted as long frames where the data field exceeds 8 bytes, and may range from 12 up to 64 bytes. They can also be configured to support bit rate switching, where the control field, the data field, and the CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame. Messages in Classical CAN format are limited to transport a maximum payload of 8 bytes at nominal rate. The following figure illustrates the message formats for Classical and FD frames with either standard or extended ID.

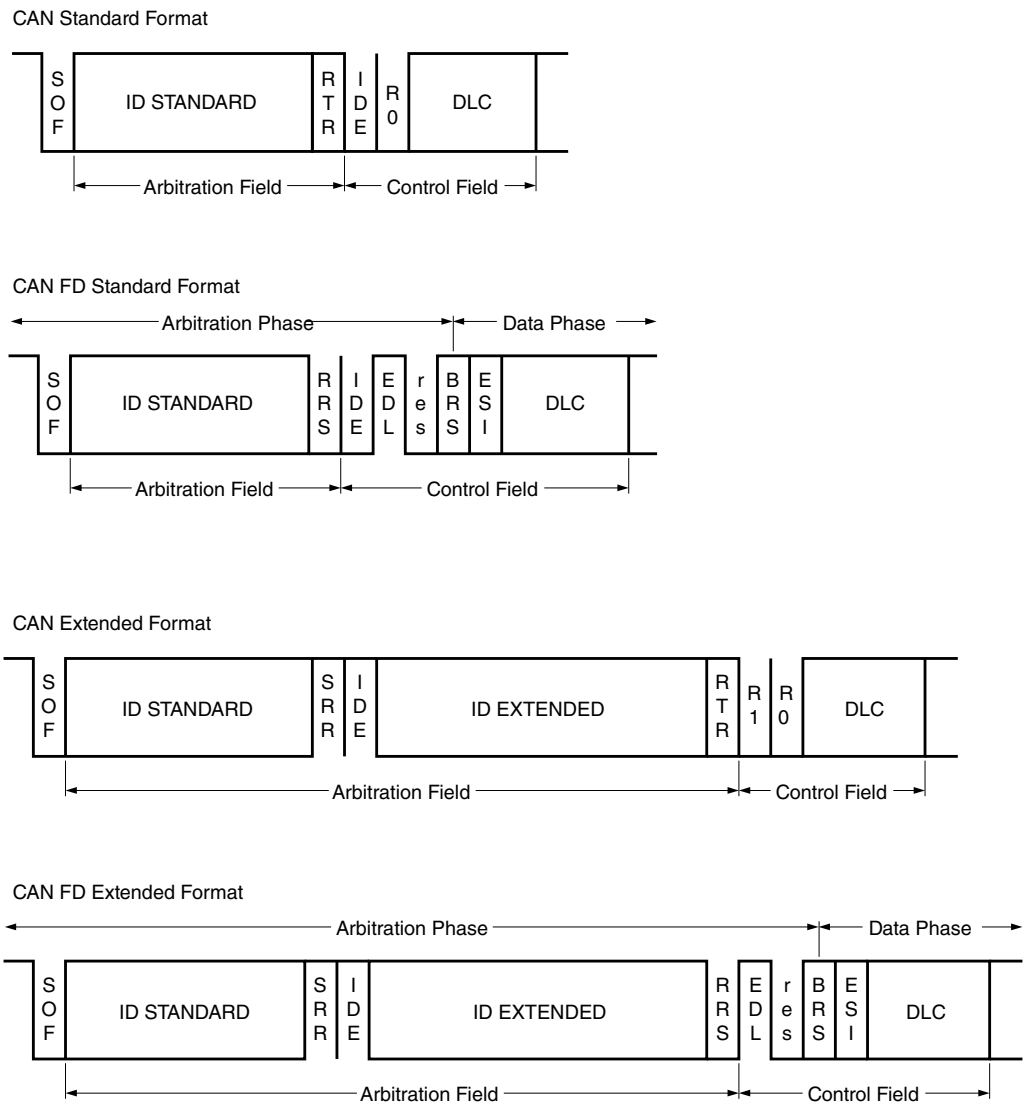


Figure 66-2. CAN message formats

The ability to receive and transmit CAN FD messages is enabled by MCR[FDEN]. Either a recessive R0 bit in CAN frames with 11-bit identifiers or a recessive R1 bit in CAN frames with 29-bit identifiers are decoded as an EDL bit (not a reserved one). A CAN FD frame is recognized by a recessive EDL bit, and a Classical CAN frame is recognized by a dominant EDL bit. The BRS bit specifies whether this frame switches the bit rate in its data phase. A long frame is decoded in accordance with the DLC field value (see DLC definition in Message buffer structure).

CAN FD messages can be transmitted with two different bit rates. The first part of a CAN FD frame, from the Start Of Frame (SOF) bit until the Bit Rate Switch (BRS) bit, also called the arbitration phase, is transmitted with the nominal bit rate based on a set of nominal CAN bit timing configuration values. The second part, from the BRS bit until the CRC Delimiter bit, also named the data phase, is transmitted with the data bit rate

defined by a second set of CAN data bit timing configuration values. Finally, from the CRC Delimiter until the Intermission bits, the transmission returns to nominal bit rate. In CAN FD frames with bit rate switching, the bit timing is changed inside the frame at the sample point of the BRS bit if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the CBT register (also by CTRL1 register for backward compatibility). Upon detecting a recessive BRS bit, the CAN data bit timing is used as defined by the FDCBT register.

NOTE

If the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase and will last until the next synchronization event. Thus, the length of the time quantum should be the same in nominal and data bit timing in order to minimize the chance of error frames on the CAN bus, and to optimize the clock tolerance in networks that use FD frames.

FDCTRL[FDRATE] enables the transmission of all frames with bit rate switching if the BRS bit in the selected Tx MB is set. If FDRATE is negated, the transmission is performed at nominal rate regardless of the BRS bit value. FDCTRL[FDRATE] can be written any time but takes effect only for the next message transmitted or received.

The nominal bit timing is resumed at either the sample point of the CRC Delimiter bit or when an error is detected, whichever occurs first. The following figure describes the mechanism for entering and leaving the data phase when BRS bit is recessive.

CAN FD Frame

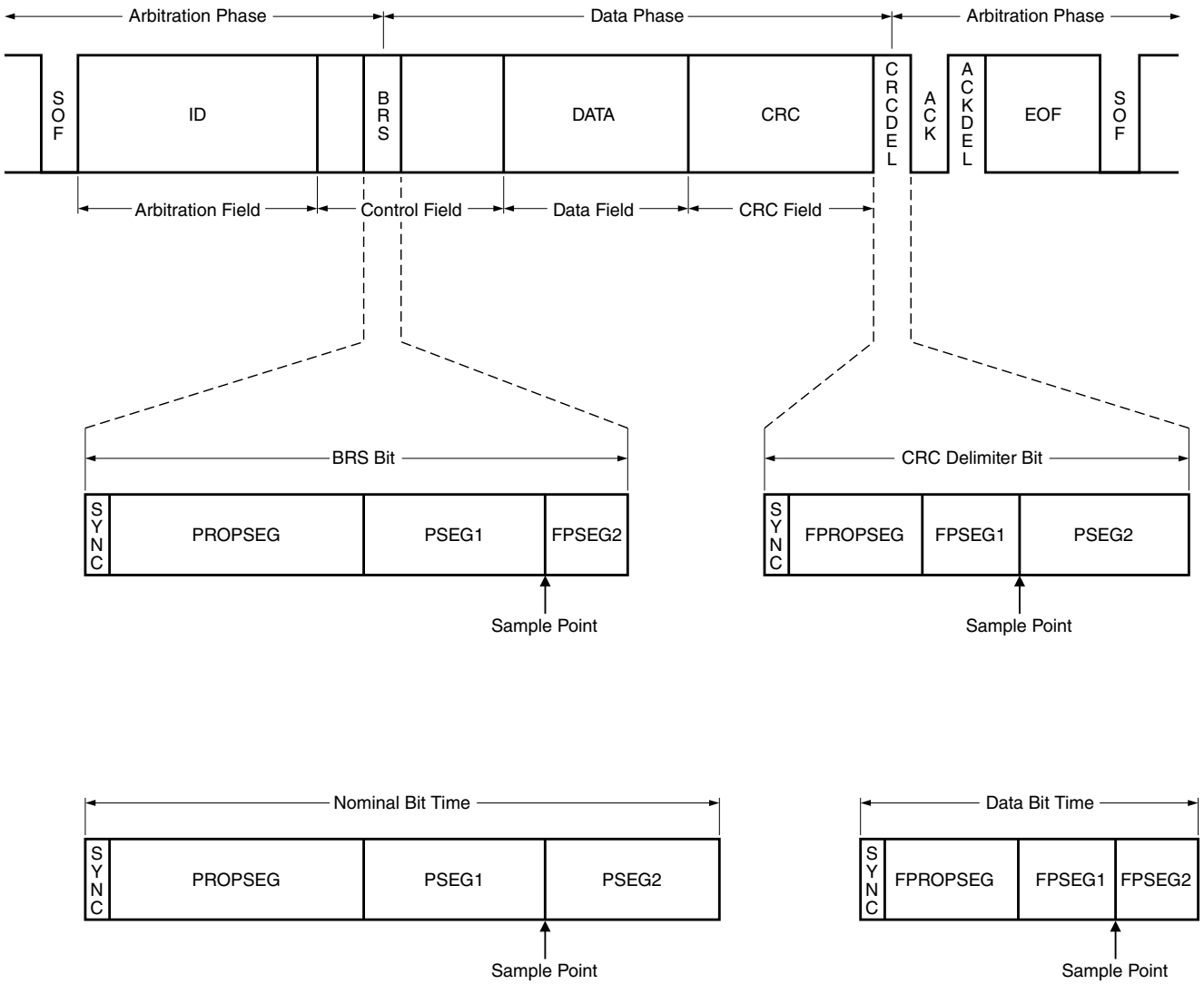


Figure 66-3. Bit rate switching mechanism for CAN FD messages

NOTE

In Classical CAN frames, the CRC delimiter is one single recessive bit. In CAN FD frames, the CRC delimiter may consist of one or two recessive bits. FlexCAN sends only one recessive bit as the CRC delimiter, but it accepts two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. As a receiver, FlexCAN sends its acknowledge bit after the first CRC delimiter bit. In CAN FD frames, FlexCAN accepts a two-bit dominant ACK slot as a valid ACK to compensate for phase shifts between the receivers.

The maximum configurable bit rate in the CAN FD data phase depends on the clock frequency of CAN_PE subblock. For example, with a CAN_PE clock frequency of 40 MHz and the shortest configurable bit time of 5 time quanta, the bit rate in the data phase is 8 Mbit/s.[3]

The value of the ESI bit is determined either by the transmitter's error state at the start of the transmission, if the frame is originated in the FlexCAN node, or by the original transmitting node in case FlexCAN is acting as a gateway for the message. If the transmitter is error passive, ESI is transmitted recessive; otherwise, it is transmitted dominant. The permutations of the relationship between the written value and the transmitted value of the ESI are shown in this table.

Table 66-8. Written versus transmitted values of ESI field

FlexCAN fault confinement status at start of frame	ESI bit Of Tx MB	Transmitted ESI
Error active	0	0 (Error Active)
Error passive	0	1 (Error Passive)
Error active	1	1 (Error Passive)
Error passive	1	1 (Error Passive)

There are different CRC polynomials for different CAN frame formats. The first polynomial, CRC_15, is used for all frames in Classical CAN format. The second, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. Each polynomial results in a Hamming distance of 6. At the start of the frame, all three CRC polynomials are calculated concurrently. The CRC sequence to be transmitted is selected by the values of the EDL bit and the DLC bit field. When receiving a message, FlexCAN decodes EDL and DLC to select the adequate CRC polynomial to check for a CRC error.

In CAN FD format frames, stuff bits are included in the bit stream for CRC calculation. In Classical CAN format frames, stuff bits are not included. After the transmission of the last bit relevant to the CRC calculation, the FDCRC register stores the calculated CRC for the transmitted message, with the adequate length in accordance to the type of message, for both CAN FD and non-FD messages. The CRCR register reports a valid CRC for Classical CAN messages only.

In CAN FD format frames, the CAN bit stuffing method is changed for the CRC sequence so that the stuff bits are inserted at fixed positions. When FlexCAN is transmitting a CAN FD frame, a fixed stuff bit is inserted just before the first bit of the CRC sequence, even if the last bits of the preceding field do not fulfill the CAN stuff

3. The frequency used in this example may not be supported on this chip; it is shown only to demonstrate how the maximum configurable bit rate is calculated.

condition. Additional stuff bits are inserted after each fourth bit of the CRC sequence. The value of any fixed stuff bit is the inverse value of its preceding bit. When FlexCAN is receiving a CAN FD frame, it discards the fixed stuff bits from the bit stream for the CRC check. A stuff error is detected if the fixed stuff bit has the same value as its preceding bit.

FlexCAN detects errors in CAN FD frames the same way as in Classical CAN frames. The error counters RXERRCNT and TXERRCNT in the ECR register accumulate the counts of Rx and Tx errors, respectively, for both FD and non-FD frames indiscriminately. There are two extra error counters (RXERRCNT_FAST and TXERRCNT_FAST) that accumulate Rx and Tx errors occurring in the data phase of CAN FD frames with the BRS bit set only. The rules for updating the error counters are the same for both CAN FD and non-FD frames (see ECR register).

Error Flags BITERR1, BITERR0, ACKERR, CRCERR, FRMERR, and STFERR in the ESR1 register report errors in both CAN FD and non-FD frames. They also generate the ERRINT interrupt if CTRL1[ERRMSK] is asserted. The ESR1 register has additional error flags (BITERR1_FAST, BITERR0_FAST, CRCERR_FAST, FRMERR_FAST, and STFERR_FAST) to individually indicate the occurrence of errors in the data phase of CAN FD frames with the BRS bit set. There is no ACKERR detected in the data phase of a CAN FD frame. Fault confinement status reported in ESR1[FLTCONF] is the same for both CAN FD and Classical CAN frames, and is based on RXERRCNT and TXERRCNT error counters only. Information contained in RXERRCNT_FAST and TXERRCNT_FAST counters may be considered as status to help detect the error nature related to the bit rate value.

When FlexCAN is in the data phase, either transmitting or receiving a CAN FD message, and detects an error, it immediately switches back to the arbitration phase and to the nominal rate to start an error flag.

Resynchronization and hard synchronization occur in CAN FD frames in the same way as in Classical CAN ones. Additionally, a hard synchronization is also performed at the recessive to dominant edge from EDL to R0 in CAN FD format frames. FlexCAN does not resynchronize while transmitting in the CAN FD data phase.

66.3.9.3 Transceiver delay compensation

The CAN FD protocol allows the transmission and reception of data at a higher bit rate than the nominal rate used in the arbitration phase when the message's BRS bit is set. This feature enables the use of rates up to 8 Mbps.

During the data phase of a CAN FD frame, the transmitter detects a bit error if it cannot receive its own latest transmitted bit at the sample point of that bit. When bit rate switching is enabled (BRS bit is asserted), the length of the CAN bit time in the data phase can become shorter than the transceiver's loop delay, thus impeding the correct comparison between the transmitted bit and the received bit within the current CAN bit time interval.

Note that the TDC process defines a secondary sample point where the transmitted bit is correctly compared with the received bit in order to check for bit errors.

The TDC mechanism can be enabled by FDCTRL[TDCEN] and is effective only during the data phase of FD frames having the BRS bit set. It has no effect either on non-FD frames, or on FD frames transmitted at normal bit rate. The TDC is active from the sample point of the BRS bit until the sample point of the CRC Delimiter bit, provided the respective message under transmission has the BRS bit set. When it is active, a comparison is done between the real received bit and the delayed transmitted bit, where the delay is calculated based on the measured transceiver loop delay.

NOTE

The actual value of the CRC Delimiter bit is disregarded by transmitters using the transceiver delay compensation mechanism. A global error at the end of the CRC field will cause the receivers to send error frames that the transmitter will detect during Acknowledge or End of Frame.

For every transmitted FD frame having the BRS bit set, the delay measurement is triggered by the transition from the recessive EDL bit to the dominant R0 bit (as shown in the next figure). The loop delay is measured in Protocol Engine (PE) clock periods (CANCLK, see Protocol timing), from the transmitted EDL-R0 edge to the received EDL-R0 edge. The position of the secondary sample point is defined by the measured loop delay time added to an offset value specified in FDCTRL[TDCOFF]. FDCTRL[TDCVAL] stores the result of this calculation. The TDCVAL value saturates at its maximum value of 63 CANCLK when the delay measurement is too long.

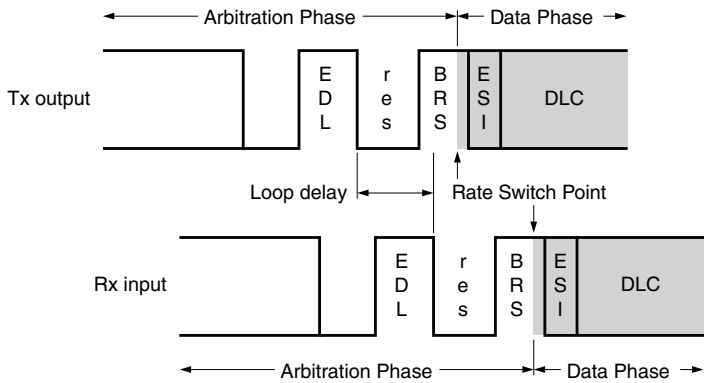


Figure 66-4. Transceiver loop delay measurement

The measured loop delay is not enough to be used to define the secondary sample point because it relates to the CAN bit edges. The transceiver delay compensation offset TDCOFF is used to shift the secondary sample point from the edge to an intermediate point inside the bit time, far away from its edges. Therefore, the TDCOFF value cannot be larger than the CAN bit duration in the data phase.

If the secondary sample point is set very near the CAN bit edge (SYNC field), then problems may occur during the bit sampling in the data phase. For the TDC to work reliably, the offset has to use optimal settings. To be sure the bit sampling is performed in the best region, the TDC offset should be configured as shown in this equation:

$$\text{Offset} = (\text{FPSEG1} + \text{FPROPSEG} + 2) \times (\text{FPRESDIV} + 1)$$

The following figure shows the SSP position when these settings are used.

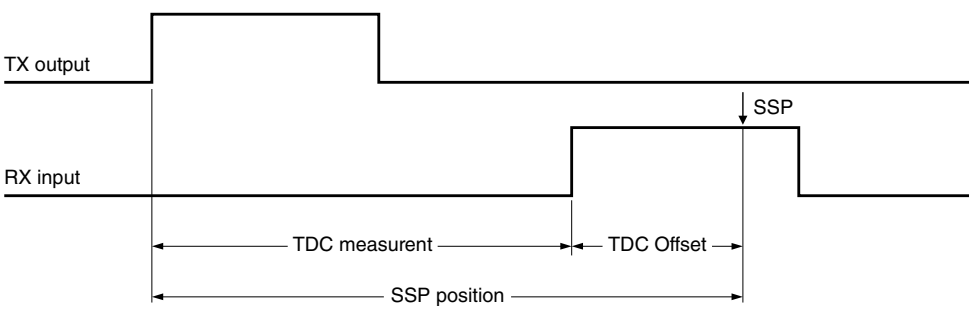


Figure 66-5. SSP position with optimal values

During the data phase of CAN FD frames with bit rate switching enabled, at the onset of every Tx CAN bit, the transmitted Tx bit value is temporarily stored in a buffer and a time countdown based on FDCTRL[TDCVAL] is started which ends with the comparison of the received Rx bit (delayed by the external loop delay plus the specified offset) with the stored Tx bit. If a bit error is detected at the secondary sample point, the FlexCAN issues an error flag to the CAN bus at the next sample point.

During the arbitration phase the delay compensation is always disabled. The maximum delay which can be compensated by the FlexCAN's transceiver delay compensation during the data phase is 3 CAN bit times – 2 Tq. Beyond this limit, the FDCTRL[TDCFAIL] flag is set to indicate when the transceiver delay compensation mechanism is out of range, unable to compensate the transceiver loop delay.

66.3.9.4 Remote frames

A remote frame is a special kind of frame. You can program a mailbox to be a remote request frame by configuring the mailbox as Transmit with the RTR bit set to one. After the remote request frame is transmitted successfully, the mailbox becomes a receive message buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on remote request storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]):

- If RRS is negated the frame's ID is compared to the IDs of the transmit message buffers with the CODE field 1010b. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame is received and matches a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as a normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0100b, 0010b, or 0110b. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No automatic remote response frame will be generated. The mask registers are used in the matching process.
- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the CPU. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote request frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

NOTE

There is no remote frame in the CAN FD format. The RTR bit is replaced by a fixed dominant RRS bit. FlexCAN receives and transmits remote frames in the Classical CAN format.

66.3.9.5 Overload frames

FlexCAN does transmit overload frames when the following conditions are detected on the CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

66.3.9.6 Time stamp

The value of the free running timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of move-in in the TIME STAMP field, providing network behavior with respect to time.

When CTRL2[TIMER_SRC] is asserted, the free running timer is continuously clocked by an external time tick.

When CTRL2[TIMER_SRC] is negated, the free running timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate.

The free running timer is not incremented during Disable, Doze, Stop, and Freeze modes. It can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in Control 1 register (CTRL1).

66.3.9.7 Protocol timing

The following figure shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source field CTRL1[CLKSRC] defines whether the internal clock is connected to the output of a crystal oscillator (oscillator clock) or to the peripheral clock. In order to guarantee reliable operation, the clock source should be selected when the module is in Disable mode (MCR[MDIS] is set).

NOTE

Please see the clock distribution chapter (module clocks table) to identify the proper clock source.

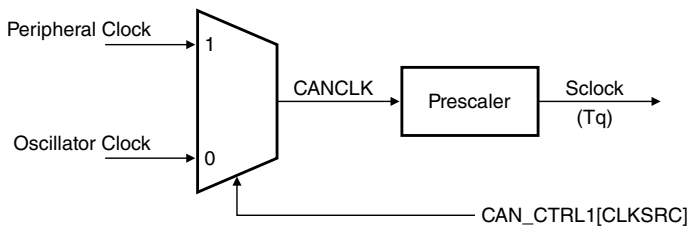


Figure 66-6. CAN engine clocking scheme

The oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than the peripheral clock.

The FlexCAN module supports a variety of means to set up bit timing parameters that are required by the CAN protocol. The Control 1 register (CTRL1) has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW.

The CAN Bit Timing register (CBT) extends the range of the CAN bit timing variables in CTRL1. The CAN FD Bit Timing register (FDCBT) provides a second set of CAN bit timing variables to be applied at the data phase of CAN FD frames with the Bit Rate Switch (BRS) set.

NOTE

When the CAN FD feature is enabled, always set CBT[BTF] and configure the CAN bit timing variables in CBT. See #d46430e7780a1310.

The PRESDIV field (as well as its extended range EPRESDIV and FDPRESDIV for the data phase bits of CAN FD messages) defines the prescaler value (see the equation below) that generates the serial clock (Sclock), whose period defines the time quantum used to compose the CAN waveform. A time quantum (Tq) is the atomic unit of time managed by the CAN engine.

$$Tq = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

The bit rate, which defines the rate the CAN message is either received or transmitted, is given by the formula:

$$\text{CAN Bit Time} = (\text{Number of Time Quanta in 1 bit time}) * Tq$$

$$\text{Bit Rate} = \frac{1}{\text{CAN Bit Time}}$$

A bit time is subdivided into three segments[1] (see Figure 66-7, Figure 66-8 and Table 66-9):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the propagation segment and the phase segment 1 of the CAN standard. It can be programmed by setting CTRL1[PROPSEG] and CTRL1[PSEG1] so that their sum (plus 2) is in the range of 2 to 16 time quanta. When CBT[BTF] is asserted, FlexCAN uses EPROPSEG and EPSEG1 fields from CBT register so that their sum (plus 2) is in the range of 2 to 96 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN uses FDPROPSEG and FDPSEG1 from FDCBT instead, so that their sum (plus 1) is in the range of 2 to 39 time quanta.
- Time Segment 2: This segment represents the phase segment 2 of the CAN standard. It can be programmed by setting CTRL1[PSEG2] (plus 1) to be 2 to 8 time quanta long. When CBT[BTF] is asserted, FlexCAN uses EPSEG2 fields of CBT register so that its value (plus 1) is in the range of 2 to 32 time quanta. For messages in CAN FD format with the BRS bit set, FlexCAN uses FDPSEG2 from FDCBT instead, so that its value (plus 1) is in the range of 2 to 8 time quanta. The time segment 2 cannot be smaller than the Information Processing Time (IPT), which value is 2 time quanta in FlexCAN.

NOTE

The bit time defined by the above time segments must not be smaller than five time quanta. For bit time calculations, use an Information Processing Time (IPT) of two, which is the value implemented in the FlexCAN module.

1. For further explanation of the underlying concepts, see ISO 11898-1. See also the CAN 2.0A/B protocol specification for bit timing.

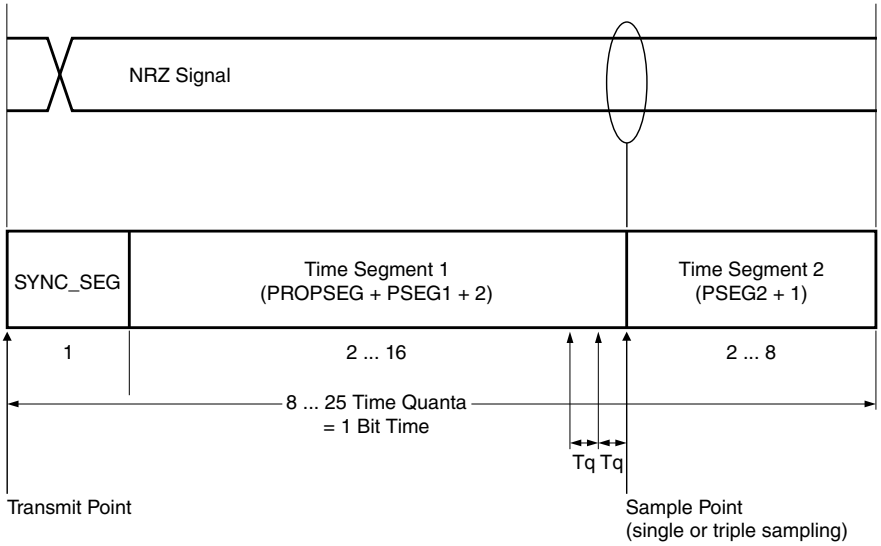


Figure 66-7. Segments within the bit time (example using CTRL1 bit timing variables for Classical CAN format)

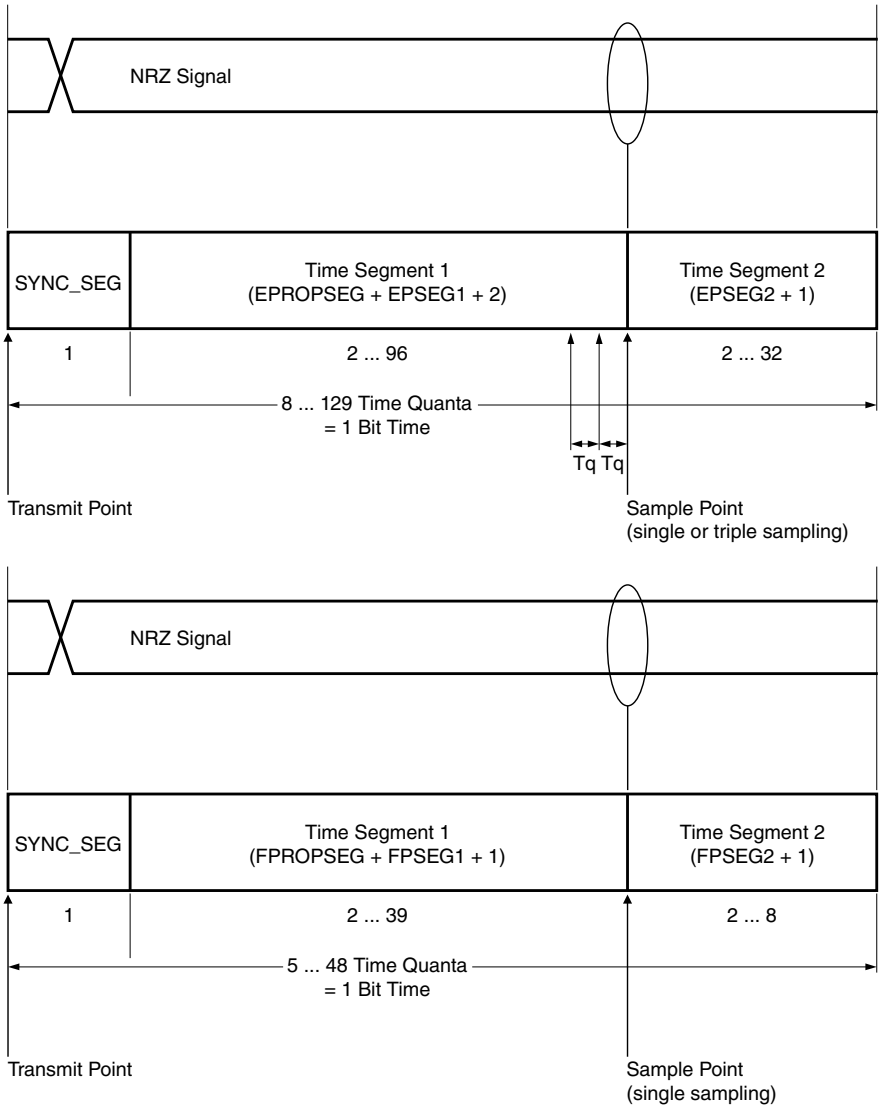


Figure 66-8. Segments within the bit time (example using CBT and FDCBT bit timing variables for CAN FD format)

Table 66-9. Time segment syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
TSEG1	Corresponds to the sum of PROPSEG and PSEG1.
TSEG2	Corresponds to the PSEG2 value.
Transmit point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The following table gives some examples of the CAN compliant segment settings for Classical CAN format (Bosch CAN 2.0B) (non-FD) messages.

Table 66-10. Bosch CAN 2.0B standard compliant bit time segment settings

Time segment 1	Time segment 2	Re-synchronization jump width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

Note

The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1).

Whenever CAN bit is used as a measure of time duration (for example, estimating the occurrence of a CAN bit event in a message), the number of peripheral clocks in one CAN bit (NumClkBit) can be calculated as:

$$\text{NumClkBit} = \frac{f_{SYS}}{f_{CANCLK}} \times (\text{PRESDIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

where:

- NumClkBit is the number of peripheral clocks in one CAN bit.
- f_{CANCLK} is the Protocol Engine (PE) Clock (see Figure 66-6), in Hz.
- f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz.
- PSEG1 is the value in CTRL1[PSEG1] field.
- PSEG2 is the value in CTRL1[PSEG2] field.
- PROPSEG is the value in CTRL1[PROPSEG] field.
- PRESDIV is the value in CTRL1[PRESDIV] field.

The formula above is also applicable to the alternative CAN bit timing variables described in the CAN Bit Timing register (CBT) and also to the CAN FD Bit Timing register (FDCBT).

For example, 180 CAN bits = (180 x NumClkBit) peripheral clock periods.

66.3.9.8 Arbitration and matching timing

During normal reception and transmission, the matching, arbitration, move-in, and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

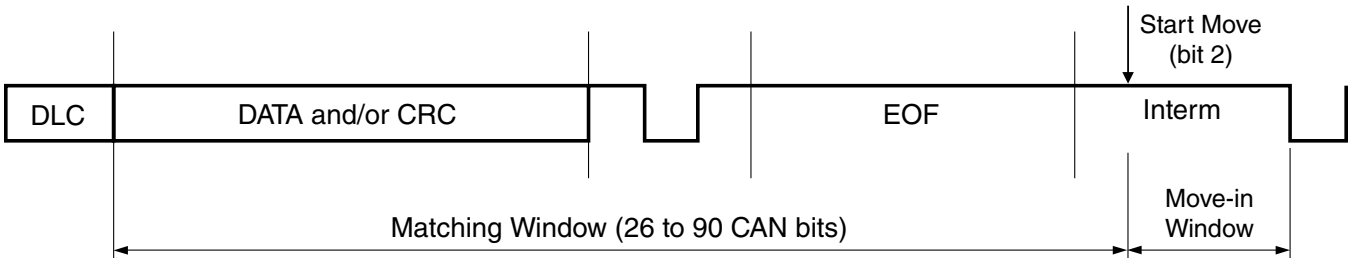


Figure 66-9. Matching and move-in time windows

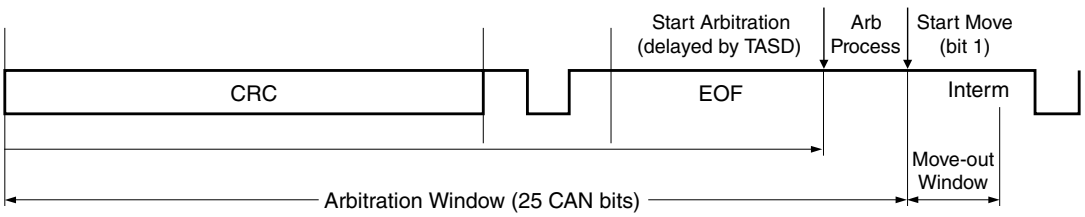


Figure 66-10. Arbitration and move-out time windows

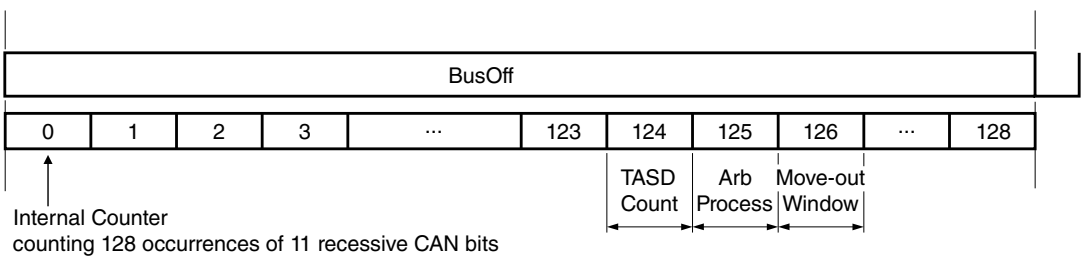


Figure 66-11. Arbitration at the end of bus off and move-out time windows

NOTE

In the preceding figures, the matching and arbitration timing does not take into account the delay caused by the concurrent memory access due to the CPU or other internal FlexCAN subblocks.

66.3.9.9 Tx arbitration start delay

CTRL2[TASD] (Tx Arbitration Start Delay) is a variable that indicates the number of CAN bits used by FlexCAN to delay the Tx arbitration process start point from the first bit of CRC field of the current frame. This variable can be written only in Freeze mode because it is blocked by hardware in other modes.

The transmission performance is impacted by the ability of the CPU to reconfigure message buffers (MBs) for transmission after the end of the internal arbitration process, where FlexCAN finds the winner MB for transmission (see Arbitration process). If the

arbitration ends too early before the first bit of Intermission field, then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is no longer the best candidate to be transmitted.

TASD is useful to optimize the transmission performance by defining the arbitration start point, as shown in the next figure, based on factors such as:

- Peripheral-to-oscillator clock ratio
- CAN bit timing variables that determine the CAN bit rate
- Number of message buffers (MBs) in use by the matching and arbitration processes

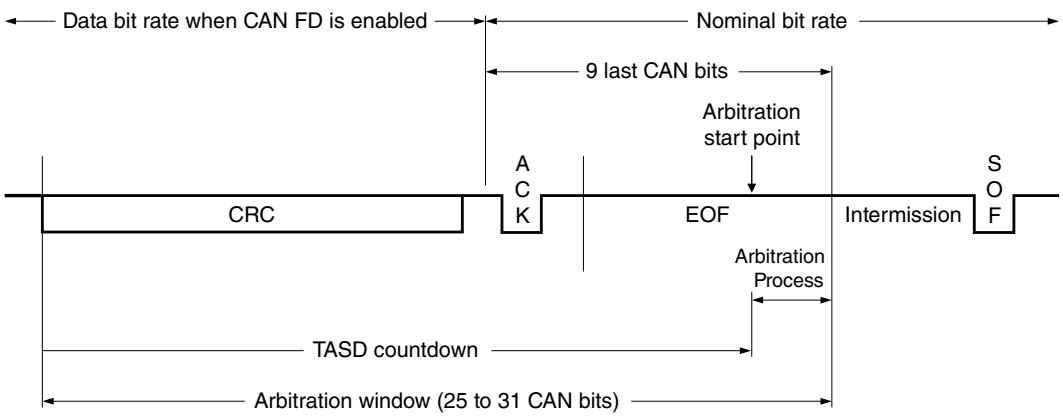


Figure 66-12. Optimal Tx Arbitration start point

The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and to the CAN bit rate, and inversely proportional to the peripheral clock frequency.

The optimal arbitration timing is that in which the last MB is scanned just before the first bit of the Intermission field of a CAN frame. For instance, if there are few MBs and the peripheral/oscillator clock ratio is high and the CAN baud rate is low, then the arbitration can be placed closer to the frame's end, adding more delay to its start point, and vice-versa.

If TASD is set to 0 then the arbitration start is not delayed and more time is reserved for arbitration. On the other hand, if TASD is close to 24 then the CPU can configure a Tx MB later and less time is reserved for arbitration. If too little time is reserved for arbitration the FlexCAN may be not be able to find a winner MB in time to be transmitted with the best chance to win the bus arbitration against external nodes on the CAN bus.

The optimal TASD value can be calculated as follows:

For CAN FD frames and (MAXMB + 1) ≤ NMB_{END}

$$TASD = 31 - \frac{2 * (MAXMB + 1) + 4}{CPCB_N}$$

For CAN FD frames and (MAXMB + 1) > NMB_{END}

$$TASD = 22 - \frac{2 * (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

For non-FD frames

$$TASD = 25 - \frac{2 * (MAXMB + 1) + 4}{CPCB}$$

where:

$$NMB_{END} = \frac{(9 * CPCB_N) - 4}{2}$$

$$BITRATE_N = \left(\frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] \times (EPRESDIV + 1)} \right)$$

$$BITRATE_F = \left(\frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] \times (FPRESDIV + 1)} \right)$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

- MAXMB is the value in CTRL1[MAXMB].
- NMB_{END} is the number of message buffers that can be scanned by the arbitration process during the 9 last CAN bits at the end of a frame (see the figure above).
- $BITRATE_N$ is the CAN bit rate in bits per second calculated by the nominal CAN bit time variables.
- $BITRATE_F$ is the CAN bit rate in bits per second calculated by the data CAN bit time variables.
- $CPCB_N$ is the number of peripheral clocks per CAN bit in nominal bit rate for CAN FD frames.
- $CPCB_F$ is the number of peripheral clocks per CAN bit in data bit rate for CAN FD frames.
- CPCB is the number of peripheral clocks per CAN bit for non-FD frames.
- f_{CANCLK} is the oscillator clock, in Hz.
- f_{SYS} is the peripheral clock, in Hz.
- EPSEG1 is the value in CBT[EPSEG1] (CTRL1[PSEG1] can also be used).
- EPSEG2 is the value in CBT[EPSEG2] (CTRL1[PSEG2] can also be used).

- EPROPSEG is the value in CBT[EPROPSEG] (CTRL1[PROPSEG] can also be used).
- EPRESDIV is the value in CBT[EPRESDIV] (CTRL1[PRESDIV] can also be used).
- FPSEG1 is the value in FDCBT[FPSEG1].
- FPSEG2 is the value in FDCBT[FPSEG2].
- FPROPSEG is the value in FDCBT[FPROPSEG].
- FPRESDIV is the value in FDCBT[FPRESDIV].

See also Protocol timing for more details.

The following tables give the TASD value calculated for some configuration cases.

Case 1:

- Clock ratio = 2:1 (example: peripheral clock 80 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 66-11. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	24	8.0
64	23	8.0

Case 2:

- Clock ratio = 1:1 (example: peripheral clock 40 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 66-12. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	23	6.67
54	22	5.0
64	21	3.33

Case 3:

- Clock ratio = 2:1 (example: peripheral clock 40 MHz and oscillator clock 20 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 66-13. TASD values:

Number of message buffers	TASD value	Maximum bit rate in data phase (MBd)
16	24	Invalid
32	23	4.0
54	22	4.0
64	21	3.33

66.3.10 Clocks

The following table describes the clock sources for FlexCAN. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 66-14. FlexCAN clocks

Clock name	Description
ipg_clk	Peripheral clock
ipg_clk_chi	Control Host Interface (CHI) clock
ipg_clk_pe	Protocol Engine (PE) clock
ipg_clk_pe_nogate	Protocol Engine clock (no gating)
ipg_clk_s	Peripheral access clock

66.3.10.1 Clock domains and restrictions

The FlexCAN module has two clock domains asynchronous to each other:

- The bus domain feeds the Control Host Interface (CHI) submodule.
- The oscillator domain feeds the CAN Protocol Engine (PE) submodule.

When CTRL1[CLKSRC] is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the bus domain clock frequency must always be greater than the oscillator domain clock frequency.

NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

When doing matching and arbitration, FlexCAN needs to scan the whole message buffer memory during the time slot of one CAN frame, comprised of a number of CAN bits. In order to have sufficient time to do that, the following requirements must be observed:

- The peripheral clock frequency cannot be smaller than the oscillator clock frequency.
- There must be a minimum number of peripheral clocks per CAN bit, as specified in the table shown below.

Table 66-15. Minimum number of peripheral clocks per CAN bit for Classical CAN format

Number of mailboxes	Value of MCR[RFEN]	Minimum number of peripheral clocks per CAN bit
16	0	16
32	0	16
64	0	25
16	1	16
32	1	17
64	1	30

For classical frame format, the minimum number of peripheral clocks per CAN bit specified in the preceding table determines the minimum peripheral clock frequency for a given number of mailboxes and for an expected CAN bit rate. The CAN bit rate depends on the number of time quanta in a CAN bit, which can be defined by adjusting one or more of the bit timing values contained in either the Control 1 register (CTRL1) or CAN Bit Time register (CBT). The time quantum (Tq) is defined in Protocol timing. The minimum number of time quanta per CAN bit must be 8; therefore, the oscillator clock frequency should be at least 8 times the CAN bit rate.

For CAN FD frame format, there are some constraints that need to be satisfied. The number of peripheral clocks per CAN bit in nominal bit rate (NumClkNomBit) can be calculated by the equation below.

$$\text{NumClkNomBit} = \frac{f_{SYS}}{f_{CANCLK}} \times (\text{PRESDIV} + 1) \times (\text{PROPSEG} + \text{PSEG1} + \text{PSEG2} + 4)$$

$$= \frac{f_{SYS}}{\text{NomBitRate}}$$

where PRESDIV, PSEG1, and PSEG2 are CAN bit time values in CTRL1 register. Alternatively, EPRESDIV, EPSEG1, and EPSEG2 values in CBT register can be used instead. NumClkNomBit can also be calculated as a function of the expected nominal bit rate used in the arbitration phase (NomBitRate), as shown in the equation above.

The number of CAN bits in the data phase of an FD frame with the BRS bit set (fast CAN bits, in short) depends on the number of data bytes in the payload. The number of fast CAN bits (NumOfFastBits) can be determined in the table below. The fewer the number of data bytes, the fewer the number of fast CAN bits, and less time is available for FlexCAN to scan the whole message buffer memory during the internal matching and arbitration processes.

Table 66-16. Number of fast CAN bits in a CAN FD frame

Minimum number of data bytes	DLC field	NumOfFastBits
0	0h	21
1	1h	29
2	2h	37
3	3h	45
4	4h	53
5	5h	61
6	6h	69
7	7h	77
8	8h	85
12	9h	117
16	Ah	149
20	Bh	186
24	Ch	218
32	Dh	282
48	Eh	410
64	Fh	538

The critical part of a CAN FD frame is during the data phase, where the CAN bit rate is faster than in the arbitration phase. The minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated to guarantee that enough time is available for FlexCAN to scan the message buffer memory during reception and transmission. The equation below calculates this constraint.

$$\text{MinNumClkFastBit}_A = \frac{(\ 8.5 \times \text{MaxNumOfMb}\ ) + 64 - (\ 9 \times \text{NumClkNomBit}\ )}{\text{NumOfFastBits}}$$

where MaxNumOfMb is the maximum number of available mailboxes defined in MCR[MAXMB].

The clock domain crossing circuit between the CHI and PE subblocks also imposes a minimum number of peripheral clocks per fast CAN bit for the handshake mechanism to work properly without losing status information through the interface, as shown in the equation below.

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{f_{SYS}}{f_{CANCLK}}\right)$$

Therefore, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) is determined by the larger of the two values calculated above.

$$MinNumClkFastBit = Maximum\ (\ MinNumClkFastBit_A, MinNumClkFastBit_B\ )$$

Then, the maximum CAN bit rate in the data phase of CAN FD frames (DataBitRateMAX) can be calculated as below.

$$DataBitrate_{MAX} = \frac{f_{CANCLK}}{ROUNDUP\left(\frac{MinNumClkFastBit \times f_{CANCLK}}{f_{SYS}}\right)}$$

The peripheral and oscillator clock frequencies, the maximum number of mailboxes, and the expected nominal bit rate affect the maximum data bit rate attainable by FlexCAN in CAN FD mode. Also, the data bit rate depends on the minimum payload size of FD frames used in a given application.

To illustrate how the CAN FD bit rate is affected by the configuration of FlexCAN variables, an application example with the peripheral and oscillator clock frequencies set to 50 MHz and 40 MHz, respectively, is considered.

1. Considering the nominal bit rate as 1 Mbps, the number of peripheral clocks per CAN bit in nominal bit rate is calculated as below.

$$NumClkNomBit = \frac{50 \times 10^6}{1 \times 10^6} = 50$$

2. The number of fast CAN bits (NumOfFastBits) is determined in the table presented above. For example, if the minimum payload in FD frames is 8 bytes, then there are 85 CAN bits in the data phase.
3. Assuming the maximum number of mailboxes is 96, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated.

$$MinNumClkFastBit_A = \frac{(8.5 \times 96) + 64 - (9 \times 50)}{85} = 5.06$$

$$MinNumClkFastBit_B = 3 \times \left(1 + \frac{50}{40}\right) = 6.75$$

$$MinNumClkFastBit = Maximum\ (5.06, 6.75) = 6.75$$

4. The maximum CAN bit rate in the data phase can finally be found.

$$\text{DataBitRate}_{MAX} = \frac{40 \times 10^6}{\text{ROUNDUP}\left(\frac{6.75 \times 40 \times 10^6}{50 \times 10^6}\right)} = 6.667 \text{ Mbps}$$

As demonstrated in this example, even though the oscillator clock frequency (40 MHz) is adequate to generate a data rate of 8 Mbps in CAN FD mode, the specific FlexCAN configuration limits this rate to 6.667 Mbps. This limitation is mainly due to the low peripheral clock frequency that imposes the MinNumClkFastBitB bound.

The table below shows the maximum data rate for CAN FD according to clock frequencies, payload size, and number of available mailboxes. See in this table that, for some cases, if the number of available mailboxes is reduced, the FlexCAN can then achieve a data rate up to 8 Mbps.

Table 66-17. Maximum CAN bit rate in data phase on CAN FD frames

Peripheral clock frequency (MHz)	Payload size	Number of available mailboxes	Maximum data rate (Mbps)
40	8	94	6.667
40	8	114	5.0
40	12	117	6.667
40	12	128	5.714
50	12 to 64	128	6.667
60	8	126	8.0
60	12	128	8.0
67	6	128	8.0
80	3	128	8.0
100	0	128	8.0

66.3.11 Reset

You can reset FlexCAN in the following ways:

1. Chip level hard reset, which resets all memory-mapped registers asynchronously.
2. MCR[SOFTRST], which resets some of the memory-mapped registers synchronously. See Table 66-19 to see what registers are affected by soft reset.

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. MCR[SOFTRST] remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset cannot be applied while clocks are shut down in a low-power mode. The low-power mode should be exited and the clocks resumed before applying soft reset.

The clock source should be selected when the module is in Disable mode (see CTRL1[CLKSRC] description in #d46430e1242a1310). After the clock source is selected and the module is enabled (MCR[MDIS] bit negated), FlexCAN automatically enters Freeze mode. In Freeze mode, FlexCAN is un-synchronized to the CAN bus, MCR[HALT] and MCR[FRZ] are set, the internal state machines are disabled, then MCR[FRZACK] and MCR[NOTRDY] are set. The Tx pin is in recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Note that the message buffers and the Rx Individual Mask registers are not affected by reset, so they are not automatically initialized.

66.3.12 Interrupts

The module has many interrupt sources: interrupts due to message buffers and interrupts due to the ORed interrupts from MBs, Bus Off, Bus Off Done, Error, Error Fast (errors detected in the data phase of CAN FD format messages with the BRS bit set), Wake Up, Tx Warning, and Rx Warning.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has an assigned flag bit in the IFLAG registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes one to it (unless another interrupt is generated at the same time).

Note

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled (MCR[RFEN] = 1) and DMA is disabled (MCR[DMA] = 0), the interrupts corresponding to MBs 0 to 7 have different meanings. Bit 7 of the IFLAG1 register becomes the "FIFO Overflow" flag; bit 6 becomes the "FIFO Warning" flag, bit 5 becomes the "Frames Available in FIFO" flag and bits 4-0 are unused. See the description of #d46430e5234a1310 for more information.

If both Rx FIFO and DMA are enabled (MCR[RFEN] and MCR[DMA] = 1) the FlexCAN does not generate any FIFO interrupt. Bit 5 of the IFLAG1 register still indicates "Frames Available in FIFO" and generates a DMA request. Bits 7, 6, 4-0 are unused.

CAUTION

FIFO cannot be enabled when CAN FD feature is enabled.

For a combined interrupt where multiple MB interrupt sources are OR'd together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the IFLAG registers to determine which MB or FIFO source caused the interrupt.

The following interrupt sources generate interrupts like the MB interrupt sources, and can be read from ESR1 register

- Bus Off
- Bus Off Done
- Error
- Error Fast
- Wake Up
- Tx Warning
- Rx Warning

The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in the CTRL1 register; the Wake-Up interrupt mask bit is located in MCR.

66.3.13 Bus interface

CPU access to FlexCAN registers is subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in Table 66-19 in Supervisor Mode or with S only) results in an access error.
- Read and write access to implemented reserved address space results in an access error.
- Write access to positions whose bits are all currently read-only results in an access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. See register and field descriptions for details.
- Read and write access to unimplemented address space results in access error.
- Read and write access to RAM located positions during Low Power mode results in an access error.
- It is possible for the RXIMR memory region to be considered as general purpose memory and available for access. There are two ways of doing this:
  a. If MCR[IRMQ] is cleared, the individual masks (RXIMR) are disabled. In this case the RXIMR memory region is considered as general purpose memory.
  b. If MCR[MAXMB] is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose

RAM space. Note that reserved words within RAM cannot be used. As an example, suppose FlexCAN's RAM can support up to 16 MBs, CTRL2[RFFN] is 0h, and MCR[MAXMB] is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0080h, and the space 0080h–008Fh is used by the one MB. The memory space 0090h–017Fh is available. The space 0180h–087Fh is reserved. The space 0880h–0883h is used by the one individual mask and the available memory in the mask registers space would be 0884h–08BFh. From 08C0h–09DFh there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.

c. If MCR[FDEN] is enabled, general purpose memory can be used out of Freeze mode only.

66.3.14 Detection and correction of memory errors

NOTE

All FlexCAN memory must be initialized before starting its operation in order to have the parity bits in memory properly updated. CTRL2[WRMFRZ] grants write access to all memory positions that require initialization, ranging from 080h–ADFh and from F28h–FFFh when the CAN FD feature is enabled. The RXMGMASK, RX14MASK, RX15MASK, and RXFGMASK registers need to be initialized as well. MCR[RFEN] must not be set during memory initialization.

FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to five parity bits that ensure a Hamming distance of four. The error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in two bits can be detected but not corrected (non-correctable errors). Errors in more than two bits may not be detected. In case of non-correctable errors, the corrupted data is not changed by the error correction logic. When a read access is performed, the parity bits are used to calculate a syndrome, which indicates the error in each byte.

FlexCAN detects a noncorrectable error in the event either an all-zeros or an all-ones read occurs. See description of #d46430e10651a1310.

Memory errors are indicated to the host through status register (Error Status Register (ERRSR)) and bus transfer errors, and reported through report registers (Error Report Address register (RERRAR), Error Report Data register (RERRDR) and Error Report Syndrome register (RERRSYNR)).

The error detection and correction mechanism can be activated or not, controlled by MECR[ECCDIS]. When disabled, updates on indications and reporting registers are stopped, but the parity bits are still calculated and written along with data in memory write operations to ensure that memory has consistent parity bits associated with the data.

To avoid accidentally changing the critical error correction configuration, this protocol must be followed to enable the update of the Memory Error Control register (MECR):

1. By default, CTRL2[ECRWRE] is zero and MECR[ECRWRDIS] is one.
2. Set CTRL2[ECRWRE].
3. Clear MECR[ECRWRDIS].
4. All writes to Memory Error Control register (MECR) must keep MECR[ECRWRDIS] cleared.
5. After configuration is done, lock the Memory Error Control register (MECR) by either setting MECR[ECRWRDIS] or clearing CTRL2[ECRWRE].

66.3.14.1 Sources of the memory access

The FlexCAN memory can be accessed by two major sources (or requestors):

- Host (CPU): the largest word accessed is 32-bit.
- FlexCAN internal processes (Rx matching, Tx arbitration, move-in on reception, move-out on transmission): the largest word accessed is 64-bit.

The way that noncorrectable errors are indicated and reported depends on the source of access.

66.3.14.2 Error indication

Memory errors are indicated by flags HANCEIF, FANCEIF, and CEIF in the Error Status Register (ERRSR). Noncorrectable errors detected in memory reads requested by the host are indicated separately than the ones detected in requests by FlexCAN internal processes. FlexCAN makes no distinction of the source of the access when correctable errors are detected. There are three independent flags for these three cases, and each flag will raise an interrupt unless it is masked by mask bits in Memory Error Control Register (MECR). If both noncorrectable and correctable errors are found in different bytes in the same read operation, both flags are set.

A noncorrectable error detected in host access is also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If the flag bit ERRSR[HANCEIF] is not masked, the same noncorrectable error will raise a bus transfer error and an interrupt request.

Each indication flag has one overrun flag in Error Status Register (ERRSR). The overrun flags do not request interrupts. Overrun flags for noncorrectable errors indicate that other errors of the same nature were detected after current error being treated; overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated.

This is the recommended handling sequence for error indication:

1. Get error report information from report registers.
2. Use this information to take proper measures in the application.
3. Clear the HANCEIF, FANCEIF, and CEIF flags.
4. If the overrun flag is active:
   a. Alert application that at least one error could not be managed.
   b. Clear the overrun flag.

The FlexCAN internal processes can access memory in transactions larger than 32 bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses. If errors are found in two or more 32-bit words the interrupt and overrun flags are set simultaneously.

66.3.14.3 Error reporting

The report registers Error Report Address register (RERRAR), Error Report Data register (RERRDR), and Error Report Syndrome register (RERRSYNR) provide detailed information about the address read, raw data, and syndrome read with error, and are indicated by the flags described in Error indication. The address, data, and syndrome registers are updated simultaneously along with the error flags, according to these rules:

1. If either of the two non-correctable error flags is currently set, the report registers are not updated (the previous non-correctable error reporting is preserved).
2. Otherwise (either no error flag is currently set or only the correctable error flag is currently set), the report registers are updated according to the new error; or according to the most severe of new errors if non-correctable and correctable errors are simultaneously detected.

Reporting of errors detected in accesses larger than 32-bit follows the rules described in Error indication and in the description of #d46430e9994a1310.

The address reported in RERRAR and defined in ERRIAR are not the same listed in the module memory map. The relation between the reported addresses and the respective ones in the module memory map is shown in the description of #d46430e8929a1310.

Addresses reported when reading memory portions organized as FIFOs, such as the Rx FIFO Structure and the Rx FIFO Information Register (RXFIR), refer to the address of the specific entry accessed in the FIFO, not to the FIFO base address.

To assure coherence of the error report registers it is necessary to turn off the report update by setting MECR[RERRDIS] before reading the report registers.

66.3.14.4 Response to errors

Correctable errors have no consequence on FlexCAN operation because affected data is corrected before its use by the host or FlexCAN internal processes.

For host-initiated reads, a noncorrectable error may affect the host, but does not affect FlexCAN operation.

Non-correctable errors detected on memory reads requested by the FlexCAN internal processes may result in incorrect operation depending on the state of MECR[NCEFAFRZ], as follows:

- During reception (either matching or move-in processes), when a noncorrectable error occurs, an incorrect destination may be selected to store the incoming frame, a corrupted frame may be stored in the correct destination, or both. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters in Freeze mode to prevent corrupted data from being treated as valid by FlexCAN internal processes. When MECR[NCEFAFRZ] is negated, FlexCAN continues working and a corrupted frame is received.
- During arbitration process, when a noncorrectable error occurs, either a non-highest priority Tx message buffer may be mistakenly selected for transmission or its data may be corrupted. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode before starting the move-out. When MECR[NCEFAFRZ] is negated, FlexCAN proceeds to move-out with a corrupted frame that will be transmitted on the CAN bus.
- During move-out process, when a noncorrectable error occurs, a corrupted frame is copied from the selected Tx MB that won the arbitration to the Tx SMB for transmission. If MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode before starting the transmission. When MECR[NCEFAFRZ]

is negated, the corrupted frame is transferred from the Tx SMB to the Protocol Engine (PE) sub-block and is transmitted on the CAN bus.

- A noncorrectable error can also be detected beyond the move-out process, when Tx data is read from Tx SMB (buffer located in RAM) to be transferred to the PE subblock for transmission. In this case, a frame with corrupted ID and/or data is transmitted on the CAN bus.

  To prevent the frame from being successfully received by the external nodes, FlexCAN inverts all bits in the CRC field (CRC sequence plus CRC delimiter), and transmits an error flag just after CRC delimiter as a result of self-detecting a Bit1 error and a form error due to the CRC field inversion. When MECR[NCEFAFRZ] is set, FlexCAN stops operation automatically and enters Freeze mode just after the error frame. When MECR[NCEFAFRZ] is negated, FlexCAN may attempt to re-transmit the same frame, as long as no other higher priority Tx MB is subsequently configured for transmission.

  In the event the noncorrectable error persists, FlexCAN eventually reaches the Bus Off state because of consecutive error detections. ECR[TXERRCNT] is updated every time FlexCAN inverts the CRC field, causing errors as described above.

When MECR[NCEFAFRZ] is set and FlexCAN enters Freeze mode, only the CPU can cause FlexCAN to exit Freeze mode and resume Normal mode. The assertion of MECR[NCEFAFRZ] is the only way to prevent corrupted frames from being transmitted on the CAN bus up to the move-out internal process.

The error report registers can provide information to the application for customized management of these situations.

66.3.14.5 Error injection

The error injection registers ERRIAR, ERRIDPR, and ERRIPPR are used to inject errors in memory reads in order to force errors and consequently update the indication and reporting registers. The relation between the error injection addresses and the respective ones in the module memory map is shown in the description of #d46430e8929a1310.

The injection is done by flipping the data and parity bits correspondent to the bits in 1 in ERRIDPR and ERRIPPR. Injection can be selected specifically for memory accesses requested by the host or by FlexCAN internal processes.

In case of accesses larger than 32-bits, the EXTERRIE bit in Memory Error Control register (MECR) extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

NOTE

It is very unlikely, but error injection may correct a bit with error. This will not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, MECR[HAERRIE] and MECR[FAERRIE] must be cleared when configuring the memory injection registers.

66.4 FlexCAN signal descriptions

The FlexCAN module has two I/O signals connected to the external chip pins. These signals are summarized in the following table and described in more detail in the next subsections.

Table 66-18. FlexCAN signal descriptions

Signal	Description	I/O
CAN Rx	CAN receive pin	Input
CAN Tx	CAN transmit pin	Output

66.4.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

66.4.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

66.5 Initialization/application information

This section provides instructions for initializing the FlexCAN module.

66.5.1 FlexCAN initialization sequence

For any configuration change/initialization it is required that FlexCAN be put into Freeze mode (see Freeze mode). Note that the module needs to be initialized after every reset.

The following is a generic initialization sequence applicable to the FlexCAN module:

1. Initialize the Module Configuration register (MCR).
   a. Enable the individual filtering per MB and reception queue features by setting IRMQ.
   b. Enable the warning interrupts by setting WRNEN.
   c. If required, disable frame self reception by setting SRXDIS.
   d. Enable the Rx FIFO by setting RFEN.
   e. If Rx FIFO is enabled and DMA is required, set DMA.
   f. Enable the abort mechanism by setting AEN.
   g. Enable the local priority feature by setting LPRIOEN.
2. Initialize the Control 1 register (CTRL1) and optionally the CAN Bit Timing register (CBT). Initialize also the CAN FD CAN Bit Timing register (FDCBT).
   a. Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, and RJW.
   b. Optionally determine the bit timing parameters: EPROPSEG, EPSEG1, EPSEG2, and ERJW.
   c. Determine the CAN FD bit timing parameters: FPROPSEG, FPSEG1, FPSEG2, and FRJW.
   d. Determine the bit rate by programming the PRESDIV field and optionally the EPRESDIV field.
   e. Determine the CAN FD bit rate by programming the FPRESDIV field.
   f. Determine the internal arbitration mode (LBUF).
3. All FlexCAN memory must be initialized if Error Code Correction (ECC) is enabled. See Detection and correction of memory errors.
4. Initialize the message buffers.
   a. The control and status word of all message buffers must be initialized.
   b. If Rx FIFO was enabled, the ID filter table must be initialized.
   c. Other entries in each message buffer should be initialized as required.
5. Initialize the Rx Individual Mask registers (RXIMRn).
6. Set required interrupt mask bits in
   - IMASK registers (for all MB interrupts)
   - MCR register (for Wake-Up interrupt)
   - CTRL1 / CTRL2 registers (for Bus Off and Error interrupts)
7. Negate MCR[HALT].

After the last step listed above, FlexCAN attempts to synchronize to the CAN bus.

66.6 Memory map/register definition

This section describes the registers and data structures in the FlexCAN module. The base address of the module depends on the particular memory map of the chip.

66.6.1 FlexCAN memory mapping

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address offset 0080h.

Each individual register is identified by its complete name and the corresponding mnemonic . The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming MCR[SUPV]. These registers are identified as S/U in the Access column of Table 66-19.

NOTE

An invalid register access will result in a bus error. This includes reading a write-only register, writing a read-only register, or accessing an invalid address.

Table 66-19. Register access and reset information

Register	Access type	Affected by hard reset	Affected by soft reset
Module Configuration Register (MCR)	S	Yes	Yes
Control 1 register (CTRL1)	S/U	Yes	No
Free Running Timer register (TIMER)	S/U	Yes	Yes
Rx Mailboxes Global Mask register (RXMGMASK)	S/U	No	No
Rx Buffer 14 Mask register (RX14MASK)	S/U	No	No
Rx Buffer 15 Mask register (RX15MASK)	S/U	No	No
Error Counter Register (ECR)	S/U	Yes	Yes
Error and Status 1 Register (ESR1)	S/U	Yes	Yes
Interrupt Masks 2 register (IMASK2)	S/U	Yes	Yes
Interrupt Masks 1 register (IMASK1)	S/U	Yes	Yes
Interrupt Flags 2 register (IFLAG2)	S/U	Yes	Yes
Interrupt Flags 1 register (IFLAG1)	S/U	Yes	Yes
Control 2 Register (CTRL2)	S/U	Yes	No
Error and Status 2 Register (ESR2)	S/U	Yes	Yes
CRC Register (CRCR)	S/U	Yes	Yes
Rx FIFO Global Mask register (RXFGMASK)	S/U	No	No

Table continues on the next page...

Table 66-19. Register access and reset information (continued)

Register	Access type	Affected by hard reset	Affected by soft reset
Rx FIFO Information Register (RXFIR)	S/U	No	No
CAN Bit Timing Register (CBT)	S/U	Yes	No
Message buffers	S/U	No	No
Rx Individual Mask Registers	S/U	No	No
Memory Error Control register (MECR)	S/U	Yes	Yes
Error Injection Address register (ERRIAR)	S/U	Yes	Yes
Error Injection Data Pattern register (ERRIDPR)	S/U	Yes	Yes
Error Injection Parity Pattern register (ERRIPPR)	S/U	Yes	Yes
Error Report Address register (RERRAR)	S/U	Yes	Yes
Error Report Data register (RERRDR)	S/U	Yes	Yes
Error Report Syndrome register (RERRSYNR)	S/U	Yes	Yes
Error Status register (ERRSR)	S/U	Yes	Yes
CAN FD Control register (FDCTRL)	S/U	Yes	No
CAN FD Bit Timing register (FDCBT)	S/U	Yes	No
CAN FD CRC register (FDCRC)	S/U	Yes	Yes

The FlexCAN module can store CAN messages for transmission and reception using mailboxes and Rx FIFO structures.

66.6.2 CAN register descriptions

The table below shows the FlexCAN memory map.

The address range from offset 80h–47Fh allocates the sixty-four 128-bit message buffers (MBs).

The memory maps for the message buffers are in FlexCAN message buffer memory map.

66.6.2.1 CAN memory map

FLEXCAN1 base address: 400C_4000h

FLEXCAN2 base address: 400C_8000h

FLEXCAN3 base address: 40C3_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Module Configuration register (MCR)	32	RW	5980_000F
4	Control 1 register (CTRL1)	32	RW	0000_0000
8	Free Running Timer (TIMER)	32	RW	0000_0000
10	Rx Mailboxes Global Mask register (RXMGMASK)	32	RW	Table 66-19
14	Rx 14 Mask register (RX14MASK)	32	RW	Table 66-19
18	Rx 15 Mask register (RX15MASK)	32	RW	Table 66-19
1C	Error Counter (ECR)	32	RW	0000_0000
20	Error and Status 1 register (ESR1)	32	R2C	0000_0000
24	Interrupt Masks 2 register (IMASK2)	32	RW	0000_0000
28	Interrupt Masks 1 register (IMASK1)	32	RW	0000_0000
2C	Interrupt Flags 2 register (IFLAG2)	32	W1C	0000_0000
30	Interrupt Flags 1 register (IFLAG1)	32	W1C	0000_0000
34	Control 2 register (CTRL2)	32	RW	0080_0000
38	Error and Status 2 register (ESR2)	32	RO	0000_0000
44	CRC register (CRCR)	32	RO	0000_0000
48	Rx FIFO Global Mask register (RXFGMASK)	32	RW	Table 66-21
4C	Rx FIFO Information register (RXFIR)	32	RO	Table 66-21
50	CAN Bit Timing register (CBT)	32	RW	0000_0000
880 - 97C	Rx Individual Mask registers (RXIMR0 - RXIMR63)	32	RW	Table 66-21
9E0	Glitch Filter Width Register (GFWR)	32	RW	0000_007F
AE0	Memory Error Control register (MECR)	32	RW	800C_0080
AE4	Error Injection Address register (ERRIAR)	32	RW	0000_0000
AE8	Error Injection Data Pattern register (ERRIDPR)	32	RW	0000_0000
AEC	Error Injection Parity Pattern register (ERRIPPR)	32	RW	0000_0000
AF0	Error Report Address register (RERRAR)	32	RO	0000_0000
AF4	Error Report Data register (RERRDR)	32	RO	0000_0000
AF8	Error Report Syndrome register (RERRSYNR)	32	RO	0000_0000
AFC	Error Status register (ERRSR)	32	W1C	0000_0000
C00	CAN FD Control register (FDCTRL)	32	RW	8000_0100
C04	CAN FD Bit Timing register (FDCBT)	32	RW	0000_0000
C08	CAN FD CRC register (FDCRC)	32	RO	0000_0000

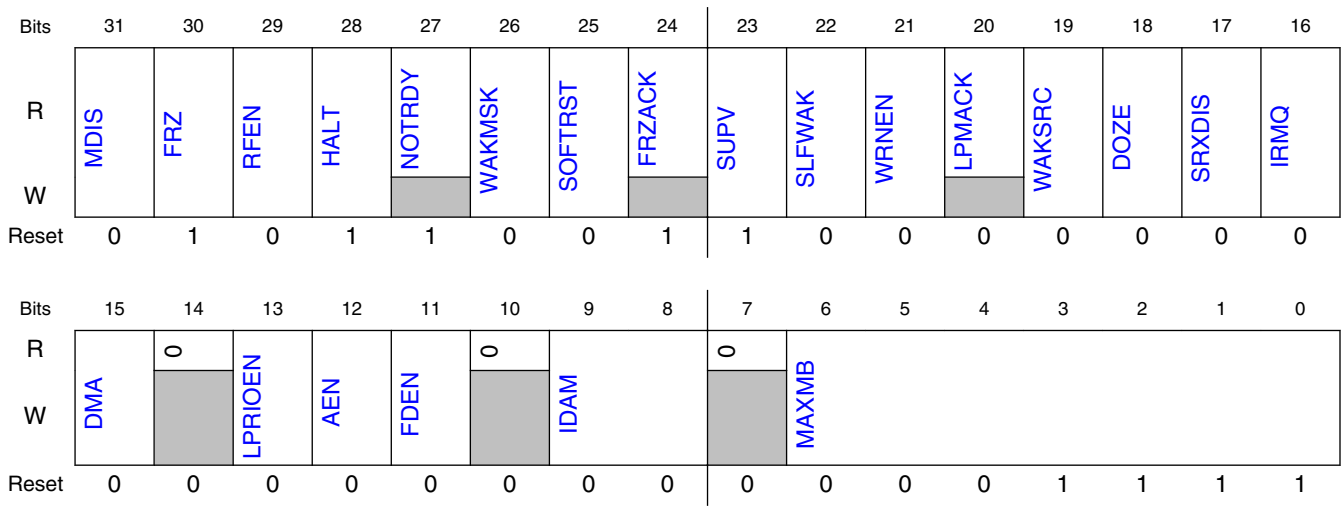
66.6.2.2 Module Configuration register (MCR)

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

66.6.2.2.1 Offset

Register	Offset
MCR	0h

66.6.2.2.2 Diagram



66.6.2.2.3 Fields

Field	Description
31 MDIS	Module Disable This bit controls whether FlexCAN is enabled or not. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface submodules. This bit is not affected by soft reset. 0 - Enable the FlexCAN module. 1 - Disable the FlexCAN module.
30 FRZ	Freeze Enable The FRZ bit specifies the FlexCAN behavior when MCR[HALT] is set or when Debug mode is requested at chip level. When FRZ is asserted, FlexCAN is enabled to enter Freeze mode. Negation of this bit field causes FlexCAN to exit from Freeze mode. This bit is set by hardware when a non-correctable error is detected (MECR[NCEFAFRZ] is asserted). 0 - Not enabled to enter Freeze mode. 1 - Enabled to enter Freeze mode.
29 RFEN	Rx FIFO Enable This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (80h–DCh) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which

Table continues on the next page...

Field	Description
	are used as Rx FIFO ID filter table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 66-15. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. **NOTE:** This bit cannot be set when CAN FD operation is enabled (see FDEN bit). 0 - Rx FIFO not enabled. 1 - Rx FIFO enabled.
28 HALT	Halt FlexCAN Assertion of this bit puts the FlexCAN module into Freeze mode. The CPU should clear it after initializing the message buffers and the Control registers CTRL1 and CTRL2. No reception or transmission is performed by FlexCAN before this bit is cleared. Freeze mode cannot be entered when FlexCAN is in a low power mode. The HALT bit is set by hardware when a non-correctable error is detected and MECR[NCEFAFRZ] is asserted. 0 - No Freeze mode request. 1 - Enters Freeze mode if the FRZ bit is asserted.
27 NOTRDY	FlexCAN Not Ready This read-only bit indicates that FlexCAN is either in Disable mode, Doze mode, Stop mode or Freeze mode. It is negated when FlexCAN has exited these modes. This bit is not affected by soft reset. 0 - FlexCAN module is either in Normal mode, Listen-Only mode, or Loop-Back mode. 1 - FlexCAN module is either in Disable mode, Doze mode, Stop mode, or Freeze mode.
26 WAKMSK	Wake Up Interrupt Mask This bit enables the Wake Up interrupt generation under Self Wake Up mechanism. 0 - Wake Up interrupt is disabled. 1 - Wake Up interrupt is enabled.
25 SOFTRST	Soft Reset When SOFTRST is asserted, FlexCAN resets its internal state machines and some of the memory-mapped registers. SOFTRST can be asserted directly by the CPU when it writes to the MCR register. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTRST bit remains asserted when reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied when clocks are shut down in a low power mode. The module should be first removed from low power mode, and then soft reset can be applied. This bit is not affected by soft reset. 0 - No reset request. 1 - Resets the registers affected by soft reset.
24 FRZACK	Freeze Mode Acknowledge This read-only bit indicates that FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZACK bit to know when FlexCAN has actually entered Freeze mode. If Freeze mode request is negated, then this bit is negated after the FlexCAN prescaler is running again. If Freeze mode is requested when FlexCAN is in a low power mode, then the FRZACK bit will be set only when the low-power mode is exited. See Freeze mode. This bit is not affected by soft reset. **NOTE:** FRZACK will be asserted within 178 CAN bits from the Freeze mode request by the CPU, and negated within 2 CAN bits after the Freeze mode request removal (see Protocol timing). 0 - FlexCAN not in Freeze mode, prescaler running.

Table continues on the next page...

Field	Description
	1 - FlexCAN in Freeze mode, prescaler stopped.
23 SUPV	Supervisor Mode This bit configures the FlexCAN to be either in Supervisor or User mode. The registers affected by this bit are marked as S/U in the "Access type" column of Table 66-19. Reset value of this bit is 1, so the affected registers start with Supervisor access allowance only. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses. 1 - FlexCAN is in Supervisor mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location.
22 SLFWAK	Self Wake Up This bit enables the Self Wake Up feature when FlexCAN is in a low-power mode other than Disable mode. When this feature is enabled, the FlexCAN module monitors the bus for wakeup event, that is, a recessive-to-dominant transition. If a wakeup event is detected during Doze mode, FlexCAN requests to resume its clocks and, if enabled to do so, generates a Wake Up interrupt to the CPU. If a wakeup event is detected during Stop mode, then FlexCAN generates, if enabled to do so, a Wake Up interrupt to the CPU so that it can exit Stop mode globally and FlexCAN can request to resume the clocks. When FlexCAN is in a low-power mode other than Disable mode, this bit cannot be written as it is blocked by hardware. 0 - FlexCAN Self Wake Up feature is disabled. 1 - FlexCAN Self Wake Up feature is enabled.
21 WRNEN	Warning Interrupt Enable When asserted, this bit enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register 1 (ESR1). If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. 0 - TWRNINT and RWRNINT bits are zero, independent of the values in the error counters. 1 - TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96.
20 LPMACK	Low-Power Mode Acknowledge This read-only bit indicates that FlexCAN is in a low-power mode (Disable mode, Doze mode, Stop mode). A low-power mode cannot be entered until all current transmission or reception processes have finished, so the CPU can poll the LPMACK bit to know when FlexCAN has actually entered low power mode. This bit is not affected by soft reset. **NOTE:** LPMACK will be asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see Protocol timing). 0 - FlexCAN is not in a low-power mode. 1 - FlexCAN is in a low-power mode.
19 WAKSRC	Wake Up Source This bit defines whether the integrated low-pass filter is applied to protect the Rx CAN input from spurious wakeup. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - FlexCAN uses the unfiltered Rx input to detect recessive to dominant edges on the CAN bus. 1 - FlexCAN uses the filtered Rx input to detect recessive to dominant edges on the CAN bus.
18	Doze Mode Enable

Table continues on the next page...

Field	Description
DOZE	This bit defines whether FlexCAN is allowed to enter low-power mode when Doze mode is requested at chip level. This bit is automatically reset when FlexCAN wakes up from Doze mode upon detecting activity on the CAN bus (self wakeup enabled). 0 - FlexCAN is not enabled to enter low-power mode when Doze mode is requested. 1 - FlexCAN is enabled to enter low-power mode when Doze mode is requested.
17 SRXDIS	Self Reception Disable This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - Self-reception enabled. 1 - Self-reception disabled.
16 IRMQ	Individual Rx Masking And Queue Enable This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK, RX15MASK, and RXFGMASK. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0 - Individual Rx masking and queue feature are disabled. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY. 1 - Individual Rx masking and queue feature are enabled.
15 DMA	DMA Enable DMA controls whether the DMA feature is enabled or not. The DMA feature can only be used in Rx FIFO, so consequently MCR[RFEN] must be asserted. When DMA and RFEN are set, IFLAG1[BUF5I] generates the DMA request and no RX FIFO interrupt is generated. This bit can be written in Freeze mode only as it is blocked by hardware in other modes. 0 - DMA feature for RX FIFO disabled. 1 - DMA feature for RX FIFO enabled.
14 —	Reserved
13 LPRIOEN	Local Priority Enable This bit is provided for backwards compatibility with legacy applications. It controls whether the local priority feature is enabled or not. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - Local Priority disabled. 1 - Local Priority enabled.
12 AEN	Abort Enable When asserted, this bit enables the Tx abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** When MCR[AEN] is asserted, only the abort mechanism (see Transmission abort mechanism) must be used for updating mailboxes configured for transmission. **CAUTION:** Writing the Abort code into Rx mailboxes can cause unpredictable results when MCR[AEN] is asserted. 0 - Abort disabled.

Table continues on the next page...

Field	Description
	1 - Abort enabled.
11 FDEN	CAN FD operation enable This bit enables the CAN with Flexible Data rate (CAN FD) operation. This bit can be written in Freeze mode only. **NOTE:** FlexCAN is able to transmit FD frame format according to ISO 11898-1. **NOTE:** The Rx FIFO Enable (RFEN) bit cannot be set if FDEN is asserted. 0 - CAN FD is disabled. FlexCAN is able to receive and transmit messages in CAN 2.0 format. 1 - CAN FD is enabled. FlexCAN is able to receive and transmit messages in both CAN FD and CAN 2.0 formats.
10 —	Reserved
9-8 IDAM	ID Acceptance Mode This 2-bit field identifies the format of the Rx FIFO ID filter table elements. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Rx FIFO structure. This field can be written only in Freeze mode because it is blocked by hardware in other modes. 00 - Format A: One full ID (standard and extended) per ID filter table element. 01 - Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID filter table element. 10 - Format C: Four partial 8-bit standard IDs per ID filter table element. 11 - Format D: All frames rejected.
7 —	Reserved
6-0 MAXMB	Number Of The Last Message Buffer This 7-bit field defines the number of the last message buffers that will take part in the matching and arbitration processes. The reset value (0Fh) is equivalent to a 16 MB configuration. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Number of the last MB = MAXMB **NOTE:** MAXMB must be programmed with a value smaller than or equal to the number of available message buffers, as described in FlexCAN memory partition for CAN FD. Additionally, the definition of MAXMB value must take into account the region of MBs occupied by Rx FIFO and its ID filters table space defined by CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table 66-15.

66.6.2.3 Control 1 register (CTRL1)

This register is defined for specific FlexCAN control features related to the CAN bus, such as bit rate, programmable sampling point within an Rx bit, Loop Back mode, Listen-Only mode, Bus Off recovery behavior, and interrupt enabling (Bus-Off, Error, Warning). It also determines the division factor for the clock prescaler.

The CAN bit timing variables (PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW) can also be configured in the CBT register, which extends the range of all these variables. If CBT[BTF] is set, PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields of CTRL1 become read only.

NOTE

When the CAN FD feature is enabled, do not use the PRESDIV, RJW, PSEG1, PSEG2, and PROPSEG fields of the CTRL1 register for CAN bit timing. Instead use the CBT register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields.

The contents of this register are not affected by soft reset.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

66.6.2.3.1 Offset

Register	Offset
CTRL1	4h

66.6.2.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRESDIV								RJW		PSEG1			PSEG2		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	Reserved	Reserved	SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.3.3 Fields

Field	Description
31-24	Prescaler Division Factor

Table continues on the next page...

Field	Description
PRESDIV	This 8-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The maximum value of this field is FFh, which gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (PRESDIV + 1).
23-22 RJW	Resync Jump Width This 2-bit field defines the maximum number of time quanta that a bit time can be changed by one resynchronization. One time quantum is equal to the Sclock period. The valid programmable values are 0–3. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = RJW + 1.
21-19 PSEG1	Phase Segment 1 This 3-bit field defines the length of phase segment 1 in the bit time. The valid programmable values are 0–7. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (PSEG1 + 1) × Time-Quanta.
18-16 PSEG2	Phase Segment 2 This 3-bit field defines the length of phase segment 2 in the bit time. The valid programmable values are 1–7. See Protocol timing for more information. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time-Quanta.
15 BOFFMSK	Bus Off Interrupt Mask This bit provides a mask for the Bus Off interrupt ESR1[BOFFINT]. 0 - Bus Off interrupt disabled. 1 - Bus Off interrupt enabled.
14 ERRMSK	Error Interrupt Mask This bit provides a mask for the Error interrupt ESR1[ERRINT]. 0 - Error interrupt disabled. 1 - Error interrupt enabled.
13 CLKSRC	CAN Engine Clock Source This bit selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock or the oscillator clock. The selected clock is the one fed to the prescaler to generate the serial clock (Sclock). In order to guarantee reliable operation, this bit can be written only in Disable mode because it is blocked by hardware in other modes. See Protocol timing. **NOTE:** The user must ensure the protocol engine clock tolerance according to the CAN Protocol standard (ISO 11898-1). **NOTE:** See the clock distribution chapter (module clocks table) to identify the proper clock source. 0 - The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock. 1 - The CAN engine clock source is the peripheral clock.
12 LPB	Loop Back Mode

Table continues on the next page...

Field	Description
	This bit configures FlexCAN to operate in Loop-Back mode. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** In this mode, MCR[SRXDIS] cannot be asserted because this will impede the self-reception of a transmitted message. **NOTE:** FDCTRL[TDCEN] must be disabled when LPB is asserted. 0 - Loop Back disabled. 1 - Loop Back enabled.
11 TWRNMSK	Tx Warning Interrupt Mask This bit provides a mask for the Tx Warning interrupt associated with the TWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when MCR[WRNEN] is negated. This bit can be written only if MCR[WRNEN] is asserted. 0 - Tx Warning interrupt disabled. 1 - Tx Warning interrupt enabled.
10 RWRNMSK	Rx Warning Interrupt Mask This bit provides a mask for the Rx Warning interrupt associated with the RWRNINT flag in the Error and Status Register 1 (ESR1). This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted. 0 - Rx Warning interrupt disabled. 1 - Rx Warning interrupt enabled.
9 —	Reserved
8 —	Reserved
7 SMP	CAN Bit Sampling This bit defines the sampling mode of CAN bits at the Rx input. It can be written in Freeze mode only, because it is blocked by hardware in other modes. **NOTE:** For proper operation, to assert SMP it is necessary to guarantee a minimum value of two TQs in CTRL1[PSEG1] (or CBT[EPSEG1]).This bit cannot be asserted when CAN FD is enabled (MCR[FDEN] = 1). 0 - Just one sample is used to determine the bit value. 1 - Three samples are used to determine the value of the received bit: the regular one (sample point) and two preceding samples; a majority rule is used.
6 BOFFREC	Bus Off Recovery This bit defines how FlexCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FlexCAN will resynchronize to the bus by waiting

Table continues on the next page...

Field	Description
	for 11 recessive bits before joining the bus. After negation, the BOFFREC bit can be reasserted again during Bus Off, but it will be effective only the next time the module enters Bus Off. If BOFFREC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. **NOTE:** See Bus off in the CAN Protocol standard (ISO 11898-1) for details. 0 - Automatic recovering from Bus Off state enabled. 1 - Automatic recovering from Bus Off state disabled.
5 TSYN	Timer Sync This bit enables a mechanism that resets the free running timer each time a message is received in message buffer 0. This feature provides means to synchronize multiple FlexCAN stations with a special “SYNC” message, that is, global network time. If MCR[RFEN] is set (Rx FIFO enabled), the first available mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0 - Timer sync feature disabled 1 - Timer sync feature enabled
4 LBUF	Lowest Buffer Transmitted First This bit defines the ordering mechanism for message buffer transmission. When asserted, MCR[LPRIOEN] does not affect the priority arbitration. This bit can be written in Freeze mode only, because it is blocked by hardware in other modes. 0 - Buffer with highest priority is transmitted first. 1 - Lowest number buffer is transmitted first.
3 LOM	Listen-Only Mode This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters described in the ECR register are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error without changing the receive error counter (ECR[RXERRCNT]), as if it was trying to acknowledge the message. Listen-Only mode is acknowledged by the state of ESR1[FLTCONF] indicating Passive Error. There can be some delay between the Listen-Only mode request and acknowledge. This bit can be written in Freeze mode only because it is blocked by hardware in other modes. 0 - Listen-Only mode is deactivated. 1 - FlexCAN module operates in Listen-Only mode.
2-0 PROPSEG	Propagation Segment This 3-bit field defines the length of the propagation segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation segment time = (PROPSEG + 1) × time-quanta. Time-quantum = one Sclock period.

66.6.2.4 Free Running Timer (TIMER)

This register represents a 16-bit free-running counter that can be read and written by the CPU. The timer starts from 0h after Reset, counts linearly to FFFFh, and wraps around.

When CTRL2[TIMER_SRC] is asserted, the timer is continuously incremented by an external time tick. The time tick must be synchronous to the peripheral clock, with a minimum pulse width of one clock cycle.

When CTRL2[TIMER_SRC] is negated, the timer is incremented by the CAN bit clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable, Doze, Stop and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the time stamp entry in a message buffer after a successful reception or transmission of a message.

If CTRL1[TSYN] is asserted, the timer is reset whenever a message is received in the first available mailbox, according to CTRL2[RFFN] setting.

The CPU can write to this register anytime. However, if the write occurs at the same time that the timer is being reset by a reception in the first mailbox, then the write value is discarded.

Reading this register affects the mailbox unlocking procedure (see Mailbox lock mechanism).

66.6.2.4.1 Offset

Register	Offset
TIMER	8h

66.6.2.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.4.3 Fields

Field	Description
31-16 —	Reserved
15-0 TIMER	Timer Value Contains the free-running counter value.

66.6.2.5 Rx Mailboxes Global Mask register (RXMGMASK)

This register is located in RAM.

RXMGMASK is provided for legacy application support.

- When MCR[IRMQ] is negated, RXMGMASK is always in effect (the bits in the MG field will mask the mailbox filter bits).
- When MCR[IRMQ] is asserted, RXMGMASK has no effect (the bits in the MG field will not mask the mailbox filter bits).

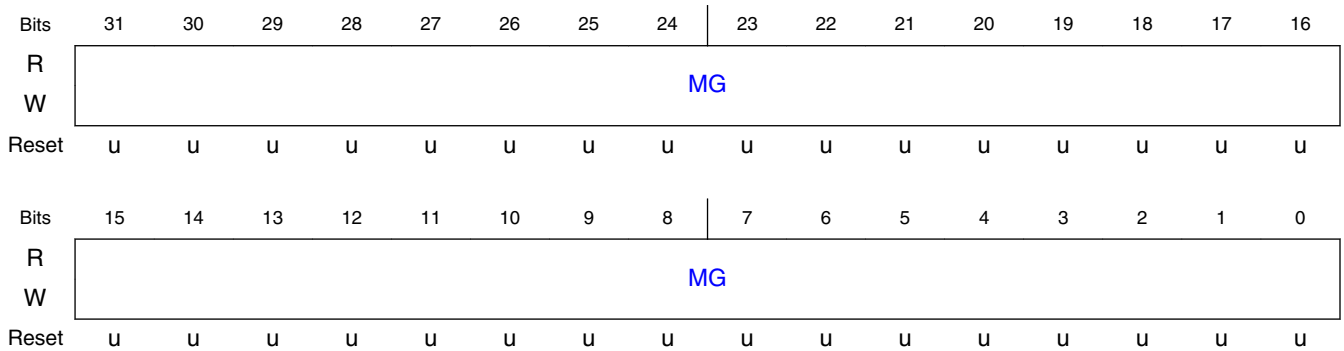
RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

66.6.2.5.1 Offset

Register	Offset
RXMGMASK	10h

66.6.2.5.2 Diagram



66.6.2.5.3 Fields

Field	Description
31-0 MG	Rx Mailboxes Global Mask Bits These bits mask the mailbox filter bits. Note that the alignment with the ID word of the mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the mailbox. The following table shows in detail which MG bits mask each mailbox filter field.

SMB[RTR] #d46731e2450a1310	CTRL2[RRS]	CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	Reserved
0	—	0	note #d46731e2478a1310	note #d46731e2483a1310	MG[28:0]	MG[31:29]
0	—	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	—	—	—	—	MG[31:0]
1	1	0	—	—	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

0b - The corresponding bit in the filter is "don't care."
1b - The corresponding bit in the filter is checked.

66.6.2.6 Rx 14 Mask register (RX14MASK)

This register is located in RAM.

RX14MASK is provided for legacy application support. When MCR[IRMQ] is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of message buffer 14.

This register can only be programmed when the module is in Freeze mode as it is blocked by hardware in other modes.

66.6.2.6.1 Offset

Register	Offset
RX14MASK	14h

66.6.2.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

66.6.2.6.3 Fields

Field	Description
31-0 RX14M	Rx Buffer 14 Mask Bits Each mask bit masks the corresponding mailbox 14 filter field in the same way that RXMGMASK masks other mailboxes' filters. See the description of the RXMGMASK register. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

66.6.2.7 Rx 15 Mask register (RX15MASK)

This register is located in RAM.

RX15MASK is provided for legacy application support. When MCR[IRMQ] is asserted, RX15MASK has no effect.

RX15MASK is used to mask the filter fields of message buffer 15.

This register can be programmed only when the module is in Freeze mode because it is blocked by hardware in other modes.

66.6.2.7.1 Offset

Register	Offset
RX15MASK	18h

66.6.2.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

66.6.2.7.3 Fields

Field	Description
31-0 RX15M	Rx Buffer 15 Mask Bits Each mask bit masks the corresponding mailbox 15 filter field in the same way that RXMGMASK masks other mailboxes' filters. See the description of the RXMGMASK register. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

66.6.2.8 Error Counter (ECR)

This register has four 8-bit fields reflecting the value of the FlexCAN error counters:

- Transmit Error Counter (TXERRCNT field)
- Receive Error Counter (RXERRCNT field)

- Transmit Error Counter for errors detected in the data phase of CAN FD messages with the BRS bit set (TXERRCNT_FAST field)
- Receive Error Counter for errors detected in the data phase of CAN FD messages with the BRS bit set (RXERRCNT_FAST field)

The TXERRCNT and RXERRCNT counters take into account all errors in both CAN FD and non-FD message formats. TXERRCNT_FAST and RXERRCNT_FAST are dedicated to count only the errors that occur in the data phase of CAN FD frames with the BRS bit set.

The Fault Confinement state (ESR1[FLTCONF]) is updated based on TXERRCNT and RXERRCNT counters only. TXERRCNT and RXERRCNT counters can be written in Freeze mode only. TXERRCNT_FAST and RXERRCNT_FAST counters are read-only except in Freeze mode when the CPU can write value zero. The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module.

The following are the basic rules for FlexCAN bus state transitions:

- If the value of TXERRCNT or RXERRCNT becomes greater than or equal to 128, ESR1[FLTCONF] is updated to reflect Error Passive state.
- If the FlexCAN state is Error Passive, and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 when the other already satisfies this condition, ESR1[FLTCONF] is updated to reflect Error Active state.
- If the value of TXERRCNT becomes greater than 255, ESR1[FLTCONF] is updated to reflect Bus Off state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in Bus Off state, then TXERRCNT is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around when incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, ESR1[FLTCONF] is updated to be Error Active and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value. The TXERRCNT_FAST counter is frozen during Bus Off.
- If during system startup only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACKERR bit in the Error and Status Register). After the transition to Error Passive state, TXERRCNT does not increment anymore by acknowledge errors. Therefore the device never goes to the Bus Off state.
- If RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected when being a receiver. At the next successful

message reception, the counter is set to a value between 119 and 127 to return to Error Active state.

- TXERRCNT_FAST and RXERRCNT_FAST error counter values increment and decrement based on errors detected only in the data phase of CAN FD frames with the BRS bit set, following the same increment and decrement rules as TXERRCNT and RXERRCNT counters. These counters do not wrap around and get stuck at their maximum value (255). They stop counting and keep their values frozen when FlexCAN is in Bus Off state. They are reset when FlexCAN leaves Bus Off state and restart counting after FlexCAN returns to Error Active state.

NOTE

See Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

66.6.2.8.1 Offset

Register	Offset
ECR	1Ch

66.6.2.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXERRCNT_FAST								TXERRCNT_FAST							
W	0								0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXERRCNT								TXERRCNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.8.3 Fields

Field	Description
31-24 RXERRCNT_FAST	Receive Error Counter for fast bits Receive error counter for errors detected in the data phase of received CAN FD messages with the BRS bit set. The RXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write an 8-bit zero value only.
23-16	Transmit Error Counter for fast bits

Table continues on the next page...

Field	Description
TXERRCNT_FAST	Transmit error counter for errors detected in the data phase of transmitted CAN FD messages with the BRS bit set. The TXERRCNT_FAST counter is read-only except in Freeze mode, where the CPU can write an 8-bit zero value only.
15-8 RXERRCNT	Receive Error Counter Receive error counter for all errors detected in received messages. The RXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.
7-0 TXERRCNT	Transmit Error Counter Transmit error counter for all errors detected in transmitted messages. The TXERRCNT counter is read-only except in Freeze mode, where it can be written by the CPU.

66.6.2.9 Error and Status 1 register (ESR1)

This register reports various error conditions detected in the reception and transmission of a CAN frame, some general status of the device, and is the source of some interrupts to the CPU. The reported error conditions are:

- Errors detected in CAN frames of any format:
  - BIT1ERR
  - BIT0ERR
  - ACKKERR
  - CRCERR
  - FRMERR
  - STFERR
- Errors detected in the data phase of CAN FD frames with the BRS bit set only:
  - BIT1ERR_FAST
  - BIT0ERR_FAST
  - CRCERR_FAST
  - FRMERR_FAST
  - STFERR_FAST

An error detected in a single CAN frame may be reported by one or more error flags. Also, error reporting is cumulative, in case more error events happen in the next frames when the CPU does not attempt to read this register.

Status bits:
- TXWRN
- RXWRN
- IDLE
- TX
- FLTCONF

- RX
- SYNCH

Interrupt bits:

- BOFFINT
- BOFFDONEINT
- ERRINT
- ERRINT_FAST
- WAKINT
- TWRNINT
- RWRNINT

It is recommended that the CPU use the following procedure when servicing interrupt requests generated by these bits:

1. Read this register to capture all error condition and status bits. This action clears the respective bits that were set since the last read access.
2. Write 1 to clear the interrupt bit that has triggered the interrupt request.
3. Write 1 to clear the ERR_OVR bit, if it is set.

Starting from all error flags cleared, a first error event sets either the ERRINT or the ERRINT_FAST (provided the corresponding mask bit is asserted). If other error events in subsequent frames happen before the CPU serves the interrupt request, the ERR_OVR bit is set to indicate that errors from different frames have accumulated.

SYNCH	IDLE	TX	RX	FlexCAN state
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

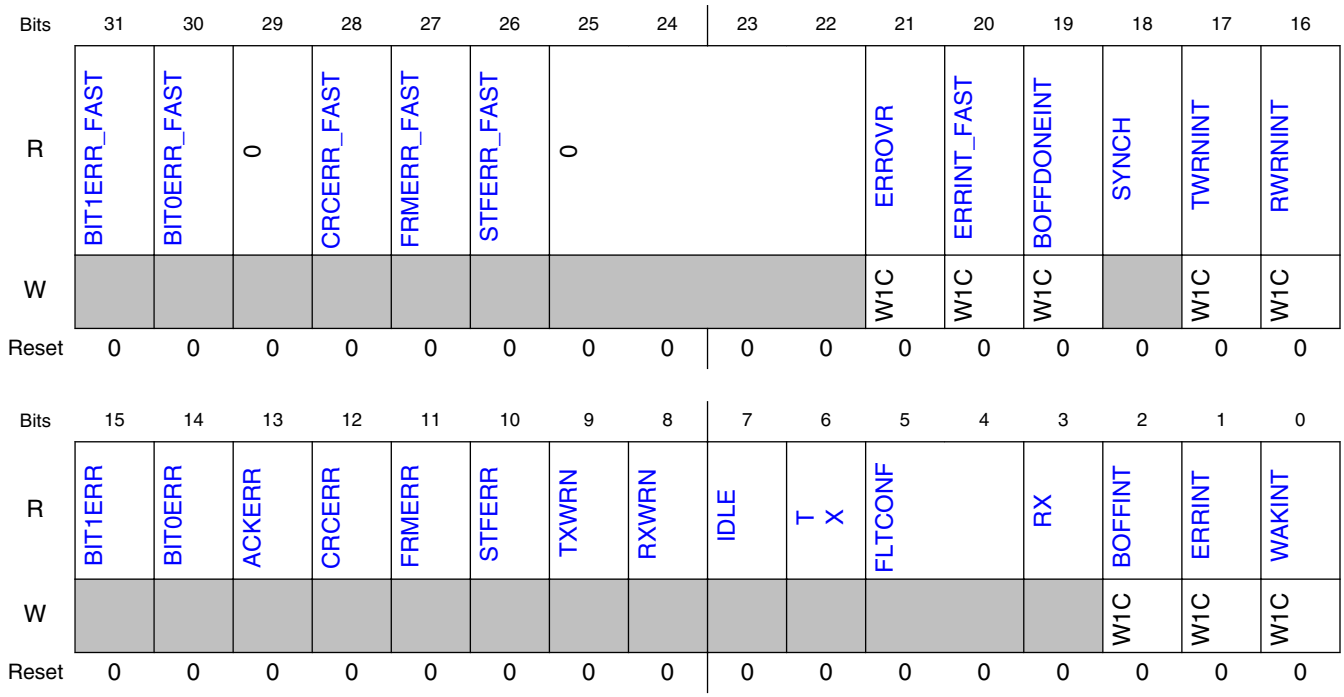
NOTE

See Fault confinement in the CAN Protocol standard (ISO 11898-1) for details.

66.6.2.9.1 Offset

Register	Offset
ESR1	20h

66.6.2.9.2 Diagram



66.6.2.9.3 Fields

Field	Description
31 BIT1ERR_FAST	Bit1 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as recessive is received as dominant.
30 BIT0ERR_FAST	Bit0 Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates when an inconsistency occurs between the transmitted and the received bit in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as dominant is received as recessive.
29 —	Reserved
28 CRCERR_FAST	Cyclic Redundancy Check Error in the CRC field of CAN FD frames with the BRS bit set This bit indicates that a CRC error has been detected by the receiver node in the CRC field of CAN FD frames with the BRS bit set, that is, the calculated CRC is different from the received. After a read operation, the field's value clears to 0.

Table continues on the next page...

Field	Description
	0 - No such occurrence. 1 - A CRC error occurred since last read of this register.
27 FRMERR_FAST	Form Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a form error has been detected by the receiver node in the data phase of CAN FD frames with the BRS bit set, that is, a fixed-form bit field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A form error occurred since last read of this register.
26 STFERR_FAST	Stuffing Error in the Data Phase of CAN FD frames with the BRS bit set This bit indicates that a stuffing error has been detected in the data phase of CAN FD frames with the BRS bit set. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A stuffing error occurred since last read of this register.
25-22 —	Reserved
21 ERROVR	Error Overrun This bit indicates that an error condition occurred when any error flag is already set. This bit is cleared by writing it to 1. 0 - Overrun has not occurred. 1 - Overrun has occurred.
20 ERRINT_FAST	Error interrupt for errors detected in Data Phase of CAN FD frames with BRS bit set This bit indicates that at least one of the error bits detected in the data phase of CAN FD frames with the BRS bit set (BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST, or STFERR_FAST) is set. If the corresponding mask bit CTRL2[ERRMSK_FAST] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 - No such occurrence. 1 - Indicates setting of any error bit detected in the data phase of CAN FD frames with the BRS bit set.
19 BOFFDONEINT	Bus Off Done Interrupt This bit is set when the Tx Error Counter (TXERRCNT) has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off. If the corresponding mask bit in the Control 2 Register (BOFFDONEMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 - No such occurrence. 1 - FlexCAN module has completed Bus Off process.
18 SYNCH	CAN Synchronization Status This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See the table in the overall ESR1 register description. 0 - FlexCAN is not synchronized to the CAN bus. 1 - FlexCAN is synchronized to the CAN bus.
17 TWRNINT	Tx Warning Interrupt Flag

Table continues on the next page...

Field	Description
	If MCR[WRNEN] is asserted, TWRNINT is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control 1 Register (CTRL1[TWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When MCR[WRNEN] is negated, this flag is masked. CPU must clear this flag before disabling MCR[WRNEN]. Otherwise it will be set when MCR[WRNEN] is set again. Writing 0 has no effect. This flag is not generated during Bus Off state. This bit is not updated during Freeze mode. 0 - No such occurrence. 1 - The Tx error counter transitioned from less than 96 to greater than or equal to 96.
16 RWRNINT	Rx Warning Interrupt Flag If MCR[WRNEN] is asserted, RWRNINT is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control 1 Register (CTRL1[RWRNMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when WRNEN is set again. Writing 0 has no effect. This bit is not updated during Freeze mode. 0 - No such occurrence. 1 - The Rx error counter transitioned from less than 96 to greater than or equal to 96.
15 BIT1ERR	Bit1 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. **NOTE:** This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as recessive is received as dominant.
14 BIT0ERR	Bit0 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or data phase of a CAN FD message. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - At least one bit sent as dominant is received as recessive.
13 ACKERR	Acknowledge Error This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - An ACK error occurred since last read of this register.
12 CRCERR	Cyclic Redundancy Check Error This bit indicates that a Cyclic Redundancy Check (CRC) error has been detected by the receiver node either in a non-FD message or in the arbitration or data phase of a frame in CAN FD format, that is, the calculated CRC is different from the received. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A CRC error occurred since last read of this register.

Table continues on the next page...

Field	Description
11 FRMERR	Form Error This bit indicates that a form error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node, that is, a fixed-form bit field contains at least one illegal bit. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A Form Error occurred since last read of this register.
10 STFERR	Stuffing Error This bit indicates that a stuffing error has been detected in a non-FD message or else in an FD message's arbitration or data phase by the receiver node. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - A stuffing error occurred since last read of this register.
9 TXWRN	TX Error Warning This bit indicates when repetitive errors are occurring during message transmission and is affected by the value of TXERRCNT in ECR register only. This bit is not updated during Freeze mode. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - TXERRCNT is greater than or equal to 96.
8 RXWRN	Rx Error Warning This bit indicates when repetitive errors are occurring during message reception and is affected by the value of RXERRCNT in ECR register only. This bit is not updated during Freeze mode. After a read operation, the field's value clears to 0. 0 - No such occurrence. 1 - RXERRCNT is greater than or equal to 96.
7 IDLE	IDLE This bit indicates when CAN bus is in IDLE state. See the table in the overall ESR1 register description. 0 - No such occurrence. 1 - CAN bus is now IDLE.
6 TX	FlexCAN In Transmission This bit indicates if FlexCAN is transmitting a message. See the table in the overall ESR1 register description. 0 - FlexCAN is not transmitting a message. 1 - FlexCAN is transmitting a message.
5-4 FLTCONF	Fault Confinement State This 2-bit field indicates the confinement state of the FlexCAN module. If CTRL1[LOM] is asserted, after a delay that depends on the CAN bit timing, ESR1[FLTCONF] will indicate Error Passive. The very same delay affects the way that ESR1[FLTCONF] reflects an update to ECR register by the CPU. It may be necessary to wait up to one CAN bit time to get them coherent again. This bit field is affected by soft reset, but if the LOM bit is asserted, its reset value lasts just one CAN bit. After this time, ESR1[FLTCONF] reports Error Passive. 00 - Error Active

Table continues on the next page...

Field	Description
	01 - Error Passive 1x - Bus Off
3 RX	FlexCAN In Reception This bit indicates if FlexCAN is receiving a message. See the table in the overall ESR1 register description. 0 - FlexCAN is not receiving a message. 1 - FlexCAN is receiving a message.
2 BOFFINT	Bus Off Interrupt This bit is set when FlexCAN enters Bus Off state. If the corresponding mask bit in the Control Register 1 (CTRL1[BOFFMSK]) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 - No such occurrence. 1 - FlexCAN module entered Bus Off state.
1 ERRINT	Error Interrupt This bit indicates that at least one of the error bits (BIT1ERR, BIT0ERR, ACKERR, CRCERR, FRMERR, or STFERR) is set. If the corresponding mask bit CTRL1[ERRMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 - No such occurrence. 1 - Indicates setting of any error bit in the Error and Status register.
0 WAKINT	Wake-Up Interrupt This field applies when FlexCAN is in low-power mode under Self Wake Up mechanism: • Doze mode • Stop mode When a recessive-to-dominant transition is detected on the CAN bus and if MCR[WAKMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When MCR[SLFWAK] is negated, this flag is masked. The CPU must clear this flag before disabling the bit. Otherwise it will be set when SLFWAK is set again. Writing 0 has no effect. 0 - No such occurrence. 1 - Indicates a recessive to dominant transition was received on the CAN bus.

66.6.2.10 Interrupt Masks 2 register (IMASK2)

This register allows any number of a range of the 32 message buffer interrupts to be enabled or disabled for MB63 to MB32. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG2 bit is set.

66.6.2.10.1 Offset

Register	Offset
IMASK2	24h

66.6.2.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.10.3 Fields

Field	Description
31-0 BUF63TO32M	Buffer MBi Mask Each bit enables or disables the corresponding FlexCAN message buffer interrupt for MB63 to MB32. **NOTE:** Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

66.6.2.11 Interrupt Masks 1 register (IMASK1)

This register allows any number of a range of the 32 message buffer interrupts to be enabled or disabled for MB31 to MB0. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG1 bit is set.

66.6.2.11.1 Offset

Register	Offset
IMASK1	28h

66.6.2.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.11.3 Fields

Field	Description
31-0 BUF31TO0M	Buffer MBi Mask Each bit enables or disables the corresponding FlexCAN message buffer interrupt for MB31 to MB0. **NOTE:** Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

66.6.2.12 Interrupt Flags 2 register (IFLAG2)

This register defines the flags for the 32 message buffer interrupts for MB63 to MB32. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating MCR[MAXMB], CPU must service the IFLAG2 bits whose MB value is greater than the MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

66.6.2.12.1 Offset

Register	Offset
IFLAG2	2Ch

66.6.2.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.12.3 Fields

Field	Description
31-0 BUF63TO32I	Buffer MBi Interrupt Each bit flags the corresponding FlexCAN message buffer interrupt for MB63 to MB32. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

66.6.2.13 Interrupt Flags 1 register (IFLAG1)

This register defines the flags for the 32 message buffer interrupts for MB31 to MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect. There is an exception when DMA for Rx FIFO is enabled, as described below.

The BUF7I to BUF5I flags are also used to represent FIFO interrupts when the Rx FIFO is enabled. When MCR[RFEN] is set and MCR[DMA] is negated, the function of the eight least significant interrupt flags changes: BUF7I, BUF6I, and BUF5I indicate operating conditions of the FIFO , BUF0I is used to empty FIFO, and BUF4I to BUF1I bits are reserved.

Before enabling MCR[RFEN], the CPU must service the IFLAG bits asserted in the Rx FIFO region; see Rx FIFO. Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When MCR[RFEN] is negated, the FIFO flags must be cleared. The same care must be taken when a CTRL2[RFFN] value is selected, extending Rx FIFO filters beyond MB7. For example, when RFFN is 8h, the MB0–23 range is occupied by Rx FIFO filters and related IFLAG bits must be cleared.

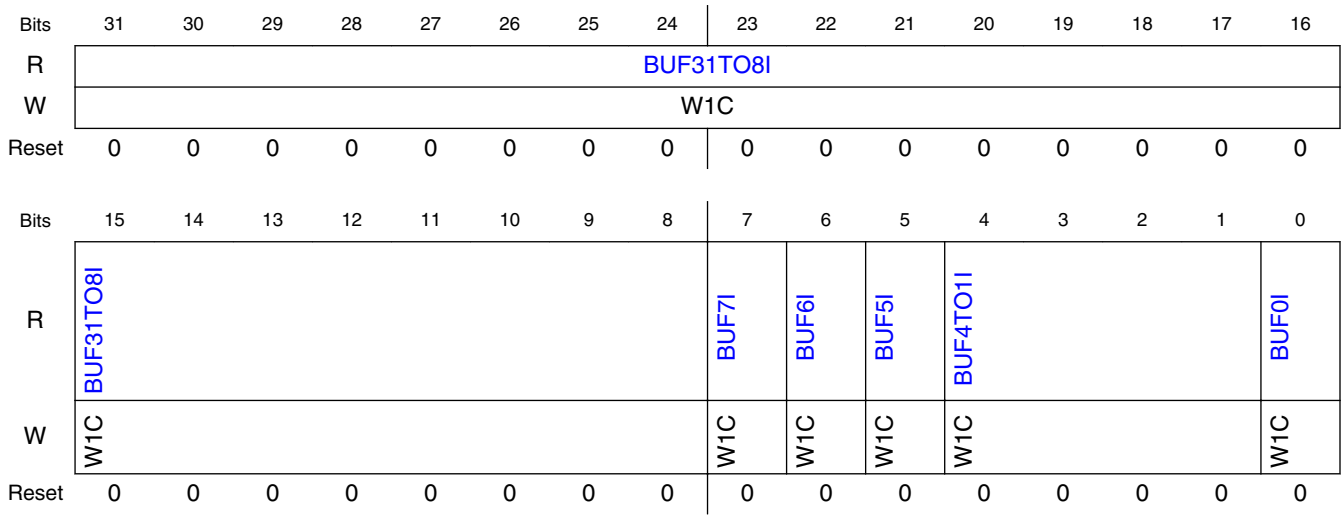
When both the MCR[RFEN] and MCR[DMA] bits are asserted (DMA feature for Rx FIFO enabled), the function of the eight least significant interrupt flags (BUF7I–BUF0I) are changed to support the DMA operation. BUF7I and BUF6I are not used, as well as BUF4I–BUF1I. BUF5I indicates operating condition of FIFO, and BUF0I is used to empty FIFO. Moreover, BUF5I does not generate a CPU interrupt, but generates a DMA request. IMASK1 bits in Rx FIFO region are not considered when bit MCR[DMA] is enabled. In addition the CPU must not clear the flag BUF5I when DMA is enabled. Before enabling MCR[DMA], the CPU must service the IFLAGs asserted in the Rx FIFO region. When MCR[DMA] is negated, the FIFO must be empty. FIFO must be disabled when MCR[FDEN] is enabled.

Before updating MCR[MAXMB], CPU must service the IFLAG1 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

66.6.2.13.1 Offset

Register	Offset
IFLAG1	30h

66.6.2.13.2 Diagram



66.6.2.13.3 Fields

Field	Description
31-8 BUF31TO8I	Buffer MBi Interrupt Each bit flags the corresponding FlexCAN message buffer interrupt for MB31 to MB8. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	Buffer MB7 Interrupt Or Rx FIFO Overflow When MCR[RFEN] is cleared (Rx FIFO disabled), this bit flags the interrupt for MB7. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF7I flag represents Rx FIFO overflow when MCR[RFEN] is set. In this case, the flag indicates that a message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a mailbox. 0 - No occurrence of MB7 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO overflow when MCR[RFEN]=1 1 - MB7 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO overflow when MCR[RFEN]=1
6 BUF6I	Buffer MB6 Interrupt Or Rx FIFO Warning When MCR[RFEN] is cleared (Rx FIFO disabled), this bit flags the interrupt for MB6. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF6I flag represents Rx FIFO warning when MCR[RFEN] is set. In this case, the flag indicates when the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared when the number of unread messages is greater than four, it does not assert again until the number of unread messages within the Rx FIFO is decreased to be equal to or less than four. 0 - No occurrence of MB6 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO almost full when MCR[RFEN]=1

Table continues on the next page...

Field	Description
	1 - MB6 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO almost full when MCR[RFEN]=1
5 BUF5I	Buffer MB5 Interrupt Or Frames available in Rx FIFO When MCR[RFEN] is cleared (Rx FIFO disabled), this field flags the interrupt for MB5. **NOTE:** This flag is cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. When MCR[RFEN] is set (Rx FIFO enabled), the BUF5I flag represents "Frames available in Rx FIFO" and indicates that at least one frame is available to be read from the Rx FIFO. When the MCR[DMA] bit is enabled, this flag generates a DMA request and the CPU must not clear this bit by writing 1 in BUF5I. 0 - No occurrence of MB5 completing transmission/reception when MCR[RFEN]=0, or of frame(s) available in the FIFO, when MCR[RFEN]=1 1 - MB5 completed transmission/reception when MCR[RFEN]=0, or frame(s) available in the Rx FIFO when MCR[RFEN]=1. It generates a DMA request in case of MCR[RFEN] and MCR[DMA] are enabled.
4-1 BUF4TO1I	Buffer MBi Interrupt Or Reserved When MCR[RFEN] is cleared (Rx FIFO disabled), these bits flag the interrupts for MB4 to MB1. **NOTE:** These flags are cleared by the FlexCAN whenever MCR[RFEN] is changed by CPU writes. The BUF4TO1I flags are reserved when MCR[RFEN] is set. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1b - The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.
0 BUF0I	Buffer MB0 Interrupt Or Clear FIFO bit When MCR[RFEN] is cleared (Rx FIFO disabled), this field flags the interrupt for MB0. If the Rx FIFO is enabled, this bit is used to trigger the clear FIFO operation. This operation empties FIFO contents. Before performing this operation the CPU must service all FIFO related IFLAGs. When MCR[DMA] is enabled this operation also clears the BUF5I flag and consequently aborts the DMA request. The clear FIFO operation occurs when the CPU writes 1 in BUF0I. It is only allowed in Freeze mode and is blocked by hardware in other conditions. 0 - The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1 - The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.

66.6.2.14 Control 2 register (CTRL2)

This register complements Control1 register, providing control bits for memory write access in Freeze mode, for extending FIFO filter quantity, and to adjust the operation of internal FlexCAN processes like matching and arbitration.

The contents of this register are not affected by soft reset.

This table shows how the Rx FIFO filter structure is determined by the value of CTRL2[RFFN]. See the CTRL2[RFFN] field description for more information.

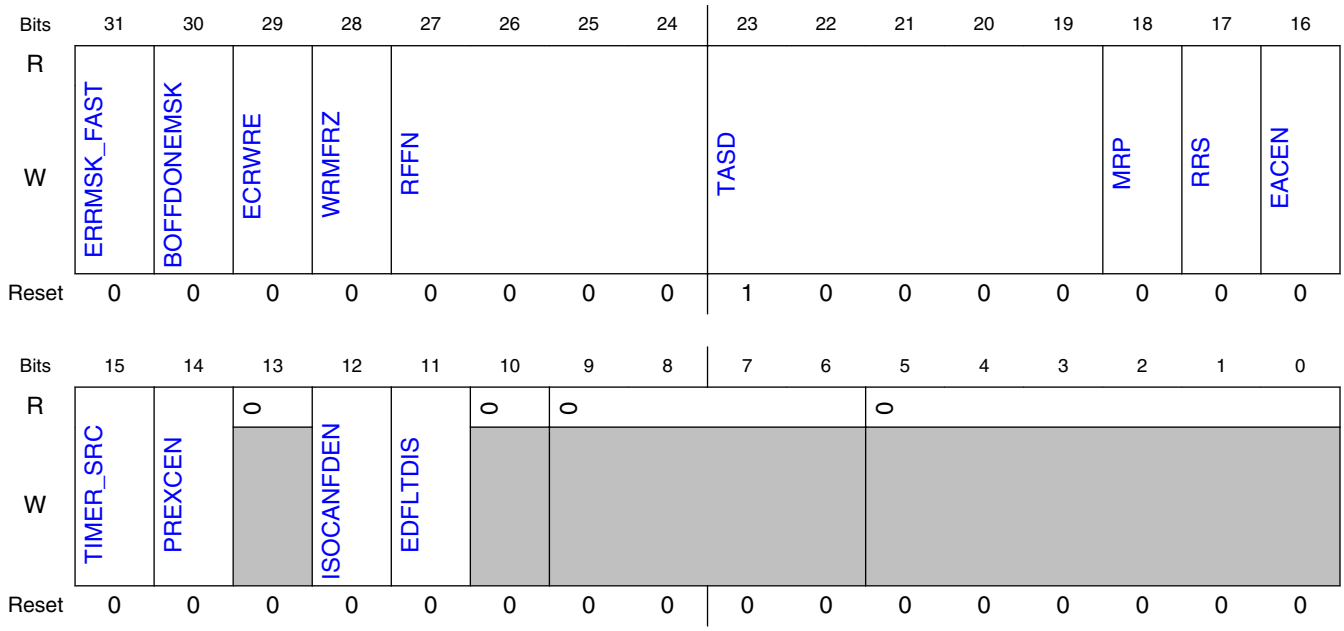
Table 66-20. Rx FIFO filter: possible structures

RFFN[3:0]	Number of Rx FIFO filter elements	Message buffers occupied by Rx FIFO and ID filter table	Remaining available mailboxes	Rx FIFO ID filter table elements affected by Rx individual masks	Rx FIFO ID filter table elements affected by Rx FIFO global mask
0h	8	MB 0-7	MB 8-63	Elements 0-7	none
1h	16	MB 0-9	MB 10-63	Elements 0-9	Elements 10-15
2h	24	MB 0-11	MB 12-63	Elements 0-11	Elements 12-23
3h	32	MB 0-13	MB 14-63	Elements 0-13	Elements 14-31
4h	40	MB 0-15	MB 16-63	Elements 0-15	Elements 16-39
5h	48	MB 0-17	MB 18-63	Elements 0-17	Elements 18-47
6h	56	MB 0-19	MB 20-63	Elements 0-19	Elements 20-55
7h	64	MB 0-21	MB 22-63	Elements 0-21	Elements 22-63
8h	72	MB 0-23	MB 24-63	Elements 0-23	Elements 24-71
9h	80	MB 0-25	MB 26-63	Elements 0-25	Elements 26-79
Ah	88	MB 0-27	MB 28-63	Elements 0-27	Elements 28-87
Bh	96	MB 0-29	MB 30-63	Elements 0-29	Elements 30-95
Ch	104	MB 0-31	MB 32-63	Elements 0-31	Elements 32-103
Dh	112	MB 0-33	MB 34-63	Elements 0-31	Elements 32-111
Eh	120	MB 0-35	MB 36-63	Elements 0-31	Elements 32-119
Fh	128	MB 0-37	MB 38-63	Elements 0-31	Elements 32-127

66.6.2.14.1 Offset

Register	Offset
CTRL2	34h

66.6.2.14.2 Diagram



66.6.2.14.3 Fields

Field	Description
31 ERRMSK_FAST	Error Interrupt Mask for errors detected in the data phase of fast CAN FD frames This bit provides a mask for the ERRINT_FAST interrupt in ESR1 register. 0 - ERRINT_FAST error interrupt disabled. 1 - ERRINT_FAST error interrupt enabled.
30 BOFFDONEMSK	Bus Off Done Interrupt Mask This bit provides a mask for the bus off done interrupt in ESR1 register. 0 - Bus off done interrupt disabled. 1 - Bus off done interrupt enabled.
29 ECRWRE	Error-correction Configuration Register Write Enable This bit enables the MECR register to be updated. This bit is automatically set to 0 if the protocol described in Detection and correction of memory errors is not followed. 0 - Disable update. 1 - Enable update.
28 WRMFRZ	Write-Access To Memory In Freeze Mode Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode. MCR[RFEN] must not be set during FlexCAN memory initialization. 0 - Maintain the write access restrictions. 1 - Enable unrestricted write access to FlexCAN memory.

Table continues on the next page...

Field	Description
27-24 RFFN	Number Of Rx FIFO Filters This 4-bit field defines the number of Rx FIFO filters, as shown in Table 66-20. The maximum selectable number of filters is determined by the chip. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field must not be programmed with values that cause the number of message buffers occupied by Rx FIFO and ID Filter to exceed the number of mailboxes present, as defined by MCR[MAXMB]. **NOTE:** Each group of eight filters occupies a memory space equivalent to two message buffers, which means that the more filters are implemented the fewer mailboxes will be available. Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words, which can be calculated as follows: (SETUP_MB - 6) × 4 where SETUP_MB is the smaller of the parameter NUMBER_OF_MB and MCR[MAXMB]. The number of remaining mailboxes available will be: (SETUP_MB - 8) - (RFFN × 2) If the number of Rx FIFO filters programmed through RFFN exceeds the SETUP_MB value (memory space available), then the exceeding ones will not be functional. **NOTE:** • The number of the last remaining available mailboxes is defined by the smaller of NUMBER_OF_MB minus 1 and MCR[MAXMB]. • If Rx Individual Mask registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.
23-19 TASD	Tx Arbitration Start Delay This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. See Tx arbitration start delay for more details. This field can be written only in Freeze mode because it is blocked by hardware in other modes.
18 MRP	Mailboxes Reception Priority If this bit is set the matching process starts from the mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - Matching starts from Rx FIFO and continues on mailboxes. 1 - Matching starts from mailboxes and continues on Rx FIFO.
17 RRS	Remote Request Storing If this bit is asserted a remote request frame is submitted to a matching process and stored in the corresponding message buffer in the same fashion as a data frame. No automatic remote response frame will be generated. If this bit is negated the remote request frame is submitted to a matching process and an automatic remote response frame is generated if a message buffer with CODE = 1010b is found with the same ID. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - Remote response frame is generated. 1 - Remote request frame is stored.
16 EACEN	Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes This bit controls the comparison of IDE and RTR bits within Rx mailbox filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 - Rx mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits.

Table continues on the next page...

Field	Description
	1 - Enables the comparison of both Rx mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply.
15 TIMER_SRC	Timer Source Selects the time tick source used for incrementing the free running timer counter. This bit can be written in Freeze mode only. 0 - The free running timer is clocked by the CAN bit clock, which defines the baud rate on the CAN bus. 1 - The free running timer is clocked by an external time tick. The period can be either adjusted to be equal to the baud rate on the CAN bus, or a different value as required. See the device-specific section for details about the external time tick.
14 PREXCEN	Protocol Exception Enable This bit enables the protocol exception feature. This field is writable only in Freeze mode. **NOTE:** See Protocol exception event in the CAN Protocol standard (ISO 11898-1) for details. 0 - Protocol exception is disabled. 1 - Protocol exception is enabled.
13 —	Reserved
12 ISOCANFDEN	ISO CAN FD Enable This field enables the CAN FD protocol according to ISO specification (ISO 11898-1) (see CAN FD ISO compliance). This field is writable only in Freeze mode. **NOTE:** FlexCAN is able to transmit FD frame format according to CAN Protocol standard (ISO 11898-1). 0 - FlexCAN operates using the non-ISO CAN FD protocol. 1 - FlexCAN operates using the ISO CAN FD protocol (ISO 11898-1).
11 EDFLTDIS	Edge Filter Disable This bit disables the edge filter used during the bus integration state. When the Edge Filter is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting of the sequence of eleven consecutive recessive bits is restarted. The edge filter prevents the dominant pulses that are shorter than a nominal bit time (present during the data phase of an FD frame) from being mistaken for an idle condition. This field is writable only in Freeze mode. **NOTE:** See Bus Integration state in the CAN Protocol standard (ISO 11898-1) for details. 0 - Edge filter is enabled 1 - Edge filter is disabled
10 —	Reserved
9-6 —	Reserved
5-0 —	Reserved

66.6.2.15 Error and Status 2 register (ESR2)

This register reports some general status information.

66.6.2.15.1 Offset

Register	Offset
ESR2	38h

66.6.2.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									LPTM						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	VPS	IMB	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.15.3 Fields

Field	Description
31-23 —	Reserved
22-16 LPTM	Lowest Priority Tx Mailbox If ESR2[VPS] is asserted, this field indicates the lowest number inactive mailbox (see the ESR2[IMB] bit description). If there is no inactive mailbox then the mailbox indicated depends on the value of CTRL1[LBUF]. If CTRL1[LBUF] is negated, then the mailbox indicated is the one that has the greatest arbitration value (see Highest-priority mailbox first). If CTRL1[LBUF] is asserted then the mailbox indicated is the highest number active Tx mailbox. If a Tx mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its mailbox number.
15 —	Reserved
14 VPS	Valid Priority Status This bit indicates whether ESR2[IMB] and ESR2[LPTM] contents are currently valid or not. It is asserted upon every complete Tx arbitration process unless the CPU writes to Control and Status word of a mailbox that has already been scanned, that is, it is behind Tx Arbitration Pointer, during the Tx

Table continues on the next page...

Field	Description
	arbitration process. If there is no inactive mailbox and only one Tx mailbox that is being transmitted then VPS is not asserted. This bit is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any mailbox. **NOTE:** ESR2[VPS] is not affected by any CPU write into Control Status (C/S) of an MB that is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of an MB that is being transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked. 0 - Contents of IMB and LPTM are invalid. 1 - Contents of IMB and LPTM are valid.
13 IMB	Inactive Mailbox If ESR2[VPS] is asserted, this bit indicates whether there is any inactive mailbox (CODE field is either 1000b or 0b). This bit is asserted in the following cases: • During arbitration, if an ESR2[LPTM] is found and it is inactive. • If ESR2[IMB] is not asserted and a frame is transmitted successfully. This bit is always cleared in start of arbitration (see Arbitration process). **NOTE:** ESR2[LPTM] mechanism has the following behavior: if an MB is successfully transmitted and ESR2[IMB]=0 (no inactive mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM]. 0 - If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive mailbox. 1 - If ESR2[VPS] is asserted, there is at least one inactive mailbox. LPTM content is the number of the first one.
12-0 —	Reserved

66.6.2.16 CRC register (CRCR)

This register provides information about the CRC of transmitted messages for non-FD messages. This register only reports the 15 low order bits of CRC calculations for messages in CAN FD format that require either 17 or 21 bits. For CAN FD format frames, the FDCRC register must be used. This register is updated at the same time the Tx interrupt flag is asserted.

NOTE

See CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

66.6.2.16.1 Offset

Register	Offset
CRCR	44h

66.6.2.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									MBCRC						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TXCRC														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.16.3 Fields

Field	Description
31-23 —	Reserved
22-16 MBCRC	CRC Mailbox This field indicates the number of the mailbox corresponding to the value in CRCR[TXCRC] field.
15 —	Reserved
14-0 TXCRC	Transmitted CRC value This field indicates the CRC value of the last transmitted message for non-FD frames. For FD frames, CRC value is reported in FDCRC register.

66.6.2.17 Rx FIFO Global Mask register (RXFGMASK)

This register is located in RAM.

If Rx FIFO is enabled, RXFGMASK is used to mask the Rx FIFO ID filter table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

The following table shows how the FGM bits correspond to each IDAF field.

Table 66-21. Correspondence of Rx FIFO global mask bits to IDF fields

Rx FIFO ID filter table elements format (MCR[IDAM])	Identifier acceptance filter fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	—	—	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	—	FGM[29:16], FGM[13:0]		—
C	—	—		—	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	

1. If MCR[IDAM] is equivalent to format B, only the fourteen most significant bits of the identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] is equivalent to format C, only the eight most significant bits of the identifier of the incoming frame are compared with the Rx FIFO filter.

66.6.2.17.1 Offset

Register	Offset
RXFGMASK	48h

66.6.2.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FGM															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FGM															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

66.6.2.17.3 Fields

Field	Description
31-0 FGM	Rx FIFO Global Mask Bits These bits mask the ID filter table elements bits in a perfect alignment. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

66.6.2.18 Rx FIFO Information register (RXFIR)

RXFIR provides information on Rx FIFO.

This register is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO. Also, its output is updated whenever the output of the Rx FIFO is updated with the next message. See Rx FIFO for instructions on reading this register.

NOTE

RXFIR can be written only during memory initialization, due to the error code correction (ECC) feature. In every other case the register is read-only.

66.6.2.18.1 Offset

Register	Offset
RXFIR	4Ch

66.6.2.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							IDHIT								
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

66.6.2.18.3 Fields

Field	Description
31-9	Reserved

Table continues on the next page...

Field	Description
—	
8-0 IDHIT	Identifier Acceptance Filter Hit Indicator This field indicates which Identifier Acceptance filter was hit by the received message that is in the output of the Rx FIFO. If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only when IFLAG1[BUF5I] is asserted.

66.6.2.19 CAN Bit Timing register (CBT)

This register is an alternative way to store the CAN bit timing variables described in CTRL1 register. EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW are extended versions of PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields respectively.

The BTF bit selects the use of the timing variables defined in this register.

The contents of this register are not affected by soft reset.

NOTE

The CAN bit variables in CTRL1 and in CBT are stored in the same register.

NOTE

When the CAN FD feature is enabled (MCR[FDEN] is set), always set CBT[BTF].

NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

66.6.2.19.1 Offset

Register	Offset
CBT	50h

66.6.2.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BTF	EPRESDIV										ERJW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPROPSEG						EPSEG1					EPSEG2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.19.3 Fields

Field	Description
31 BTF	Bit Timing Format Enable Enables the use of extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2. and ERJW replacing the CAN bit timing variables defined in CTRL1 register. This field can be written in Freeze mode only. 0 - Extended bit time definitions disabled. 1 - Extended bit time definitions enabled.
30-21 EPRESDIV	Extended Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency when CBT[BTF] is asserted, otherwise it has no effect. It extends the CTRL1[PRESDIV] value range. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency (see Protocol timing). This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (EPRESDIV + 1)
20-16 ERJW	Extended Resync Jump Width This 5-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[RJW] value range. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = ERJW + 1.
15-10 EPROPSEG	Extended Propagation Segment This 6-bit field defines the length of the propagation segment in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PROPSEG] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = (EPROPSEG + 1) × Time-Quanta. Time-Quantum = one Sclock period.
9-5 EPSEG1	Extended Phase Segment 1

Table continues on the next page...

Field	Description
	This 5-bit field defines the length of phase segment 1 in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PSEG1] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-0 EPSEG2	Extended Phase Segment 2 This 5-bit field defines the length of phase segment 2 in the bit time when CBT[BTF] bit is asserted, otherwise it has no effect. It extends the CTRL1[PSEG2] value range. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (EPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

66.6.2.20 Rx Individual Mask registers (RXIMR0 - RXIMR63)

The RX Individual Mask Registers are used to store the acceptance masks for ID filtering in Rx MBs and the Rx FIFO.

When the Rx FIFO is disabled (MCR[RFEN] bit is negated), an individual mask is provided for each available Rx mailbox on a one-to-one correspondence. When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), an individual mask is provided for each Rx FIFO ID filter table element on a one-to-one correspondence depending on the setting of CTRL2[RFFN] (see Rx FIFO).

RXIMR0 stores the individual mask associated with either MB0 or ID filter table element 0, RXIMR1 stores the individual mask associated with either MB1 or ID filter table element 1, and so on.

RXIMR registers can only be accessed by the CPU when the module is in Freeze mode; otherwise, they are blocked by hardware. These registers are not affected by reset. They are located in RAM and must be explicitly initialized prior to any reception.

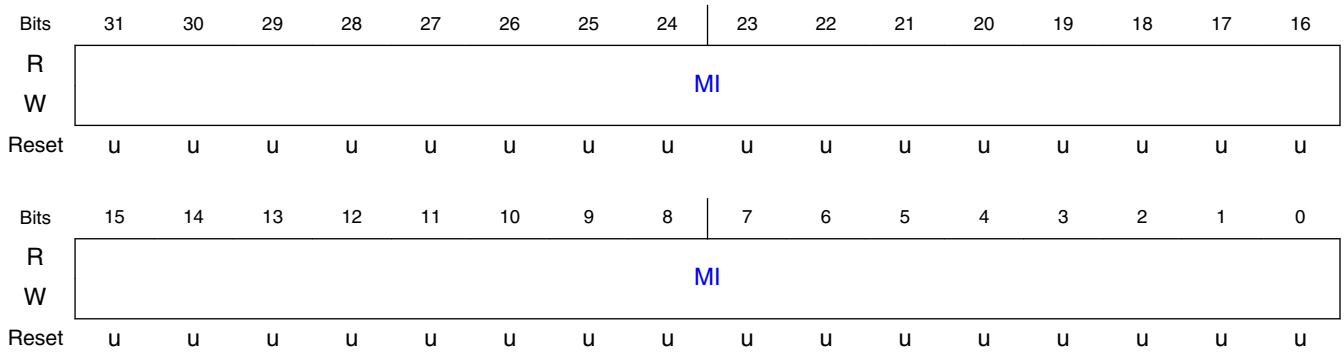
It is possible for the RXIMR memory region to be accessed as general purpose memory. See Bus interface for more information.

66.6.2.20.1 Offset

For n = 0 to 63:

Register	Offset
RXIMRn	880h + (n × 4h)

66.6.2.20.2 Diagram



66.6.2.20.3 Fields

Field	Description
31-0 MI	Individual Mask Bits Each individual mask bit masks the corresponding bit in both the mailbox filter and Rx FIFO ID filter table element in distinct ways. For mailbox filters, see the RXMGMASK register description. For Rx FIFO ID filter table elements, see the RXFGMASK register description. 0b - The corresponding bit in the filter is "don't care." 1b - The corresponding bit in the filter is checked.

66.6.2.21 Glitch Filter Width Register (GFWR)

The Glitch Filter just takes effect when FLEXCAN enters the STOP mode.

66.6.2.21.1 Offset

Register	Offset
GFWR	9E0h

66.6.2.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								GFWR							
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

66.6.2.21.3 Fields

Field	Description
31-8 —	- Reserved
7-0 GFWR	Glitch Filter Width This bit determines the Glitch Filter width. The width will be divided from Oscillator clock by GFWR values. By default, it is 5.33 µs when the oscillator is 24 MHz. Filter Pulse Width = [(GFWR FIELD + 1) x (1 / Osc. Frequency)]

66.6.2.22 Memory Error Control register (MECR)

This register contains control bits for memory error detection and correction (ECC).

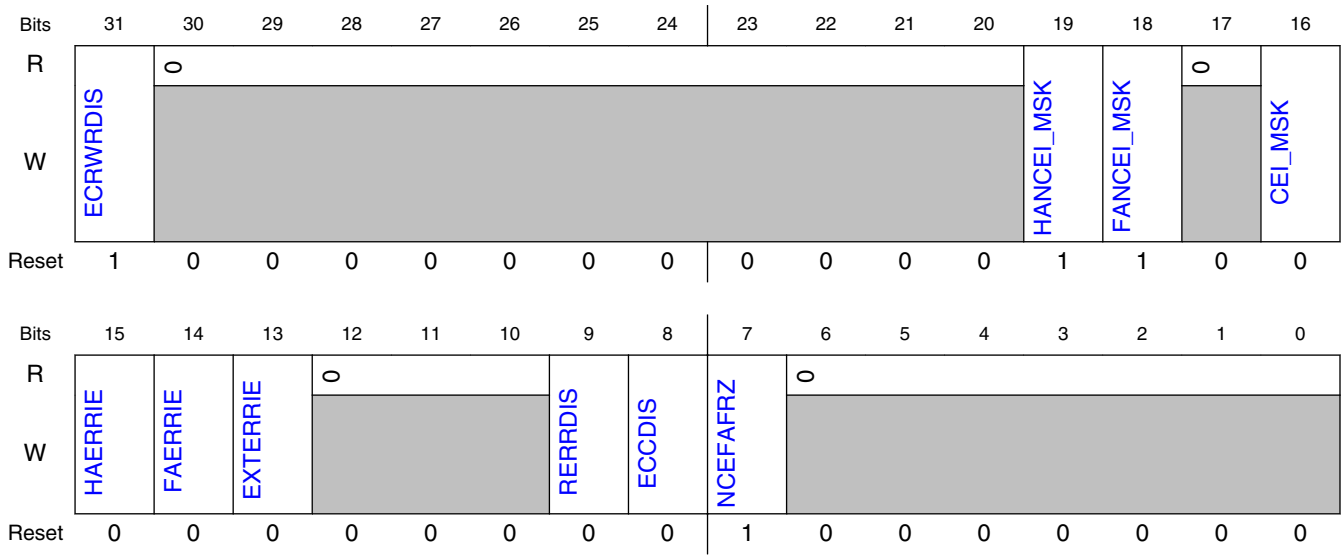
NOTE

When bit CTRL2[ECRWRE] is 0, writes in this register are blocked, except in the ECRWRDIS bit.

66.6.2.22.1 Offset

Register	Offset
MECR	AE0h

66.6.2.22.2 Diagram



66.6.2.22.3 Fields

Field	Description
31 ECRWRDIS	Error Configuration Register Write Disable Disables writes on this register. This bit is automatically set to 1 (disabled) when CTRL2[ECRWRE] is enabled. The protocol described in section Detection and correction of memory errors must be followed. 0 - Write is enabled. 1 - Write is disabled.
30-20 —	Reserved
19 HANCEI_MSK	Host Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the host (CPU). 0 - Interrupt is disabled. 1 - Interrupt is enabled.
18 FANCEI_MSK	FlexCAN Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the FlexCAN internal processes. 0 - Interrupt is disabled. 1 - Interrupt is enabled.
17 —	Reserved
16	Correctable Errors Interrupt Mask

Table continues on the next page...

Field	Description
CEI_MSK	Enables the interrupt in case of correctable errors detected in memory reads issued by the host or FlexCAN internal processes. 0 - Interrupt is disabled. 1 - Interrupt is enabled.
15 HAERRIE	Host Access Error Injection Enable Enables the injection of errors only in memory reads issued by the host (CPU). 0 - Injection is disabled. 1 - Injection is enabled.
14 FAERRIE	FlexCAN Access Error Injection Enable Enables the injection of errors only in memory reads issued by the FlexCAN internal processes. 0 - Injection is disabled. 1 - Injection is enabled.
13 EXTERRIE	Extended Error Injection Enable Memory accesses performed by internal FlexCAN processes are 64-bit. This bit extends the error injection on 32-bit memory accesses to the complementary 32-bit word using the same 32-bit error injection data and parity words. See Error Injection Data Pattern register (ERRIDPR) and Error Injection Parity Pattern register (ERRIPPR). 0 - Error injection is applied only to the 32-bit word. 1 - Error injection is applied to the 64-bit word.
12-10 —	Reserved
9 RERRDIS	Error Report Disable Disables the update of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active. **NOTE:** When reading the report registers, this bit must be set to assure coherence on the consecutive register reads. 0 - Enable updates of the error report registers. 1 - Disable updates of the error report registers.
8 ECCDIS	Error Correction Disable Disables completely the memory detection and correction mechanism. Besides disabling the error report mechanism, it also stops the update of the error-related flags and generation of bus transfer errors. The parity bits continue being calculated and written into memory on write transactions. 0 - Enable memory error correction. 1 - Disable memory error correction.
7 NCEFAFRZ	Non-Correctable Errors In FlexCAN Access Put Device In Freeze Mode Determines the response when a non-correctable error is detected in a memory read performed by FlexCAN internal processes. In this case, entering Freeze mode prevents corrupted data from being treated as valid by FlexCAN internal processes. 0 - Keep normal operation. 1 - Put FlexCAN in Freeze mode (see section "Freeze mode").
6-0 —	Reserved

66.6.2.23 Error Injection Address register (ERRIAR)

This register holds the address where error is to be injected.

Use the following table to convert from the memory map address to the location in the physical FlexCAN RAM (where pairs of values are provided, the first is the address for MCR[FDEN] negated, the second is for MCR[FDEN] asserted):

RAM contents	Injection address	Memory map
FlexCAN Registers	Not mapped	—
MBs	0000h	0080h
Reserved	—	0480h
RXIMRs	0400h	0880h
Reserved	—	0980h
RXFIR_0	0500h	0A80h
RXFIR_1	0504h	0A84h
RXFIR_2	0508h	0A88h
RXFIR_3	050Ch	0A8Ch
RXFIR_4	0510h	0A90h
RXFIR_5	0514h	0A94h
Reserved	—	0A98h
RXMGMASK	0520h	0AA0h
RXFGMASK	0524h	0AA4h
RX14MASK	0528h	0AA8h
RX15MASK	052Ch	0AACh
Tx_SMB	0530h	0AB0h / 0F28h
Rx_SMB0	0540h / 0578h	0AC0h / 0F70h
Rx_SMB1	0550h / 05C0h	0AD0h / 0FB8h
ECC Registers	Not mapped	0AE0h
Reserved	—	0B00h
CAN FD Registers	Not mapped	0C00h
Reserved	—	0C0Ch

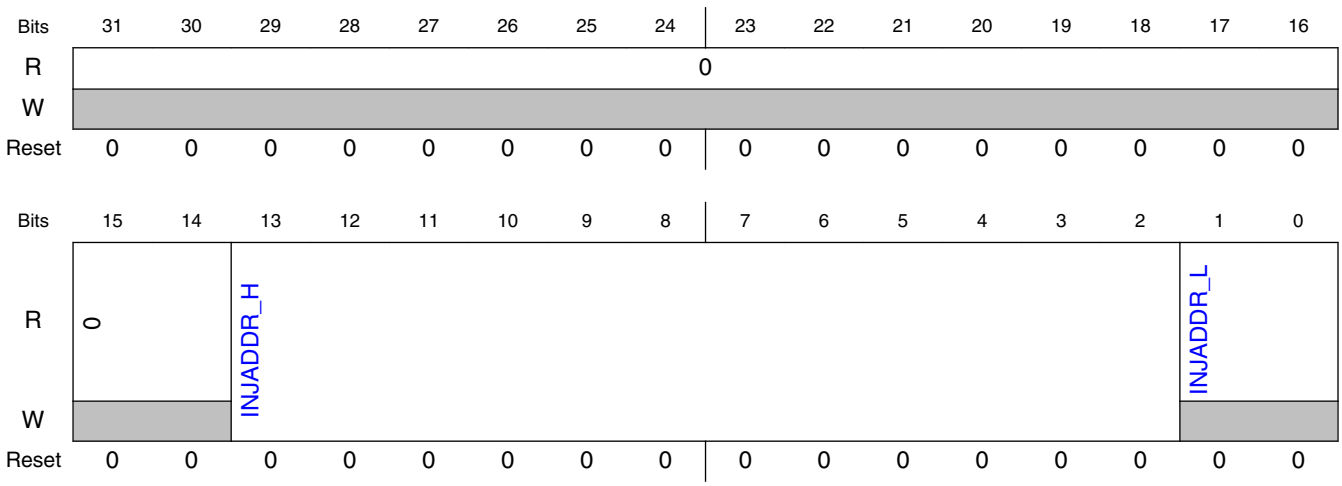
NOTE

For RXFIR addresses, you must inject ECC errors in host access only.

66.6.2.23.1 Offset

Register	Offset
ERRIAR	AE4h

66.6.2.23.2 Diagram



66.6.2.23.3 Fields

Field	Description
31-14 —	Reserved
13-2 INJADDR_H	Error Injection Address High This read-write field defines the twelve most significant bits of the physical RAM address where error is to be injected (see table above).
1-0 INJADDR_L	Error Injection Address Low This read-only field defines the two least significant bits of the physical RAM address where error is to be injected. They ensure that the address is on a thirty-two-bit boundary.

66.6.2.24 Error Injection Data Pattern register (ERRIDPR)

Holds the error pattern to be injected in the data word read from memory.

66.6.2.24.1 Offset

Register	Offset
ERRIDPR	AE8h

66.6.2.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DFLIP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DFLIP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.24.3 Fields

Field	Description
31-0 DFLIP	Data flip pattern Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert.

66.6.2.25 Error Injection Parity Pattern register (ERRIPPR)

Holds the error pattern to be injected in parity bits read from memory along with data word.

Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.

66.6.2.25.1 Offset

Register	Offset
ERRIPPR	AECh

66.6.2.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			PFLIP3					0			PFLIP2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PFLIP1					0			PFLIP0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.25.3 Fields

Field	Description
31-29 —	Reserved
28-24 PFLIP3	Parity Flip Pattern For Byte 3 (most significant)
23-21 —	Reserved
20-16 PFLIP2	Parity Flip Pattern For Byte 2
15-13 —	Reserved
12-8 PFLIP1	Parity Flip Pattern For Byte 1
7-5 —	Reserved
4-0 PFLIP0	Parity Flip Pattern For Byte 0 (Least Significant)

66.6.2.26 Error Report Address register (RERRAR)

Reports the address used for an access in which an error (correctable or non-correctable) was detected, and also reports the identification of the source of that access.

This address is always reported using a 32-bit alignment. Non-aligned accesses (ERRADDR[1:0] non-0) are reported with the address aligned and data is reported in RERRDR accordingly shifted. In case of errors detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word which the error was detected is reported. In case of errors detected in more than one 32-bit word, only the least significant address is reported.

Table 66-22. Source of memory access

SAID[2:0]	Error during...
0	Move-out FlexCAN access
1	Move-in
2	Tx Arbitration
3	Rx Matching
4	Move-out Host access
5-7	Reserved

66.6.2.26.1 Offset

Register	Offset
RERRAR	AF0h

66.6.2.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							NCE	0					SAID		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ERRADDR													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.26.3 Fields

Field	Description
31-25 —	Reserved

Table continues on the next page...

Field	Description
24 NCE	Non-Correctable Error Indicates that the report is due to a non-correctable error. 0 - Reporting a correctable error 1 - Reporting a non-correctable error
23-19 —	Reserved
18-16 SAID	SAID SAID[2] — Identification of the requester of the memory read request: • 0 = Requested by FlexCAN internal processes • 1 = Requested by host (CPU) SAID[1] — Details of FlexCAN operation: • 0 = Move • 1 = Scanning SAID[0] — Operation that requested the memory read: • 0 = Transmission • 1 = Reception For more information see Table 66-22.
15-14 —	Reserved
13-0 ERRADDR	Address Where Error Detected See the description of the Error Injection Address Register (ERRIAR).

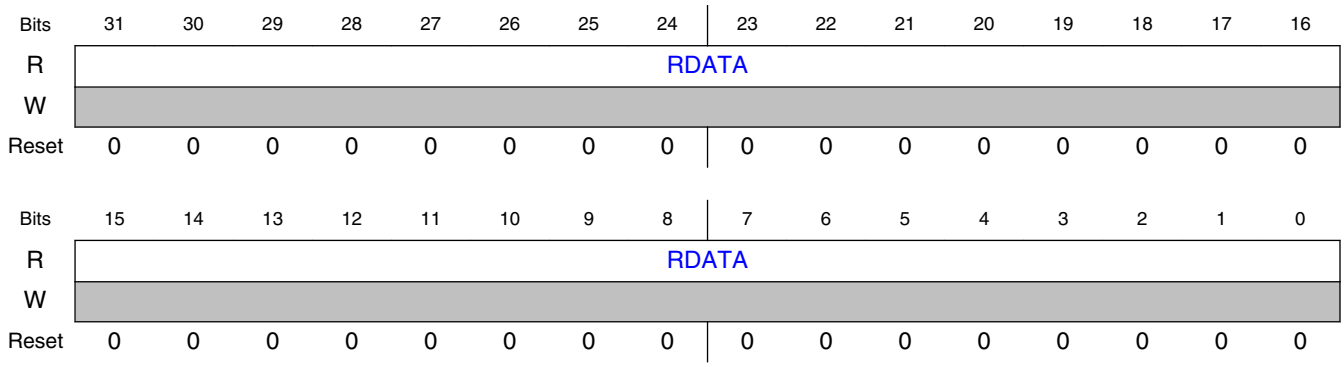
66.6.2.27 Error Report Data register (RERRDR)

Reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit (see Table 66-26) when reading a message buffer.

66.6.2.27.1 Offset

Register	Offset
RERRDR	AF4h

66.6.2.27.2 Diagram



66.6.2.27.3 Fields

Field	Description
31-0 RDATA	Raw data word read from memory with error

66.6.2.28 Error Report Syndrome register (RERRSYNR)

Holds the syndrome detected in a memory read with error. It also reports the bytes which were read in this 32-bit read transaction.

Each SYNDn field indicates the type of error and which bit in byte (n) is affected by the error. (SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.)

Table 66-23. Syndrome definition

SYNDn (hex)	Type	Bit affected
00	—	none (no error)
01	Code	0
02	Code	1
04	Code	2
07	Data	5
08	Code	3
0E	Data	7
10	Code	4
13	Data	2

Table continues on the next page...

Table 66-23. Syndrome definition (continued)

SYNDn (hex)	Type	Bit affected
15	Data	6
16	Data	1
19	Data	3
1A	Data	4
1C	Data	0
06	—	All-zeros non-correctable error
1F	—	All-ones non-correctable error
All others	—	Non-correctable error

Each BEn field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, even for the non-read ones. Errors detected in non-read bytes are indicated (see Error indication) and reported (see Error reporting).

66.6.2.28.1 Offset

Register	Offset
RERRSYNR	AF8h

66.6.2.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BE3	0		SYND3					BE2	0		SYND2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BE1	0		SYND1					BE0	0		SYND0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.6.2.28.3 Fields

Field	Description
31 BE3	Byte Enabled For Byte 3 (most significant) 0 - The byte was not read. 1 - The byte was read.

Table continues on the next page...

Field	Description
30-29 —	Reserved
28-24 SYND3	Error Syndrome For Byte 3 (most significant) See Table 66-23.
23 BE2	Byte Enabled For Byte 2 0 - The byte was not read. 1 - The byte was read.
22-21 —	Reserved
20-16 SYND2	Error Syndrome For Byte 2 See Table 66-23.
15 BE1	Byte Enabled For Byte 1 0 - The byte was not read. 1 - The byte was read.
14-13 —	Reserved
12-8 SYND1	Error Syndrome for Byte 1 See Table 66-23.
7 BE0	Byte Enabled For Byte 0 (least significant) 0 - The byte was not read. 1 - The byte was read.
6-5 —	Reserved
4-0 SYND0	Error Syndrome For Byte 0 (least significant) See Table 66-23.

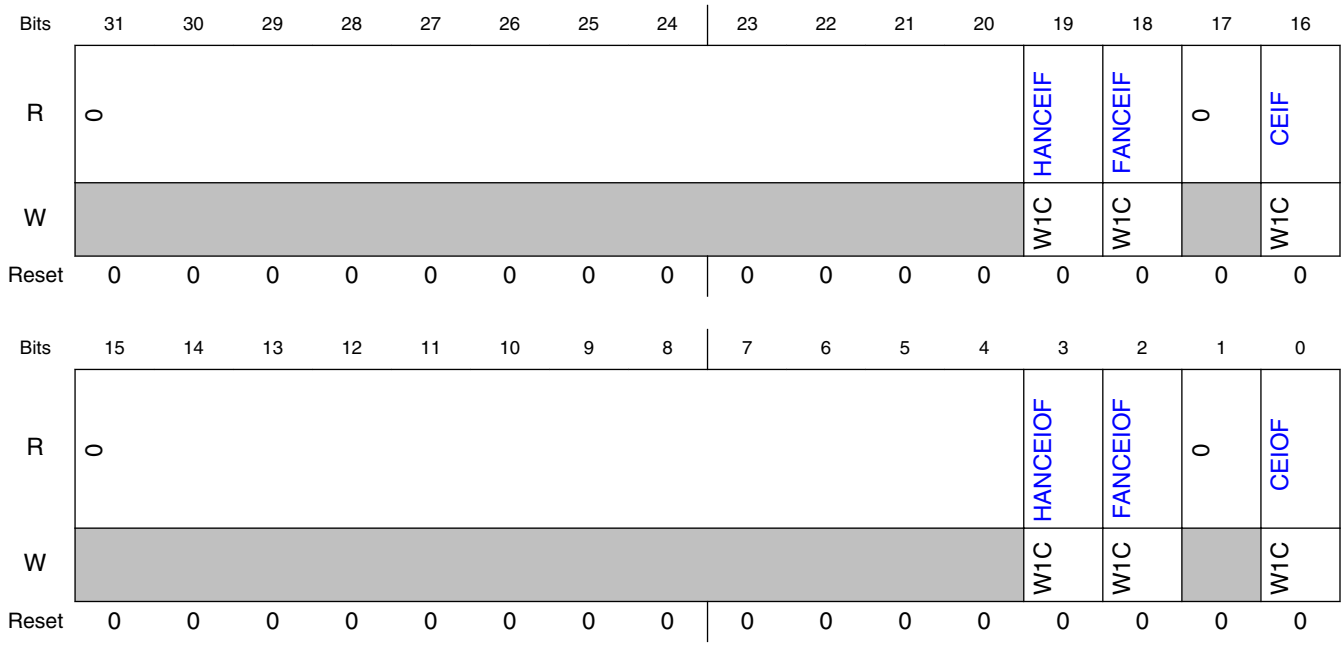
66.6.2.29 Error Status register (ERRSR)

Holds the status bits of the error correction and detection operations. These flags can be cleared by writing 1 to them. Writing 0 has no effect.

66.6.2.29.1 Offset

Register	Offset
ERRSR	AFCh

66.6.2.29.2 Diagram



66.6.2.29.3 Fields

Field	Description
31-20 —	Reserved
19 HANCEIF	Host Access With Non-Correctable Error Interrupt Flag Indicates that a non-correctable error was detected in a memory read initiated by host. A bus transfer error is asserted for that access. If MECR[HANCEI_MSK] is set, the interrupt is asserted. 0 - No non-correctable errors were detected in host accesses so far. 1 - A non-correctable error was detected in a host access.
18 FANCEIF	FlexCAN Access With Non-Correctable Error Interrupt Flag Indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes. If MECR[FANCEI_MSK] is set, the interrupt is asserted. 0 - No non-correctable errors were detected in FlexCAN accesses so far. 1 - A non-correctable error was detected in a FlexCAN access.
17 —	Reserved
16 CEIF	Correctable Error Interrupt Flag Indicates that a correctable error was detected in a memory read. If MECR[CEI_MSK] is set, the interrupt is asserted. 0 - No correctable errors were detected so far.

Table continues on the next page...

Field	Description
	1 - A correctable error was detected.
15-4 —	Reserved
3 HANCEIOF	Host Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory read initiated by host when HANCEIF was set. No interrupt is associated with this flag. See Error indication. 0 - No overrun on non-correctable errors in host access 1 - Overrun on non-correctable errors in host access
2 FANCEIOF	FlexCAN Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes when FANCEIF was set. No interrupt is associated to this flag. See Error indication. 0 - No overrun on non-correctable errors in FlexCAN access 1 - Overrun on non-correctable errors in FlexCAN access
1 —	Reserved
0 CEIOF	Correctable Error Interrupt Overrun Flag Indicates that a correctable error was detected in a memory read when CEIF was set. No interrupt is associated to this flag. See Error indication. 0 - No overrun on correctable errors 1 - Overrun on correctable errors

66.6.2.30 CAN FD Control register (FDCTRL)

This register contains control bits for the CAN FD operation. It also defines the data size of message buffers allocated in different partitions of RAM (memory blocks) as described in the table below.

When an 8-byte payload is selected:

- Block R0 allocates MB0 to MB31.
- Block R1 allocates MB32 to MB63.

When a payload larger than 8 bytes is selected, the maximum number of MBs in a block is limited as described below:

Table 66-24. Number of message buffers

Payload size	Maximum number of message buffers per RAM block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

NOTE

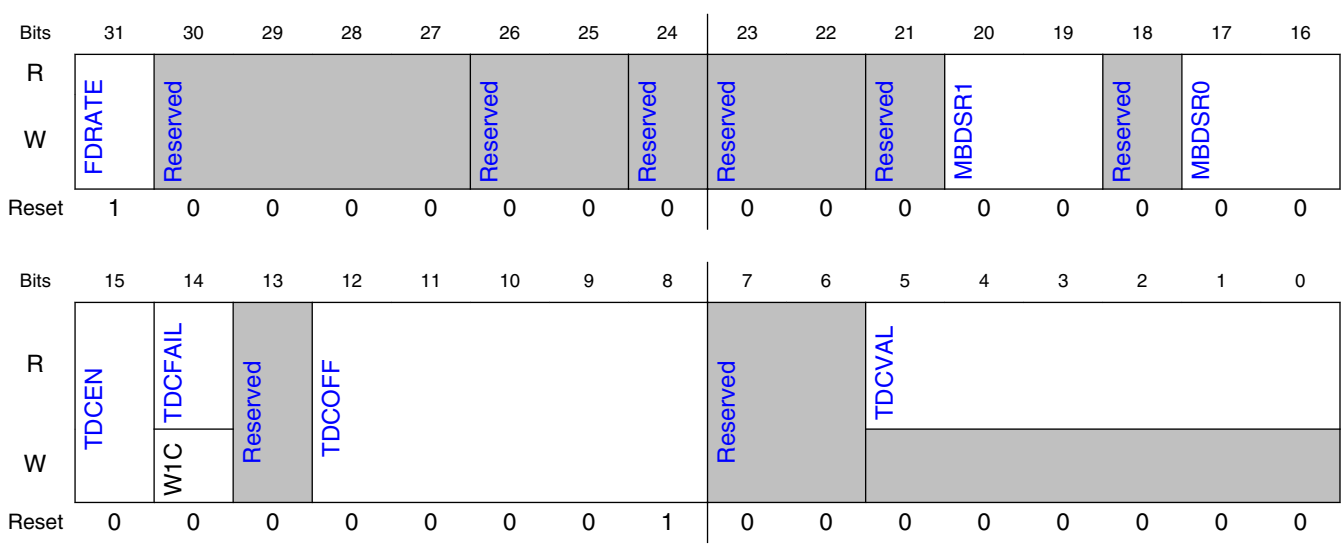
One memory block fits exactly 32 MBs with an 8-byte payload. For other possible payload sizes, empty memory may exist between the last MB in a block and the beginning of the next block. This empty memory corresponds to less than one MB, and must not be used.

The contents of this register are not affected by soft reset.

66.6.2.30.1 Offset

Register	Offset
FDCTRL	C00h

66.6.2.30.2 Diagram



66.6.2.30.3 Fields

Field	Description
31 FDRATE	Bit Rate Switch Enable This bit enables the effect of the Bit Rate Switch (BRS bit) during the data phase of Tx messages. The CPU can write this bit any time. However, its effect turns active only when the CAN bus is in Wait for Bus Idle, Bus Idle, or Bus Off state, or when the current frame under reception or transmission reaches the interframe space. By negating FDCTRL[FDRATE], the CPU can force all bits in CAN FD messages to be transmitted in nominal bit rate, no matter what the value is in the BRS bit of the Tx MBs. 0 - Transmit a frame in nominal rate. The BRS bit in the Tx MB has no effect. 1 - Transmit a frame with bit rate switching if the BRS bit in the Tx MB is recessive.
30-27 —	Reserved
26-25 —	Reserved
24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20-19 MBDSR1	Message Buffer Data Size for Region 1 This two-bit field selects the data size (8, 16, 32, or 64 bytes) for region R1 of message buffers allocated in RAM. It can be written in Freeze mode only. 00 - Selects 8 bytes per message buffer. 01 - Selects 16 bytes per message buffer. 10 - Selects 32 bytes per message buffer. 11 - Selects 64 bytes per message buffer.
18 —	Reserved
17-16 MBDSR0	Message Buffer Data Size for Region 0 This two-bit field selects the data size (8, 16, 32, or 64 bytes) for region R0 of message buffers allocated in RAM. It can be written in Freeze mode only. 00 - Selects 8 bytes per message buffer. 01 - Selects 16 bytes per message buffer. 10 - Selects 32 bytes per message buffer. 11 - Selects 64 bytes per message buffer.
15 TDCEN	Transceiver Delay Compensation Enable This bit can be used to enable and disable the TDC feature. It can be written in Freeze mode only.

Table continues on the next page...

Field	Description
	NOTE: See Transmitter delay compensation in the CAN Protocol standard (ISO 11898-1) for details. **NOTE:** TDC must be disabled when Loop Back mode is enabled (see CTRL1[LPB]). 0 - TDC is disabled 1 - TDC is enabled
14 TDCFAIL	Transceiver Delay Compensation Fail This bit indicates when the Transceiver Delay Compensation (TDC) mechanism is out of range, unable to compensate the transceiver's loop delay and successfully compare the delayed received bits to the transmitted ones (see Transceiver delay compensation). TDCFAIL sets in the first time FlexCAN detects the out of range condition. The CPU needs to write one to clear it. 0 - Measured loop delay is in range. 1 - Measured loop delay is out of range.
13 —	Reserved
12-8 TDCOFF	Transceiver Delay Compensation Offset This bit field contains the offset value to be added to the measured transceiver's loop delay in order to define the position of the delayed comparison point when bit rate switching is active. See Transceiver delay compensation for more details on how the loop delay measurement is performed. TDCOFF can be written in Freeze mode only. Its value can be defined in Protocol Engine (PE) Clock periods (CANCLK, see Protocol timing for more details), and must be selected to be smaller than the CAN bit duration in the data bit rate for proper operation. **NOTE:** It is not recommended to have TDCOFF equal to zero.
7-6 —	Reserved
5-0 TDCVAL	Transceiver Delay Compensation Value This register contains the value of the transceiver loop delay measured from the transmitted EDL to R0 transition edge to the respective received one added to the TDCOFF value specified in the FDCTRL register. This value is an integer multiple of the Protocol Engine (PE) Clock period (CANCLK). See Protocol timing for more details on how the loop delay measurement is performed.

66.6.2.31 CAN FD Bit Timing register (FDCBT)

This register stores the CAN bit timing variables used in the data phase of CAN FD messages when the FDCTRL[FDRATE] is set, compatible with CAN FD specification. FPRESDIV, FPROPSEG, FPSEG1, FPSEG2, and FRJW are used to define the time quantum duration, the number of time quanta per CAN bit, and the sample point position for the data bit rate portion of a CAN FD message with the BRS bit set.

The contents of this register are not affected by soft reset.

NOTE

The sum of the Fast Propagation Segment (FPROPSEG) and Fast Phase Segment 1 (FPSEG1) must be at least two time quanta.

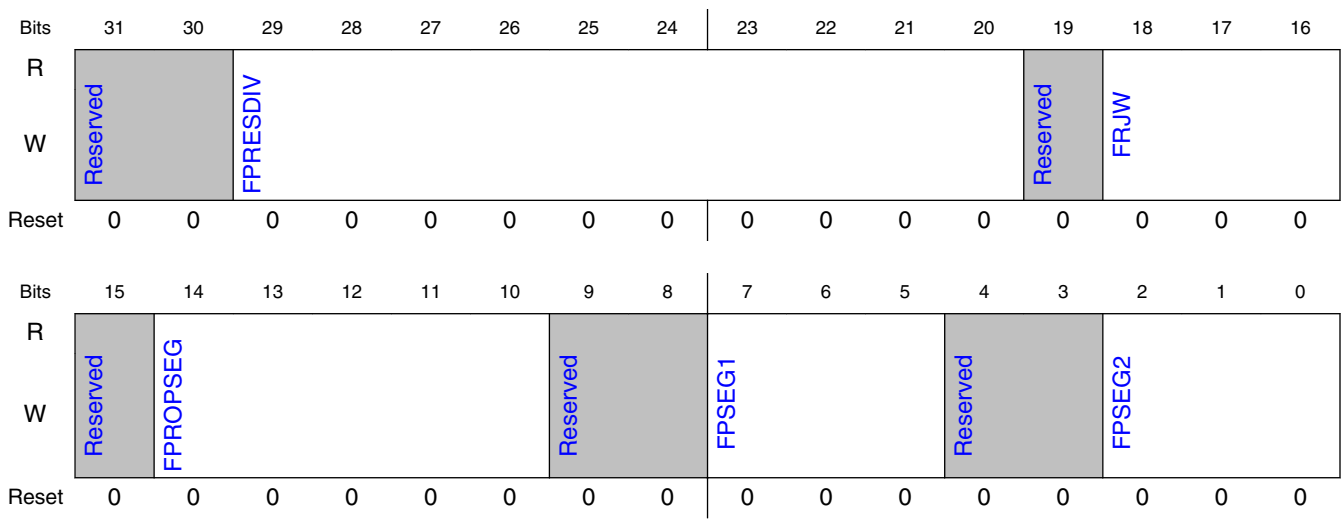
NOTE

The user must ensure bit time settings and protocol engine tolerance are in compliance with the CAN Protocol standard (ISO 11898-1).

66.6.2.31.1 Offset

Register	Offset
FDCBT	C04h

66.6.2.31.2 Diagram



66.6.2.31.3 Fields

Field	Description
31-30 —	Reserved
29-20 FPRESDIV	Fast Prescaler Division Factor This 10-bit field defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency in the data bit rate portion of a CAN FD message with the BRS bit set.

Table continues on the next page...

Field	Description
	The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (FPRESDIV + 1). **NOTE:** To minimize errors when processing FD frames, use the same value for FPRESDIV and PRESDIV (in CBT or CTRL1). For more details see the first NOTE in section CAN FD frames.
19 —	Reserved
18-16 FRJW	Fast Resync Jump Width This 3-bit field defines the maximum number of time quanta that a bit time can be changed by one resynchronization in the data bit rate portion of a CAN FD message with the BRS bit set. One time quantum is equal to the Sclock period. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = FSJW + 1.
15 —	Reserved
14-10 FPROPSEG	Fast Propagation Segment This 5-bit field defines the length of the propagation segment in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = FPROPSEG × Time-Quanta. Time-Quantum = one Sclock period.
9-8 —	Reserved
7-5 FPSEG1	Fast Phase Segment 1 This 3-bit field defines the length of phase segment 1 in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Segment 1 = (FPSEG1 + 1) × Time-Quanta. Time-Quantum = one Sclock period.
4-3 —	Reserved
2-0 FPSEG2	Fast Phase Segment 2 This 3-bit field defines the length of phase segment 2 in the data bit rate portion of a CAN FD message with the BRS bit set. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Segment 2 = (FPSEG2 + 1) × Time-Quanta. Time-Quantum = one Sclock period.

66.6.2.32 CAN FD CRC register (FDCRC)

This register provides information about the CRC of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats, as shown below.

The CRC_15 polynomial is used for all frames in CAN format. The CRC_17 polynomial is used for frames in CAN FD format with a DATA FIELD up to sixteen bytes. The CRC_21 polynomial is used for frames in CAN FD format with a DATA FIELD longer than sixteen bytes. Each polynomial shown below results in a Hamming distance of 6. This register is updated at the same time the Tx Interrupt flag is asserted.

$$CRC_15 = C599h:\ (x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1)$$

$$CRC_17 = 3685Bh:\ (x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1)$$

$$CRC_21 = 302899h:\ (x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1)$$

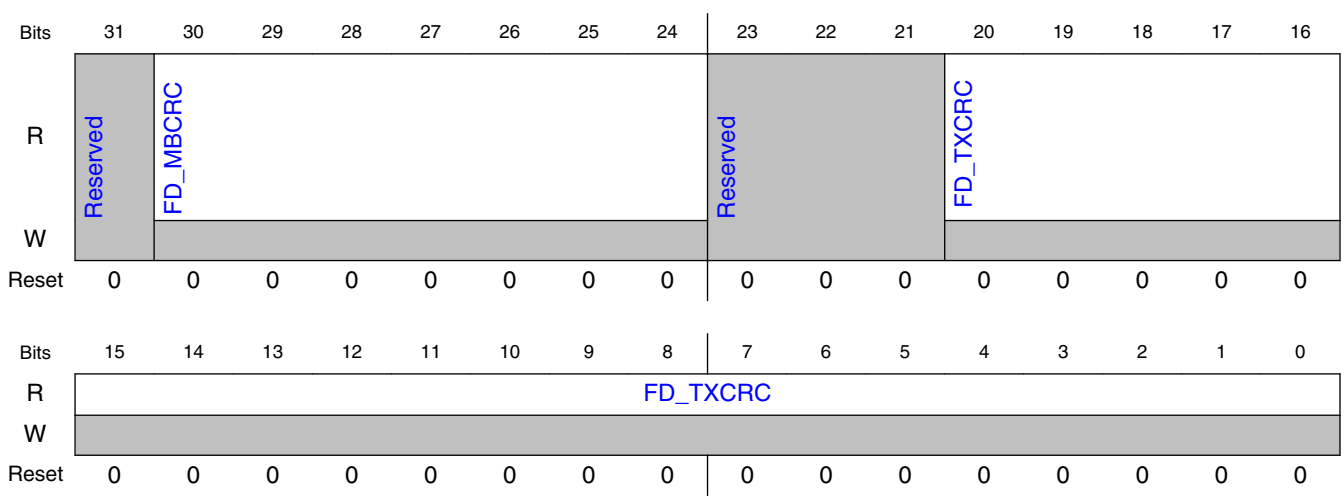
NOTE

See CRC sequence calculation in the CAN Protocol standard (ISO 11898-1) for details.

66.6.2.32.1 Offset

Register	Offset
FDCRC	C08h

66.6.2.32.2 Diagram



66.6.2.32.3 Fields

Field	Description
31 —	Reserved
30-24 FD_MBCRC	CRC Mailbox Number for FD_TXCRC This field indicates the number of the mailbox corresponding to the value in the FD_TXCRC field, for both FD and non-FD frames. It reports the same information as in CRCR[MBCRC].
23-21 —	Reserved
20-0 FD_TXCRC	Extended Transmitted CRC value This 21-bit field contains the CRC value calculated over the most recent transmitted message. Different CRC polynomials are used for different frame formats. A 15-bit polynomial, CRC_15, is used for all frames in CAN format. The second 17-bit polynomial, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third 21-bit polynomial, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. For CRC_15 and CRC_17, the 6 most significant bits and the 4 most significant bits are reported as zeros, respectively. For CRC_15, this register has the same content as CRC register.

66.6.3 Message buffer structure

The message buffer structure used by the FlexCAN module is represented in the following figure. Both extended (29-bit identifier) and standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16, 24, 40, or 72 bytes, depending on the quantity of data bytes allocated for the message payload: 8, 16, 32, or 64 data bytes, respectively.

The memory area 80h–47Fh is used by the mailboxes. When CAN FD is enabled, the exact address for each MB depends on the size of its payload. See FlexCAN memory partition for CAN FD for more detailed information.

Table 66-25. Message buffer structure — example with 64-byte payload

	31	30	29	28	27	24	23	22	21	20	19	18	17	16	15	8	7	0
0h	EDL	BRS	ESI		CODE			SRR	IDE	RTR	DLC				TIME STAMP			
4h	PRIO			ID (standard/extended)											ID (extended)			
8h	Data byte 0						Data byte 1								Data byte 2		Data byte 3	
Ch	Data byte 4						Data byte 5								Data byte 6		Data byte 7	
10h	Data byte 8						Data byte 9								Data byte 10		Data byte 11	
14h	Data byte 12						Data byte 13								Data byte 14		Data byte 15	

Table continues on the next page...

Table 66-25. Message buffer structure — example with 64-byte payload (continued)

18h	Data byte 16	Data byte 17	Data byte 18	Data byte 19
1Ch	Data byte 20	Data byte 21	Data byte 22	Data byte 23
20h	Data byte 24	Data byte 25	Data byte 26	Data byte 27
24h	Data byte 28	Data byte 29	Data byte 30	Data byte 31
28h	Data byte 32	Data byte 33	Data byte 34	Data byte 35
2Ch	Data byte 36	Data byte 37	Data byte 38	Data byte 39
30h	Data byte 40	Data byte 41	Data byte 42	Data byte 43
34h	Data byte 44	Data byte 45	Data byte 46	Data byte 47
38h	Data byte 48	Data byte 49	Data byte 50	Data byte 51
3Ch	Data byte 52	Data byte 53	Data byte 54	Data byte 55
40h	Data byte 56	Data byte 57	Data byte 58	Data byte 59
44h	Data byte 60	Data byte 61	Data byte 62	Data byte 63

= Unimplemented or reserved

EDL — Extended Data Length

This bit distinguishes between CAN format and CAN FD format frames. The EDL bit must not be set for message buffers configured to RANSWER with code field 1010b (see table below).

BRS — Bit Rate Switch

This bit defines whether the bit rate is switched inside a CAN FD format frame.

ESI — Error State Indicator

This bit indicates if the transmitting node is error active or error passive.

CODE — Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 66-26 and Table 66-27. See Functional description for additional information.

Table 66-26. Message buffer code for Rx buffers

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0000b: INACTIVE — MB is not active.	INACTIVE	—	—	—	MB does not participate in the matching process.

Table continues on the next page...

Table 66-26. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0100b: EMPTY — MB is active and empty.	EMPTY	—	FULL	—	When a frame is received successfully (after the Move-in process), the CODE field is automatically updated to FULL.
0010b: FULL — MB is full.	FULL	Yes	FULL	—	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. See Matching process for matching details related to FULL code.
		No	OVERRUN	—	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See Matching process for details about overrun behavior.
0110b: OVERRUN — MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	—	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB then the code returns to FULL.
		No	OVERRUN	—	If the CODE field already indicates OVERRUN, and another new frame must be moved, the

Table continues on the next page...

Table 66-26. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
					MB will be overwritten again, and the code will remain OVERRUN. See Matching process for details about overrun behavior.
1010b: RANSWER[4] — A frame was configured to recognize a Remote Request frame and transmit a Response frame in return.[5]	RANSWER	—	TANSWER(1110b)	0	A Remote Answer was configured to recognize a remote request frame received. After that an MB is set to transmit a response frame. The code is automatically changed to TANSWER (1110b). See Matching process for details. If CTRL2[RRS] is negated, transmit a response frame whenever a remote request frame with the same ID is received.
		—	—	1	This code is ignored during matching and arbitration process. See Matching process for details.
CODE[0]=1: BUSY — FlexCAN is updating the contents of the MB. The CPU must not access the MB.	BUSY[6]	—	FULL	—	Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE.
		—	OVERRUN	—	

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered a successful reception after the frame to be moved to MB (move-in process). See Move-in for details.
3. Remote Request Stored bit, see "Control 2 register (CTRL2)" for details.
4. Code 1010b is not considered Tx and an MB with this code should not be aborted.
5. Code 1010b must be used in message buffers configured in CAN FD format, having the EDL bit set.
6. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 66-27. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
1000b: INACTIVE — MB is not active	INACTIVE	—	—	MB does not participate in arbitration process.
1001b: ABORT — MB is aborted	ABORT	—	—	MB does not participate in arbitration process.
1100b: DATA — MB is a Tx data frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.
1100b: REMOTE — MB is a Tx Remote Request frame (MB RTR must be 1)	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.
1110b: TANSWER — MB is a Tx Response frame from an incoming Remote Request frame	TANSWER	—	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once, and then the code will automatically return to RANSWER (1010b). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See Matching process and Arbitration process for details.

SRR — Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to one by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

1 = Recessive value is compulsory for transmission in extended format frames

0 = Dominant is not a valid value for transmission in extended format frames

IDE — ID Extended Bit

This field identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

RTR — Remote Transmission Request

This bit affects the behavior of remote frames and is part of the reception filter. See Table 66-26, Table 66-27, and the description of the RRS field in Control 2 register (CTRL2) for additional details.

If FlexCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as an arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FlexCAN module treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

1 = Indicates the current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx then incoming remote request frames may be stored.

0 = Indicates the current MB has a data frame to be transmitted. In Rx MB it may be considered in matching processes.

NOTE

When configuring CAN FD frames, the RTR bit must be negated.

DLC — Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 8h–Fh of the MB space (see Table 66-25). In reception, this field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see Table 66-28).

TIME STAMP — Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO — Local priority

This 3-bit field is used only when MCR[LPRIO_EN] is set, and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.

ID — Frame Identifier

In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.

DATA BYTE 0 to 63 — Data Field

Up to sixty four bytes can be used for a data frame, depending on the size of payload selected for the message buffers.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC as shown in the table below.

Table 66-28. DATA BYTEs validity

DLC	Valid DATA BYTEs
0	none
1	DATA BYTE 0
2	DATA BYTE 0 to 1
3	DATA BYTE 0 to 2
4	DATA BYTE 0 to 3
5	DATA BYTE 0 to 4
6	DATA BYTE 0 to 5
7	DATA BYTE 0 to 6
8	DATA BYTE 0 to 7
9	DATA BYTE 0 to 11
10	DATA BYTE 0 to 15
11	DATA BYTE 0 to 19
12	DATA BYTE 0 to 23
13	DATA BYTE 0 to 31
14	DATA BYTE 0 to 47
15	DATA BYTE 0 to 63

66.6.4 FlexCAN memory partition for CAN FD

When CAN FD is enabled, the FlexCAN RAM can be partitioned in blocks of 512 bytes. Each block can accommodate a number of message buffers which depends on the configuration provided by FDCTRL[MBDSRn] bit fields as shown in table below.

Table 66-29. RAM partition

RAM block	Number of MBs with 8 bytes (default range)	Size control bit field in FDCTRL register	Number of MBs of different sizes, per block
0	0 to 31	MBDSR0	MBDSR0=00, 32 MBs with 8 bytes payload MBDSR0=01, 21 MBs with 16 bytes payload MBDSR0=10, 12 MBs with 32 bytes payload MBDSR0=11, 7 MBs with 64 bytes payload
1	32 to 63	MBDSR1	MBDSR1=00, 32 MBs with 8 bytes payload MBDSR1=01, 21 MBs with 16 bytes payload MBDSR1=10, 12 MBs with 32 bytes payload MBDSR1=11, 7 MBs with 64 bytes payload

When payload sizes of 16, 32, or 64 bytes are configured in some or all RAM blocks, the total number of MBs and its respective number order may differ from the default configuration of 8 bytes. For example, suppose Block0 is configured to 8 bytes payload, Block1 to 16 bytes, Then the following table indicates how the message buffers will be arranged in RAM.

Table 66-30. RAM partition example

RAM block	Payload size	Number of MBs in the RAM block	Message buffer range
0	FDCTRL[MBDSR0]=00, 8 bytes payload	32	0 to 31
1	FDCTRL[MBDSR1]=01, 16 bytes payload	21	32 to 52

66.6.5 FlexCAN message buffer memory map

The FlexCAN memory buffers are allocated in memory according to the tables below.

Table 66-31. 8-byte message buffers

Address offset (hex)	MBDSR=b00 8-byte payload
0080	MB0
0090	MB1
00A0	MB2
00B0	MB3
00C0	MB4
00D0	MB5
00E0	MB6
00F0	MB7
0100	MB8
0110	MB9
0120	MB10
0130	MB11
0140	MB12
0150	MB13
0160	MB14
0170	MB15
0180	MB16
0190	MB17
01A0	MB18
01B0	MB19
01C0	MB20
01D0	MB21
01E0	MB22
01F0	MB23
0200	MB24
0210	MB25
0220	MB26
0230	MB27
0240	MB28
0250	MB29
0260	MB30
0270	MB31
0280	MB32
0290	MB33
02A0	MB34
02B0	MB35
02C0	MB36
02D0	MB37

Table continues on the next page...

Table 66-31. 8-byte message buffers (continued)

Address offset (hex)	MBDSR=b00 8-byte payload
02E0	MB38
02F0	MB39
0300	MB40
0310	MB41
0320	MB42
0330	MB43
0340	MB44
0350	MB45
0360	MB46
0370	MB47
0380	MB48
0390	MB49
03A0	MB50
03B0	MB51
03C0	MB52
03D0	MB53
03E0	MB54
03F0	MB55
0400	MB56
0410	MB57
0420	MB58
0430	MB59
0440	MB60
0450	MB61
0460	MB62
0470	MB63

Table 66-32. 16-byte message buffers

Address offset (hex)	MBDSR=b01 16-byte payload
0080	MB0
0098	MB1
00B0	MB2
00C8	MB3
00E0	MB4
00F8	MB5
0110	MB6

Table continues on the next page...

Table 66-32. 16-byte message buffers (continued)

Address offset (hex)	MBDSR=b01 16-byte payload
0128	MB7
0140	MB8
0158	MB9
0170	MB10
0188	MB11
01A0	MB12
01B8	MB13
01D0	MB14
01E8	MB15
0200	MB16
0218	MB17
0230	MB18
0248	MB19
0260	MB20
0280	MB21
0298	MB22
02B0	MB23
02C8	MB24
02E0	MB25
02F8	MB26
0310	MB27
0328	MB28
0340	MB29
0358	MB30
0370	MB31
0388	MB32
03A0	MB33
03B8	MB34
03D0	MB35
03E8	MB36
0400	MB37
0418	MB38
0430	MB39
0448	MB40
0460	MB41

Table 66-33. 32-byte message buffers

Address offset (hex)	MBDSR=b10 32-byte payload
0080	MB0
00A8	MB1
00D0	MB2
00F8	MB3
0120	MB4
0148	MB5
0170	MB6
0198	MB7
01C0	MB8
01E8	MB9
0210	MB10
0238	MB11
0280	MB12
02A8	MB13
02D0	MB14
02F8	MB15
0320	MB16
0348	MB17
0370	MB18
0398	MB19
03C0	MB20
03E8	MB21
0410	MB22
0438	MB23

Table 66-34. 64-byte message buffers

Address offset (hex)	MBDSR=b11 64-byte payload
0080	MB0
00C8	MB1
0110	MB2
0158	MB3
01A0	MB4
01E8	MB5
0230	MB6
0280	MB7
02C8	MB8

Table continues on the next page...

Table 66-34. 64-byte message buffers (continued)

Address offset (hex)	MBDSR=b11 64-byte payload
0310	MB9
0358	MB10
03A0	MB11
03E8	MB12
0430	MB13

66.6.6 Rx FIFO structure

When MCR[RFEN] is set, the memory area 80h–DCh (which is normally occupied by MBs 0–5) is used by the reception FIFO engine.

The region 80h–8Ch contains the output of the FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. The region 90h–DCh is reserved for internal use of the FIFO engine.

An additional memory area, which starts at E0h and may extend up to 2DCh (normally occupied by MBs 6–37) depending on the CTRL2[RFFN] field setting, contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the FIFO.

Out of reset, the ID filter table flexible memory area defaults to E0h and extends only to FCh, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Rx FIFO data structure.

Table 66-35. Rx FIFO structure

	31 28 24 23	22	21	20	19 18 17 16	15 8	7 0
80h	IDHIT	SRR	IDE	RTR	DLC	TIME STAMP	
84h		ID standard			ID extended		
88h	Data byte 0	Data byte 1				Data byte 2	Data byte 3
8Ch	Data byte 4	Data byte 5				Data byte 6	Data byte 7
90h–DCh	Reserved						
E0h	ID filter table element 0						
E4h	ID filter table element 1						
E8h–2D4h	ID filter table elements 2 to 125						
2D8h	ID filter table element 126						

Table continues on the next page...

Table 66-35. Rx FIFO structure (continued)

2DCh	ID filter table element 127
	= Unimplemented or reserved

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the MCR[IDAM] field setting. The following figures show the IDAF indexation.

The following table shows the three different formats of the ID table elements. Note that all elements of the table must have the same format. See Rx FIFO for more information.

Table 66-36. ID table structure

Format	31	30	29 24	23 16	15	14	13 8	7 1	0
A	RTR	IDE	RXIDA (standard = 29–19, extended = 29–1)						
B	RTR	IDE	RXIDB_0 (standard = 29–19, extended = 29–16)		RTR	IDE	RXIDB_1 (standard = 13–3, extended = 13–0)		
C	RXIDC_0 (std/ext = 31–24)			RXIDC_1 (std/ext = 23–16)	RXIDC_2 (std/ext = 15–8)			RXIDC_3 (std/ext = 7–0)	
	= Unimplemented or Reserved								

RTR — Remote frame

This bit specifies if remote frames are accepted into the FIFO if they match the target ID.

1 = Remote frames can be accepted and data frames are rejected

0 = Remote frames are rejected and data frames can be accepted

IDE — Extended frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected

0 = Extended frames are rejected and standard frames can be accepted

RXIDA — Rx frame identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 — Rx frame identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 — Rx frame identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

IDHIT — Identifier acceptance filter hit indicator

This 9-bit field indicates which identifier acceptance filter was hit by the received message that is in the output of the Rx FIFO. See Rx FIFO for more information.

Chapter 67
Flexible I/O (FlexIO)

67.1 Chip-specific FlexIO information

Table 67-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	—	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Trigger mapping	-	

NOTE

For each channel, FlexIO DMA requests have been ORed together at the SoC level. Only one should be enabled at a time.

67.2 Overview

The FlexIO is a highly configurable module providing a wide range of functionality including:

- Emulation of a variety of serial/parallel communication protocols
- Flexible 16-bit timers with support for a variety of trigger, reset, enable and disable conditions
- Programmable logic blocks allowing the implementation of digital logic functions on-chip and configurable interaction of internal and external modules
- Programmable state machine for offloading basic system control functions from CPU

67.2.1 Block Diagram

The Figure 67-1 gives a high-level overview of the configuration of FlexIO timers and shifters.

The FlexIO uses shifters, timers and external triggers to shift data into or out of the FlexIO. The timing of the data shift is controlled by the timers, as shown in the block diagram. The timers can be configured to use generic timer functions, external triggers, or various other conditions to determine this control.

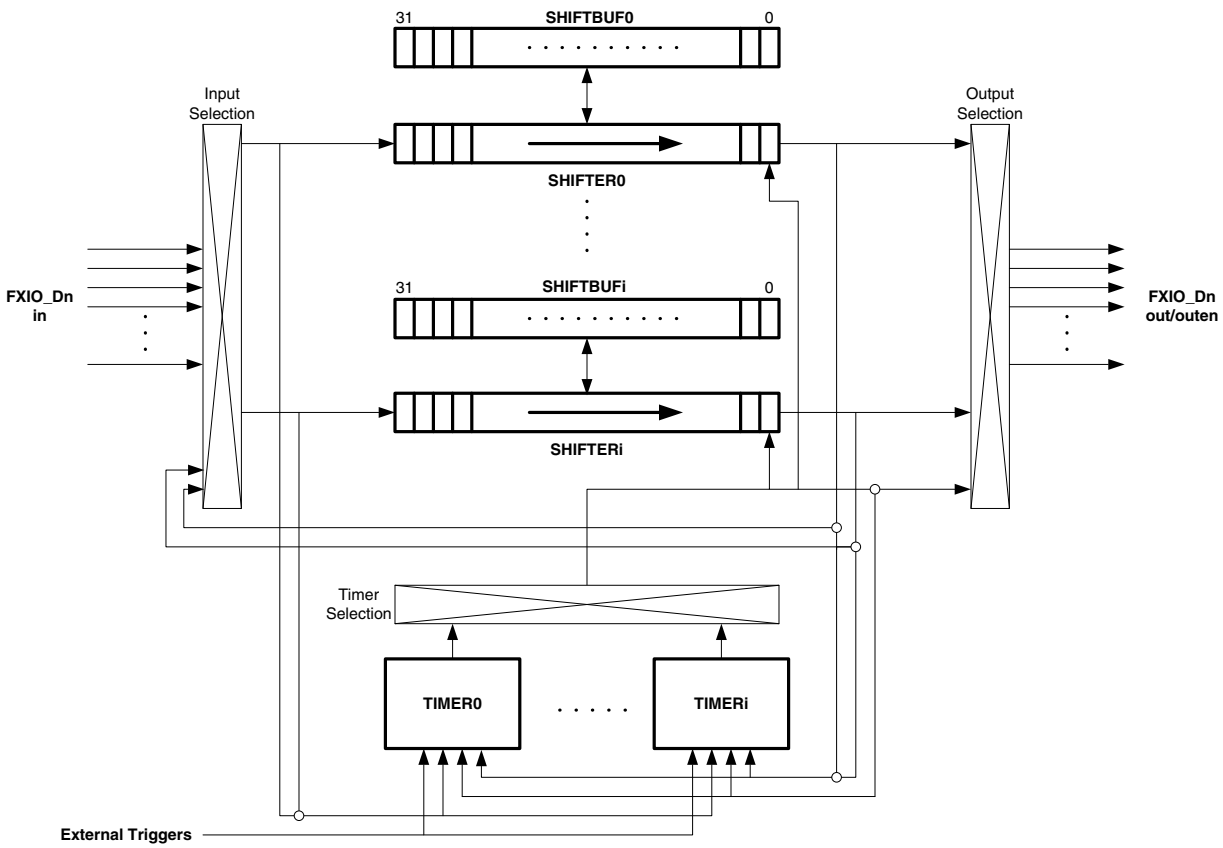


Figure 67-1. FlexIO block diagram

67.2.2 Features

The FlexIO module is capable of supporting a wide range of protocols including, but not limited to:

- UART
- I2C
- SPI

- I2S
- Camera IF
- Motorola 68K/Intel 8080 bus
- PWM/Waveform generation
- Input-capture (pulse edge interval measurement) - Example: SENT

The following key features are provided:

- Array of 32-bit shift registers with transmit, receive, data match, logic mode, and state modes
- Double buffered shifter operation for continuous data transfer
- Shifter concatenation to support large transfer sizes
- Automatic start/stop bit generation
- 1, 2, 4, 8, 16 or 32 multi-bit shift widths for parallel interface support
- Interrupt, DMA or polled transmit/receive operation
- Programmable baud rates independent of bus clock frequency, with support for asynchronous operation during stop modes
- Highly flexible 16-bit timers with support for a variety of internal or external trigger, reset, enable and disable conditions
- Programmable logic mode for integrating external digital logic functions on-chip or combining pin/shifter/timer functions to generate complex outputs
- Programmable state machine for offloading basic system control functions from CPU with support for up to 8 states, 8 outputs and 3 selectable inputs per state

67.3 Functional description

67.3.1 Shifter operation

Shifters are responsible for buffering and shifting data into or out of the FlexIO. The timing of shift, load and store events are controlled by the Timer assigned to the Shifter via the SHIFTCTL[TIMSEL] register. The Shifters are designed to support either DMA, interrupt or polled operation. The following block diagram provides a detailed view of the Shifter microarchitecture.

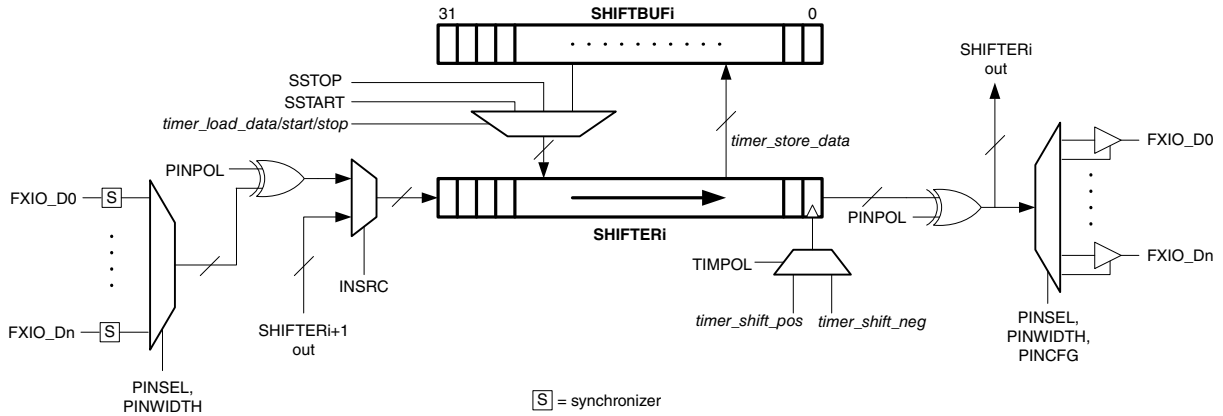


Figure 67-2. Shifter Microarchitecture

67.3.1.1 Transmit Mode

When configured for Transmit mode (SHIFTCTL[SMOD] =Transmit), the shifter loads data from the SHIFTBUF register and shift data out when a load event is signalled by the assigned Timer. An optional start/stop bit can also be automatically loaded before/ afterSHIFTBUF data by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer.

NOTE

The shifter immediately loads a stop bit when the Shifter is initially configured for Transmit mode if a stop bit is enabled.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data has been loaded from the SHIFTBUF register into the Shifter or when the Shifter is initially configured into Transmit mode. The flag clears when new data has been written into the SHIFTBUF register or a logic 1 is written to this flag. In transmit mode, any write of the SHIFTBUF register clears the corresponding Shifter Status Flag. It does not matter what is writing or the state of the DMA/interrupt enables, the flag is cleared. How the flag is set/cleared for each mode is documented in the SSF register description.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to load data from an empty SHIFTBUF register occurs (buffer underrun). The flag can be cleared by writing it with logic 1.

67.3.1.2 Receive Mode

When configured for Receive mode (SHIFTCTL[SMOD] =Receive), the shifter shifts data in and store data into the SHIFTBUF register when a store event is signalled by the assigned Timer. Checking for a start/stop bit can be enabled before/after shifter data is sampled by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when data has been stored into the SHIFTBUF register from the Shifter. The flag clears when the data has been read from theSHIFTBUF register or a logic 1 is written to this flag. Any read of the SHIFTBUF register clears the corresponding Shifter Status Flag when Shifter is configured in Receive mode. It does not matter what is reading or the state of the DMA/interrupt enables, the flag is cleared. How the flag is set/clear for each mode is documented in the SSF register description.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to store data into a fullSHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start/stop bit check. The flag can be cleared by writing it with logic 1.

67.3.1.3 Match Store Mode

When configured for Match Store mode (SHIFTCTL[SMOD] =Match Store), the shifter shifts data in, check for a match result and store matched data into the SHIFTBUF register when a store event is signalled by the assigned Timer. Checking for a start/stop bit can be enabled before/after shifter data is sampled by configuring the SHIFTCFG[SSTART], TIMCFG[TSTART] or SHIFTCFG[SSTOP], TIMCFG[TSTOP] registers in the Shifter and Timer. Up to 16-bits of data can be compared using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs and matched data has been stored into the SHIFTBUF register from the Shifter. The flag clears when the matched data has been read from the SHIFTBUF register or a logic 1 is written to this flag. Any read of the SHIFTBUF register clears the corresponding Shifter Status Flag when Shifter is configured in Match Store mode. It does not matter what is reading or the state of the DMA/interrupt enables, the flag is cleared. How the flag is set/clear for each mode is documented in the SSF register description.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts are set when an attempt to store matched data into a full SHIFTBUF register occurs (buffer overrun) or when a mismatch occurs on a start/stop bit check. The flag can be cleared by writing it with logic 1.

67.3.1.4 Match Continuous Mode

When configured for Match Continuous mode (SHIFTCTL[SMOD] =Match Continuous), the shifter shifts data in and continuously check for a match result whenever a shift event is signalled by the assigned Timer. Up to 16-bits of data can be compared using SHIFTBUF[31:16] to configure the data to be matched and SHIFTBUF[15:0] to mask the match result.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set when a match occurs. The flag clears automatically as soon as there is no longer a match between Shifter data and SHIFTBUF register. **The flag cannot be cleared by reading the SHIFTBUF register.**

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts are set when a match occurs. The flag clears when there is a read from the SHIFTBUF register or it written with logic 1.

67.3.1.5 State Mode

Using State mode enables the user to implement any state machine with up to 8 states, 8 outputs and 3 selectable inputs per state. This feature allows basic control functions to be offloaded from the CPU, which could potentially remain in a low power mode.

When configured for State mode (SHIFTCTL[SMOD] =State), the SHIFTBUF register is used to drive the output and compute next state values when the Shifter has been selected by the current state pointer (SHIFTSTATE[STATE]). The following diagram provides a detailed view of Shifter microarchitecture when configured for State mode.

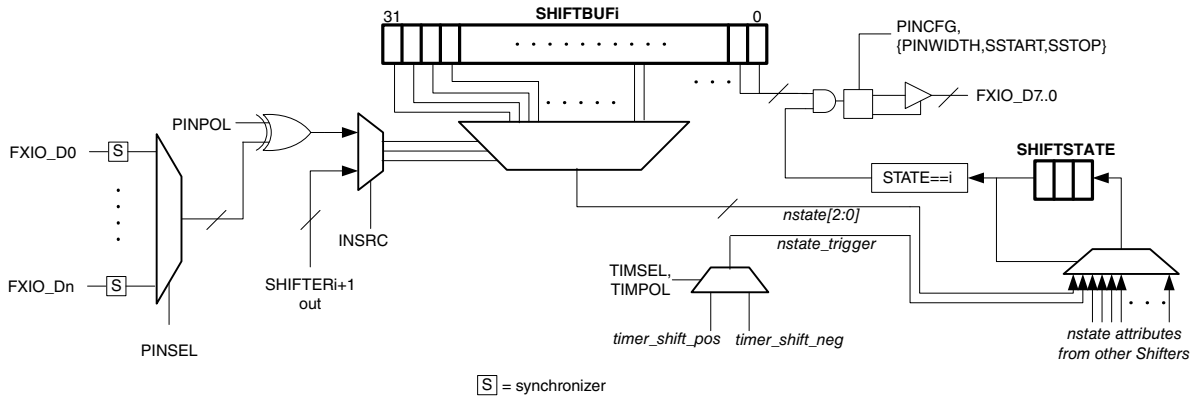


Figure 67-3. State Microarchitecture

When Shifter i has been selected by the current state pointer, output pins FXIO_D[7:0] are driven by SHIFTBUFi[31:24] using the configuration set by SHIFTCTLi[PINCFG]. When set, SHIFTCFGi {PWIDTH[3:0],SSTOP[1:0],SSTART[1:0]} are respectively used to disable the output drive on pins FXIO_D[7:0] for state machine applications which require less than 8 output pins.

The next state value is computed using the 3 input pins selected by SHIFTCTLi[PINSEL] together with SHIFTBUFi[23:0].

NOTE

Each state could potentially use a different set of 3 input pins.

The following table details how the next state value is computed when the current state pointer is pointing to Shifter i.

Table 67-2. Next State computation for SHIFTSTATE[STATE]=i

FXIO_D[PINSEL+2]	FXIO_D[PINSEL+1]	FXIO_D[PINSEL]	Next State Value
0	0	0	SHIFTBUFi[2:0]
0	0	1	SHIFTBUFi[5:3]
0	1	0	SHIFTBUFi[8:6]
0	1	1	SHIFTBUFi[11:9]
...	...	...	...
1	1	1	SHIFTBUFi[23:21]

Note that other shifters/timers could potentially be configured to drive the input pins of a given state, allowing the user to create complex combinations of shifters/timers as desired e.g. the output of a Shifter configured for logic mode could potentially be used to drive a state machine input.

The next state transition is triggered using the Timer output selected by SHIFTCTLi[TIMSEL] with polarity controlled by SHIFTCTLi[TIMPOL]. Note that each state could potentially use a different Timer to trigger each next state transition, allowing a variety of internal/external trigger sources and clocking configurations to be used (see Timer section for more detail).

The current state pointer defaults to Shifter 0 at reset, however it can be written by the user to select a different Shifter for the initial state. If the current state pointer selects a Shifter which is not configured for State mode, then outputs are not driven and a next state transition is never triggered.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set whenever the Shifter has been selected by the current state pointer. The flag clears when the current state pointer is updated to a different Shifter.

67.3.1.6 Logic Mode

Using logic mode enables the user to implement a small amount of programmable digital logic within a FlexIO Shifter.

When configured for Logic mode (SHIFTCTL[SMOD] =Logic), theSHIFTBUF register is used to implement a 5-input, 32-bit programmable logic look-up table. The following diagram provides a detailed view of Shifter microarchitecture when configured for Logic mode.

The SHIFTBUFi configures the lookup table for the 4 pin inputs and the SHIFTERi can optionally be used to configure a feedback or delayed pin source as the 5th input to the look up table.

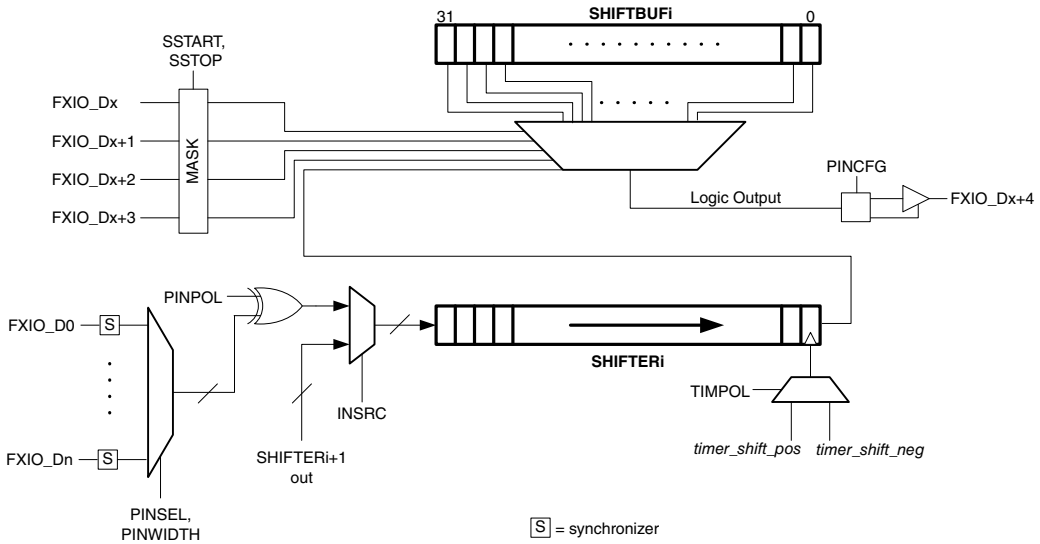


Figure 67-4. Logic Microarchitecture

The look-up table is driven using 4 pin inputs (maskable using SHIFTCFG[SSTOP] and SHIFTCFG[SSTART]) plus 1 input from the internal shifter and can be configured to drive an output pin using the SHIFTCTL[PINCFG] field. Pin inputs and outputs are fixed for each logic look-up table and are not selectable. The following table lists the logic output value selected by the look-up table for Shifter 'i'.

Table 67-3. Logic Look-up table for Shifter 'i'

SHIFTERi[0]	FXIO_D[x+3][1]	FXIO_D[x+2]	FXIO_D[x+1]	FXIO_D[x]	Logic Output to FXIO_D[x+4]
0	0	0	0	0	SHIFTBUFi[0]
0	0	0	0	1	SHIFTBUFi[1]
0	0	0	1	0	SHIFTBUFi[2]
0	0	0	1	1	SHIFTBUFi[3]
...	...	...	...	...	...
1	1	1	1	1	SHIFTBUFi[31]

1. for Shifter i=0...3, x=i

   for Shifter i=4...7, x=i+4

To minimize output glitches, SHIFTCFG[SSTOP] and SHIFTCFG[SSTART] can be used to mask unused input pins. When set, {SSTOP[1:0], SSTART[1:0]} mask FXIO_D[x+3]...FXIO_D[x] inputs respectively, so that any transitions on these pins does not cause the logic output to glitch.

Note that other shifters/timers could potentially be configured to drive the input pins of a given look-up table, allowing the user to concatenate look-up tables or create complex combinations of shifters/timers as desired.

SHIFTCFG[PWIDTH] controls the number of delay stages introduced by the internal shifter input (SHIFTERi[0]). For example, when configured for 1-bit shift (PWIDTH=0), the internal shifter introduces a 32 Shift clock delay before passing its input (selected by SHIFTCTL[PINSEL]) to the look-up table. When configured for 32-bit shift (PWIDTH=16...31), the internal shifter introduces a 1 Shift clock delay to its input.

The Shifter Status Flag (SHIFTSTAT[SSF]) and any enabled interrupts or DMA requests are set whenever the output pin allocated to the logic look-up table has a value of 1 (after being synchronized to the FlexIO clock). The flag clears when the output pin has a value of 0. This also allows the SSF flag to be used as a trigger to a Timer if desired.

The Shifter Error Flag (SHIFTERR[SEF]) and any enabled interrupts are set when the output pin allocated to the logic look-up table has asserted. The flag can be cleared by writing it with logic 1.

The logic mode input pins (including pins driven by other shifters/timers) are first synchronized to the FlexIO functional clock before they are input to the programmable logic look-up table.

67.3.2 Timer Operation

The FlexIO 16-bit timers control the loading, shifting and storing of the shift registers, the counters load the contents of the compare register and decrement down to zero on the FlexIO clock. They can perform generic timer functions such as generating a clock or select output or a PWM waveform. Timers can be configured to enable in response to a trigger, pin or shifter condition; decrement always or only on a trigger or pin edge; reset in response to a trigger or pin condition; and disable on a trigger or pin condition or on a timer compare. Timers can optionally include a start condition and/or stop condition.

Each timer operates independently, although a timer can be configured to enable or disable at the same time as the previous timer (eg: timer1 can enable or disable at the same time as timer 0) and a timer output can be used to trigger any other timer. The trigger used by each timer is configured independently and can be configured to be a timer output, shifter status flag, pin input or an external trigger input (refer to the chip-specific FlexIO information for details on the external trigger connections). The trigger configuration is separate from the pin configuration, which can be configured for input, output data or output enable.

The Timer Configuration Register (TIMCFGn) should be configured before setting the Timer Mode (TIMOD).

67.3.2.1 Timer 8-bit Baud Counter Mode

In 8-bit Baud Counter Mode, the 16-bit counter is divided into two 8-bit counters. The lower 8-bits are used to configure the baud rate of the shift clock and the upper 8-bits are used to configure the number of shift clock edges in the transfer. When the lower 8-bits decrement to zero, the timer output is toggled and the lower 8-bits reload from the compare register. The upper 8-bits decrement when the lower 8-bits equal zero and decrement.

Note that a timer reset event in 8-bit Baud Counter Mode only resets the lower 8-bit counter, the upper 8-bit counter is not affected and can decrement if the timer reset is configured to update the state of the timer output, and the timer output toggles as a result of the timer reset event.

A timer compare event occurs when the upper 8-bits equal zero and decrements. The timer status flag is set on a timer compare event.

67.3.2.2 Timer 8-bit High PWM Mode

In 8-bit High PWM Mode, the 16-bit counter is divided into two 8-bit counters. The lower 8-bits are used to configure the timer output high period and the upper 8-bits are used to configure the timer output low period. The lower 8-bits decrement when the output is high. When the lower 8-bits equal zero and decrement, the timer output is cleared and the lower 8-bits are reloaded from the compare register. The upper 8-bits decrement when the output is low. When the upper 8-bits equal zero and decrement, the timer output is set and the upper 8-bits are reloaded from the compare register.

A timer compare event occurs when the upper 8-bits equal zero and decrements. The timer status flag is set on a timer compare event.

67.3.2.3 Timer 16-bit Counter Mode

In 16-bit Counter Mode, the 16-bit counter can be used to configure either the baud rate of the shift clock (eg: TIMDEC[1:0] != 10 or 11) or the number of shift clock edges in the transfer (eg: TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer compare event.

67.3.2.4 Timer 16-bit Counter Disable Mode

In 16-bit Counter Disable Mode, the 16-bit counter can be used to configure either the baud rate of the shift clock (eg: TIMDEC[1:0] != 10 or 11) or the number of shift clock edges in the transfer (eg: TIMDEC[1:0] = 10 or 11). When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads from the compare register.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer disable event.

67.3.2.5 Timer 8-bit Word Counter Mode

In 8-bit Word Counter Mode, the 16-bit counter is divided into two 8-bit counters. The lower 8-bits are used to configure the number of shift clock edges in each word and the upper 8-bits are used to configure the number number of words in the transfer. When the lower 8-bits decrement to zero, the timer output is toggled and the lower 8-bits reload from the compare register. The upper 8-bits only decrement when the lower 8-bits equal zero and decrement.

A timer compare event occurs when the lower 8-bits equal zero and decrements. The timer status flag is set when the upper 8-bits equal zero and decrements.

67.3.2.6 Timer 8-bit Low PWM Mode

In 8-bit Low PWM Mode, the 16-bit counter is divided into two 8-bit counters. The lower 8-bits are used to configure the timer output low period and the upper 8-bits are used to configure the timer output high period. The lower 8-bits decrement when the output is low. When the lower 8-bits equal zero and decrement, the timer output is set and the lower 8-bits are reloaded from the compare register. The upper 8-bits decrement when the output is high. When the upper 8-bits equal zero and decrement, the timer output is cleared and the upper 8-bits are reloaded from the compare register.

A timer compare event occurs when the upper 8-bits equal zero and decrements. The timer status flag is set on a timer compare event.

67.3.2.7 Timer 16-bit Input Capture Mode

In 16-bit Input Capture Mode, the 16-bit counter uses a fixed initial value of 0xFFFF to load the counter. Whenever a timer counter disable condition is detected (as configured by TIMDIS), the timer status flag is set and the inverted timer counter value is stored in the timer compare register, then the timer counter is immediately restarted from 0xFFFF. The timer output is always output on the FlexIO trigger outputs (which can also be used by other timers as an input trigger) and can also be output to a pin (PINCFG/PINPOL/PINSEL) or a shifter.

If the timer status flag is already set, then the input capture event is missed. When the 16-bit counter equals zero and decrements, the timer output toggles and the counter reloads with 0xFFFF. The timer output can be used to count the number of counter overflows between two input capture events. Input capture mode is not intended to be used to control shift registers, it is to enable software to count the number of bits in a variable length transfer or to monitor the period or frequency of an input.

A timer compare event occurs when the 16-bit counter equals zero and decrements. The timer status flag is set on a timer disable event. Whenever the status flag asserts, the inverted timer counter value is stored in the timer compare register.

67.3.2.8 Timer Enable and Start Bit

When the TIMOD is configured for the desired mode, and the condition configured by timer enable (TIMENA) is detected then the following events occur. When ONETIM is set, then the timer status flag must also be clear to generate a timer enable event, otherwise the timer enable event is blocked. This can be used to enforce software intervention after each timer iteration.

- Timer counter loads the current value of the compare Register and start decrementing as configured by TIMDEC.
- Timer output may update to its initial state depending on the TIMOUT configuration. Shifters that are controlled by this timer do not see this as a rising edge on the timer shift clock.
- Transmit shifters controlled by this timer either output their start bit value, or load the shift register from the shift buffer and output the first bit, as configured by SSTART.

If the Timer start bit is enabled, the timer counter reloads with the compare register on the first rising edge of the shift clock after the timer starts decrementing. If there is no falling edge on the shift clock before the first rising edge (for example, when TIMOUT =1), a shifter that is configured to shift on falling edge and load on the first shift does not load correctly.

67.3.2.9 Timer Decrement and Reset

The Timer then generates the timer output and timer shift clock depending on the TIMOD and TIMDEC fields. The shifter clock is either equal to the timer output (when TIMDEC != 10 or 11) or equal to the decrement clock (when TIMDEC = 10 or 11). When TIMDEC is configured to decrement from a pin or trigger, the timer decrements on both rising and falling edges.

If a Timer is configured to decrement on the FlexIO functional clock divided by 16 or 256 (when TIMDEC = 100 or 101), then a common prescaler that is shared by all Timers is used to generate the two divide ratios. This prescaler is reset when all Timers are either idle or configured to not use the prescaler (TIMDEC != 100 or 101).

When the Timer is configured to reset as configured in the TIMRST field then the Timer counter loads the current value of the compare register again. The timer output and timer shift clock can be configured to update on timer reset, as configured by TIMOUT. This can result in a timer shift clock edge if the timer output toggles as a result of the timer reset. In 8-bit Baud Counter mode this would also decrement the upper 8-bits of the counter.

In general, when the timer counter decrements to zero a timer compare event is triggered. The timer compare event causes the timer counter to load the contents of the timer compare register, the timer output to toggle, any configured transmit shift registers to load, and any configured receive shift registers to store. Depending on the timer mode, the timer status flag may also be set.

67.3.2.10 Timer Disable and Stop Bit

When the is Timer is configured to add a stop bit on each compare, the following additional events occur.

- Transmit shifters controlled by this timer output their stop bit value (if configured by SSTOP).

- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SSTOP.
- On the first rising edge of the shifter clock after the compare, the timer counter reloads the current value of the Compare Register.

Transmit shifters must be configured to load on the first shift when the timer is configured to insert a stop bit on each compare.

When the condition configured by timer disable (TIMDIS) is detected, the following events occur.

- Timer counter reloads the current value of the Compare Register and start decrementing as configured by TIMDEC.
- Timer output clears. Shifters that are controlled by this timer do not see this as a falling edge on the timer shift clock, but can generate a shift event if the timer shift clock would otherwise generate one.
- Transmit shifters controlled by this timer output their stop bit value (if configured by SSTOP).
- Receive shifters controlled by this timer store the contents of the shift register in their shift buffer, as configured by SSTOP.

If the timer stop bit is enabled, the timer counter continues decrementing until the next rising edge of the shift clock is detected, at which point it finishes. Although the timer output is forced low during the stop bit, the timer shift clock can toggle during the stop bit, but does not generate shift events.

A timer enable condition can be detected in the same cycle as a timer disable condition (if timer stop bit is disabled), or on the first rising edge of the shift clock after the disable condition (if stop bit is enabled). When ONETIM is set, the timer status flag must be clear before the next timer enable condition is detected. Receive shift registers with stop bit enabled store the contents of the shift register into the shift buffer and verify the state of the input data on the configured shift edge while the timer is in the stop state condition. If there is no configured edge between the timer disable and the next rising edge of the shift clock then the final store and verify do not occur.

67.3.3 Pin operation

The pin configuration for each timer and shifter can be configured to use any FlexIO pin with either polarity. Each timer and shifter can be configured as an input, output data, output enable or bidirectional output. A pin configured for output enable can be used as an open drain (with inverted polarity, since the output enable assertion would cause logic

zero to be output on the pin) or to control the enable on the bidirectional output. Any timer or shifter could be configured to control the output enable for a pin where the bidirectional output data is driven by another timer or shifter.

67.3.3.1 Parallel Interface

Shifters can be configured to use multiple FlexIO pins in parallel using the SHIFTCFG[PWIDTH] field. PWIDTH is used to configure the following settings of a shifter:

1. Number of bits shifted per Shift clock.
2. Number of pins driven by the shifter per Shift clock (only on shifters supporting parallel transmit i.e. SHIFTER0, SHIFTER4).
3. Number of pins sampled by the shifter per Shift clock (only on Shifter supporting parallel receive i.e. SHIFTER3, SHIFTER7).

When configured for parallel shift, either 4, 8, 16 or 32-bits can be shifted on every Shift clock. If an adjacent shifter is selected as the input source (SHIFTCFG[INSRC] =1), the least significant 4, 8, 16 or 32-bits from the adjacent shifter is sampled on each Shift clock.

For shifters supporting parallel receive (SHIFTER3, SHIFTER7), the shifter can be configured to sample multiple pins (SHIFTCFG[INSRC] =0), with PWIDTH and PINSEL selecting the pins as follows: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL]. Note that if PWIDTH is less than the number of bits being shifted on each Shift clock, then the most significant bits are masked with 0 e.g. if PINSEL=7 and PWIDTH=6, then SHIFTER[31:24] samples {0,0,FXIO_D[12:7]} on each Shift clock.

For shifters supporting parallel transmit (SHIFTER0, SHIFTER4), the shifter can be configured to drive multiple pins using SHIFTCTL[PINCFG], with PWIDTH and PINSEL selecting the pins as follows: FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL]. Note that if PWIDTH is less than the number of bits being shifted on each Shift clock, then the most significant pins are not driven e.g. if PINSEL=7 and PWIDTH=6, then SHIFTER[5:0] drives only FXIO_D[12:7] on each Shift clock.

67.3.3.2 Pin Synchronization

When configuring a pin as an input (this includes a timer trigger configured as a pin input), the input signal is first synchronized to the FlexIO clock before the signal is used by a timer or shifter. This introduces a small latency of between 0.5 to 1.5 FlexIO clock cycles when using an external pin input to generate an output or control a shifter. This sets the maximum setup time at 1.5 FlexIO clock cycles.

If an input is used by more than one timer or shifter then the synchronization occurs once to ensure any edge is seen on the same cycle by all timers and shifters using that input.

Note that FlexIO pins are also connected internally, configuring a FlexIO shifter or timer to output data on an unused pin makes an internal connection that allows other shifters and timer to use this pin as an input. This allows a shifter output to be used to trigger a timer or a timer output to be shifted into a shifter. This path is also synchronized to the FlexIO clock and therefore incurs a 1 cycle latency.

So when using a Pin input as a Timer Trigger, Timer Clock or Shifter Data Input, the following synchronization delays occur:

1. 0.5 to 1.5 FlexIO clock cycles for external pin
2. 1 FlexIO clock cycle for an internally driven pin

For timing considerations such as output valid time and input setup time for specific applications (SPI Master, SPI Slave, I2C Master, I2S Master, I2S Slave) please refer to the FlexIO Application Information Section.

67.3.4 Low Power Modes

The FlexIO remains functional during low power modes, provided the Doze Enable bit (CTRL[DOZEN]) is clear and the FlexIO functional clock remains enabled.

67.3.5 Debug Mode

The FlexIO remains functional in debug mode provided the Debug Enable bit (CTRL[DBGE]) is set.

67.3.6 Clocks

67.3.6.1 Functional clock

The FlexIO functional clock is asynchronous to the bus clock and can remain enabled in low power modes. The FlexIO functional clock must be enabled before accessing any FlexIO registers. Provided the FlexIO functional clock is at least two times faster than the bus clock, the CTRL[FASTACC] bit can be set to support fast register accesses.

67.3.6.2 Bus clock

The bus clock is only used for bus accesses to the control and configuration registers.

67.3.7 Reset

67.3.7.1 Chip reset

The logic and registers for the FlexIO are reset to their default state on a chip reset.

67.3.7.2 Software reset

The FlexIO implements a software reset bit in its Control Register. The CTRL[SWRST] resets all logic and registers to their default state, except for the CTRL itself.

67.3.8 Interrupts and DMA Requests

The following table illustrates the status flags that can generate the FlexIO interrupt and DMA requests.

Table 67-4. FlexIO Interrupts and DMA Requests

Flag	Description	Interrupt	DMA Request	Low Power Wakeup
SSF	Shifter Status Flag.	Y	Y	Y
SEF	Shifter Error Flag.	Y	N	Y
TSF	Timer Status Flag.	Y	Y	Y

67.3.9 Peripheral Triggers

The connection of the FlexIO peripheral triggers with other peripherals are device specific.

67.3.9.1 Output Triggers

Each FlexIO Timer generates an output trigger equal to the timer output. The output trigger is not affected by the timer pin polarity configuration.

67.3.9.2 Input Trigger

FlexIO supports multiple external trigger inputs that can be used to trigger one or more FlexIO timers. The external triggers are synchronized to the FlexIO functional clock and must assert for at least two cycles of the FlexIO functional clock to be sampled correctly.

67.4 Application Information

This section provides examples for a variety of FlexIO module applications.

67.4.1 UART Transmit

UART transmit can be supported using one Timer, one Shifter and one Pin (two Pins if supporting CTS). The start and stop bit insertion is handled automatically and multiple transfers can be supported using DMA controller. The timer status flag can be used to indicate when the stop bit of each word is transmitted.

Break and idle characters require software intervention, before transmitting a break or idle character the SSTART and SSTOP fields should be altered to transmit the required state and the data to transmit must equal 0xFF or 0x00. Supporting a second stop bit requires the stop bit to be inserted into the data stream using software (and increasing the number of bits to transmit). Note that when performing byte writes to SHIFTBUFn (or SHIFTBUFBIS for transmitting MSB first), the rest of the register remains unaltered allowing an address mark bit or additional stop bit to remain undisturbed.

FlexIO does not support automatic insertion of parity bits.

Table 67-5. UART Transmit Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with output data on Pin 0. Can invert output data by setting PINPOL, or can support open drain by setting PINPOL=0x1 and PINCFG =0x1.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.

Table continues on the next page...

Table 67-5. UART Transmit Configuration (continued)

Register	Value	Comments
TIMCFGn	0x0000_2222	Configure start bit, stop bit, enable on trigger asserted and disable on compare. Can support CTS by configuring TIMEN=0x3.
TIMCTLn	0x01C0_0001	Configure dual 8-bit counter using Shifter 0 status flag as inverted internal trigger source. Can support CTS by configuring PINSEL=0x1 (for Pin 1) and PINPOL=0x1.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS[7:0] register instead.

An alternative configuration is possible that supports slower baud rates but requires two Timers.

Table 67-6. UART Transmit Configuration for Slow Baud Rate

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with output data on Pin 0. Can invert output data by setting PINPOL, or can support open drain by setting PINPOL=0x1 and PINCFG =0x1.
TIMCMPn	0x0000_000F	Configure for 8-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFGn	0x0030_2622	Configure start bit, stop bit, enable on trigger rising edge, decrement on trigger and disable on compare.
TIMCTLn	0x0740_0003	Configure 16-bit counter using Timer 1 output as internal trigger source.
TIMCMP(n+1)	0x0000_0001	Configure baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:0] = (baud rate divider / 2) - 1.
TIMCFG(n+1)	0x0000_1200	Configure enable on trigger asserted and disable on Timer 0 disable. Can support CTS by configuring TIMEN=0x3.
TIMCTL(n+1)	0x01C0_0003	Configure 16-bit counter using Shifter 0 status flag as inverted internal trigger source. Can support CTS by configuring PINSEL =0x1 (for Pin 1) and PINPOL =0x1.

Table continues on the next page...

Table 67-6. UART Transmit Configuration for Slow Baud Rate (continued)

Register	Value	Comments
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF[7:0] to initiate an 8-bit transfer, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS[7:0] register instead.

67.4.2 UART Receive

UART receive can be supported using one Timer, one Shifter and one Pin (two Timers and two Pins if supporting RTS). The start and stop bit verification is handled automatically and multiple transfers can be supported using the DMA controller. The timer status flag can be used to indicate when the stop bit of each word is received.

Triple voting of the received data is not supported by FlexIO, data is sampled only once in the middle of each bit. Another timer can be used to implement a glitch filter on the incoming data, another Timer can also be used to detect an idle line of programmable length. Break characters cause the error flag to set and the shifter buffer register returns 0x00.

FlexIO does not support automatic verification of parity bits.

Table 67-7. UART Receiver Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0080_0001	Configure receive using Timer 0 on negedge of clock with input data on Pin 0. Can invert input data by setting PINPOL.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0204_2422	Configure start bit, stop bit, enable on pin posedge and disable on compare. Enable resynchronization to received data with TIMOUT=0x2 and TIMRST=0x4.
TIMCTLn	0x0000_0081	Configure dual 8-bit counter using inverted Pin 0 input.

Table continues on the next page...

Table 67-7. UART Receiver Configuration (continued)

Register	Value	Comments
SHIFTBUFn	Data to receive	Received data can be read from SHIFTBUFBYS[7:0], use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS[7:0] register instead.

The UART Receiver with RTS configuration uses a 2nd Timer to generate the RTS output. The RTS asserts when the start bit is detected and negate when the data is read from the shifter buffer register. No start bit is detected while the RTS is asserted, the received data is simply ignored.

Table 67-8. UART Receiver with RTS Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Configure start bit of 0 and stop bit of 1.
SHIFTCTLn	0x0080_0001	Configure receive using Timer 0 on negedge of clock with input data on Pin 0. Can invert input data by setting PINPOL.
TIMCMPn	0x0000_0F01	Configure 8-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0204_2522	Configure start bit, stop bit, enable on pin posedge with trigger asserted and disable on compare. Enable resynchronization to received data with TIMOUT=0x2 and TIMRST=0x4.
TIMCTLn	0x02C0_0081	Configure dual 8-bit counter using inverted Pin 0 input. Trigger is internal using inverted Pin 1 input.
TIMCMP(n+1)	0x0000_FFFF	Never compare.
TIMCFG(n+1)	0x0030_6100	Enable on Timer N enable and disable on trigger falling edge. Decrement on trigger to ensure no compare.
TIMCTL(n+1)	0x0143_0003	Configure 16-bit counter and output on Pin 1. Trigger is internal using Shifter 0 flag.
SHIFTBUFn	Data to receive	Received data can be read from SHIFTBUFBYS[7:0], use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS[7:0] register instead.

67.4.3 SPI Master

SPI master mode can be supported using two Timers, two Shifters and four Pins. Either CPHA=0 or CPHA=1 can be supported and transfers can be supported using the DMA controller. For CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The stop bit is used to guarantee a minimum of 1 clock cycle between the slave select negating and before the next transfer. Writing to the transmit buffer by either core or DMA is used to initiate each transfer.

NOTE

Due to synchronization delays, the setup time for the serial input data is 1.5 FlexIO clock cycles, so the maximum baud rate is divide by 4 of the FlexIO clock frequency.

Table 67-9. SPI Master (CPHA=0) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0083_0002	Configure transmit using Timer 0 on negedge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0000_0101	Configure receive using Timer 0 on posedge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0100_2222	Configure start bit, stop bit, enable on trigger high and disable on compare, initial clock state is logic 0.
TIMCTLn	0x01C3_0201	Configure dual 8-bit counter using Pin 2 output (shift clock), with Shifter 0 flag as the inverted trigger. Set PINPOL to invert the output shift clock.
TIMCMP(n+1)	0x0000_FFFF	Never compare.
TIMCFG(n+1)	0x0000_1100	Enable when Timer 0 is enabled and disable when Timer 0 is disabled.
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter (never compare) using inverted Pin 3 output (as slave select).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag

Table continues on the next page...

Table 67-9. SPI Master (CPHA=0) Configuration (continued)

Register	Value	Comments
		to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

Table 67-10. SPI Master (CPHA=1) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0021	Start bit loads data on first shift.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on negedge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0100_2222	Configure start bit, stop bit, enable on trigger high and disable on compare, initial clock state is logic 0.
TIMCTLn	0x01C3_0201	Configure dual 8-bit counter using Pin 2 output (shift clock), with Shifter 0 flag as the inverted trigger. Set PINPOL to invert the output shift clock. Set TIMDIS=3 to keep slave select asserted for as long as there is data in the transmit buffer.
TIMCMP(n+1)	0x0000_FFFF	Never compare.
TIMCFG(n+1)	0x0000_1100	Enable when Timer 0 is enabled and disable when Timer 0 is disabled.
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter (never compare) using inverted Pin 3 output (as slave select).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.

Table continues on the next page...

Table 67-10. SPI Master (CPHA=1) Configuration (continued)

Register	Value	Comments
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

67.4.4 SPI Slave

SPI slave mode can be supported using one Timer, two Shifters and four Pins. Either CPHA=0 or CPHA=1 can be supported and transfers can be supported using the DMA controller. For CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer.

The transmit data must be written to the transmit buffer register before the external slave select asserts, otherwise the shifter error flag is set.

NOTE

Due to synchronization delays, the output valid time for the serial output data is 2.5 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

Table 67-11. SPI Slave (CPHA=0) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0083_0002	Configure transmit using Timer 0 on falling edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0000_0101	Configure receive using Timer 0 on rising edge of shift clock with input data on Pin 1.
TIMCMPn	0x0000_003F	Configure 32-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFGn	0x0120_0600	Configure enable on trigger rising edge, initial clock state is logic 0 and decrement on pin input.
TIMCTLn	0x06C0_0203	Configure 16-bit counter using Pin 2 input (shift clock), with Pin 3 input (slave select) as the inverted trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag

Table continues on the next page...

Table 67-11. SPI Slave (CPHA=0) Configuration (continued)

Register	Value	Comments
		to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

Table 67-12. SPI Slave (CPHA=1) Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0001	Shifter configured to load on first shift and stop bit disabled.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on rising edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on falling edge of shift clock with input data on Pin 1.
TIMCMPn	0x0000_003F	Configure 32-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFGn	0x0120_6602	Configure start bit, enable on trigger rising edge, disable on trigger falling edge, initial clock state is logic 0 and decrement on pin input.
TIMCTLn	0x06C0_0203	Configure 16-bit counter using Pin 2 input (shift clock), with Pin 3 input (slave select) as the inverted trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUF, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support MSB first transfer by writing to SHIFTBUFBBS register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBYS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support MSB first transfer by reading from SHIFTBUFBIS register instead.

67.4.5 I2C Master

I2C master mode can be supported using two Timers, two Shifters and two Pins. One timer is used to generate the SCL output and one timer is used to control the shifters. The two shifters are used to transmit and receive for every word, when receiving the transmitter must transmit 0xFF to tristate the output. FlexIO inserts a stop bit after every word to generate/verify the ACK/NACK. FlexIO waits for the first write to the transmit data buffer before enabling SCL generation. Data transfers can be supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The first timer generates the bit clock for the entire packet (START to Repeated START/ STOP), so the compare register needs to be programmed with the total number of clock edges in the packet (minus one). The timer supports clock stretching using the reset counter when pin equal to output (although this increases both the clock high and clock low periods by at least 1 FlexIO clock cycle each). The second timer uses the SCL input pin to control the transmit/receive shift registers, this enforces an SDA data hold time by an extra 2 FlexIO clock cycles.

Both the transmit and receive shifters need to be serviced for each word in the transfer, the transmit shifter must transmit 0xFF when receiving and the receive shifter returns the data actually present on the SDA pin. The transmit shifter loads 1 additional word on the last falling edge of SCL pin, this word should be 0x00 if generating a STOP condition or 0xFF if generating a repeated START condition. During the last word of a master-receiver transfer, the transmit SSTOP bit should be set by software to generate a NACK.

The receive shift register asserts an error interrupt if a NACK is detected, but software is responsible for generating the STOP or repeated START condition. If a NACK is detected during master-transmit, the interrupt routine should immediately write the transmit shifter register with 0x00 (if generating STOP) or 0xFF (if generating repeated START). Software should then wait for the next rising edge on SCL and then disable both timers. The transmit shifter should then be disabled after waiting the setup delay for a repeated START or STOP condition.

NOTE

Due to synchronization delays, the data valid time for the transmit output is 2 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

The I2C master data valid is delayed 2 cycles because the clock output is passed through a synchronizer before clocking the transmit/receive shifter (to guarantee some SDA hold time). Since the SCL output is synchronous with FlexIO clock, the synchronization delay is 1 cycle and then 1 cycle to generate the output.

Table 67-13. I2C Master Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0032	Start bit enabled (logic 0) and stop bit enabled (logic 1).
SHIFTCTLn	0x0101_0082	Configure transmit using Timer 1 on rising edge of clock with inverted output enable (open drain output) on Pin 0.
SHIFTCFG(n+1)	0x0000_0020	Start bit disabled and stop bit enabled (logic 0) for ACK/NACK detection.
SHIFTCTL(n+1)	0x0180_0001	Configure receive using Timer 1 on falling edge of clock with input data on Pin 0.
TIMCMPn	0x0000_2501	Configure 2 word transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of words x 18) + 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0102_2222	Configure start bit, stop bit, enable on trigger high, disable on compare, reset if output equals pin. Initial clock state is logic 0 and is not affected by reset.
TIMCTLn	0x01C1_0101	Configure dual 8-bit counter using Pin 1 output enable (SCL open drain), with Shifter 0 flag as the inverted trigger.
TIMCMP(n+1)	0x0000_000F	Configure 8-bit transfer. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFG(n+1)	0x0020_1112	Enable when Timer 0 is enabled, disable when Timer 0 is disabled, enable start bit and stop bit at end of each word, decrement on pin input.
TIMCTL(n+1)	0x01C0_0183	Configure 16-bit counter using inverted Pin 1 input (SCL).
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBBS[7:0], use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS[7:0], use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request.

67.4.6 I2S Master

I2S master mode can be supported using two Timers, two Shifters and four Pins. One timer is used to generate the bit clock and control the shifters and one timer is used to generate the frame sync. FlexIO waits for the first write to the transmit data buffer before enabling bit clock and frame sync generation. Data transfers can be supported using the DMA controller and the shifter error flag sets on transmit underrun or receive overflow.

The bit clock frequency is an even integer divide of the FlexIO clock frequency, and the initial frame sync assertion occurs at the same time as the first bit clock edge. The timer uses the start bit to ensure the frame sync is generated one clock cycle before the first output data.

NOTE

Due to synchronization delays, the setup time for the receiver input is 1.5 FlexIO clock cycles, so the maximum baud rate is divide by 4 of the FlexIO clock frequency.

Table 67-14. I2S Master Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0001	Load transmit data on first shift and stop bit disabled.
SHIFTCTLn	0x0003_0002	Configure transmit using Timer 0 on rising edge of clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0080_0101	Configure receive using Timer 0 on falling edge of clock with input data on Pin 1.
TIMCMPn	0x0000_3F01	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of bits x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFGn	0x0000_0202	Configure start bit, enable on trigger high and never disable. Initial clock state is logic 1.
TIMCTLn	0x01C3_0281	Configure dual 8-bit counter using inverted Pin 2 output (bit clock), with Shifter 0 flag as the inverted trigger. Clear PINPOL to invert the polarity of the output shift clock.
TIMCMP(n+1)	0x0000_007F	Configure 32-bit transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:0] = (number of bits x baud rate divider) - 1.
TIMCFG(n+1)	0x0000_0100	Enable when Timer 0 is enabled and never disable.
TIMCTL(n+1)	0x0003_0383	Configure 16-bit counter using inverted Pin 3 output (as frame sync). Clear PINPOL to invert the polarity of the output frame sync
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support LSB first transfer by writing to SHIFTBUF register instead.

Table continues on the next page...

Table 67-14. I2S Master Configuration (continued)

Register	Value	Comments
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support LSB first transfer by reading from SHIFTBUF register instead.

67.4.7 I2S Slave

I2S slave mode can be supported using three Timers, two Shifters and four Pins (for single transmit and single receive, other combinations of transmit and receive are possible).

The transmit data must be written to the transmit buffer register before the external frame sync asserts, otherwise the shifter error flag is set.

NOTE

Due to synchronization delays, the output valid time for the serial output data is 2.5 FlexIO clock cycles, so the maximum baud rate is divide by 6 of the FlexIO clock frequency.

The output valid time of I2S slave is max 2.5 cycles because there is a maximum 1.5 cycle delay on the clock synchronization plus 1 cycle to output the data

Timer 2 detects the falling edge of frame sync (start of new frame) and asserts output until rising edge of bit clock (when frame sync is normally sampled). Timer 0 detects rising edge of bit clock with Timer 2 output asserted and asserts output for length of frame. Timer 1 detects falling edge of bit clock with Timer 0 output asserted and controls shift registers for 32-bit transfers.

Table 67-15. I2S Slave Configuration

Register	Value	Comments
SHIFTCFGn	0x0000_0000	Start and stop bit disabled.
SHIFTCTLn	0x0103_0002	Configure transmit using Timer 1 on rising edge of shift clock with output data on Pin 0.
SHIFTCFG(n+1)	0x0000_0000	Start and stop bit disabled.
SHIFTCTL(n+1)	0x0180_0101	Configure receive using Timer 1 on falling edge of shift clock with input data on Pin 1.

Table continues on the next page...

Table 67-15. I2S Slave Configuration (continued)

Register	Value	Comments
TIMCMPn	0x0000_007F	Configure two 32-bit transfers per frame. Set TIMCMP[15:0] = (number of bits x 4) - 1.
TIMCFGn	0x0020_2500	Configure enable on pin rising edge (inverted bit clock) with trigger high (Timer 2) and disable on compare, initial clock state is logic 1 and decrement on pin input (bit clock).
TIMCTLn	0x0B40_0203	Configure 16-bit counter using Pin 2 input (bit clock), with Timer 2 output as the trigger.
TIMCMP(n+1)	0x0000_003F	Configure 32-bit transfers. Set TIMCMP[15:0] = (number of bits x 2) - 1.
TIMCFG(n+1)	0x0020_2500	Configure enable on pin (bit clock) rising edge with trigger (Timer 0) high and disable on compare, initial clock state is logic 1 and decrement on pin input (bit clock).
TIMCTL(n+1)	0x0340_0283	Configure 16-bit counter using inverted Pin 2 input (bit clock), with Timer 0 output as the trigger.
TIMCMP(n+2)	0x0000_0000	Compare on zero (first edge).
TIMCFG(n+2)	0x0020_6400	Configure enable on inverted pin (frame sync) rising edge and disable on trigger falling edge (bit clock), initial clock state is logic 1 and decrement on inverted pin input (frame sync).
TIMCTL(n+2)	0x04C0_0383	Configure 16-bit counter using inverted Pin 3 input (frame sync), with Pin 2 inverted input (bit clock) as the trigger.
SHIFTBUFn	Data to transmit	Transmit data can be written to SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request. Can support LSB first transfer by writing to SHIFTBUF register instead.
SHIFTBUF(n+1)	Data to receive	Received data can be read from SHIFTBUFBIS, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request. Can support LSB first transfer by reading from SHIFTBUF register instead.

67.4.8 SENT Receiver

SENT receiver can be supported by using one Timer and one Pin. The Timer is configured as Input-Capture mode, and captures the conter value at falling edge of the Pin input. Once the counter value is captured, the counter is automatically restarted from counter value 0. Therefore, the captured value is always indicating the period between previous falling edge and current falling edge. CPU interrupt or DMA trigger can be configured at each capture of the counter, and the entire SENT frame decoding with the latest tick width adustment is performed by CPU software.

Table 67-16. SENT Receiver Configuration

Register	Value	Comments
TIMCMPn	-	Counter value at the falling edge of the Pin is stored in this register.
TIMCFGn	0x0000_6000	Timer always enabled, Timer disabled on Trigger falling edge, Decrement counter on FlexIO clock.
TIMCTLn	0xnn40_0007	Single 16-bit input capture mode, Timer pin input and output are selected by PINSEL, Timer pin output disabled, Internal trigger selected, Select Pin nn as Trigger.

67.4.9 Camera Interface

Camera Interface can be supported using one Timer, one or more Shifters and multiple Pins. Multiple transfers can be supported using DMA controller.

The example below describes FlexIO configuration for interfacing to an 8-bit CMOS sensor with PCLK, VSYNC, HREF and D[7:0] outputs. The example uses a 128-bit buffer to capture 16-pixels of image data before interrupt or DMA transfer, however a bigger or smaller buffer may be used depending on system DMA performance and FlexIO resource usage by other applications. Note that additional timers may be used to track number of pixels per row and number of rows per frame or HREF/VSYNC may be assigned as GPIO interrupts for software tracking.

Table 67-17. Camera Interface Configuration for 8-bit CMOS sensor

Register	Value	Comments
SHIFTCFGn...n+2[1]	0x0007_0100	Configure 8-bit parallel shift in from adjacent shifter.
SHIFTCFGn+3	0x0007_0000	Configure 8-bit parallel shift in from pins FXIO_D[7:0] (D[7:0]).

Table continues on the next page...

Table 67-17. Camera Interface Configuration for 8-bit CMOS sensor (continued)

Register	Value	Comments
SHIFTCTLn...n+3	0x0080_0001	Configure receive using Timer 0 on negedge of clock.
TIMCMPn	0x0000_001F	Configure 16-pixel (8 bits/pixel x 16 pixels = 128-bits) transfer. Set TIMCMP[15:0] = (number of pixels x 2) - 1.
TIMCFGn	0x0120_6600	Configure enable on trigger (HREF) rising edge and disable on trigger falling edge, initial Shift clock state is logic 0 and decrement on PCLK input.
TIMCTLn	0x12C0_0803	Configure 16-bit counter using FXIO_D[8] input (PCLK), with FXIO_D[9] input (HREF) as the inverted trigger.
SHIFTBUFn...n+3	Data to receive	Received data can be read from SHIFTBUFn...n+3, use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request.

1. n=0 or 4

67.4.10 Motorola 68K/Intel 8080 Bus Interface

The Motorola 68K and Intel 8080 bus are two parallel interfaces commonly used by Smart/Asynchronous LCD controllers. In conjunction with GPIO, FlexIO is able to drive these interfaces using one Timer and one Shifter, although additional Shifters could be used to support large transfers via the DMA controller.

The configuration below provides an example of how to drive a 16-bit 68K or 8080 bus. For a 8080 bus, two GPIO are used to drive the nCS and RS pins. For a 68K bus, an additional GPIO is required to drive the RDWR pin.

Table 67-18. Motorola 68K/Intel 8080 Write Configuration

Register	Value	Comments
SHIFTCFG0...7	0x000F_0100	Configure 16-bit parallel shift in from adjacent shifter.
SHIFTCTL0	0x0003_0002	Configure transmit using Timer 0 on posedge of clock with data output to FXIO_D[15:0].
SHIFTCTL1...7	0x0000_0002	Configure transmit using Timer 0 on posedge of clock.
TIMCMP0	0x0000_0101 (1-beat) 0x0000_1F01 (16-beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of beats x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.

Table continues on the next page...

Table 67-18. Motorola 68K/Intel 8080 Write Configuration (continued)

Register	Value	Comments
TIMCFG0	0x0000_2200	Configure enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	0x01C3_1001 (Motorola 68K, 1-beat) 0x1DC3_1001 (Motorola 68K, 16-beats) 0x01C3_1081 (Intel 8080, 1-beat) 0x1DC3_1081 (Intel 8080, 16-beats)	Configure dual 8-bit counter using FXIO_D[16] output (EN pin for 68K, nWR pin for 8080), with Shifter 0 (1-beat) or Shifter 7 (16-beats) flag as the inverted trigger.
SHIFTBUF0...7	Data to transmit	Transmit data can be written to SHIFTBUF0 (1-beat) or SHIFTBUF0...7 (16-beats) to initiate a transfer, use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request.

Table 67-19. Motorola 68K/Intel 8080 Read Configuration

Register	Value	Comments
SHIFTCFG0...6	0x000F_0100	Configure 16-bit parallel shift in from adjacent shifter.
SHIFTCFG7	0x000F_0000	Configure 16-bit parallel shift in from pin.
SHIFTCTL0...7	0x0080_0001	Configure receive using Timer 0 on negedge of clock with data input from FXIO_D[15:0].
TIMCMP0	0x0000_0101 (1-beat) 0x0000_1F01 (16-beats)	Configure 1 or 16-beat transfer with baud rate of divide by 4 of the FlexIO clock. Set TIMCMP[15:8] = (number of beats x 2) - 1. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFG0	0x0000_2220	Configure stop_bit, enable on trigger high and disable on compare. Timer output high on enable.
TIMCTL0	0x1DC3_1001 (Motorola 68K, 1 beat) 0x01C3_1001 (Motorola 68K, 16 beats) 0x1DC3_1181 (Intel 8080, 1 beat) 0x01C3_1181 (Intel 8080, 16 beats)	Configure dual 8-bit counter using either FXIO_D[16] output (EN pin for 68K) or FXIO_D[17] output (nRD pin for 8080), with Shifter 7 flag (1-beat) or Shifter 0 flag (16-beats) as the inverted trigger.
SHIFTBUF0...7	Data received	Received data can be read from SHIFTBUF0 (1-beat) or SHIFTBUF0...7 (16-beats), use the Shifter Status Flag to indicate when data can be read using interrupt or DMA request.

In general, any operation to a 68K/8080 bus slave begins with a register write cycle followed by one or more data read or write cycles. To accomplish this, the following program flow should be used:

1. Configure FlexIO with 1-beat write configuration
2. Configure GPIO to assert nCS, RS pins (and deassert RDWR pin for 68K)
3. Write register index data to SHIFTBUF0[15:0]
4. Configure GPIO to deassert RS pin (and assert RDWR pin for 68K data read)
5. Configure FlexIO with desired read or write configuration (e.g. 1 or 16-beats)
6. Use the Shifter Status Flag to trigger interrupt or DMA driven data transfers to/from SHIFTBUF registers
7. Configure GPIO to deassert nCS pin

67.4.11 Low Power State Machine

The configuration below details a hypothetical state machine example to illustrate the flexibility allowed when using Shifter state mode.

In this example, FlexIO waits for the FXIO_D[2] pin to assert and then drives a complementary square wave output at a frequency of FLEXIO_CLK/131072 on the FXIO_D[1:0] pins while the comparator output is asserted (This assumes comparator is connected to external trigger 15. See the chip-specific FlexIO information for actual FlexIO trigger mappings). Throughout this operation, the CPU can be kept in a STOP/ VLPS mode by clearing the CTRL[DOZEN] bit and ensuring the FLEXIO_CLK is enabled. The state diagram below shows the states and transitions implemented by this example.

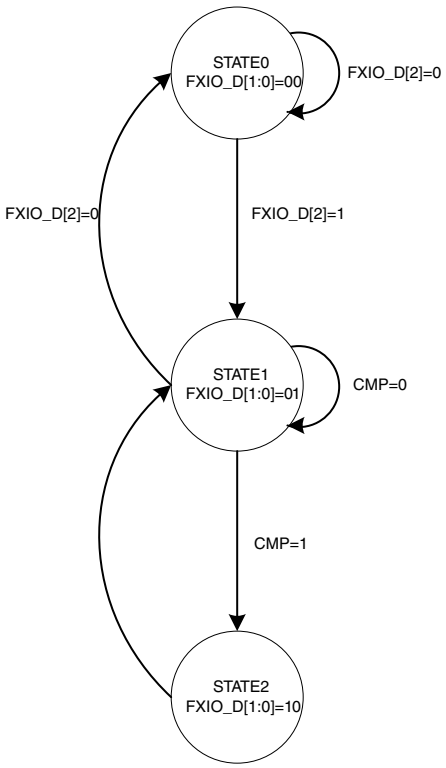


Figure 67-5. State Diagram

Table 67-20. State Machine Configuration

Register	Value	Comments
SHIFTCFG0...2	0x0000_0003	Enable FXIO_D[1:0] as outputs.
SHIFTCTL0	0x0080_0206	Configure for State mode using FXIO_D[4:2] as inputs to select next state and Timer0 output low to trigger next state.
SHIFTBUF0	0x0020_8208	State0: Drive FXIO_D[1:0]=00, transition to State0 if FXIO_D[2]=0, State1 if FXIO_D[2]=1.
SHIFTCTL1	0x0000_0206	Configure for State mode using FXIO_D[4:2] as inputs to select next state and Timer0 output high to trigger next state.
SHIFTBUF1	0x0140_8408	State1: Drive FXIO_D[1:0]=01, transition to State0 if FXIO_D[2]=0, State1 if CMP=0, State2 if CMP=1 (FXIO_D[3]=1)
SHIFTCTL2	0x0080_0206	Configure for State mode using FXIO_D[4:2] as inputs to select next state and Timer0 output low to trigger next state.
SHIFTBUF2	0x0224_9249	State2: Drive FXIO_D[1:0]=10, transition to State1 when Timer0 output low

Table continues on the next page...

Table 67-20. State Machine Configuration (continued)

Register	Value	Comments
TIMCMP0	0x0000_FFFF	Configure baud rate of divide by 131072 of the FlexIO clock. Set TIMCMP[7:0] = (baud rate divider / 2) - 1.
TIMCFG0	0x0000_0000	Configure timer always enabled.
TIMCTL0	0x0000_0003	Configure single 16-bit counter.
TIMCFG1	0x0010_7600	Configure timer enabled on trigger rising edge, disabled on trigger falling edge, decrement on trigger edge.
TIMCTL1	0x0F03_0303	Configure timer output enabled on FXIO_D[3], external trigger 15 (comparator output) selected.

67.4.12 Keypad Interface

The Keypad Interface can support a 3x4 keypad matrix using 3 timers and 3 shifters, although a larger matrix could be supported using additional shifters. The configuration is designed for 4 columns which are configured as active low open drain pins, and 3 rows which are configured as input pins with pullup resistors enabled.

One shifter is configured in logic mode and is configured to assert its output when any of the row inputs are low, indicating a key is pressed. A timer is used to filter the shifter output to ensure the key is pressed for a minimum amount of time before performing the column scan.

Another shifter is configured for parallel transmit, this shifter is used to scan each column when a key press is detected. Whenever not scanning, the shifter output is configured to assert all active low open drain column outputs to detect any keypress. A dedicated timer is used to control the transmit shifter.

The last shifter is configured for parallel receive, this shifter is used to capture the result of the column scanning for software to decode which key or keys was pressed. This configuration captures the state of both row and colum pins for each scan, although the row state could also be deduced by the shift order. A dedicated timer is used to control the receive shifter, it shifts at half the frequency of the transmit shifter.

Once the result of the key scan is available in the receive shifter register, FlexIO continues to monitor the row inputs and can trigger multiple scans from the one key press. To support deboucing, software can decide how many consecutive scans should be considered a single key press.

Since the pins used in logic mode are fixed per shifter, and the shifters that support parallel shifts are limited, this configuration is restricted in what pins and shifters it can use. Increasing the matrix beyond 4 row inputs requires multiple shifters in logic mode and increasing beyond 4 column outputs requires concatenating transmit shifters to create a larger shift register.

Table 67-21. Keypad Interface Configuration

Register	Value	Comments
SHIFTCFG0	0x0003_0032	Start bit enabled (logic 0) and stop bit enabled (logic 1), configired for 4-bit shift.
SHIFTCTL0	0x0101_0402	Configure transmit using Timer 1 on rising edge of clock generating open drain output on Pins [7:4] (column outputs).
SHIFTBUF0	0x0804_0201	Static data containing the column scan pattern, each column is scanned one-hot with deadtime inbetween.
SHIFTCFG1	0x0000_0020	In logic mode, mask input 3.
SHIFTCTL1	0x0000_0007	Configure logic mode.
SHIFTBUF1	0x07ff_07ff	Static data configuring logic mode LUT. Output asserts if pins [3:1] are logic 0.
SHIFTCFG3	0x0007_0000	Configured for 8-bit shift.
SHIFTCTL3	0x0280_0001	Configure receive using Timer 2 on falling edge of clock with input data on Pins [7:0] (both rows and columns, pin 0 is don't care).
TIMCMP0	0x0000_00ff	Configures pre-scanning glitch filter to 256 FlexIO clock cycles. For different filter cycles, set TIMCMP[15:0] = (filter cycles) - 1.
TIMCFG0	0x0103_6600	Configure enable on trigger rising edge and disable on trigger falling edge. Initial clock state is logic 0 and is not affected by reset.
TIMCTL0	0x0540_0003	Configure 16-bit counter using Shifter 1 flag (logic state) as trigger.
TIMCMP1	0x0000_0f3f	Configure 8 shifts (twice for each column) at column scan rate of divide by 128. For different scan frequency, set TIMCMP[7:0] = (scan divider / 2) - 1.
TIMCFG1	0x0020_2622	Enable on trigger rising edge, disable on compare, start and stop bits are enabled.
TIMCTL1	0x0340_0001	Configure dual 8-bit counter using Timer 0 output as the trigger.
TIMCMP2	0x0000_0801	Configure 4 shifts at half the frequency of Timer 1 trigger.

Table continues on the next page...

Table 67-21. Keypad Interface Configuration (continued)

Register	Value	Comments
TIMCFG2	0x0110_2100	Enable on Timer 1 enable, disable on compare, decrement on trigger input with output initially negated and not affected by reset
TIMCTL2	0x0740_0001	Configure dual 8-bit counter using Timer 1 output as the trigger.
SHIFTBUF3	Keypad Scan Result	Keypad scan result can be read from SHIFTBUF3, each byte is the result of one scan with both row and column pin state from the scan (note that pin 0 is not used). Use the Shifter Status Flag to indicate when data can be written using interrupt or DMA request.

67.5 FlexIO Signal Descriptions

Signal	Description	I/O
FXIO_Dn (n=0...31)	Bidirectional FlexIO Shifter and Timer pin inputs/outputs	I/O

67.6 Memory Map and Registers

67.6.1 FLEXIO register descriptions

NOTE

An invalid register access results in a bus error. This includes reading a write-only register, writing a read-only register or accessing an invalid address.

67.6.1.1 FLEXIO memory map

FLEXIO1 base address: 400A_C000h

FLEXIO2 base address: 400B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID Register (VERID)	32	RO	0200_0001
4	Parameter Register (PARAM)	32	RO	0220_0808
8	FlexIO Control Register (CTRL)	32	RW	0000_0000
C	Pin State Register (PIN)	32	RO	0000_0000
10	Shifter Status Register (SHIFTSTAT)	32	W1C	0000_0000
14	Shifter Error Register (SHIFTERR)	32	W1C	0000_0000
18	Timer Status Register (TIMSTAT)	32	W1C	0000_0000
20	Shifter Status Interrupt Enable (SHIFTSIEN)	32	RW	0000_0000
24	Shifter Error Interrupt Enable (SHIFTEIEN)	32	RW	0000_0000
28	Timer Interrupt Enable Register (TIMIEN)	32	RW	0000_0000
30	Shifter Status DMA Enable (SHIFTSDEN)	32	RW	0000_0000
38	Timer Status DMA Enable (TIMERSDEN)	32	RW	0000_0000
40	Shifter State Register (SHIFTSTATE)	32	RW	0000_0000
80 - 9C	Shifter Control N Register (SHIFTCTL0 - SHIFTCTL7)	32	RW	0000_0000
100 - 11C	Shifter Configuration N Register (SHIFTCFG0 - SHIFTCFG7)	32	RW	0000_0000
200 - 21C	Shifter Buffer N Register (SHIFTBUF0 - SHIFTBUF7)	32	RW	0000_0000
280 - 29C	Shifter Buffer N Bit Swapped Register (SHIFTBUFBIS0 - SHIFTBUFBIS7)	32	RW	0000_0000
300 - 31C	Shifter Buffer N Byte Swapped Register (SHIFTBUFBYS0 - SHIFTBUFBYS7)	32	RW	0000_0000
380 - 39C	Shifter Buffer N Bit Byte Swapped Register (SHIFTBUFBBS0 - SHIFTBUFBBS7)	32	RW	0000_0000
400 - 41C	Timer Control N Register (TIMCTL0 - TIMCTL7)	32	RW	0000_0000
480 - 49C	Timer Configuration N Register (TIMCFG0 - TIMCFG7)	32	RW	0000_0000
500 - 51C	Timer Compare N Register (TIMCMP0 - TIMCMP7)	32	RW	0000_0000
680 - 69C	Shifter Buffer N Nibble Byte Swapped Register (SHIFTBUFNBS0 - SHIFTBUFNBS7)	32	RW	0000_0000
700 - 71C	Shifter Buffer N Half Word Swapped Register (SHIFTBUFHWS0 - SHIFTBUFHWS7)	32	RW	0000_0000
780 - 79C	Shifter Buffer N Nibble Swapped Register (SHIFTBUFNIS0 - SHIFTBUFNIS7)	32	RW	0000_0000
800 - 81C	Shifter Buffer N Odd Even Swapped Register (SHIFTBUFOES0 - SHIFTBUFOES7)	32	RW	0000_0000
880 - 89C	Shifter Buffer N Even Odd Swapped Register (SHIFTBUFEOS0 - SHIFTBUFEOS7)	32	RW	0000_0000

67.6.1.2 Version ID Register (VERID)

67.6.1.2.1 Offset

Register	Offset
VERID	0h

67.6.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

67.6.1.2.3 Fields

Field	Description
31-24 MAJOR	Major Version Number This read only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000000 - Standard features implemented. 0000000000000001 - Supports state, logic and parallel modes. 0000000000000010 - Supports pin control registers. 0000000000000011 - Supports state, logic and parallel modes; plus pin control registers.

67.6.1.3 Parameter Register (PARAM)

67.6.1.3.1 Offset

Register	Offset
PARAM	4h

67.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRIGGER								PIN							
W																
Reset	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TIMER								SHIFTER							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

67.6.1.3.3 Fields

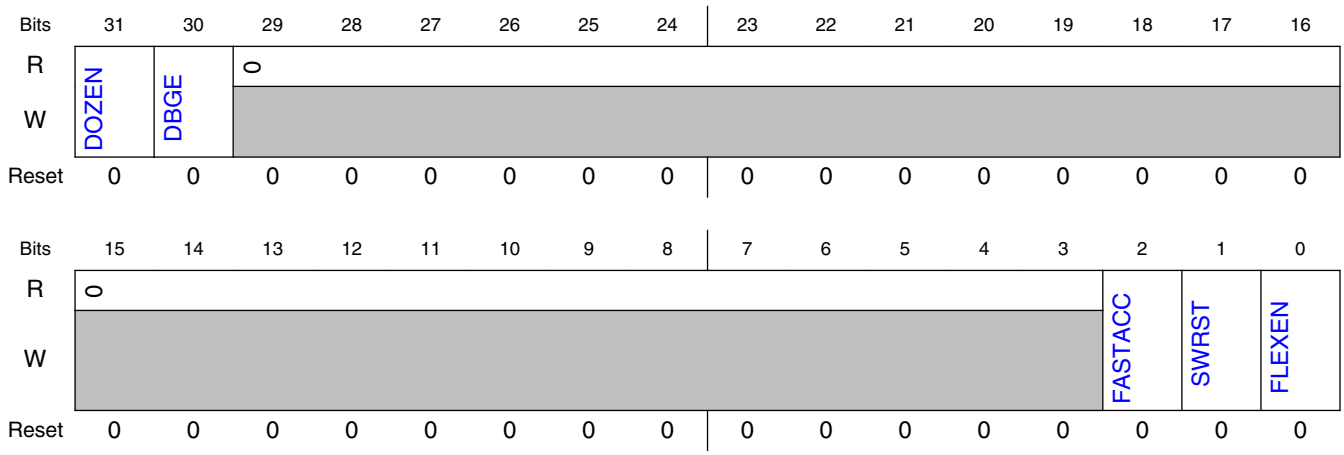
Field	Description
31-24 TRIGGER	Trigger Number Number of external triggers implemented.
23-16 PIN	Pin Number Number of Pins implemented.
15-8 TIMER	Timer Number Number of Timers implemented.
7-0 SHIFTER	Shifter Number Number of Shifters implemented.

67.6.1.4 FlexIO Control Register (CTRL)

67.6.1.4.1 Offset

Register	Offset
CTRL	8h

67.6.1.4.2 Diagram



67.6.1.4.3 Fields

Field	Description
31 DOZEN	Doze Enable Disables FlexIO operation in Doze modes. 0 - FlexIO enabled in Doze modes. 1 - FlexIO disabled in Doze modes.
30 DBGE	Debug Enable Enables FlexIO operation in Debug mode. 0 - FlexIO is disabled in debug modes. 1 - FlexIO is enabled in debug modes
29-3 —	Reserved
2 FASTACC	Fast Access Enables fast register accesses to FlexIO registers, but requires the FlexIO functional clock to be at least two times faster than the frequency of the bus clock. 0 - Configures for normal register accesses to FlexIO 1 - Configures for fast register accesses to FlexIO
1 SWRST	Software Reset The FlexIO Control Register is not affected by the software reset, all other logic in the FlexIO is affected by the software reset and all other register accesses are ignored until this bit is cleared. This register bit remains set until cleared by software, and the reset has cleared in the FlexIO clock domain. 0 - Software reset is disabled 1 - Software reset is enabled, all FlexIO registers except the Control Register are reset.
0 FLEXEN	FlexIO Enable 0 - FlexIO module is disabled. 1 - FlexIO module is enabled.

67.6.1.5 Pin State Register (PIN)

67.6.1.5.1 Offset

Register	Offset
PIN	Ch

67.6.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.5.3 Fields

Field	Description
31-0 PDI	Pin Data Input Returns the input data on each of the FlexIO pins.

67.6.1.6 Shifter Status Register (SHIFTSTAT)

67.6.1.6.1 Offset

Register	Offset
SHIFTSTAT	10h

67.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SSF							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.6.3 Fields

Field	Description
31-8 —	Reserved
7-0 SSF	Shifter Status Flag The shifter status flag is updated when one of the following events occurs: For SMOD =Receive, the status flag is set when SHIFTBUF has been loaded with data from Shifter (SHIFTBUF is full), and the status flag is cleared when SHIFTBUF register is read. For SMOD =Transmit, the status flag is set when SHIFTBUF data has been transferred to the Shifter (SHIFTBUF is empty) or when initially configured for SMOD=Transmit, and the status flag is cleared when the SHIFTBUF register is written. For SMOD =Match Store, the status flag is set when a match has occured between SHIFTBUF and Shifter, and the status flag is cleared when the SHIFTBUF register is read. For SMOD =Match Continuous, returns the current match result between the SHIFTBUF and Shifter. The status flag cannot be cleared by reading SHIFTBUF. For SMOD =State, the status flag for a shifter sets when it is selected by the current state pointer. For SMOD =Logic, returns the current value of the programmable logic block output. The status flag can also be cleared by writing a logic one to the flag for all modes except Match Continuous/State/Logic. 0b - Status flag is clear. 1b - Status flag is set.

67.6.1.7 Shifter Error Register (SHIFTERR)

67.6.1.7.1 Offset

Register	Offset
SHIFTERR	14h

67.6.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SEF							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.7.3 Fields

Field	Description
31-8 —	Reserved
7-0 SEF	Shifter Error Flags The shifter error flag is set when one of the following events occurs: For SMOD =Receive, indicates Shifter was ready to store new data into SHIFTBUF before the previous data was read from SHIFTBUF (SHIFTBUF Overrun), or indicates that the received start or stop bit does not match the expected value. For SMOD =Transmit, indicates Shifter was ready to load new data from SHIFTBUF before new data had been written into SHIFTBUF (SHIFTBUF Underrun). For SMOD =Match Store, indicates a match event occured before the previous match data was read from SHIFTBUF (SHIFTBUF Overrun). For SMOD =Match Continuous, the error flag is set when a match has occured between SHIFTBUF and Shifter. For SMOD =Logic, the error flag is set when the output of the programmable logic block has asserted. Can be cleared by writing logic one to the flag. For SMOD=Match Continuous, can also be cleared when the SHIFTBUF register is read. 0b - Shifter Error Flag is clear. 1b - Shifter Error Flag is set.

67.6.1.8 Timer Status Register (TIMSTAT)

67.6.1.8.1 Offset

Register	Offset
TIMSTAT	18h

67.6.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TSF							
W									W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.8.3 Fields

Field	Description
31-8 —	Reserved
7-0 TSF	Timer Status Flags The timer status flag sets depending on the timer mode, and can be cleared by writing logic one to the flag. In 8-bit baud counter mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements. In 8-bit high PWM mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements. In 16-bit counter mode, the timer status flag is set when the 16-bit counter equals zero and decrements. In 16-bit counter disable mode, the timer status flag is set when a timer disable event is detected. In 8-bit word counter mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements. In 8-bit low PWM mode, the timer status flag is set when the upper 8-bit counter equals zero and decrements.

Field	Description
	In 16-bit input capture mode, the timer status flag is set when a timer disable event is detected and the timer status flag is clear. In this mode, the timer compare register should only be read when the timer status flag is set. 0b - Timer Status Flag is clear. 1b - Timer Status Flag is set.

67.6.1.9 Shifter Status Interrupt Enable (SHIFTSIEN)

67.6.1.9.1 Offset

Register	Offset
SHIFTSIEN	20h

67.6.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SSIE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.9.3 Fields

Field	Description
31-8 —	Reserved
7-0 SSIE	Shifter Status Interrupt Enable Enables interrupt generation when corresponding SSF is set. 0b - Shifter Status Flag interrupt disabled. 1b - Shifter Status Flag interrupt enabled.

67.6.1.10 Shifter Error Interrupt Enable (SHIFTEIEN)

67.6.1.10.1 Offset

Register	Offset
SHIFTEIEN	24h

67.6.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SEIE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.10.3 Fields

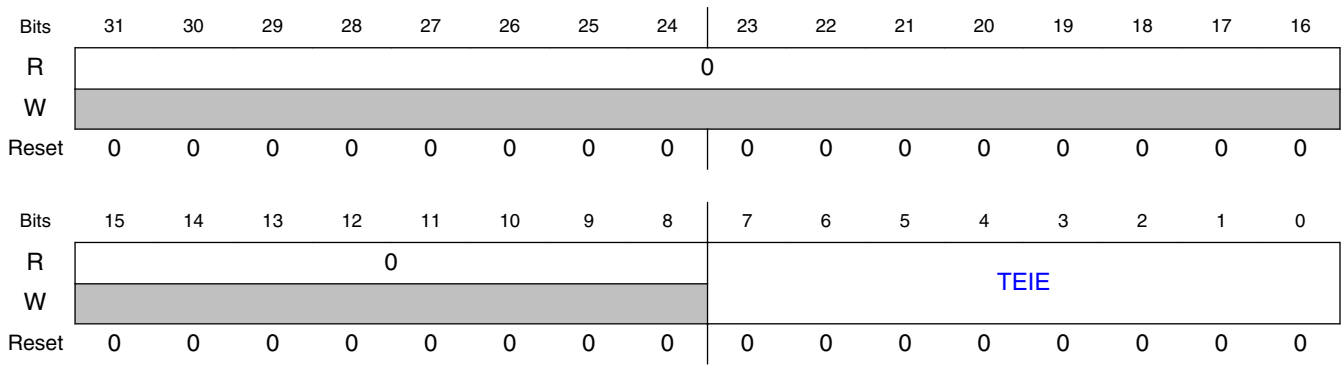
Field	Description
31-8 —	Reserved
7-0 SEIE	Shifter Error Interrupt Enable Enables interrupt generation when corresponding SEF is set. 0b - Shifter Error Flag interrupt disabled. 1b - Shifter Error Flag interrupt enabled.

67.6.1.11 Timer Interrupt Enable Register (TIMIEN)

67.6.1.11.1 Offset

Register	Offset
TIMIEN	28h

67.6.1.11.2 Diagram



67.6.1.11.3 Fields

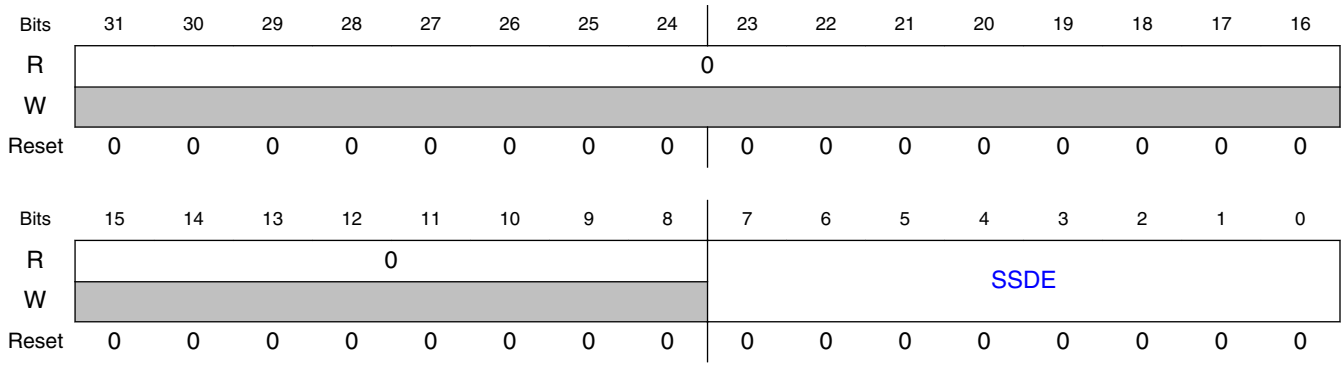
Field	Description
31-8 —	Reserved
7-0 TEIE	Timer Status Interrupt Enable Enables interrupt generation when corresponding TSF is set. 0b - Timer Status Flag interrupt is disabled. 1b - Timer Status Flag interrupt is enabled.

67.6.1.12 Shifter Status DMA Enable (SHIFTSDEN)

67.6.1.12.1 Offset

Register	Offset
SHIFTSDEN	30h

67.6.1.12.2 Diagram



67.6.1.12.3 Fields

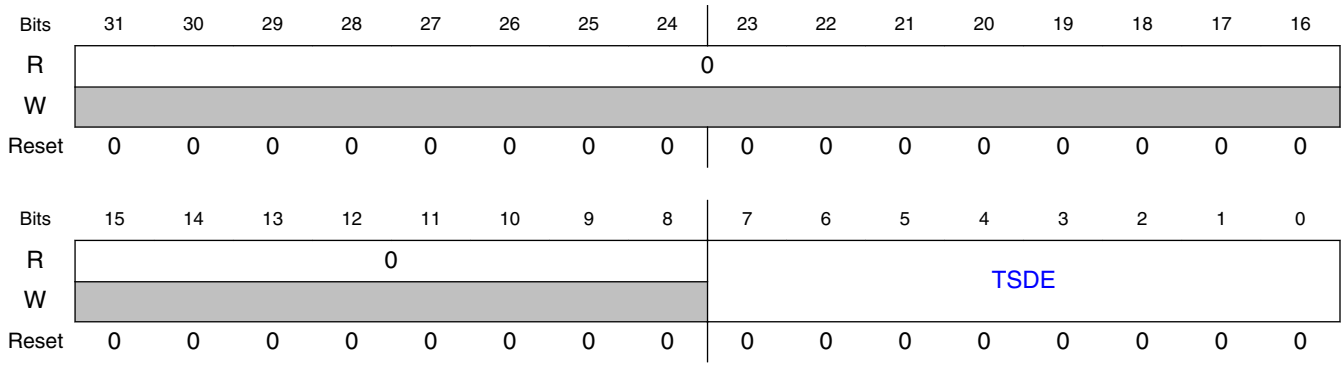
Field	Description
31-8 —	Reserved
7-0 SSDE	Shifter Status DMA Enable Enables DMA request generation when corresponding SSF is set. 0b - Shifter Status Flag DMA request is disabled. 1b - Shifter Status Flag DMA request is enabled.

67.6.1.13 Timer Status DMA Enable (TIMERSDEN)

67.6.1.13.1 Offset

Register	Offset
TIMERSDEN	38h

67.6.1.13.2 Diagram



67.6.1.13.3 Fields

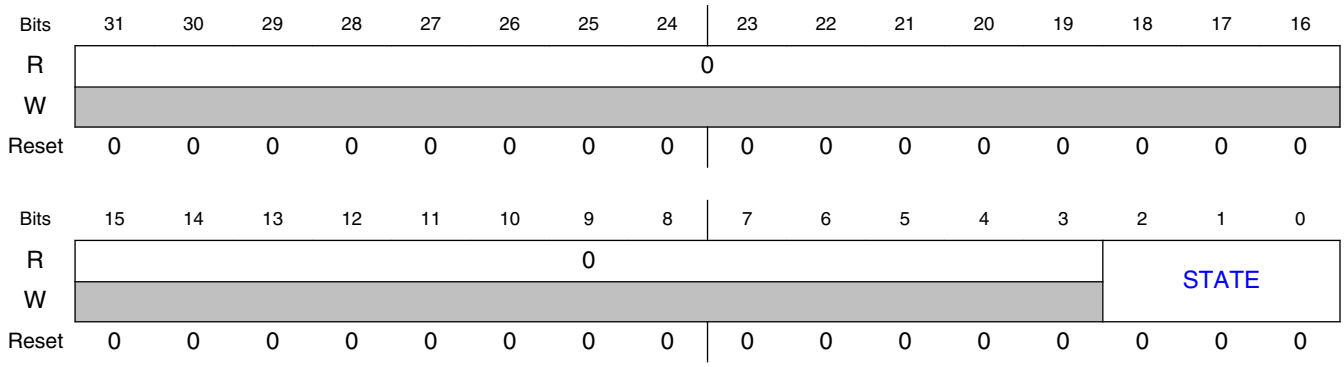
Field	Description
31-8 —	Reserved
7-0 TSDE	Timer Status DMA Enable Enables DMA request generation when corresponding TSF is set. When the Timer Status DMA request is enabled, reading or writing a Timer Compare Register clears the corresponding Timer Status Register. The DMA must therefore read or write the Timer Compare Register as part of the DMA transfer, otherwise the DMA request remains asserted. 0b - Timer Status Flag DMA request is disabled. 1b - Timer Status Flag DMA request is enabled.

67.6.1.14 Shifter State Register (SHIFTSTATE)

67.6.1.14.1 Offset

Register	Offset
SHIFTSTATE	40h

67.6.1.14.2 Diagram



67.6.1.14.3 Fields

Field	Description
31-3 —	Reserved
2-0 STATE	Current State Pointer The current state field maintains a pointer to keep track of the current Shifter (configured for State mode) enabled to drive outputs and compute the next state. Reading this register when the state pointer is updating can result in the incorrect state returned.

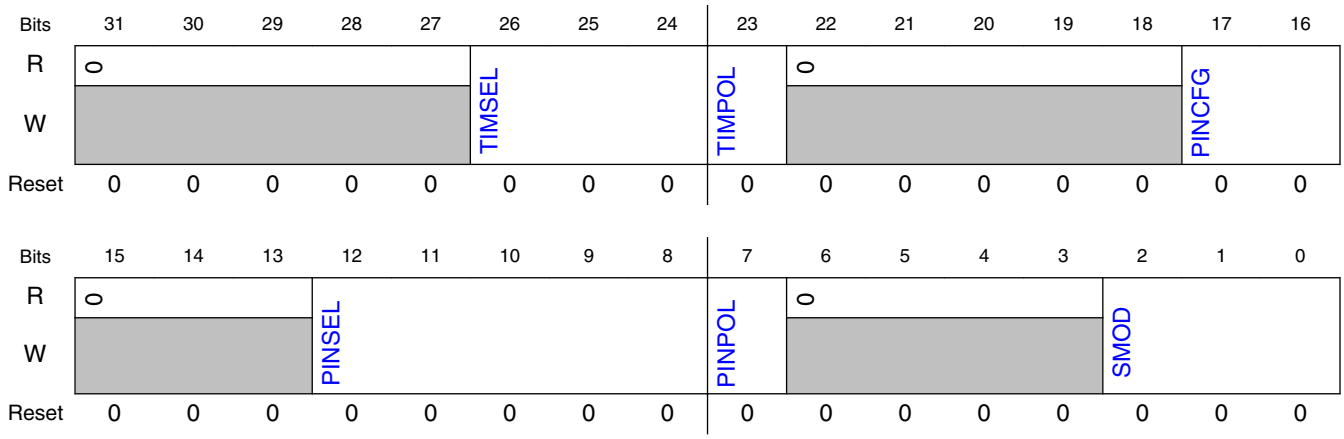
67.6.1.15 Shifter Control N Register (SHIFTCTL0 - SHIFTCTL7)

67.6.1.15.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTCTLa	80h + (a × 4h)

67.6.1.15.2 Diagram



67.6.1.15.3 Fields

Field	Description
31-27 —	Reserved
26-24 TIMSEL	Timer Select Selects which Timer is used for controlling the logic/shift register and generating the Shift clock. TIMSEL=i selects TIMERi.
23 TIMPOL	Timer Polarity 0 - Shift on posedge of Shift clock 1 - Shift on negedge of Shift clock
22-18 —	Reserved
17-16 PINCFG	Shifter Pin Configuration For pins configured as an output (PINCFG=11), this field takes effect when the register is written. 00 - Shifter pin output disabled 01 - Shifter pin open drain or bidirectional output enable 10 - Shifter pin bidirectional output data 11 - Shifter pin output
15-13 —	Reserved
12-8 PINSEL	Shifter Pin Select Selects which pin is used by the Shifter input or output. PINSEL=i selects the FXIO_Di pin. For pins configured as an output (PINCFG=11), this field takes effect when the register is written.
7 PINPOL	Shifter Pin Polarity For pins configured as an output (PINCFG=11), this field takes effect when the register is written. 0 - Pin is active high

Table continues on the next page...

Field	Description
	1 - Pin is active low
6-3 —	Reserved
2-0 SMOD	Shifter Mode Configures the mode of the Shifter. 000 - Disabled. 001 - Receive mode. Captures the current Shifter content into the SHIFTBUF on expiration of the Timer. 010 - Transmit mode. Load SHIFTBUF contents into the Shifter on expiration of the Timer. 011 - Reserved. 100 - Match Store mode. Shifter data is compared to SHIFTBUF content on expiration of the Timer. 101 - Match Continuous mode. Shifter data is continuously compared to SHIFTBUF contents. 110 - State mode. SHIFTBUF contents are used for storing programmable state attributes. 111 - Logic mode. SHIFTBUF contents are used for implementing programmable logic look up table.

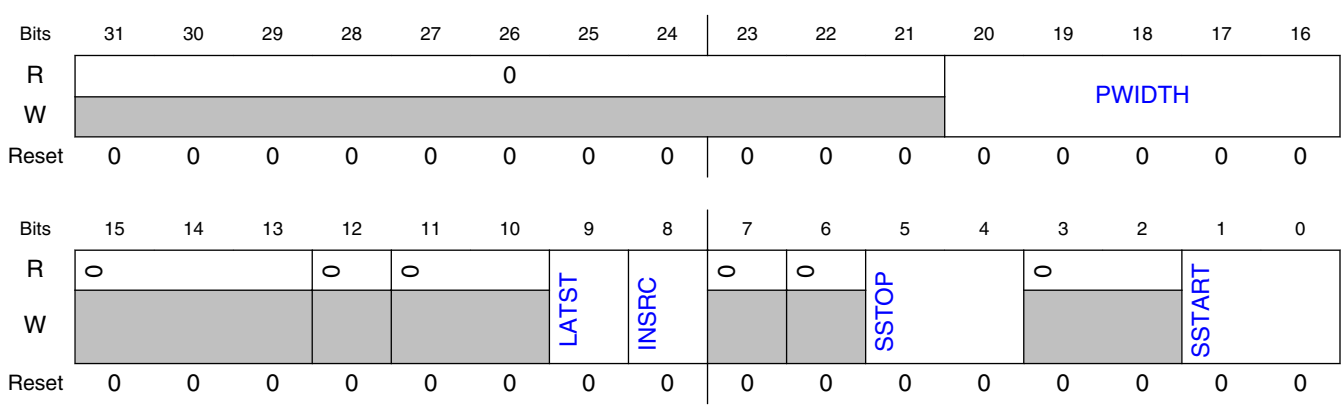
67.6.1.16 Shifter Configuration N Register (SHIFTCFG0 - SHIFTCFG7)

67.6.1.16.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTCFGa	100h + (a × 4h)

67.6.1.16.2 Diagram



67.6.1.16.3 Fields

Field	Description
31-21 —	Reserved
20-16 PWIDTH	Parallel Width For all Shifters, this register field configures the number of bits to be shifted on each Shift clock as follows: 1-bit shift for PWIDTH=0 2-bit shift for PWIDTH=1 4-bit shift for PWIDTH=2...3 8-bit shift for PWIDTH=4...7 16-bit shift for PWIDTH=8...15 32-bit shift for PWIDTH=16...31 For Shifters which support parallel transmit (SHIFTER0, SHIFTER4, ...) or parallel receive (SHIFTER3, SHIFTER7, ...), this register field, together with SHIFTCTL[PINSEL], also selects the pins to be driven or sampled on each Shift clock as follows: • FXIO_D[PINSEL+PWIDTH]:FXIO_D[PINSEL] Shifters which do not support parallel transmit or parallel receive only support parallel shift when INSRC =1. For SMOD =State, this field is used to disable state outputs. See State Mode section for more detail.
15-13 —	Reserved
12 —	Reserved
11-10 —	Reserved
9 LATST	Late Store Configures what happens when a receive or match shift register is configured to both shift and store on the same cycle. 0 - Shift register stores the pre-shift register state. 1 - Shift register stores the post-shift register state.
8 INSRC	Input Source Selects the input source for the shifter. Configuring INSRC=1 is not supported for the last shifter. 0 - Pin 1 - Shifter N+1 Output
7 —	Reserved
6 —	Reserved
5-4	Shifter Stop bit

Table continues on the next page...

Field	Description
SSTOP	For SMOD =Transmit, this field allows automatic stop bit insertion if the selected timer has also enabled a stop bit. For SMOD =Receive or Match Store, this field allows automatic stop bit checking if the selected timer has also enabled a stop bit. For SMOD =State, this field is used to disable state outputs. See 'State Mode' section for more detail. For SMOD =Logic, this field is used to mask logic pin inputs. See 'Logic Mode' section for more detail. 00 - Stop bit disabled for transmitter/receiver/match store 01 - Reserved for transmitter/receiver/match store 10 - Transmitter outputs stop bit value 0 on store, receiver/match store sets error flag if stop bit is not 0 11 - Transmitter outputs stop bit value 1 on store, receiver/match store sets error flag if stop bit is not 1
3-2 —	Reserved
1-0 SSTART	Shifter Start bit For SMOD =Transmit, this field allows automatic start bit insertion if the selected timer has also enabled a start bit. For SMOD =Receive or Match Store, this field allows automatic start bit checking if the selected timer has also enabled a start bit. For SMOD =State, this field is used to disable state outputs. See 'State Mode' section for more detail. For SMOD =Logic, this field is used to mask logic pin inputs. See 'Logic Mode' section for more detail. 00 - Start bit disabled for transmitter/receiver/match store, transmitter loads data on enable 01 - Start bit disabled for transmitter/receiver/match store, transmitter loads data on first shift 10 - Transmitter outputs start bit value 0 before loading data on first shift, receiver/match store sets error flag if start bit is not 0 11 - Transmitter outputs start bit value 1 before loading data on first shift, receiver/match store sets error flag if start bit is not 1

67.6.1.17 Shifter Buffer N Register (SHIFTBUF0 - SHIFTBUF7)

67.6.1.17.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFa	200h + (a × 4h)

67.6.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.17.3 Fields

Field	Description
31-0 SHIFTBUF	Shift Buffer Shift buffer data is used for a variety of functions depending on the SMOD setting: For SMOD=Receive, Shifter data is transferred into SHIFTBUF at the expiration of Timer. The SHIFTBUF register should only be read when the corresponding shifter status flag is set, indicating new Shifter data is available. For SMOD=Transmit, SHIFTBUF data is transferred into the Shifter before the Timer begins. For SMOD=Match Store, SHIFTBUF[31:16] contains the data to be matched with the Shifter contents and SHIFTBUF[15:0] can be used to mask the match result (1=mask, 0=no mask). The Match is checked when the Timer expires and Shifter data [31:16] is written to SHIFTBUF[31:16] whenever a match event occurs. The SHIFTBUF register should only be read when the corresponding shifter status flag is set, indicating new Shifter data is available. For SMOD=Match Continuous, SHIFTBUF[31:16] contains the data to be matched with the Shifter contents and SHIFTBUF[15:0] can be used to mask the match result (1=mask, 0=no mask). For SMOD=Logic, SHIFTBUF[31:0] is used to implement a 5-input, 32-bit programmable logic look-up table. See Logic Mode section for more detail. For SMOD=State, SHIFTBUF[31:24] is used to drive the output value when this shifter is selected by the current state pointer and SHIFTBUF[23:0] is used to configure the value of the next state transition. See State Mode section for more detail.

67.6.1.18 Shifter Buffer N Bit Swapped Register (SHIFTBUFBIS0 - SHIFTBUFBIS7)

67.6.1.18.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFBISa	280h + (a × 4h)

67.6.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.18.3 Fields

Field	Description
31-0 SHIFTBUFBIS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are bit swapped. Reads return SHIFTBUF[0:31].

67.6.1.19 Shifter Buffer N Byte Swapped Register (SHIFTBUFBYS0 - SHIFTBUFBYS7)

67.6.1.19.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFBYSa	300h + (a × 4h)

67.6.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBYS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBYS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.19.3 Fields

Field	Description
31-0 SHIFTBUFBYS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are byte swapped. Reads return { SHIFTBUF[7:0], SHIFTBUF[15:8], SHIFTBUF[23:16], SHIFTBUF[31:24] }.

67.6.1.20 Shifter Buffer N Bit Byte Swapped Register (SHIFTBUFBBS0 - SHIFTBUFBBS7)

67.6.1.20.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFBBSa	380h + (a × 4h)

67.6.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFBBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFBBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.20.3 Fields

Field	Description
31-0 SHIFTBUFBBS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are bit swapped within each byte. Reads return { SHIFTBUF[24:31], SHIFTBUF[16:23], SHIFTBUF[8:15], SHIFTBUF[0:7] }.

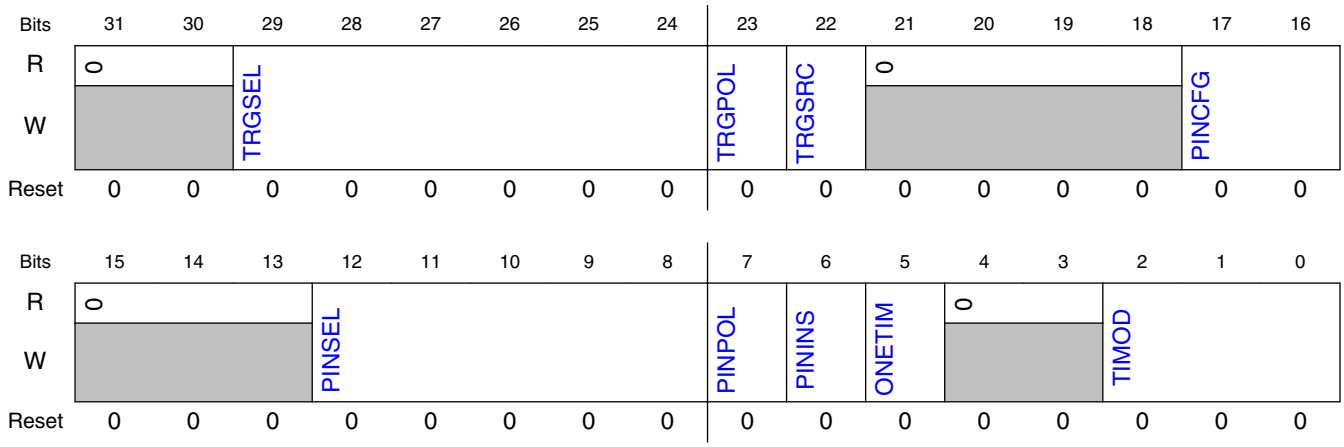
67.6.1.21 Timer Control N Register (TIMCTL0 - TIMCTL7)

67.6.1.21.1 Offset

For a = 0 to 7:

Register	Offset
TIMCTLa	400h + (a × 4h)

67.6.1.21.2 Diagram



67.6.1.21.3 Fields

Field	Description
31-30 —	Reserved
29-24 TRGSEL	Trigger Select The valid values for TRGSEL depends on the FLEXIO_PARAM register. • When TRGSRC = 1, the valid values for N depends on PIN, TIMER, SHIFTER fields in the FLEXIO_PARAM register. • When TRGSRC = 0, the valid values for N depends on TRIGGER field in FLEXIO_PARAM register. Refer to the chip-specific FlexIO information for external trigger selection. **NOTE:** For a pin, N=0 to 31. For a Shifter, N=0 to 7. For a Timer, N=0 to 7. When TRGSRC = 0 the trigger selection is configured as follows: • N - External trigger N input When TRGSRC = 1 the internal trigger can be configured to select an input pin as follows: • 2*N - Pin N input When TRGSRC = 1 the internal trigger can be configured to select a shifter/timer signal as follows: • 4*N+1 - Shifter N status flag • 4*N+3 - Timer N trigger output In other words, the expanded internal trigger selection (TRGSRC = 1) is as follows: • 0000 = Pin 0 • 0001 = Shifter 0 Flag • 0010 = Pin 1 • 0011 = Timer 0 Trigger • 0100 = Pin 2 • 0101 = Shifter 1 Flag • 0110 = Pin 3 • 0111 = Timer 1 Trigger • ... • This continues up to the Pin 31, Shifter 7 and Timer 7.

Table continues on the next page...

Field	Description
23 TRGPOL	Trigger Polarity 0 - Trigger active high 1 - Trigger active low
22 TRGSRC	Trigger Source 0 - External trigger selected 1 - Internal trigger selected
21-18 —	Reserved
17-16 PINCFG	Timer Pin Configuration Configures the direction of the Timer pin. For pins configured as an output (PINCFG=11), this field takes effect when the register is written. 00 - Timer pin output disabled 01 - Timer pin open drain or bidirectional output enable 10 - Timer pin bidirectional output data 11 - Timer pin output
15-13 —	Reserved
12-8 PINSEL	Timer Pin Select Selects which pin is used by the Timer input or output. PINSEL=i selects the FXIO_Di pin. For pins configured as an output (PINCFG=11), this field takes effect when the register is written.
7 PINPOL	Timer Pin Polarity For pins configured as an output (PINCFG=11), this field takes effect when the register is written. 0 - Pin is active high 1 - Pin is active low
6 PININS	Timer Pin Input Select When set, the timer input pin is a different pin from the timer output pin. PINSEL must select an even numbered pin when this bit is set, so the output pin is even numbered and input pin is odd numbered. 0 - Timer pin input and output are selected by PINSEL. 1 - Timer pin input is selected by PINSEL+1, timer pin output remains selected by PINSEL.
5 ONETIM	Timer One Time Operation When set, configures the timer to perform a single enable/disable iteration, after which software must clear the timer status flag for the timer to be enabled again. 0 - The timer enable event is generated as normal. 1 - The timer enable event is blocked unless timer status flag is clear.
4-3 —	Reserved
2-0 TIMOD	Timer Mode In 8-bit baud counter mode, the lower 8-bits of the counter and compare register are used to configure the baud rate of the timer shift clock, and the upper 8-bits are used to configure the shifter bit count.

Field	Description
	In 8-bit PWM high mode, the lower 8-bits of the counter and compare register are used to configure the high period of the timer shift clock, and the upper 8-bits are used to configure the low period of the timer shift clock. The shifter bit count is configured using another timer or external signal. In 16-bit counter mode, the full 16-bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. In 16-bit counter disable mode, the full 16-bits of the counter and compare register are used to configure either the baud rate of the shift clock or the shifter bit count. In 8-bit word counter mode, the lower 8-bits of the counter and compare register are used to configure the shifter bit count, and the upper 8-bits are used to configure the shifter word count. In 8-bit PWM low mode, the lower 8-bits of the counter and compare register are used to configure the low period of the timer shift clock and the upper 8-bits are used to configure the high period of the timer shift clock. The shifter bit count is configured using another timer or external signal. In 16-bit input capture mode, the inverted value of 16-bit counter is latched into the compare register when a timer counter disable condition is detected (as configured by TIMDIS). This also sets the timer status flag. The timer counter is immediately restarted from 0xFFFF 000 - Timer Disabled. 001 - Dual 8-bit counters baud mode. 010 - Dual 8-bit counters PWM high mode. 011 - Single 16-bit counter mode. 100 - Single 16-bit counter disable mode. 101 - Dual 8-bit counters word mode. 110 - Dual 8-bit counters PWM low mode. 111 - Single 16-bit input capture mode.

67.6.1.22 Timer Configuration N Register (TIMCFG0 - TIMCFG7)

The options to enable or disable the timer using the Timer N-1 enable or disable are reserved when N is evenly divisible by 4 (eg: Timer 0).

NOTE

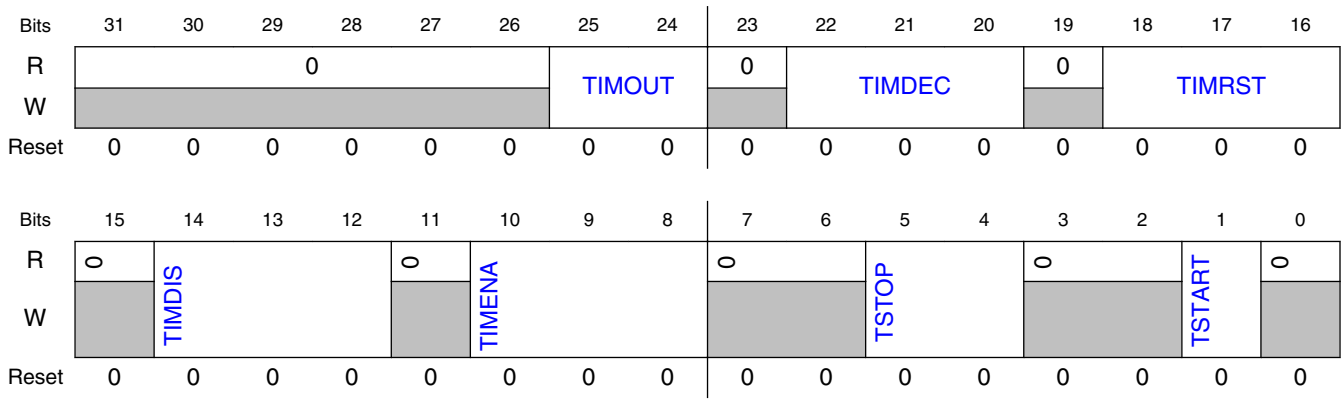
The pin and trigger level/edges specified below refer to the signal state after being modified by the PINPOL or TRGPOL setting in the TIMCTL register. For example, "Trigger low" means a trigger is actually at logic level 1 if the TRGPOL is set to 1 (active low).

67.6.1.22.1 Offset

For a = 0 to 7:

Register	Offset
TIMCFGa	480h + (a × 4h)

67.6.1.22.2 Diagram



67.6.1.22.3 Fields

Field	Description
31-26 —	Reserved
25-24 TIMOUT	Timer Output Configures the initial state of the Timer Output and whether it is affected by the Timer reset. 00 - Timer output is logic one when enabled and is not affected by timer reset 01 - Timer output is logic zero when enabled and is not affected by timer reset 10 - Timer output is logic one when enabled and on timer reset 11 - Timer output is logic zero when enabled and on timer reset
23 —	Reserved
22-20 TIMDEC	Timer Decrement Configures the source of the Timer decrement and the source of the Shift clock. 000 - Decrement counter on FlexIO clock, Shift clock equals Timer output. 001 - Decrement counter on Trigger input (both edges), Shift clock equals Timer output. 010 - Decrement counter on Pin input (both edges), Shift clock equals Pin input. 011 - Decrement counter on Trigger input (both edges), Shift clock equals Trigger input. 100 - Decrement counter on FlexIO clock divided by 16, Shift clock equals Timer output. 101 - Decrement counter on FlexIO clock divided by 256, Shift clock equals Timer output. 110 - Decrement counter on Pin input (rising edge), Shift clock equals Pin input.

Table continues on the next page...

Field	Description
	111 - Decrement counter on Trigger input (rising edge), Shift clock equals Trigger input.
19 —	Reserved
18-16 TIMRST	Timer Reset Configures the condition that causes the timer counter (and optionally the timer output) to be reset. In 8-bit counter mode, the timer reset only resets the lower 8-bits that configure the baud rate. In all other modes, the timer reset resets the full 16-bits of the counter. 000 - Timer never reset 001 - Timer reset on Timer Output high. 010 - Timer reset on Timer Pin equal to Timer Output 011 - Timer reset on Timer Trigger equal to Timer Output 100 - Timer reset on Timer Pin rising edge 101 - Reserved 110 - Timer reset on Trigger rising edge 111 - Timer reset on Trigger rising or falling edge
15 —	Reserved
14-12 TIMDIS	Timer Disable Configures the condition that causes the Timer to be disabled and stop decrementing. 000 - Timer never disabled 001 - Timer disabled on Timer N-1 disable 010 - Timer disabled on Timer compare (upper 8-bits match and decrement) 011 - Timer disabled on Timer compare (upper 8-bits match and decrement) and Trigger Low 100 - Timer disabled on Pin rising or falling edge 101 - Timer disabled on Pin rising or falling edge provided Trigger is high 110 - Timer disabled on Trigger falling edge 111 - Reserved
11 —	Reserved
10-8 TIMENA	Timer Enable Configures the condition that causes the Timer to be enabled and start decrementing. 000 - Timer always enabled 001 - Timer enabled on Timer N-1 enable 010 - Timer enabled on Trigger high 011 - Timer enabled on Trigger high and Pin high 100 - Timer enabled on Pin rising edge 101 - Timer enabled on Pin rising edge and Trigger high 110 - Timer enabled on Trigger rising edge 111 - Timer enabled on Trigger rising or falling edge
7-6	Reserved

Table continues on the next page...

Field	Description
—	
5-4 TSTOP	Timer Stop Bit The stop bit can be added on a timer compare (between each word) or on a timer disable. When stop bit is enabled, configured shifters output the contents of the stop bit when the timer is disabled. When stop bit is enabled on timer disable, the timer remains disabled until the next rising edge of the shift clock. If configured for both timer compare and timer disable, only one stop bit is inserted on timer disable. 00 - Stop bit disabled 01 - Stop bit is enabled on timer compare 10 - Stop bit is enabled on timer disable 11 - Stop bit is enabled on timer compare and timer disable
3-2 —	Reserved
1 TSTART	Timer Start Bit When start bit is enabled, configured shifters outputs the contents of the start bit when the timer is enabled and the timer counter reloads from the compare register on the first rising edge of the shift clock. 0 - Start bit disabled 1 - Start bit enabled
0 —	Reserved

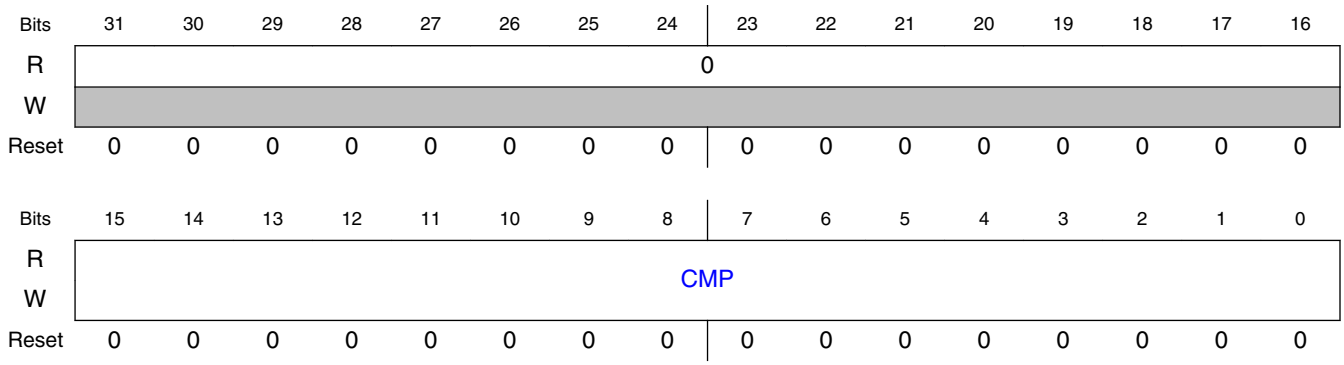
67.6.1.23 Timer Compare N Register (TIMCMP0 - TIMCMP7)

67.6.1.23.1 Offset

For a = 0 to 7:

Register	Offset
TIMCMPa	500h + (a × 4h)

67.6.1.23.2 Diagram



67.6.1.23.3 Fields

Field	Description
31-16 —	Reserved
15-0 CMP	Timer Compare Value The timer compare value is loaded into the timer counter when the timer is first enabled, when the timer is reset and when the timer decrements down to zero. In 8-bit baud counter mode, the lower 8-bits configure the baud rate divider equal to (CMP[7:0] + 1) * 2. The upper 8-bits configure the number of bits in each word equal to (CMP[15:8] + 1) / 2. In 8-bit PWM high mode, the lower 8-bits configure the high period of the output to (CMP[7:0] + 1) and the upper 8-bits configure the low period of the output to (CMP[15:8] + 1). In 16-bit counter mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) to equal (CMP[15:0] + 1) * 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word equal to (CMP[15:0] + 1) / 2. In 16-bit counter disable mode, the compare value can be used to generate the baud rate divider (if shift clock source is timer output) to equal (CMP[15:0] + 1) * 2. When the shift clock source is a pin or trigger input, the compare register is used to set the number of bits in each word equal to (CMP[15:0] + 1) / 2. In 8-bit word counter mode, the lower 8-bits configure the number of bits in each word equal to (CMP[7:0] + 1) / 2. The upper 8-bits configure the number of words to transfer equal to (CMP[15:8] + 1) / 2. In 8-bit PWM low mode, the lower 8-bits configure the low period of the output to (CMP[7:0] + 1) and the upper 8-bits configure the high period of the output to (CMP[15:8] + 1). In 16-bit input capture mode, the compare register is updated with the inverse of the timer counter value whenever the timer status flag is set. The timer compare register should only be read when the timer status flag is set.

67.6.1.24 Shifter Buffer N Nibble Byte Swapped Register (SHIFTBUFNBS0 - SHIFTBUFNBS7)

67.6.1.24.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFNBSa	680h + (a × 4h)

67.6.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFNBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFNBS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.24.3 Fields

Field	Description
31-0 SHIFTBUFNBS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are nibble swapped within each byte. Reads return { SHIFTBUF[27:24], SHIFTBUF[31:28], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[3:0], SHIFTBUF[7:4] }.

67.6.1.25 Shifter Buffer N Half Word Swapped Register (SHIFTBUFHWS0 - SHIFTBUFHWS7)

67.6.1.25.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFHWSa	700h + (a × 4h)

67.6.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFHWS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFHWS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.25.3 Fields

Field	Description
31-0 SHIFTBUFHWS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are half word swapped. Reads return { SHIFTBUF[15:0], SHIFTBUF[31:16] }.

67.6.1.26 Shifter Buffer N Nibble Swapped Register (SHIFTBUFNIS0 - SHIFTBUFNIS7)

67.6.1.26.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFNISa	780h + (a × 4h)

67.6.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFNIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFNIS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.26.3 Fields

Field	Description
31-0 SHIFTBUFNIS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are nibble swapped. Reads return { SHIFTBUF[3:0], SHIFTBUF[7:4], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[27:24], SHIFTBUF[31:28] }.

67.6.1.27 Shifter Buffer N Odd Even Swapped Register (SHIFTBUFOES0 - SHIFTBUFOES7)

67.6.1.27.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFOESa	800h + (a × 4h)

67.6.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFOES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFOES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.27.3 Fields

Field	Description
31-0 SHIFTBUFOES	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register have the odd and even bits partitioned separately. Only 32-bit accesses are supported to this register. Reads return { SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1], SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0] }.

67.6.1.28 Shifter Buffer N Even Odd Swapped Register (SHIFTBUFEOS0 - SHIFTBUFEOS7)

67.6.1.28.1 Offset

For a = 0 to 7:

Register	Offset
SHIFTBUFEOSa	880h + (a × 4h)

67.6.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SHIFTBUFEOS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SHIFTBUFEOS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

67.6.1.28.3 Fields

Field	Description
31-0 SHIFTBUFEOS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register have the even and odd bits partitioned separately. Only 32-bit accesses are supported to this register. Reads return { SHIFTBUF[30], SHIFTBUF[28], SHIFTBUF[26], SHIFTBUF[24], SHIFTBUF[22], SHIFTBUF[20], SHIFTBUF[18], SHIFTBUF[16], SHIFTBUF[14], SHIFTBUF[12], SHIFTBUF[10], SHIFTBUF[8], SHIFTBUF[6], SHIFTBUF[4], SHIFTBUF[2], SHIFTBUF[0], SHIFTBUF[31], SHIFTBUF[29], SHIFTBUF[27],SHIFTBUF[25],SHIFTBUF[23],SHIFTBUF[21],SHIFTBUF[19],SHIFTBUF[17],SHIFTBUF[15],SHIFTBUF[13],SHIFTBUF[11],SHIFTBUF[9], SHIFTBUF[7], SHIFTBUF[5], SHIFTBUF[3], SHIFTBUF[1] }.

Chapter 68
Keypad Port (KPP)

68.1 Chip-specific KPP information

Table 68-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

68.2 Overview

The Keypad Port (KPP) is a 16-bit peripheral that can be used as a keypad matrix interface or as general purpose input/output (I/O). The figure below shows the KPP block diagram. The KPP provides interface for the keypad matrix with 2-point contact or 3-point contact keys. The KPP is designed to simplify the software task of scanning a keypad matrix. With appropriate software support, the KPP is capable of detecting, debouncing, and decoding one or multiple keys pressed simultaneously on the keypad.

68.2.1 Block diagram

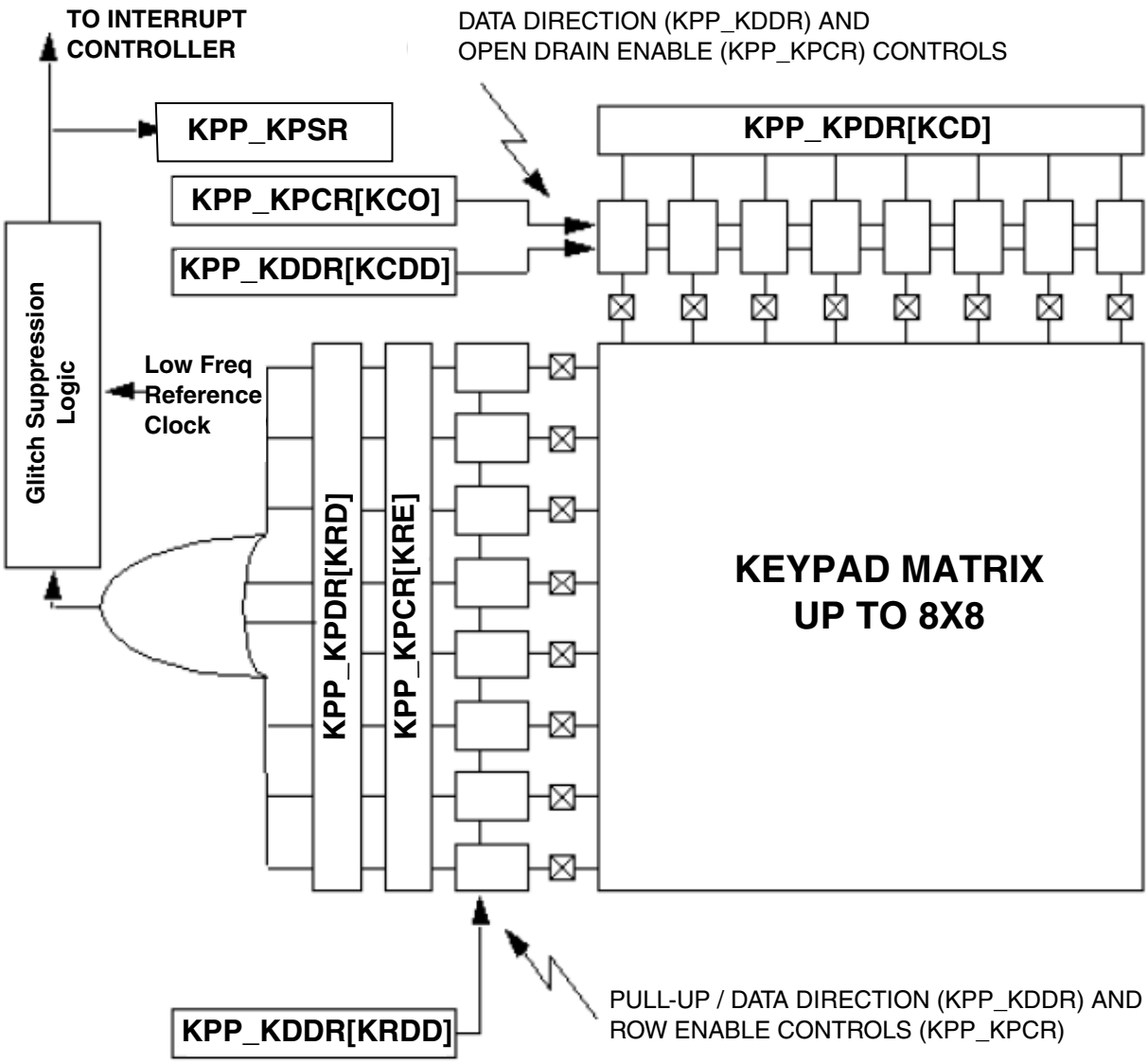


Figure 68-1. KPP Peripheral Block Diagram

68.2.2 Features

The KPP includes these distinctive features:

- Supports up to an 8 x 8 external key pad matrix
- Port pins can be used as general purpose I/O
- Open drain design
- Glitch suppression circuit design
- Multiple-key detection

- Long key-press detection
- Standby key-press detection
- Synchronizer chain clear
- Supports a 2-point and 3-point contact key matrix

68.3 Functional Description

The Keypad Port (KPP) is designed to simplify the software task of scanning a keypad matrix. With appropriate software support and matrix organization, the KPP is capable of detecting, debouncing, and decoding one or more keys pressed simultaneously on the keypad.

Logic in the KPP is capable of detecting a key press even while the processor is in one of the low power standby modes provided that a low frequency reference clock (ipg_clk_32k) is on. The KPP may generate an ARM platform interrupt any time a key press or key release is detected. This interrupt is capable of forcing the processor out of a low power mode.

68.3.1 Modes and Operations

This block supports the following modes:

- Run Mode-This is the normal functional mode in which the KPP can detect any key press event.
- Low Power Mode-The keypad can detect any key press even in low power modes (when there is no MCU clock).

68.3.2 Keypad Matrix Construction

The KPP is designed to interface to a keypad matrix, which shorts the intersecting row and column lines together whenever a key is depressed. The interface is not optimized for any other switch configuration.

68.3.3 Keypad Port Configuration

The software must initialize the KPP for the size of the keypad matrix. Pins connected to the keypad columns should be configured as open-drain outputs. Pins connected to the keypad rows should be configured as inputs. On-chip, pull-up resistors should be implemented for active keypad rows.

In addition to enabled row inputs in the Keypad Control register, corresponding interrupt (depress or/and release) must also be enabled to generate an interrupt.

Discrete switches that are not part of the matrix may be connected to any unused row inputs. The second terminal of the discrete switch is connected to ground. The hardware detects closures of these switches without the need for software polling.

68.3.4 Keypad Matrix Scanning

Keypad scanning is performed by a software loop that walks a zero across each of the keypad columns, reading the value on the rows at each step. The process is repeated several times in succession, with the results of each pass optionally compared to those from the previous pass. When several consecutive scans yield the same key closures, a valid key press has been detected. Software then can decode exactly which switch was depressed and pass the value up to the next software layer.

The KPP may be used to control the debounce period, but it must be defined in software by the user's application. The basic period is the period between the scan of two consecutive columns, so the debouncing time between two consecutive scans of the whole matrix shall be the number of columns multiplied by the basic period.

68.3.5 Keypad Standby

There is no need for the ARM platform to continually scan the keypad. Between key presses, the keypad can be left in a state that requires no software intervention until the next key press is detected. To place the keypad in a standby state, software should write all column outputs low. Row inputs are left enabled. At this point, the ARM platform can attend to other tasks or revert to a low power standby mode. The KPP can interrupt the ARM platform if any key is pressed.

Upon receiving a keypad interrupt, the ARM platform should set all the column strobes high, and begin a normal keypad scanning routine to determine which key was pressed. It is important that open-drain drivers be used when scanning to prevent a possible DC path between power and ground through two or more switches.

68.3.6 Glitch Suppression on Keypad Inputs

A glitch suppression circuit qualifies the keypad inputs to prevent noise from inadvertently interrupting the ARM platform. The circuit is a 4-state synchronizer clocked from a low frequency reference clock (ipg_clk_32k) source. This clock must continue to run in any low power mode where the keypad is a wake-up source, as the ARM platform interrupt is generated from the synchronized input. An interrupt is not generated until all four synchronizer stages have latched a valid key assertion. This guarantees the filtering out of any noise less than three clock periods in duration of a low frequency reference clock. Noise filtering of the duration between three to four clock periods cannot be guaranteed. The interrupt output is latched in an S-R latch and remains asserted until cleared by the software. The Set input of the latch is rising-edge clocked. See the figure below.

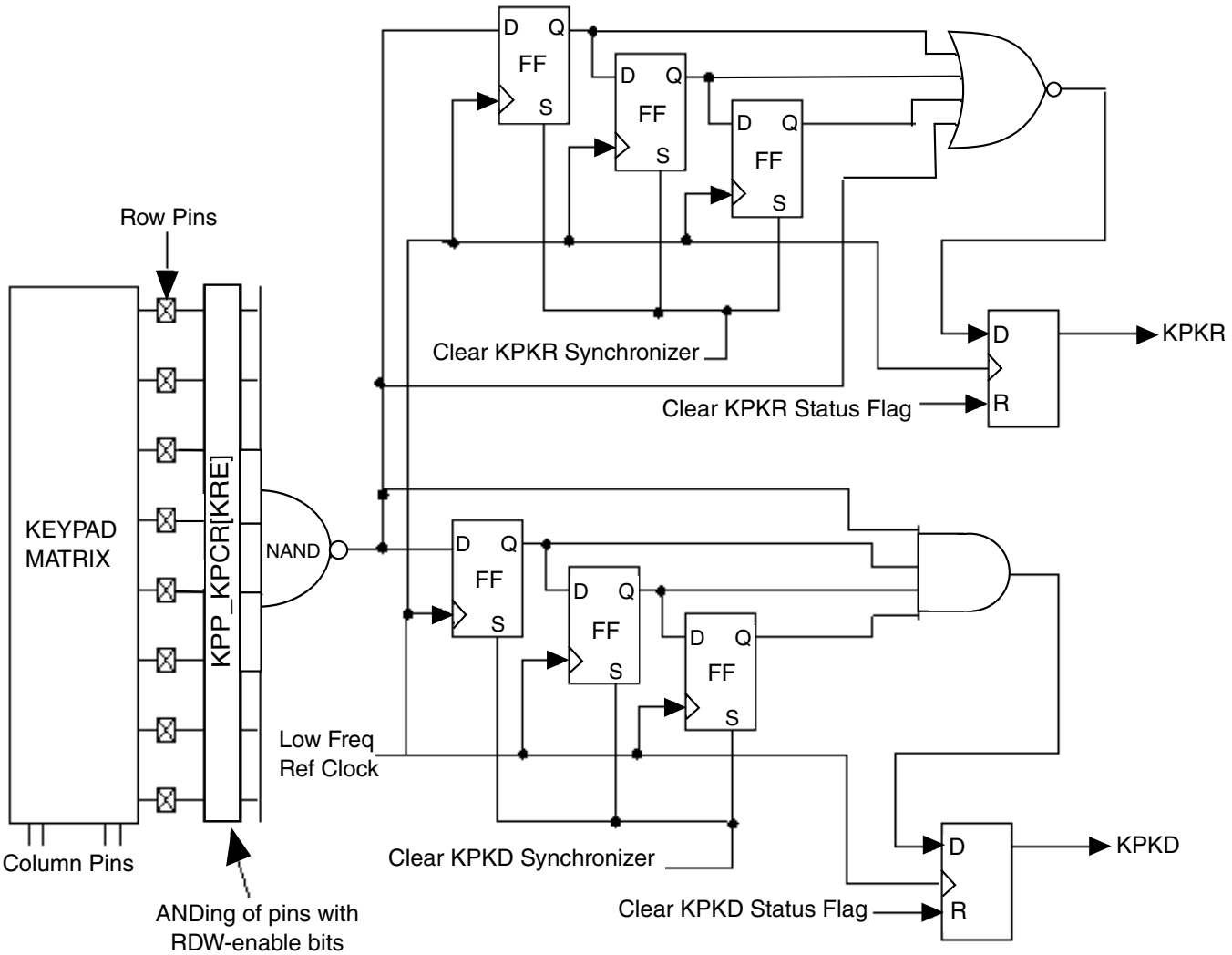


Figure 68-2. Keypad Synchronizer Functional Diagram

68.3.7 Multiple Key Closures

Using the key press and key release interrupts, the software can detect multiple keys or achieve n key rollover. The key scanning routine can be programmed accordingly (See Initialization/Application Information for more information).

The following figures illustrate the interface of a 2-contact keypad matrix with the KPP controller. With proper enabling of row lines and the performing scan-routine, multiple key presses can be detected. When keys present on the same row are pressed, corresponding row lines (multiple lines) become low when the column is driven low during a scan-routine. By reading the data-register, pressed keys can be detected.

Similarly, when keys present on same column line are pressed, the corresponding row line (only one line) becomes low when logic "0" is driven on the column line during a scan-routine.

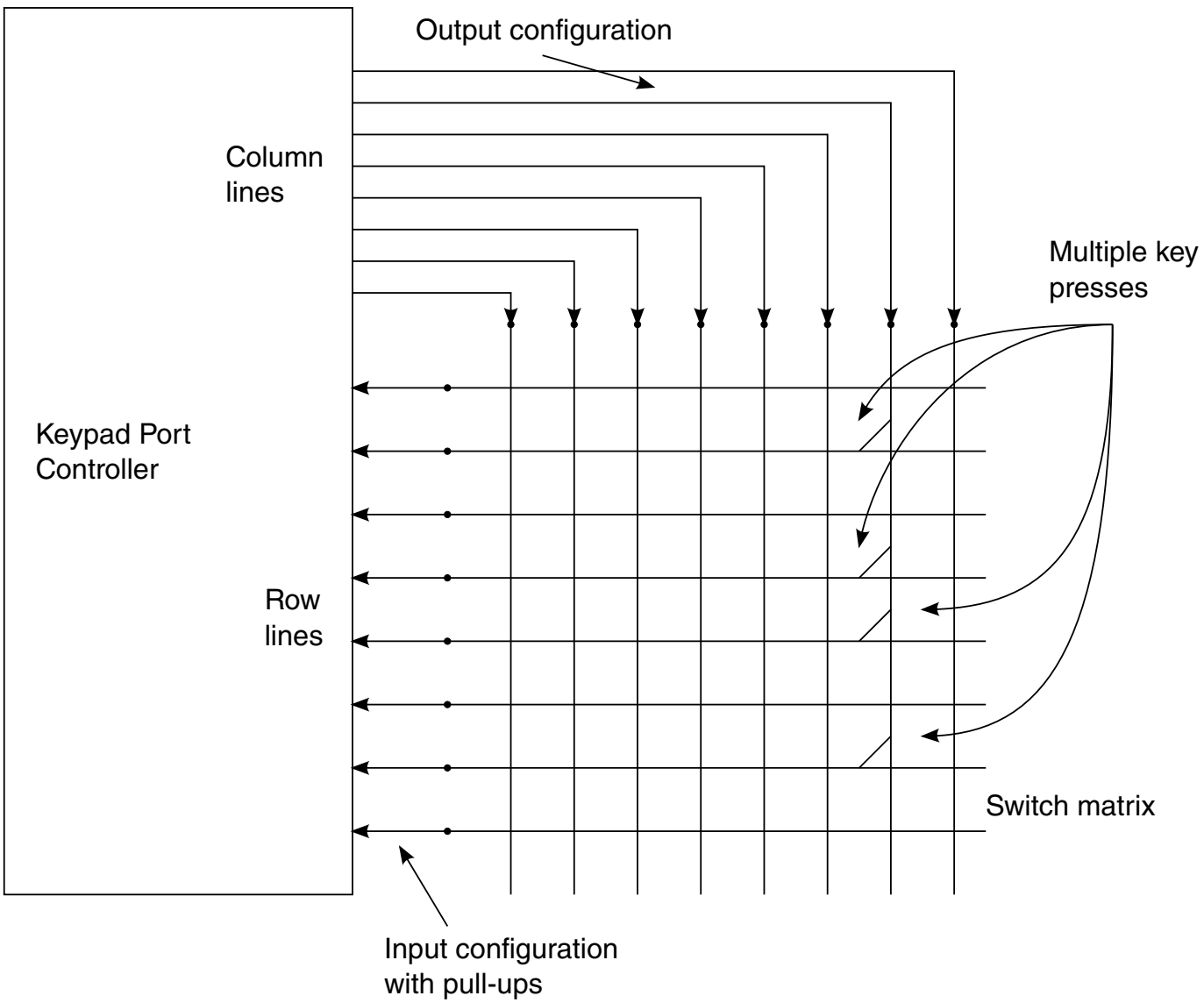


Figure 68-3. Multiple Key Presses on Same Column Line (Simplified View)

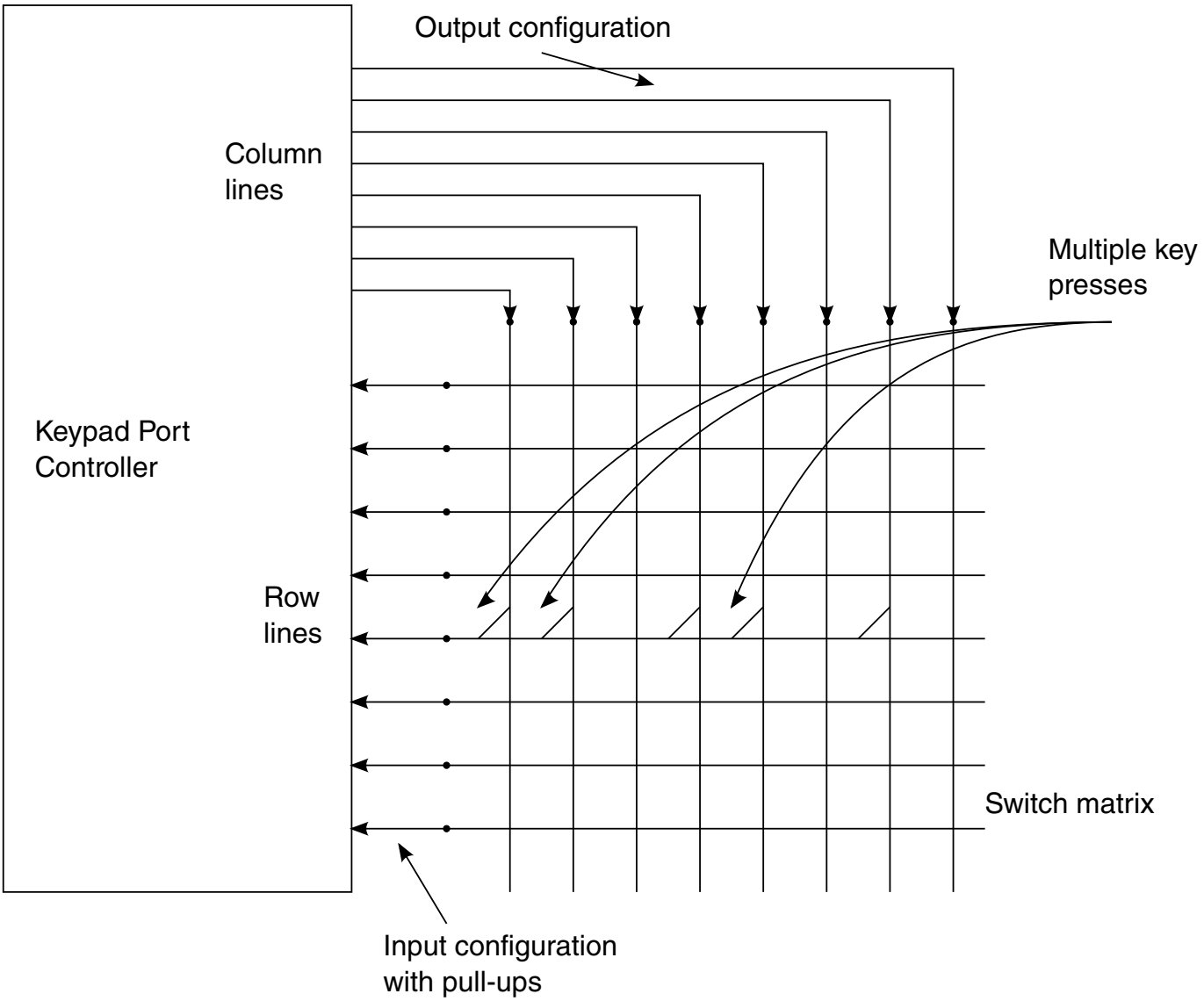


Figure 68-4. Multiple Key Presses on Same Row Line (Simplified View)

NOTE

An n key rollover is a technique with which the system can recognize the order in which keys are pressed.

68.3.7.1 Ghost Key Problem and Correction

The KPP detects if one or multiple keys are pressed or released. In the case where a simple keypad matrix with two-contact switches is used, there is a chance of "ghost" key detection when three or more keys are pressed. This is a limitation imposed by such a keypad matrix.

As seen in Figure 68-5, three keys pressed simultaneously can cause a short between the column currently "scanned" by the software and another column. Depending on the location of the third key pressed, a "ghost" key press may be detected.

However, this can be corrected by using a keypad matrix that provides "ghost" key protection. Such a matrix implements a one-way "diode" at all keypad points between rows and columns. This way, the multiple pressing of three keys will not cause a short at a fourth key (see Figure 68-6).

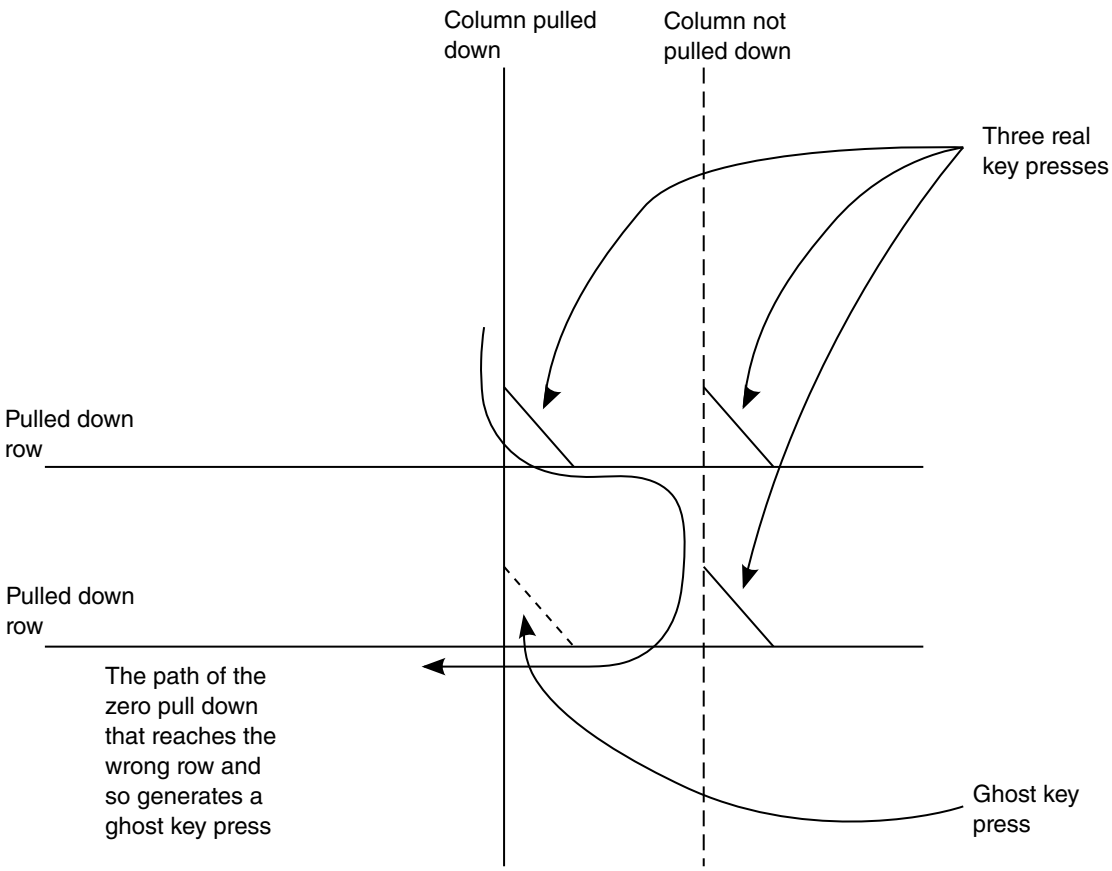


Figure 68-5. Decoding Wrong Three- Key-Presses

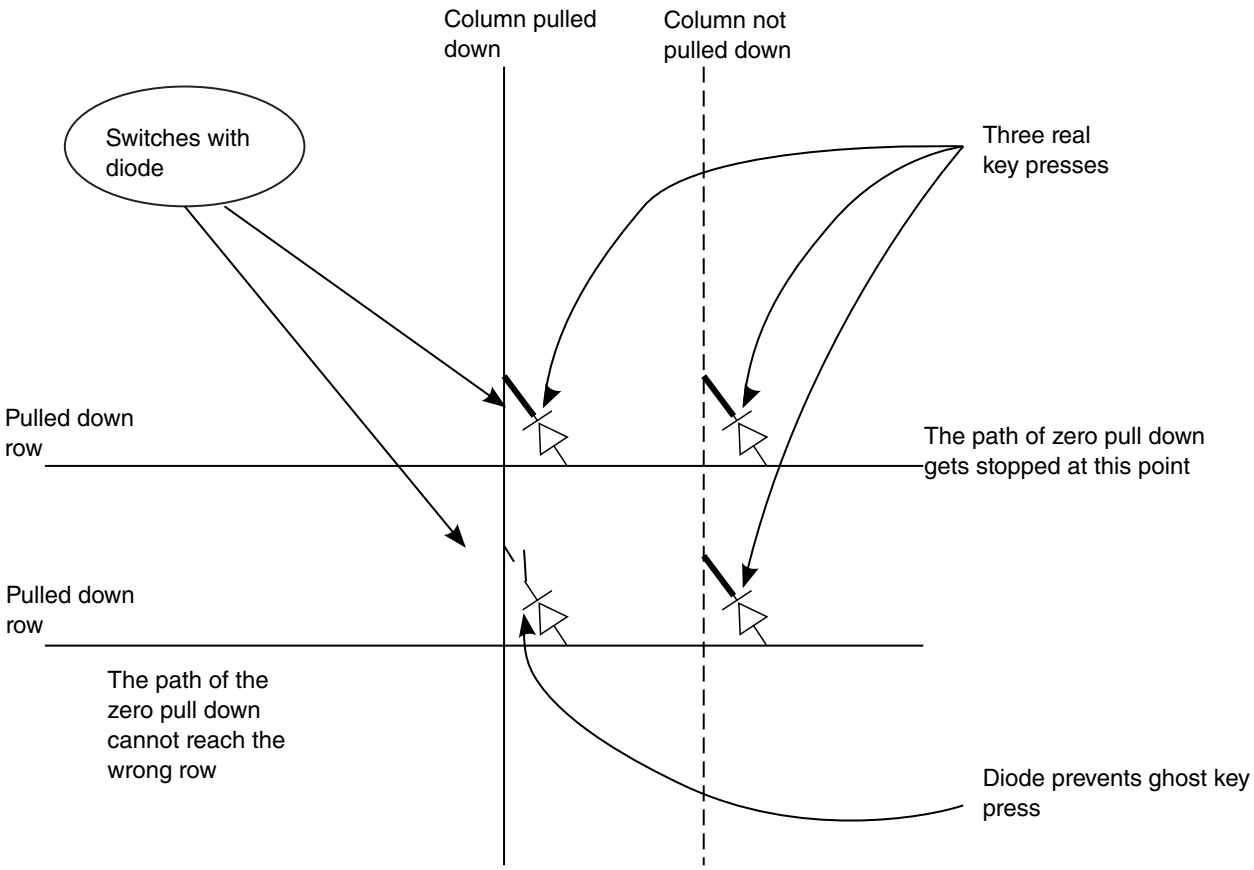


Figure 68-6. Matrix with "Ghost" Key Protections

68.3.8 3-Point Contact Keys Support

The KPP supports interfacing to a matrix consisting of 3-point contact keys. As shown in Figure 68-7, two points of such a key are connected to keypad lines, while a third point is connected to ground (low logic).

The keypad lines should be configured as input and a pull-up should to be present on these lines. When such a key is pressed, corresponding keypad lines go low and an interrupt is generated. There is no need to perform a scanning routine for identification of pressed key as it can be done by reading the keypad data-register. A limitation with such a matrix is that for every key at least one keypad row line should be used.

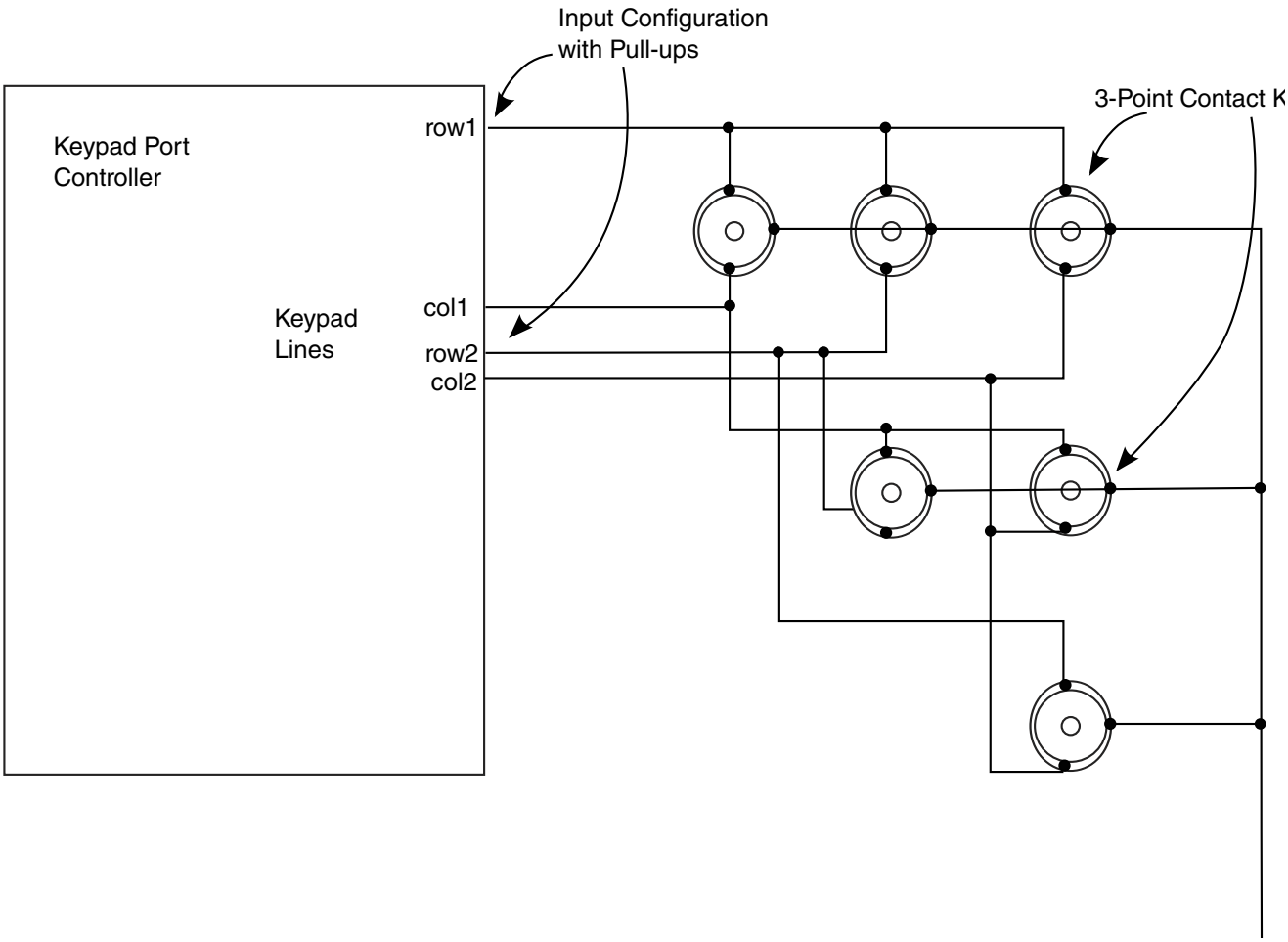


Figure 68-7. KPP Interface with 3-point Contact Key Matrix (Simplified View)

68.3.9 Clocks

The table found here describes the clock sources for KPP. Please see clock control block for clock setting, configuration and gating information.

Table 68-2. KPP Clocks

Clock name	Description
ipg_clk_32k	Low-frequency reference clock (32 kHz)
ipg_clk_s	Peripheral access clock

68.4 External Signals

There are several pins dedicated to the KPP. Keypads of any configuration up to eight rows and eight columns are supported through the software configuration of the peripheral pins. Any pins not used for the keypad are available as general purpose I/O. The registers are configured such that the pins can be treated as an I/O port up to 16 bits wide. Please see IOMUX Controller for KPP pin mux assignments.

See the table below for the list of external signals.

Table 68-3. KPP External Signals

Name	Direction	Function	Reset State	Pull-Up
KEY_COL[7:0]	IO	Column input or output pin, from chip	0	Active[1]
KEY_ROW[7:0]	IO	Row input or output pin, from chip	1	Active[1]

1. The corresponding pads are required to be pull-up enabled.

68.4.1 Input Pins

Any of the 16 pins associated with the KPP can be configured as inputs by writing a "0" to the appropriate bits in the KPP_KDDR. Additionally, the least significant 8 bits (ROW inputs) corresponding to KPP_KDDR[KRDD] have internal pull-ups, which are enabled when the pin is used as an input.

68.4.2 Output Pins

Any of the 16 pins associated with the KPP can be configured as outputs by writing the appropriate bits in the KPP_KDDR to a "1". Additionally, the 8 most significant bits (15-8) can be designated as open drain outputs by writing a "1" to the appropriate bits in the KPP_KPCR. The lower 8 bits (7-0) are always in "totem pole" style, driven when configured as outputs.

See the table below.

Table 68-4. Keypad Port Column Modes

KPP_KDDR (15:8)	KPP_KPCR (15:8)	Pin Function
0	x	Input
1	0	Totem-Pole Output
1	1	Open-Drain Output

NOTE

Totem pole configuration helps for a faster discharge of keypad capacitance when all columns need to be quickly brought to a "1" during the scan routine. With this configuration, delay between the scanning of two subsequent columns is reduced. Totem pole mode is to be used temporarily on the column pins to discharge the lines only, but should be switched to open-drain mode for normal operation.

68.4.3 Generation of Transfer Error Signal on Peripheral Bus

If there is an access to an address which is not implemented, then the KPP asserts a transfer error signal on Peripheral Bus.

68.5 Initialization/Application Information

68.5.1 Typical Keypad Configuration and Scanning Sequence

Perform the following steps to configure the keypad:

1. Enable the number of rows in the keypad (KPP_KPCR[KRE]).
2. Write 0s to KPP_KPDR[KCD].
3. Configure the keypad columns as open-drain (KPP_KPCR[KCO]).
4. Configure columns as output (KPP_KDDR[KCDD]) and rows as input (KPP_KDDR[KRDD]).
5. Clear the KPKD Status Flag and Synchronizer chain.
6. Set the KDIE control bit, and clear the KRIE control bit (avoid false release events).
7. (The system is now in standby mode, and awaiting a key press.)

68.5.2 Key Press Interrupt Scanning Sequence

Perform the following steps to perform a keypad scanning routine:

1. Disable both (depress and release) keypad interrupts.
2. Write 1s to KPP_KPDR[KCD], setting column data to 1s.
3. Configure columns as totem pole outputs (for quick discharging of keypad capacitance).

4. Configure columns as open-drain.
5. Write a single column to 0, and other columns to 1.
6. Sample row inputs and save data. Multiple key presses can be detected on a single column.
7. Repeat Steps 2-6 for remaining columns.
8. Return all columns to 0 in preparation for standby mode.
9. Clear KPKD and KPKR status bit(s) by writing to a "1"; set the KPKR synchronizer chain by writing a "1" to the KPP_KRSS register; and clear the KPKD synchronizer chain by writing a "1" to the KDSC register.
10. Re-enable the appropriate keypad interrupt(s) so that the KDIE detects a key hold condition, or the KRIE detects a key-release event.

68.5.3 Additional Comments

The order of key press detection can be done in software only. Therefore, the software may need to run the scan routines at very short intervals of time per the application's demands. The reason that such functionality cannot be put in the KPP is that the block is limited by the number of external pins.

For the keys that require a very precise order (such as game keys), individual GPIO pins may be more useful.

68.6 Memory Map and Register Definition

68.6.1 KPP register descriptions

The KPP contains four registers.

68.6.1.1 KPP memory map

KPP base address: 400E_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Keypad Control Register (KPCR)	16	RW	0000
2	Keypad Status Register (KPSR)	16	RW	0400
4	Keypad Data Direction Register (KDDR)	16	RW	0000
6	Keypad Data Register (KPDR)	16	RW	0000

68.6.1.2 Keypad Control Register (KPCR)

The Keypad Control Register determines which of the eight possible column strobes are to be open drain when configured as outputs, and which of the eight row sense lines are considered in generating an interrupt to the core.

It is up to the programmer to ensure that pins being used for functions other than the keypad are properly disabled. The KPP_KPCR register is byte- or half-word-addressable.

68.6.1.2.1 Offset

Register	Offset
KPCR	0h

68.6.1.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCO								KRE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

68.6.1.2.3 Fields

Field	Description
15-8 KCO	KCO Keypad Column Strobe Open-Drain Enable. Setting a column open-drain enable bit (KCO7-KCO0) disables the pull-up driver on that pin. Clearing the bit allows the pin to drive to the high state. This bit has no effect when the pin is configured as an input. **NOTE:** Configuration of external port control logic (for example, IOMUX) should be done properly so that the KPP controls an open-drain enable of the pin.

Table continues on the next page...

Field	Description
	00000000 - Column strobe output is totem pole drive. 00000001 - Column strobe output is open drain.
7-0 KRE	KRE Keypad Row Enable. Setting a row enable control bit in this register enables the corresponding row line to participate in interrupt generation. Likewise, clearing a bit disables that row from being used to generate an interrupt. This register is cleared by a reset, disabling all rows. The row-enable logic is independent of the programmed direction of the pin. Writing a "0" to the data register of the pins configured as outputs will cause a keypad interrupt to be generated if the row enable associated with that bit is set. 00000000 - Row is not included in the keypad key press detect. 00000001 - Row is included in the keypad key press detect.

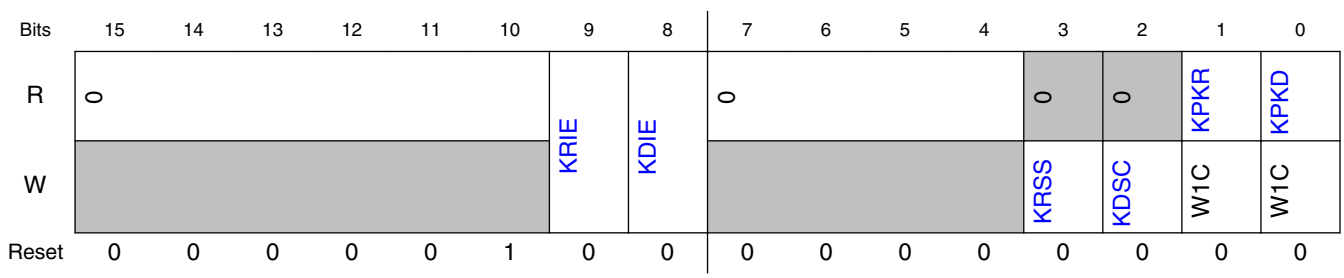
68.6.1.3 Keypad Status Register (KPSR)

The Keypad Status Register reflects the state of the key press detect circuit. The KPP_KPSR register is byte- or half-word-addressable.

68.6.1.3.1 Offset

Register	Offset
KPSR	2h

68.6.1.3.2 Diagram



68.6.1.3.3 Fields

Field	Description
15-10	-

Table continues on the next page...

Field	Description
—	Reserved
9 KRIE	KRIE Keypad Release Interrupt Enable. The software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 - No interrupt request is generated when KPKR is set. 1 - An interrupt request is generated when KPKR is set.
8 KDIE	KDIE Keypad Key Depress Interrupt Enable. Software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice-versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 - No interrupt request is generated when KPKD is set. 1 - An interrupt request is generated when KPKD is set.
7-4 —	- Reserved, should be cleared
3 KRSS	KRSS Key Release Synchronizer Set. Self-clear bit. The Key release synchronizer is set by writing a logic one into this bit. Reads return a value of "0". 0 - No effect 1 - Set bits which sets keypad release synchronizer chain
2 KDSC	KDSC Key Depress Synchronizer Clear. Self-clear bit. The Key depress synchronizer is cleared by writing a logic "1" into this bit. Reads return a value of "0". 0 - No effect 1 - Set bits that clear the keypad depress synchronizer chain
1 KPKR	KPKR Keypad Key Release. The keypad key release (KPKR) status bit is set when all enabled rows are detected high after synchronization (the KPKR status bit will be set when cleared by a reset). The KPKR bit may be used to generate a maskable key release interrupt. The key release synchronizer may be set high by software after scanning the keypad to ensure a known state. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents. Reset value of register is "0" as long as reset is asserted. However when reset is de-asserted, the value of the register depends upon the external row pins and can become "1". 0 - No key release detected 1 - All keys have been released
0	KPKD

Field	Description
KPKD	Keypad Key Depress. The keypad key depress (KPKD) status bit is set when one or more enabled rows are detected low after synchronization. The KPKD status bit remains set until cleared by the software. The KPKD bit may be used to generate a maskable key depress interrupt. If desired, the software may clear the key press synchronizer chain to allow a repeated interrupt to be generated while a key remains pressed. In this case, a new interrupt will be generated after the synchronizer delay (4 cycles of the low frequency reference clock (ipg_clk_32k) elapses if a key remains pressed. This functionality can be used to detect a long key press. This allows detection of additional key presses of the same key or other keys. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by the software prior to the system exiting the state it represents. 0 - No key presses detected 1 - A key has been depressed

68.6.1.4 Keypad Data Direction Register (KDDR)

The bits in the KPP_KDDR control the direction of the keypad port pins. The upper eight bits in the register affect the pins designated as column strobes, while the lower eight bits affect the row sense pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding port pin as an input. For the Keypad Row DDR, an internal pull-up is enabled if the corresponding bit is clear. This register is cleared by a reset, configuring all pins as inputs. The KPP_KDDR register is byte- or half-word addressable.

NOTE

When a pin is used as row pin for keypad purposes, all corresponding pull-ups should be enabled at the upper level (for example, IOMUX) when the bit in KRDD is cleared.

68.6.1.4.1 Offset

Register	Offset
KDDR	4h

68.6.1.4.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCDD								KRDD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

68.6.1.4.3 Fields

Field	Description
15-8 KCDD	KCDD Keypad Column Data Direction Register. Setting a bit configures the corresponding COL*n* pin as an output (where *n* = 7 through 0). 00000000 - COLn pin is configured as an input. 00000001 - COLn pin is configured as an output.
7-0 KRDD	KRDD Keypad Row Data Direction. Setting a bit configures the corresponding ROW*n* pin as an output (where *n* = 7 through 0). 00000000 - ROWn pin configured as an input. 00000001 - ROWn pin configured as an output.

68.6.1.5 Keypad Data Register (KPDR)

This 16-bit register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPP_KPDR register is byte- or half-word addressable. This register is not initialized by a reset. Valid data should be written to this register before any bits are configured as outputs.

68.6.1.5.1 Offset

Register	Offset
KPDR	6h

68.6.1.5.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KCD								KRD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

68.6.1.5.3 Fields

Field	Description
15-8 KCD	KCD Keypad Column Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to column ports 1 Read/Write "1" from/to column ports
7-0 KRD	KRD Keypad Row Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to row ports 1 Read/Write "1" from/to row ports

Chapter 69
Low Power Inter-Integrated Circuit (LPI2C)

69.1 Chip-specific LPI2C information

Table 69-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

For LPI2C on this device, only LPI2C1 fully supports the 5 wires, while other instances (LPI2C2, ...) have only 2 wires.

69.2 Overview

The LPI2C is a low power Inter-Integrated Circuit (I2C) module that supports an efficient interface to an I^2C bus as a master and/or as a slave.

- Implements logic support for standard-mode, fast-mode, fast-mode plus and ultra-fast modes of operation.
- Uses little CPU overhead, with DMA offloading of FIFO register accesses.
- Continues operating in stop modes if an appropriate clock is available.

The LPI2C module also complies with the System Management Bus (SMBus) Specification, version 3. The SMBus is a single-ended simple two-wire bus, which is typically used for low bandwidth communications.

NOTE

The I^2C (Inter-Integrated Circuit) serial bus is multi-master, multi-slave, packet-switched, and single-ended, and is often used to attach microcontroller ICs to lower-speed peripheral ICs.

69.2.1 Block diagram

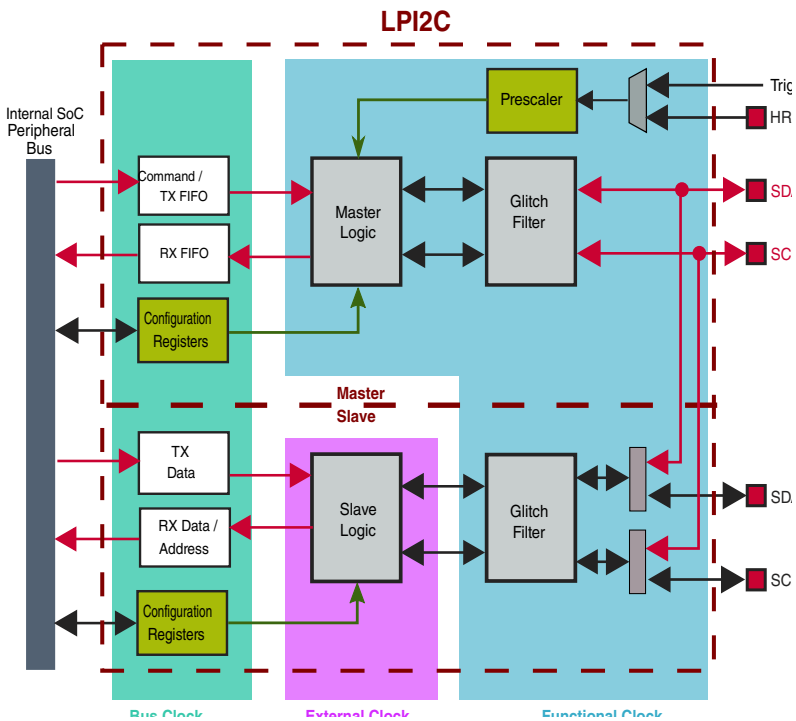


Figure 69-1. LPI2C block diagram

69.2.2 Features

The LPI2C supports:

- Standard, Fast, Fast+ and Ultra Fast modes are supported
- High speed mode (HS) in slave mode
- Multi-master support, including synchronization and arbitration. Multi-master means any number of master nodes can be present. Additionally, master and slave roles may be changed between messages (after a STOP is sent).

- Clock stretching: Sometimes multiple I2C nodes may be driving the lines at the same time. If any I2C node is driving a line low, then that line will be low. I2C nodes that are starting to transmit a logical one (by letting the line float high) can detect that the line is low, and thereby know that another I2C node is active at the same time.
  - When node detection is used on the SCL line, it is called *clock stretching*, and clock stretching is used as a I2C flow control mechanism for multiple slaves.
  - When node detection is used on the SDA line, it is called *arbitration*, and arbitration ensures that there is only one I2C node transmitter at a time.
- General call, 7-bit and 10-bit addressing
- Software reset, START byte and Device ID (also require software support)

The LPI2C master supports:

- Command/transmit FIFO of 4words (8-bit transmit data + 3-bit command).
- Receive FIFO of 4words (8-bit receive data).
- Command FIFO will wait for idle I2C bus before initiating transfer
- Command FIFO can initiate (repeated) START and STOP conditions and one or more master-receiver transfers
- STOP condition can be generated from command FIFO, or generated automatically when the transmit FIFO is empty
- Host request input to control the start time of an I2C bus transfer
- Flexible receive data match can generate interrupt on data match and/or discard unwanted data
- Flag and optional interrupt to signal Repeated START condition, STOP condition, loss of arbitration, unexpected NACK, and command word errors
- Supports configurable bus idle timeout and pin-stuck-low timeout

The LPI2C slave supports:

- Separate I2C slave registers to minimize software overhead because of master/slave switching
- Support for 7-bit or 10-bit addressing, address range, SMBus alert and general call address
- Transmit data register that supports interrupt or DMA requests
- Receive data register that supports interrupt or DMA requests
- Software-controllable ACK or NACK, with optional clock stretching on ACK/NACK bit
- Configurable clock stretching, to avoid transmit FIFO underrun and receive FIFO overrun errors
- Flag and optional interrupt at end of packet, STOP condition, or bit error detection

69.3 Functional description

69.3.1 Master Mode

The LPI2C master logic operates independently from the slave logic to perform all master mode transfers on the I2C bus.

69.3.1.1 Transmit and Command FIFO commands

The transmit FIFO stores command data to initiate the various I2C operations. The following operations can be initiated through commands in the transmit FIFO:

- START or Repeated START condition with address byte and expecting ACK or NACK.
- Transmit data (this is the default for zero extended byte writes to the transmit FIFO).
- Receive 1-256 bytes of data (can also be configured to discard receive data and not store in receive FIFO).
- STOP condition (can also be configured to send STOP condition when transmit FIFO is empty).

Multiple transmit and receive commands can be inserted between the START condition and STOP conditon; transmit and receive commands must not be interleaved (to comply with the I2C specification). The receive data command and the receive data and discard commands can be interleaved, to ensure that only the desired received data is stored in the receive FIFO (or compared with the data match logic).

The LPI2C master automatically transmits a NACK on the last byte of a receive data command unless the next command in the FIFO is also a receive data command. A NACK is also automatically transmitted if the transmit FIFO is empty when a receive data command completes.

The LPI2C master supports 10-bit addressing through a (repeated) START condition, followed by a transmit data byte containing the second address byte, followed by any number of data bytes with the master-transmit data.

A START or Repeated START condition that is expecting a NACK (for example,HS-mode master code) must be followed by a STOP or (repeated) START condition.

69.3.1.2 Master operations

Whenever the LPI2C is enabled, it monitors the I2C bus to detect when the I2C bus is idle (MSR[BBF]). The I2C bus is no longer considered idle if either SCL or SDA are low, and the I2C bus becomes idle if a STOP condition is detected or if a bus idle timeout is detected (as configured by MCFGR2[BUSIDLE]). After the I2C bus is idle, the transmit FIFO is not empty, and the host request is either asserted or disabled, then the LPI2C master initiates a transfer on the I2C bus. This involves the following steps:

- Wait the bus idle time equal to (MCCR0[CLKLO] + 1) multiplied by the prescaler (MCFGR1[PRESCALE]).
- Transmit a START condition and address byte using the timing configuration in Master Clock Configuration 0 (MCCR0); if a high speed mode transfer is configured, then the timing configuration from Master Clock Configuration 1 (MCCR1) is used instead.
- Perform master-transmit or master-receive transfers, as configured by the transmit FIFO.
- Transmit NACK on the last byte of a master-receive transfer, unless the next command in the transmit FIFO is also a receive data command and the transmit FIFO is not empty.
- Transmit a Repeated START or STOP condition as configured by the transmit FIFO and/or MCFGR1[AUTOSTOP]. A repeated START can change which timing configuration register is used.

When the LPI2C master is disabled (either due to MCR[MEN] being clear or automatically due to mode entry), the LPI2C continues to empty the transmit FIFO until a STOP condition is transmitted. However, the LPI2C will no longer stall the I2C bus waiting for the transmit or receive FIFO, and after the transmit FIFO is empty, the LPI2C generates a STOP condition automatically.

The LPI2C master can stall the I2C bus under certain conditions; this results in SCL pulled low continuously on the first bit of a byte, until the condition is removed:

- LPI2C master is enabled and busy, the transmit FIFO is empty, and MCFGR1[AUTOSTOP] is clear.
- LPI2C master is enabled and receiving data, receive data is not being discarded (due to command or receive data match), and the receive FIFO is full.

69.3.1.3 Receive FIFO and Data Matching

The receive FIFO is used to store receive data during master-receiver transfers. Receive data can also be configured to discard receive data instead of storing in the receive FIFO; this is configured by the command word in the transmit FIFO.

Receive data supports a receive data match function that can match received data against one of two bytes or against a masked data byte. The data match function can also be configured to compare only the first one or two received data words since the last (repeated) START condition. Receive data that is already discarded due to the command word cannot cause the data match to set, and delaya the match on the first received data word until after the discarded data is received. The receiver match function can also be configured to discard all receive data until a data match is detected, using the MCFGR0[RDMO] control bit. When clearing the MCFGR0[RDMO] control bit following a data match, clear MCFGR0[RDMO] before clearing MSR[DMF], to allow all subsequent data to be received.

69.3.1.4 Timing Parameters

The following timing parameters can be configured by the LPI2C master. Parameters are configured separately for high speed mode (Master Clock Configuration 1 (MCCR1)) and other modes (Master Clock Configuration 0 (MCCR0)). This allows the high speed mode master code to be sent using the regular timing parameters, and then switch to the high speed mode timing (following a repeated START) until the next STOP condition.

The LPI2C master timing parameters in LPI2C functional clock cycles are configured as follows. They must be configured to meet the I2C timing specification for the required mode.

Table 69-2. Timing Parameters

I2C Specification Timing Parameter	I2C Specification Timing Symbol	LPI2C Timing Parameter (LPI2C functional clock cycles)
SCL clock period	tSCL	(CLKHI + CLKLO + 2 + SCL_LATENCY) x (2 ^ PRESCALE)
hold time (repeated) START condition	tHD:STA	(SETHOLD +1) x (2 ^ PRESCALE)
LOW period of the SCL clock	tLOW	(CLKLO + 1) x (2 ^ PRESCALE)
HIGH period of the SCL clock	tHIGH	(CLKHI + 1 + SCL_LATENCY) x (2 ^ PRESCALE)
setup time for a repeated START condition or STOP condition	tSU:STA, tSU:STO	(SETHOLD + 1 + SCL_LATENCY) x (2 ^ PRESCALE)
data hold time	tHD:DAT	(DATAVD + 1) x (2 ^ PRESCALE)
data setup time	tSU:DAT	(SDA_LATENCY + 1) x (2 ^ PRESCALE)

Table continues on the next page...

Table 69-2. Timing Parameters (continued)

I2C Specification Timing Parameter	I2C Specification Timing Symbol	LPI2C Timing Parameter (LPI2C functional clock cycles)
bus free time between a STOP and START condition	tBUF	(CLKLO + 1 + SDA_LATENCY) x (2 ^ PRESCALE)
data valid time, data valid acknowledge time	tVD:DAT, tVD:ACK	(DATAVD + 1) x (2 ^ PRESCALE)

The latency parameters are defined in the following table, these parameters assume the risetime is less than one LPI2C functional clock cycle. The risetime depends on a number of factors, including the I/O propagation delay, the I2C bus loading and the external pull-up resistor sizing. A larger rise time increasea the number of cycles that the signal takes to propagate through the synchronizer (and glitch filter), which increases the latency.

Table 69-3. Synchronization Latency

Timing Parameter	Timing Definition
SCL_LATENCY	ROUNDDOWN ((2 + FILTSCL + SCL_RISETIME) / (2 ^ PRESCALE))
SDA_LATENCY	ROUNDDOWN ((2 + FILTSDA + SDA_RISETIME) / (2 ^ PRESCALE))

The following timing restrictions must be enforced to avoid unexpected START or STOP conditions on the I2C bus, or to avoid unexpected START or STOP conditions detected by the LPI2C master. The timing restrictions can be summarized as **SDA cannot change when SCL is high outside of a transmitted (repeated) START or STOP condition**.

Table 69-4. LPI2C Timing Parameter Restrictions

Timing Parameter	Minimum	Maximum	Comment
CLKLO	0x03	-	Also: CLKLO x (2 ^ PRESCALE) > SCL_LATENCY
CLKHI	0x01	-	Configure CLKHI to meet the duty cycle requirements in the I2C specification
SETHOLD	0x02	-	Also: SETHOLD x (2 ^ PRESCALE) > SDA_LATENCY
DATAVD	0x01	CLKLO - SDA_LATENCY - 1	Configure DATAVD to meet the data hold requirement in the I2C specification
FILTSCL	0x00	[CLKLO x (2 ^ PRESCALE)] - 3	FILTSCL and FILTSDA are the only parameters not multiplied by (2 ^ PRESCALE)
FILTSDA	FILTSCL	[CLKLO x (2 ^ PRESCALE)] - 3	Configuring FILTSDA greater than FILTSCL can delay the SDA input to compensate for board level skew
BUSIDLE	(CLKLO+SETHOLD+2) x 2	-	Must also be greater than (CLKHI+1)

The timing parameters must be configured to meet the requirements of the I2C specification; this depends on the mode being supported and the LPI2C functional clock frequency. When switching between two modes using the different clock configuration registers (for example, Fast and HS-mode), the PRESCALE factor must remain constant between the modes. Some example timing configurations are provided below.

Table 69-5. LPI2C Example Timing Configurations

I2C Mode	Clock Frequency	Baud Rate	PRESCALE	FILTSCL / FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Fast	8 MHz	400 kbps	0x0	0x0/0x0	0x04	0x0B	0x05	0x02
Fast+	8 MHz	1 Mbps	0x0	0x0/0x0	0x02	0x03	0x01	0x01
Fast	48 MHz	400 kbps	0x0	0x1/0x1	0x1D	0x3E	0x35	0x0F
Fast	48 MHz	400 kbps	0x2	0x1/0x1	0x07	0x11	0x0B	0x03
Fast+	48 MHz	1 Mbps	0x2	0x1/0x1	0x03	0x06	0x04	0x04
HS-mode	48 MHz	3.2 Mbps	0x0	0x0/0x0	0x07	0x08	0x03	0x01
Fast	60 MHz	400 kbps	0x1	0x2/0x2	0x11	0x28	0x1F	0x08
Fast+	60 MHz	1 Mbps	0x1	0x2/0x2	0x07	0x0F	0x0B	0x01
HS-mode	60 MHz	3.33 Mbps	0x1	0x0/0x0	0x04	0x04	0x02	0x01

69.3.1.5 Error Conditions

The LPI2C master monitors for errors while it is active, the following conditions generates an error flag and block a new START condition from being sent, until the flag is cleared by software:

- A START or STOP condition is detected and is not generated by the LPI2C master (sets MSR[ALF]).
- Transmitting data on SDA and different values are being received (sets MSR[ALF]).
- NACK is detected when transmitting data, and MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- NACK is detected and is expecting ACK for the address byte, and MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- ACK is detected and is expecting NACK for the address byte, and MCFGR1[IGNACK] is clear (sets MSR[NDF]).
- Transmit FIFO is requesting to transmit or receive data without a START condition (sets MSR[FEF]).
- SCL (or SDA if MCFGR1[TIMECFG] is set) is low for (MCFGR2[TIMELOW] * 256) prescaler cycles without a pin transition (sets MSR[PLTF]).

Software must respond to the MSR[PLTF] flag to terminate the existing command either cleanly (by clearing MCR[MEN]), or abruptly (by setting MCR[RST]).

The MCFGR2[BUSIDLE] field can be used to force the I2C bus to be considered idle when SCL and SDA remain high for (BUSIDLE+1) prescaler cycles. The I2C bus is normally considered idle when the LPI2C master is first enabled, but when BUSIDLE is configured greater than zero then SCL and/or SDA must be high for (BUSIDLE+1) prescaler cycles before the I2C bus is first considered idle.

69.3.1.6 Pin Configuration

- **Open-drain support:** The LPI2C master defaults to open-drain configuration of the SDA and SCL pins. Support for true open drain depends on the specific device, and requires the pins where LPI2C pins are muxed to support true open drain.
- **High Speed mode support:** Support for high speed mode also depends on the specific device, and requires the SCL pin to support the current source pull-up required in the I2C specification.
- **Ultra-Fast mode support:** The LPI2C master also supports the output-only push-pull function required for I2C ultra-fast mode using the SDA and SCL pins. Support for ultra-fast mode also requires the MCFGR1[IGNACK] bit to be set.
- **Push-pull 2-wire support:** A push-pull 2-wire configuration is also available to the LPI2C master that may support a partial high speed mode, if the LPI2C is the only master and all I2C pins on the bus are at the same voltage. This configures the SCL pin as push-pull for every clock except the 9th clock pulse, to allow high speed mode compatible slaves to perform clock stretching. In this mode, the SDA pin is tristated for master-receive data bits and master-transmit ACK/NACK bits, and is configured as push-pull at other times. To avoid the risk of contention when SDA is push-pull, the pin can be configured for open-drain operation, as part of the device-specific configuration.
- **Push-pull 4-wire support:** The push-pull 4-wire configuration separates the SCL input data and output data into separate pins, and separates the SDA input data and output data into separate pins. The SCL/SDA pins are used for input data; the SCLS/SDAS pins are used for output data, with configurable polarity. This simplifies external connections when connecting the LPI2C to the I2C bus through external level shifters or discrete components. When using this 4-wire configuration, the LPI2C master logic and LPI2C slave logic are not able to connect to separate I2C buses.

69.3.2 Slave Mode

To perform all slave mode transfers on the I2C bus, the LPI2C slave logic operates independently from the LPI2C master logic.

69.3.2.1 Address Matching

The LPI2C slave can be configured:

- To match one of two addresses, using either 7-bit or 10-bit addressing modes for each address
- To match a range of addresses in either 7-bit or 10-bit addressing modes
- To match the General Call Address, and generate appropriate flags
- To match the SMBus Alert Address, and generate appropriate flags
- To detect the high speed mode master code, and to disable the digital filters and output valid delay time until the next STOP condition is detected

After a valid address is matched, the LPI2C slave automatically performs slave-transmit or slave-receive transfers until:

- A NACK is detected (unless MCFGR1[IGNACK] is set)
- A bit error is detected (the LPI2C slave is driving SDA, but a different value is sampled)
- A (repeated) START or STOP condition is detected

69.3.2.2 Transmit and Receive Data

- The Transmit and Receive Data registers are double-buffered and only update during a slave-transmit and slave-receive transfer, respectively.
- The slave address *that was received* can be configured to be read from either the Receive Data register (for example, when using DMA to transfer data), or from the Address Status register.
- The Transmit Data register can be configured to only request data after a slave-transmit transfer is detected, or to request new data whenever the Transmit Data register is empty.
- The Transmit Data register should only be written when the Transmit Data flag is set.
- The Receive Data register should only be read when the Received Data flag is set (or the Address Valid flag is set and SCFGR1[RXCFG] = 1).
- The Address Status register should only be read when the Address Valid flag is set.

69.3.2.3 Clock Stretching

The LPI2C slave supports many configurable options for when clock stretching is performed. The following conditions can be configured to perform clock stretching:

- During the 9th clock pulse of the address byte and the Address Valid flag is set.

- During the 9th clock pulse of a slave-transmit transfer and the Transmit Data flag is set.
- During the 9th clock pulse of a slave-receive transfer and the Receive Data flag is set.
- During the 8th clock pulse of an address byte or a slave-receive transfer and the Transmit ACK flag is set. In high speed mode, this is disabled.
- Clock stretching can also be extended for CLKHOLD cycles, to allow additional setup time to sample the SDA pin externally. In high speed mode, this is disabled.

Unless extended by the CLKHOLD configuration, clock stretching extends for one peripheral bus clock cycle after SDA updates when clock stretching is enabled.

69.3.2.4 Timing Parameters

The LPI2C slave can configure the following timing parameters. These parameters are disabled when SCR[FILTEN] is clear, when SCR[FILTDZ] is set in Doze mode, and when LPI2C slave detects high speed mode. When disabled, the LPI2C slave is clocked directly from the I2C bus, and may not satisfy all timing requirements of the I2C specification (such as SDA minimum hold time in Standard/Fast mode).

- SDA data valid time from SCL negation to SDA update
- SCL hold time when clock stretching is enabled to increase setup time when sampling SDA externally
- SCL glitch filter time
- SDA glitch filter time

The LPI2C slave imposes the following restrictions on the timing parameters.

- FILTSDA must be configured to greater than or equal to FILTSCL (unless compensating for board level skew between SDA and SCL).
- DATAVD must be configured less than the minimum SCL low period.

69.3.2.5 Error Conditions

The LPI2C slave can detect the following error conditions:

- Bit error flag sets when the LPI2C slave is driving SDA, but samples a different value than what is expected.

- FIFO error flag sets due to a transmit data underrun or a receive data overrun. To eliminate the possibility of underrun and overrun occurring, enable clock stretching.
- FIFO error flag also sets due to an address overrun when SCFGR1[RXCFG] is set, otherwise an address overrun is not flagged. To eliminate the possibility of overrun occurring, enable clock stretching.

The LPI2C slave does not implement a timeout due to SCL and/or SDA being stuck low. If this detection is required, then the LPI2C master logic should be used and so software can reset the LPI2C slave when this condition is detected.

69.3.3 Modes of operation

Table 69-6. Chip modes supported by the LPI2C module

Chip mode	LPI2C Operation
Run	Normal operations
Stop	Can continue operating in stop mode if the Doze Enable bit (MCR[DOZEN]) is clear and the LPI2C is using an external or internal clock source that remains operating during stop mode.
Low Power Stop (also called Low Leakage Stop or LLS)	Before entering low power stop mode, the LPI2C will wait for the current transfer to finish any pending operation, while temporarily ignoring The Doze Enable (MCR[DOZEN]) bit.
Debug	Can continue operating in debug mode if the Debug Enable bit (MCR[DBGE]) is set.

69.3.4 Interrupts and DMA Requests

Depending on the specific device, interrupts and DMA requests can be combined in some ways:

- The LPI2C master and slave interrupts may be combined
- The LPI2C master and slave transmit DMA requests may be combined
- The LPI2C master and slave receive DMA requests may be combined

69.3.4.1 Master mode

The next table lists the master mode sources that can generate LPI2C master interrupts and LPI2C master transmit/receive DMA requests.

Table 69-7. Master Interrupts and DMA Requests

Master Status Register (MSR)		Description	Can generate		
Flag	Name		Interrupt?	DMA Request?	Low Power Wakeup?
TDF	Transmit Data Flag	Data can be written to Transmit FIFO, as configured by the Transmit FIFO Watermark MFCR[TXWATER]	Y	TX	Y
RDF	Receive Data Flag	Data can be read from the Receive FIFO, as configured by the Receive FIFO Watermark MFCR[RXWATER]	Y	RX	Y
EPF	End Packet Flag	Master has transmitted a Repeated START or STOP condition	Y	N	Y
SDF	STOP Detect Flag	Master has transmitted a STOP condition	Y	N	Y
NDF	NACK Detect Flag	• During an address byte, the master expected an ACK but detected a NACK • During an address byte, the master expected a NACK but detected an ACK • During a master-transmitter data byte, the master detected a NACK	Y	N	Y
ALF	Arbitration Lost Flag	• The master lost arbitration due to a START/STOP condition detected at the wrong time • Or the master was transmitting data but received different data than the data that was transmitted	Y	N	Y
FEF	FIFO Error Flag	The master is expecting a START condition in the Command FIFO, but the next entry in the Command FIFO is not a START condition	Y	N	Y
PLTF	Pin Low Timeout Flag	Pin low timeout is enabled and SCL (or SDA if configured) is low for longer than the configured timeout	Y	N	Y
DMF	Data Match Flag	The received data matches the configured data match, but the received data is not discarded due to a command FIFO entry	Y	N	Y
MBF	Master Busy Flag	LPI2C master is busy transmitting/receiving data	N	N	N
BBF	Bus Busy Flag	LPI2C master is enabled and activity is detected on I2C bus, but a STOP condition has not been detected and a bus idle timeout (if enabled) has not occurred.	N	N	N

69.3.4.2 Slave mode

The next table lists the slave mode sources that can generate LPI2C slave interrupts and the LPI2C slave transmit/receive DMA requests.

Table 69-8. Slave Interrupts and DMA Requests

Slave Status Register (SSR)		Description	Can generate		
Flag	Name		Interrupt?	DMA Request?	Low Power Wakeup?
TDF	Transmit Data Flag	Data can be written to the Slave Transmit Data Register (STDR)	Y	TX	Y
RDF	Receive Data Flag	Data can be read from the Slave Receive Data Register (SRDR)	Y	RX	Y
AVF	Address Valid Flag	Address can be read from the Slave Address Status Register (SASR)	Y	RX	Y
TAF	Transmit ACK Flag	ACK/NACK can be written to the Slave Transmit ACK Register (STAR)	Y	N	Y
RSF	Repeated Start Flag	Slave has detected an address match followed by a Repeated START condition	Y	N	Y
SDF	STOP Detect Flag	Slave has detected an address match followed by a STOP condition	Y	N	Y
BEF	Bit Error Flag	Slave was transmitting data, but received different data than what was transmitted	Y	N	Y
FEF	FIFO Error Flag	• Transmit data underrun • Receive data overrun • Address status overrun (when Receive Data Configuration SCFGR1[RXCFG] = 1,) FEF flag can only set when clock stretching is disabled.	Y	N	Y
AM0F	Address Match 0 Flag	Slave detected an address match with SAMR[ADDR0] field	Y	N	N
AM1F	Address Match 1 Flag	Slave detected an address match with SAMR[ADDR1] field or using an address range	Y	N	N
GCF	General Call Flag	Slave detected an address match with the General Call address	Y	N	N
SARF	SMBus Alert Response Flag	Slave detected an address match with the SMBus Alert address	Y	N	N
SBF	Slave Busy Flag	LPI2C slave is busy receiving an address byte or is transmitting/receiving data	N	N	N
BBF	Bus Busy Flag	LPI2C slave is enabled and a START condition is detected on I2C bus, but a STOP condition has not been detected	N	N	N

69.3.5 Clocking and Resets

For device-specific clocking information, refer to the Clocking chapter.

Table 69-9. Clocks

LPI2C Functional clock	The LPI2C functional clock is asynchronous to the bus clock and can remain enabled in low power modes to support I2C bus transfers by the LPI2C master. The functional clock is also used by the LPI2C slave to support digital filter and data hold time configurations. The LPI2C master divides the functional clock by a prescaler and the resulting frequency must be at least 8 times faster than the I2C bus bandwidth.
External clock	The LPI2C slave logic is clocked directly from the external pins SCL and SDA (or SCLS and SDAS if master and slave are implemented on separate pins). This allows the LPI2C slave to remain operational, even when the LPI2C functional clock is disabled. **NOTE:** The LPI2C slave digital filter must be disabled if the LPI2C functional clock is disabled, and this can affect compliance with some of the timing parameters of the I2C specification, such as the data hold time.
Bus clock	The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPI2C master and slave registers.

Table 69-10. Resets

Chip reset	The logic and registers for the LPI2C master and slave are reset to their default state on a chip reset.
Software reset	• The LPI2C master implements a software reset bit in its Control Register. The MCR[RST] will reset all master logic and registers to their default state, except for the MCR itself. • The LPI2C slave implements a software reset bit in its Control Register. The SCR[RST] will reset all slave logic and registers to their default state, except for the SCR itself.
FIFO reset	• The LPI2C master implements write-only control bits that reset the transmit FIFO (MCR[RTF]) and receive FIFO (MCR[RRF]). After a FIFO is reset, that FIFO is empty. • The LPI2C slave implements write-only control bits that reset the transmit data register (SCR[RTF] and receive data register (SCR[RRF]). After a data register is reset, that data register is empty.

69.3.6 Peripheral Triggers

The connection of the LPI2C peripheral triggers to other peripherals depend upon the specific device being used.

Table 69-11. LPI2C Triggers

Trigger	Description
Master Output Trigger	The LPI2C master generates an output trigger that can be connected to other peripherals on the device. The master output trigger asserts on both a Repeated START or STOP condition, and the master output trigger remains asserted for one cycle of the LPI2C functional clock divided by the prescaler.
Slave Output Trigger	The LPI2C slave generates an output trigger that can be connected to other peripherals on the device. The slave output trigger asserts on both a Repeated START or STOP condition that occurs after a slave address match, and the slave output trigger remains asserted until the next slave SCL pin negation.

Table continues on the next page...

Table 69-11. LPI2C Triggers (continued)

Trigger	Description
Input Trigger	To control the start of a LPI2C bus transfer, the LPI2C input trigger can be selected instead of the HREQ input. The input trigger is synchronized and to be detected, the input trigger must assert for at least 2 cycles of the LPI2C functional clock divided by the PRESCALE configuration. When the LPI2C is busy, the HREQ input (and therefore the input trigger) is ignored.

69.4 Signal descriptions

Table 69-12. Signals

Signal	Name	2-Wire Scheme	4-Wire Scheme	I/O
SCL	LPI2C clock line	SCL	In 4-wire mode, this is the SCL input pin.	I/O
SDA	LPI2C data line	SDA	In 4-wire mode, this is the SDA input pin.	I/O
SCLS	Secondary I2C clock line	Not used	In 4-wire mode, this is the SCLS output pin. If LPI2C master/slave are configured to use separate pins, then this is the LPI2C slave SCL pin.	I/O
SDAS	Secondary I2C data line	Not used	In 4-wire mode, this is the SDAS output pin. If LPI2C master/slave are configured to use separate pins, then this is the LPI2C slave SDA pin.	I/O
HREQ	Host request	If host request is asserted and the I2C bus is idle, then it initiates an LPI2C master transfer. HREQ is an additional pin separate from the 2-wire or 4-wire scheme.		I

The I^2C 2-wire serial bus diagram shows the 2 signal connection.

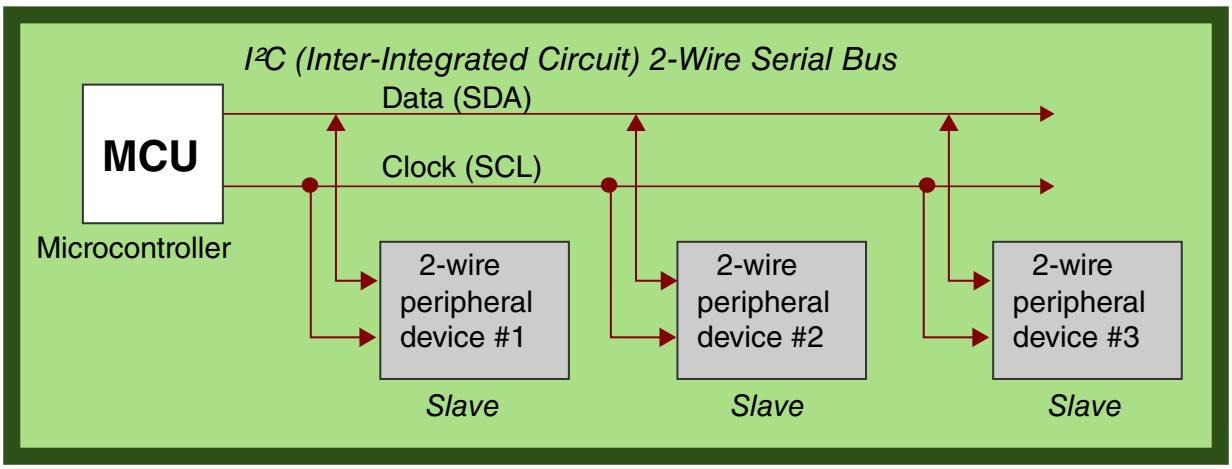


Figure 69-2. I²C 2-wire serial bus

The I²C 4-wire serial bus diagram shows a possible 4 signal connection.

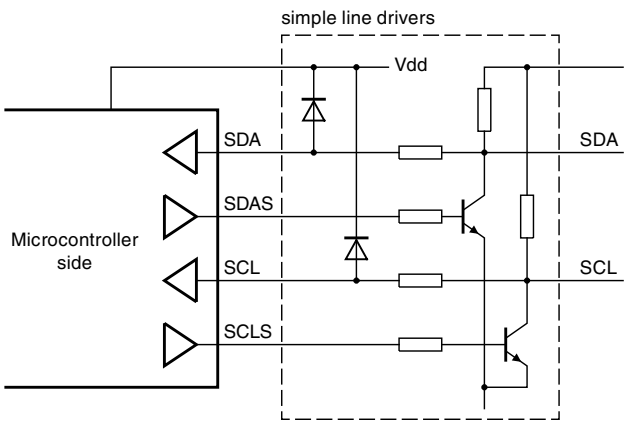


Figure 69-3. I²C 4-wire serial bus

69.5 Memory Map and Registers

NOTE

Writing a Read-Only (RO) register or reading a Write-Only (WO) register can cause bus errors. This module will not check if programmed values in the registers are correct; the application software must ensure that valid programmed values are being written.

69.5.1 LPI2C register descriptions

69.5.1.1 LPI2C memory map

LPI2C1 base address: 4010_4000h

LPI2C2 base address: 4010_8000h

LPI2C3 base address: 4010_C000h

LPI2C4 base address: 4011_0000h

LPI2C5 base address: 40C3_4000h

LPI2C6 base address: 40C3_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID (VERID)	32	RO	0101_0003
4	Parameter (PARAM)	32	RO	0000_0202
10	Master Control (MCR)	32	RW	0000_0000
14	Master Status (MSR)	32	W1C	0000_0001
18	Master Interrupt Enable (MIER)	32	RW	0000_0000
1C	Master DMA Enable (MDER)	32	RW	0000_0000
20	Master Configuration 0 (MCFGR0)	32	RW	0000_0000
24	Master Configuration 1 (MCFGR1)	32	RW	0000_0000
28	Master Configuration 2 (MCFGR2)	32	RW	0000_0000
2C	Master Configuration 3 (MCFGR3)	32	RW	0000_0000
40	Master Data Match (MDMR)	32	RW	0000_0000
48	Master Clock Configuration 0 (MCCR0)	32	RW	0000_0000
50	Master Clock Configuration 1 (MCCR1)	32	RW	0000_0000
58	Master FIFO Control (MFCR)	32	RW	0000_0000
5C	Master FIFO Status (MFSR)	32	RO	0000_0000
60	Master Transmit Data (MTDR)	32	WO	0000_0000
70	Master Receive Data (MRDR)	32	RO	0000_4000
110	Slave Control (SCR)	32	RW	0000_0000
114	Slave Status (SSR)	32	W1C	0000_0000
118	Slave Interrupt Enable (SIER)	32	RW	0000_0000
11C	Slave DMA Enable (SDER)	32	RW	0000_0000
124	Slave Configuration 1 (SCFGR1)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
128	Slave Configuration 2 (SCFGR2)	32	RW	0000_0000
140	Slave Address Match (SAMR)	32	RW	0000_0000
150	Slave Address Status (SASR)	32	RO	0000_4000
154	Slave Transmit ACK (STAR)	32	RW	0000_0000
160	Slave Transmit Data (STDR)	32	WO	0000_0000
170	Slave Receive Data (SRDR)	32	RO	0000_4000

69.5.1.2 Version ID (VERID)

Contains version numbers for the module design and feature set.

69.5.1.2.1 Offset

Register	Offset
VERID	0h

69.5.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

69.5.1.2.3 Fields

Field	Description
31-24	Major Version Number
MAJOR	Returns the major version number for the module design specification. Read-only field.
23-16	Minor Version Number

Table continues on the next page...

Field	Description
MINOR	Returns the minor version number for the module design specification. Read-only field.
15-0 FEATURE	Feature Specification Number Returns the feature set number. Read-only field. 0000000000000010 - Master only, with standard feature set 0000000000000011 - Master and slave, with standard feature set

69.5.1.3 Parameter (PARAM)

Contains parameter values that were implemented in the module.

69.5.1.3.1 Offset

Register	Offset
PARAM	4h

69.5.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MRXFIFO				0				MTXFIFO			
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

69.5.1.3.3 Fields

Field	Description
31-16 —	Reserved
15-12 —	Reserved
11-8	Master Receive FIFO Size

Table continues on the next page...

Field	Description
MRXFIFO	Configures the number of words in the master receive FIFO to $2^{MRXFIFO}$
7-4 —	Reserved
3-0 MTXFIFO	Master Transmit FIFO Size Configures the number of words in the master transmit FIFO to $2^{MTXFIFO}$

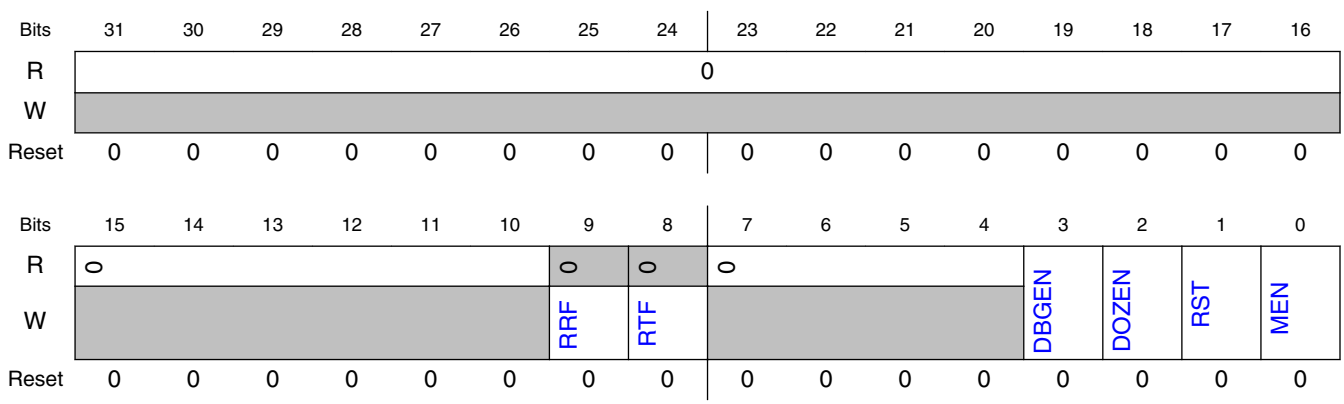
69.5.1.4 Master Control (MCR)

The MCR register contains resets, debug enable, and other master control settings.

69.5.1.4.1 Offset

Register	Offset
MCR	10h

69.5.1.4.2 Diagram



69.5.1.4.3 Fields

Field	Description
31-10 —	Reserved
9 RRF	Reset Receive FIFO Resets the Receive FIFO. 0 - No effect

Table continues on the next page...

Field	Description
	1 - Receive FIFO is reset
8 RTF	Reset Transmit FIFO Resets the Transmit FIFO. 0 - No effect 1 - Transmit FIFO is reset
7-4 —	Reserved
3 DBGEN	Debug Enable Enables the Master in Debug mode. 0 - Master is disabled in debug mode 1 - Master is enabled in debug mode
2 DOZEN	Doze mode enable Enables or disables the master in Doze mode. 0 - Master is enabled in Doze mode 1 - Master is disabled in Doze mode
1 RST	Software Reset Resets all internal master logic and registers, except the Master Control (MCR) register. RST remains set until cleared by software. The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0 - Master logic is not reset 1 - Master logic is reset
0 MEN	Master Enable Enables the Master logic. 0 - Master logic is disabled 1 - Master logic is enabled

69.5.1.5 Master Status (MSR)

MSR contains status flags for transmit and receive data; Start and Stop conditions; and bus and master busy or idle status.

69.5.1.5.1 Offset

Register	Offset
MSR	14h

69.5.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

69.5.1.5.3 Fields

Field	Description
31-26 —	Reserved
25 BBF	Bus Busy Flag Indicates that the I2C bus is busy. 0 - I2C Bus is idle 1 - I2C Bus is busy
24 MBF	Master Busy Flag Indicates that the I2C Master is busy. 0 - I2C Master is idle 1 - I2C Master is busy
23-16 —	Reserved
15 —	Reserved
14 DMF	Data Match Flag Indicates that the received data has matched the MDMR[MATCH0] and/or MDMR[MATCH1] fields (as configured by MCFGR1[MATCFG]. Received data *that is discarded due to CMD field* does not cause DMF to set. 0 - Have not received matching data 1 - Have received matching data
13 PLTF	Pin Low Timeout Flag Sets when the SCL and/or SDA input is low for more than PINLOW cycles (Pin Low Timeout, MCFGR3[PINLOW]), even when the LPI2C master is idle. • Software is responsible for resolving the pin low condition.

Table continues on the next page...

Field	Description
	• PLTF cannot be cleared as long as the pin low timeout continues. • Before the LPI2C can initiate a START condition, the PLTF must be cleared. 0 - Pin low timeout has not occurred or is disabled 1 - Pin low timeout has occurred
12 FEF	FIFO Error Flag Detects an attempt to send or receive data without first generating a (repeated) START condition. This can occur if the transmit FIFO underflows when MCFGR1[AUTOSTOP] = 1. When FEF is set, the LPI2C master sends a STOP condition (if busy), and does not initiate a new START condition until FEF has been cleared. 0 - No error 1 - Master sending or receiving data without a START condition
11 ALF	Arbitration Lost Flag Set if either of these conditions exist: • The LPI2C master transmits a logic one and detects a logic zero on the I2C bus • The LPI2C master detects a START or STOP condition while the LPI2C master is transmitting data When ALF sets, the LPI2C master releases the I2C bus (goes idle), and the LPI2C master does not initiate a new START condition until the ALF has been cleared. 0 - Master has not lost arbitration 1 - Master has lost arbitration
10 NDF	NACK Detect Flag Set if the LPI2C master detects a NACK it was not expecting when transmitting an address or data. When set, the master does not initiate a new START condition until NDF has been cleared. If a NACK is expected for a given address (as configured by the command word), then the NDF sets if a NACK is not generated. When NDF is set, the LPI2C master automatically transmits a STOP condition if MCFGR1[AUTOSTOP] = 1, or the transmit FIFO is not empty. 0 - Unexpected NACK was not detected 1 - Unexpected NACK was detected
9 SDF	STOP Detect Flag Set when the LPI2C master generates a STOP condition. 0 - Master has not generated a STOP condition 1 - Master has generated a STOP condition
8 EPF	End Packet Flag Set when the LPI2C master generates either a repeated START condition or a STOP condition. Does not set when the master first generates a START condition. 0 - Master has not generated a STOP or Repeated START condition 1 - Master has generated a STOP or Repeated START condition
7-2 —	Reserved
1 RDF	Receive Data Flag RDF sets whenever the number of words in the receive FIFO is greater than MFCR[RXWATER]. 0 - Receive Data is not ready

Table continues on the next page...

Field	Description
	1 - Receive data is ready
0 TDF	Transmit Data Flag TDF sets whenever the number of words in the transmit FIFO is equal or less than MFCR[TXWATER]. 0 - Transmit data is not requested 1 - Transmit data is requested

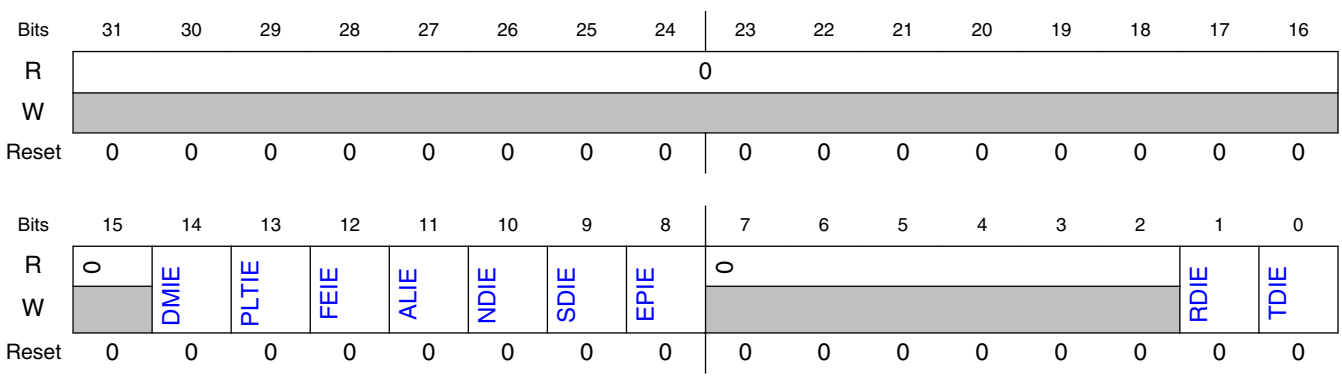
69.5.1.6 Master Interrupt Enable (MIER)

MIER contains transmit and receive data interrupt enables; Start, Stop, and Nack detect interrupt enables; and DMA interrupt enables.

69.5.1.6.1 Offset

Register	Offset
MIER	18h

69.5.1.6.2 Diagram



69.5.1.6.3 Fields

Field	Description
31-16 —	Reserved
15 —	Reserved

Table continues on the next page...

Field	Description
14 DMIE	Data Match Interrupt Enable Enables interrupt for data match. 0 - Disabled 1 - Enabled
13 PLTIE	Pin Low Timeout Interrupt Enable Enables interrupt for pin low timeout. 0 - Disabled 1 - Enabled
12 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0 - Enabled 1 - Disabled
11 ALIE	Arbitration Lost Interrupt Enable Enables interrupt for arbitration lost. 0 - Disabled 1 - Enabled
10 NDIE	NACK Detect Interrupt Enable Enables interrupt for NACK detect. 0 - Disabled 1 - Enabled
9 SDIE	STOP Detect Interrupt Enable Enables interrupt for STOP detect. 0 - Disabled 1 - Enabled
8 EPIE	End Packet Interrupt Enable Enables interrupt for end packet. 0 - Disabled 1 - Enabled
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0 - Disabled 1 - Enabled
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0 - Disabled 1 - Enabled

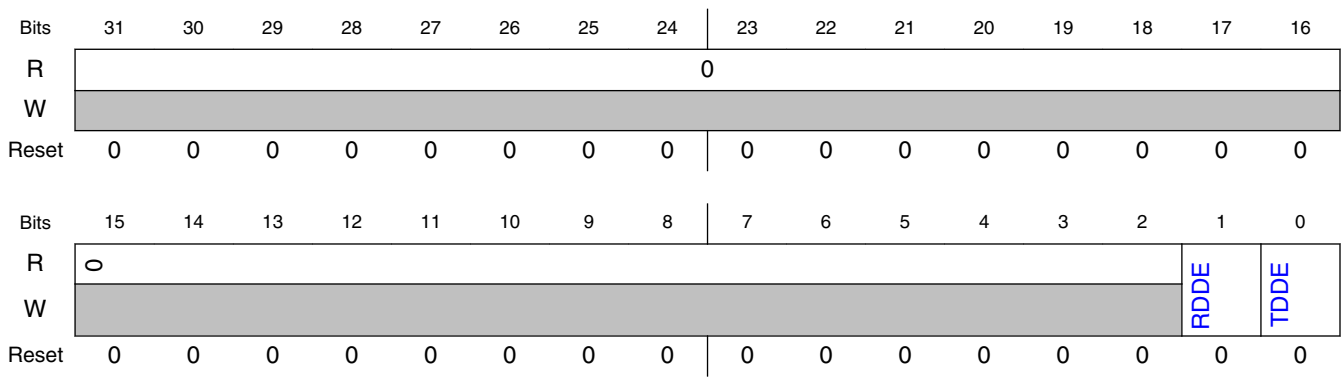
69.5.1.7 Master DMA Enable (MDER)

MDER contains DMA transmit, request, and receive enables.

69.5.1.7.1 Offset

Register	Offset
MDER	1Ch

69.5.1.7.2 Diagram



69.5.1.7.3 Fields

Field	Description
31-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables DMA receive data. 0 - DMA request is disabled 1 - DMA request is enabled
0 TDDE	Transmit Data DMA Enable Enables DMA transmit data. 0 - DMA request is disabled 1 - DMA request is enabled

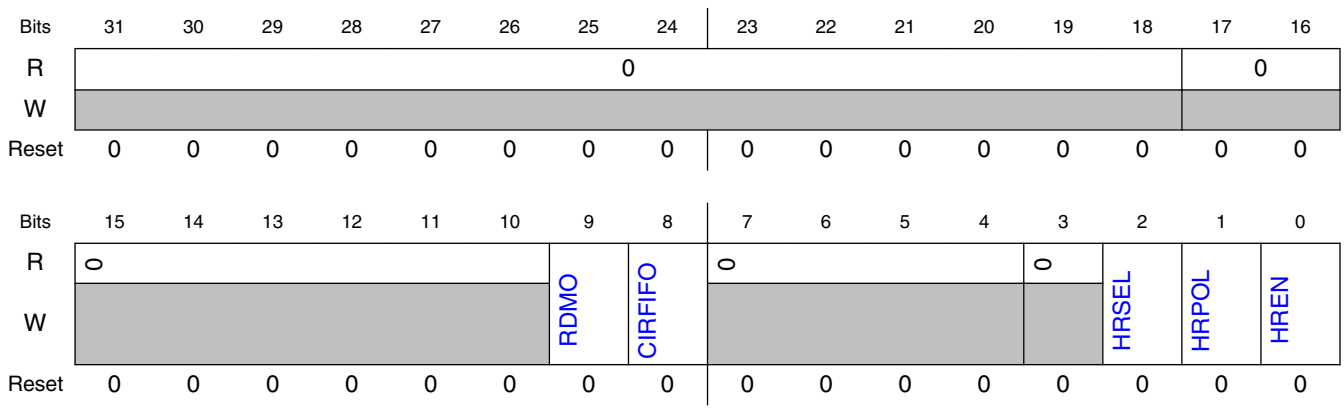
69.5.1.8 Master Configuration 0 (MCFGR0)

MCFGR0 contains host settings and other receive and transfer settings.

69.5.1.8.1 Offset

Register	Offset
MCFGR0	20h

69.5.1.8.2 Diagram



69.5.1.8.3 Fields

Field	Description
31-18 —	Reserved
17-16 —	Reserved
15-10 —	Reserved
9 RDMO	Receive Data Match Only When enabled, all received data that does not cause the Data Match Flag (MSR[DMF]) to set is discarded. After MSR[DMF] is set, the RDMO configuration is ignored. When disabling RDMO, clear RDMO before clearing MSR[DMF], to ensure that no receive data is lost. 0 - Received data is stored in the receive FIFO 1 - Received data is discarded unless the the Data Match Flag (MSR[DMF]) is set
8 CIRFIFO	Circular FIFO Enable When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO empties as normal, but after the LPI2C master is idle and the transmit FIFO is empty, then the read

Table continues on the next page...

Field	Description
	pointer value will be restored from the temporary register. This causes the contents of the transmit FIFO to be cycled through repeatedly. If MCFGR1[AUTOSTOP] is set, then a STOP condition is sent whenever the transmit FIFO is empty and the read pointer is restored. 0 - Circular FIFO is disabled 1 - Circular FIFO is enabled
7-4 —	Reserved
3 —	Reserved
2 HRSEL	Host Request Select Selects the source of the host request input. When host request is enabled, HRSEL should remain static (the Host Request Select should not change). 0 - Host request input is pin HREQ 1 - Host request input is input trigger
1 HRPOL	Host Request Polarity Configures the polarity of the host request pin. When host request is enabled, HRPOL should remain static (Host Request Polarity should not change). 0 - Active low 1 - Active high
0 HREN	Host Request Enable When enabled, the LPI2C master only initiates a START condition if the host request input is asserted and the bus is idle. A repeated START is not affected by the host request. 0 - Host request input is disabled 1 - Host request input is enabled

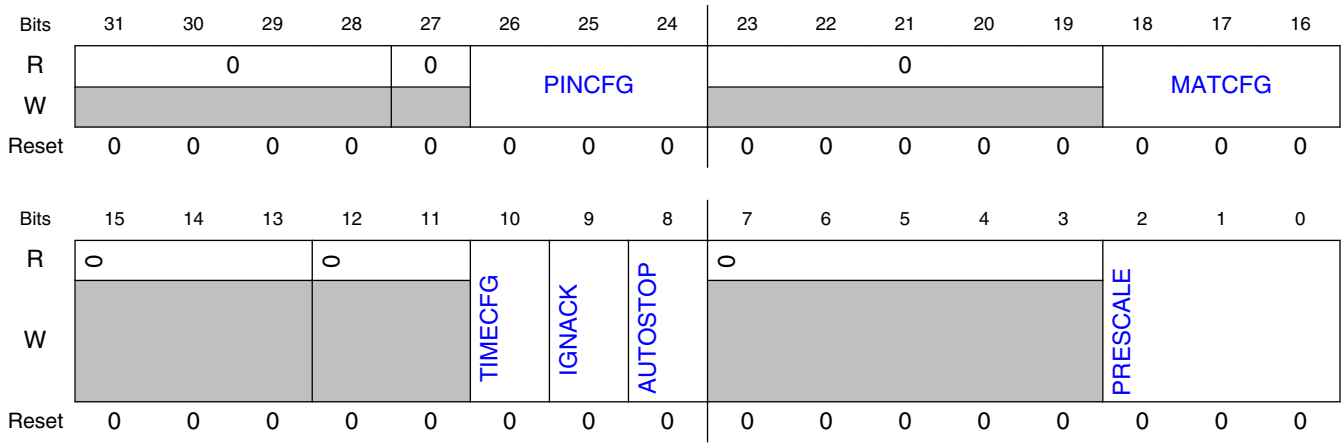
69.5.1.9 Master Configuration 1 (MCFGR1)

The MCFGR1 should only be written when the I2C Master is disabled.

69.5.1.9.1 Offset

Register	Offset
MCFGR1	24h

69.5.1.9.2 Diagram



69.5.1.9.3 Fields

Field	Description
31-28 —	Reserved
27 —	Reserved
26-24 PINCFG	Pin Configuration Configures the pin mode for LPI2C. **Table 69-13. 2-pin / 4-pin pin configurations for masters and slaves** 000 - 2-pin open drain mode 001 - 2-pin output only mode (ultra-fast mode) 010 - 2-pin push-pull mode

PINCFG	SCL / SDA pins	SCLS / SDAS pins
000	Bi-directional open drain for master and slave	Not used
001	Output-only (ultra-fast mode) open drain for master and slave	Not used
010	Bi-directional push-pull for master and slave	Not used
011	Input only for master and slave	Output-only push-pull for master and slave
100	Bi-directional open drain for master	Bi-directional open drain for slave
101	Output-only (ultra-fast mode) open drain for master	Output-only open drain for slave
110	Bi-directional push-pull for master	Bi-directional push-pull for slave
111	Input only for master and slave	Inverted output-only push-pull for master and slave

Table continues on the next page...

Field	Description
	011 - 4-pin push-pull mode 100 - 2-pin open drain mode with separate LPI2C slave 101 - 2-pin output only mode (ultra-fast mode) with separate LPI2C slave 110 - 2-pin push-pull mode with separate LPI2C slave 111 - 4-pin push-pull mode (inverted outputs)
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that causes MSR[DMF] to set. See Master Data Match (MDMR). 000 - Match is disabled 001 - Reserved 010 - Match is enabled (1st data word equals MDMR[MATCH0] OR MDMR[MATCH1]) 011 - Match is enabled (any data word equals MDMR[MATCH0] OR MDMR[MATCH1]) 100 - Match is enabled (1st data word equals MDMR[MATCH0] AND 2nd data word equals MDMR[MATCH1) 101 - Match is enabled (any data word equals MDMR[MATCH0] AND next data word equals MDMR[MATCH1) 110 - Match is enabled (1st data word AND MDMR[MATCH1] equals MDMR[MATCH0] AND MDMR[MATCH1]) 111 - Match is enabled (any data word AND MDMR[MATCH1] equals MDMR[MATCH0] AND MDMR[MATCH1])
15-13 —	Reserved
12-11 —	Reserved
10 TIMECFG	Timeout Configuration TIMECFG configures the timeout settings for the Pin Low Timeout Flag field (MSR[PLTF]). 0 - MSR[PLTF] sets if SCL is low for longer than the configured timeout 1 - MSR[PLTF] sets if either SCL or SDA is low for longer than the configured timeout
9 IGNACK	IGNACK When set, the received NACK field is ignored and assumed to be ACK. IGNACK bit is required to be set in Ultra-Fast Mode. 0 - LPI2C Master receives ACK and NACK normally 1 - LPI2C Master treats a received NACK as if it (NACK) was an ACK
8 AUTOSTOP	Automatic STOP Generation When enabled, a STOP condition is generated whenever the LPI2C master is busy and the transmit FIFO is empty. The STOP condition can also be generated using a transmit FIFO command. 0 - No effect 1 - STOP condition is automatically generated whenever the transmit FIFO is empty and the LPI2C master is busy
7-3	Reserved

Table continues on the next page...

Field	Description
—	
2-0 PRESCALE	Prescaler Configures the clock prescaler used for all LPI2C master logic, except for the digital glitch filters. 000 - Divide by 1 001 - Divide by 2 010 - Divide by 4 011 - Divide by 8 100 - Divide by 16 101 - Divide by 32 110 - Divide by 64 111 - Divide by 128

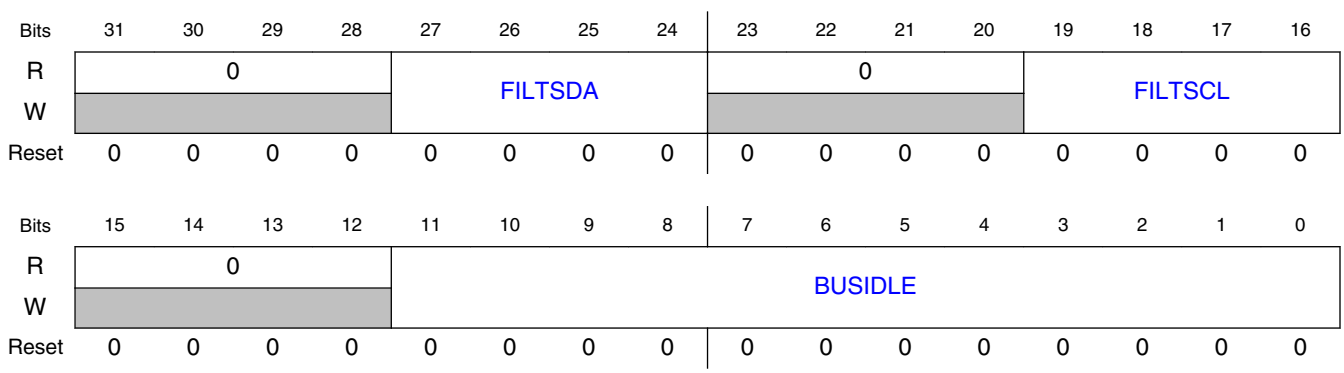
69.5.1.10 Master Configuration 2 (MCFGR2)

The Master Configuration Register 2 should only be written when the I2C Master is disabled.

69.5.1.10.1 Offset

Register	Offset
MCFGR2	28h

69.5.1.10.2 Diagram



69.5.1.10.3 Fields

Field	Description
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C master digital glitch filters for SDA input. • A configuration of 0 disables the glitch filter • Glitches equal to or less than FILTSDA cycles long are filtered out and ignored • The latency through the glitch filter is equal to FILTSDA cycles, and must be configured to be less than the minimum SCL low or high period • The glitch filter cycle count is not affected by the PRESCALE configuration, and the glitch filter cycle count is automatically bypassed in High Speed mode
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C master digital glitch filters for SCL input. • A configuration of 0 disables the glitch filter • Glitches equal to or less than FILTSCL cycles long are filtered out and ignored.The FILTSCL cycles are based on the functional clock. • The latency through the glitch filter is equal to FILTSCL cycles, and must be configured to be less than the minimum SCL low or high period • The glitch filter cycle count is not affected by the PRESCALE configuration, and the glitch filter cycle count is automatically bypassed in High Speed mode
15-12 —	Reserved
11-0 BUSIDLE	Bus Idle Timeout Configures the bus idle timeout period in clock cycles. • If both SCL and SDA are high for longer than BUSIDLE cycles, then the I2C bus is assumed to be idle and the master can generate a START condition • When Bus Idle Timeout is set to zero, the Bus Idle Timeout is disabled

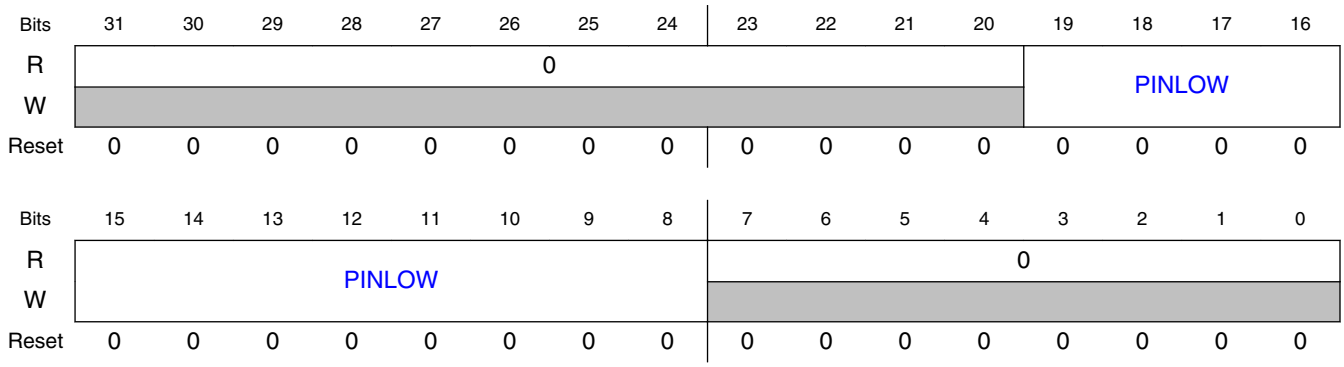
69.5.1.11 Master Configuration 3 (MCFGR3)

The MCFGR3 should only be written when the I2C Master is disabled.

69.5.1.11.1 Offset

Register	Offset
MCFGR3	2Ch

69.5.1.11.2 Diagram



69.5.1.11.3 Fields

Field	Description
31-20 —	Reserved
19-8 PINLOW	Pin Low Timeout Configures the pin low timeout flag in clock cycles. • If SCL or, either SCL or SDA, is low for longer than (PINLOW * 256) cycles, then MSR[PLTF] is set. • When Pin Low Timeout is set to zero, the Pin Low Timeout feature is disabled
7-0 —	Reserved

69.5.1.12 Master Data Match (MDMR)

The MDMR should only be written when the I2C Master is disabled or idle.

69.5.1.12.1 Offset

Register	Offset
MDMR	40h

69.5.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								MATCH1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MATCH0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.12.3 Fields

Field	Description
31-24 —	Reserved
23-16 MATCH1	Match 1 Value Compared against the received data when receive data match is enabled.
15-8 —	Reserved
7-0 MATCH0	Match 0 Value Compared against the received data when receive data match is enabled.

69.5.1.13 Master Clock Configuration 0 (MCCR0)

The MCCR0 cannot be changed when the I2C master is enabled and is used for standard, fast, fast-mode plus and ultra-fast transfers.

69.5.1.13.1 Offset

Register	Offset
MCCR0	48h

69.5.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.13.3 Fields

Field	Description
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Minimum number of cycles (minus one) that is used as the data hold time for SDA. Must be configured less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Minimum number of cycles (minus one) that is used by the master for these conditions: • Hold time for a START • Setup and hold time for a repeated START • Setup time for a STOP The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Minimum number of cycles (minus one) that the SCL clock is driven high by the master. The SCL high time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Minimum number of cycles (minus one) that the SCL clock is driven low by the master. The Clock Low Period value is also used for the minimum bus free time between a STOP and a START condition; this is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.

69.5.1.14 Master Clock Configuration 1 (MCCR1)

The MCCR1 cannot be changed when the I2C master is enabled and is used for high speed mode transfers. The separate clock configuration for high speed mode allows arbitration to take place in Fast mode (with timing configured by Master Clock Configuration 0 (MCCR0)), before switching to high speed mode (with timing configured by MCCR1).

69.5.1.14.1 Offset

Register	Offset
MCCR1	50h

69.5.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DATAVD						0		SETHOLD					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLKHI						0		CLKLO					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.14.3 Fields

Field	Description
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Minimum number of cycles (minus one) that is used as the data hold time for SDA. The DATAVD must be configured to be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Minimum number of cycles (minus one) that is used by the master for these conditions: • Hold time for a START condition

Table continues on the next page...

Field	Description
	• Setup and hold time for a repeated START condition • Setup time for a STOP condition The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Minimum number of cycles (minus one) that the SCL clock is driven high by the master. The SCL high time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Minimum number of cycles (minus one) that the SCL clock is driven low by the master. The CLKLO value is also used for the minimum bus free time between a STOP and a START condition; this is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) / 2^PRESCALE cycles.

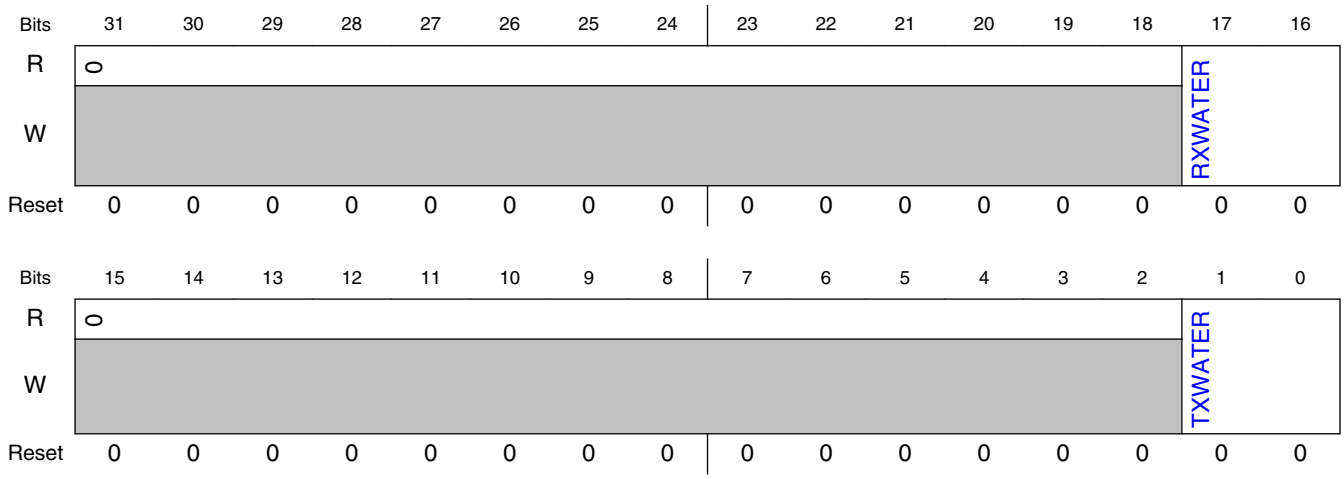
69.5.1.15 Master FIFO Control (MFCR)

The Master FIFO control register is only used in Stop mode when the MFCR register is static (i.e., the MFCR register is not changing).

69.5.1.15.1 Offset

Register	Offset
MFCR	58h

69.5.1.15.2 Diagram



69.5.1.15.3 Fields

Field	Description
31-18 —	Reserved
17-16 RXWATER	Receive FIFO Watermark The Receive Data Flag (SSR[RDF]) is set whenever the number of words in the receive FIFO is greater than RXWATER. Writing a value equal to or greater than the FIFO size truncates the value.
15-2 —	Reserved
1-0 TXWATER	Transmit FIFO Watermark The Transmit Data Flag (SSR[TDF]) is set whenever the number of words in the transmit FIFO is equal or less than TXWATER. Writing a value equal to or greater than the FIFO size truncates the value.

69.5.1.16 Master FIFO Status (MFSR)

MFSR gives the number of words in the transmit and receive FIFO.

69.5.1.16.1 Offset

Register	Offset
MFSR	5Ch

69.5.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													RXCOUNT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													TXCOUNT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.16.3 Fields

Field	Description
31-19 —	Reserved
18-16 RXCOUNT	Receive FIFO Count Returns the number of words in the receive FIFO.
15-3 —	Reserved
2-0 TXCOUNT	Transmit FIFO Count Returns the number of words in the transmit FIFO.

69.5.1.17 Master Transmit Data (MTDR)

- An 8-bit write to the CMD field is ignored and does not increment the FIFO write pointer.
- An 8-bit write to the DATA field zero-extends the CMD field and increments the FIFO write pointer.
- A 16-bit or 32-bit writes both the CMD and DATA fields and increments the FIFO write pointer.

69.5.1.17.1 Offset

Register	Offset
MTDR	60h

69.5.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0					CMD			DATA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.17.3 Fields

Field	Description
31-11 —	Reserved
10-8 CMD	Command Data 000 - Transmit DATA[7:0] 001 - Receive (DATA[7:0] + 1) bytes 010 - Generate STOP condition 011 - Receive and discard (DATA[7:0] + 1) bytes 100 - Generate (repeated) START and transmit address in DATA[7:0] 101 - Generate (repeated) START and transmit address in DATA[7:0]. This transfer expects a NACK to be returned. 110 - Generate (repeated) START and transmit address in DATA[7:0] using high speed mode 111 - Generate (repeated) START and transmit address in DATA[7:0] using high speed mode. This transfer expects a NACK to be returned.
7-0 DATA	Transmit Data Performing an 8-bit write to DATA zero-extends the CMD field.

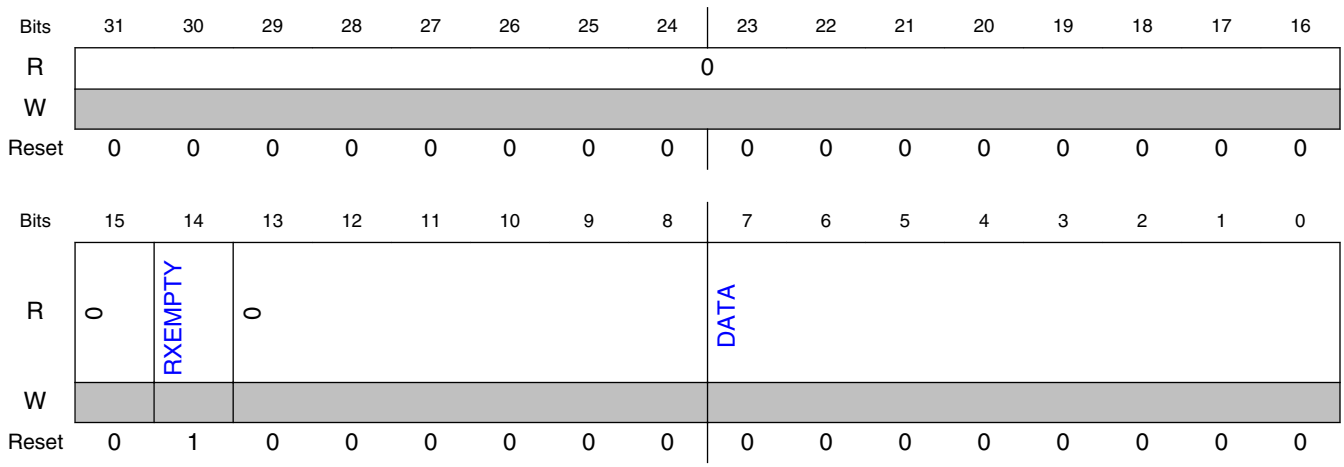
69.5.1.18 Master Receive Data (MRDR)

Reading the Receive Data register returns the data received by the I2C master that has not been discarded.

69.5.1.18.1 Offset

Register	Offset
MRDR	70h

69.5.1.18.2 Diagram



69.5.1.18.3 Fields

Field	Description
31-15 —	Reserved
14 RXEMPTY	RX Empty RXEMPTY gives the empty status of the Master receive data FIFO. 0 - Receive FIFO is not empty 1 - Receive FIFO is empty
13-8 —	Reserved
7-0 DATA	Receive Data Receive data can be discarded due to the CMD field, or the master can be configured to discard non-matching data.

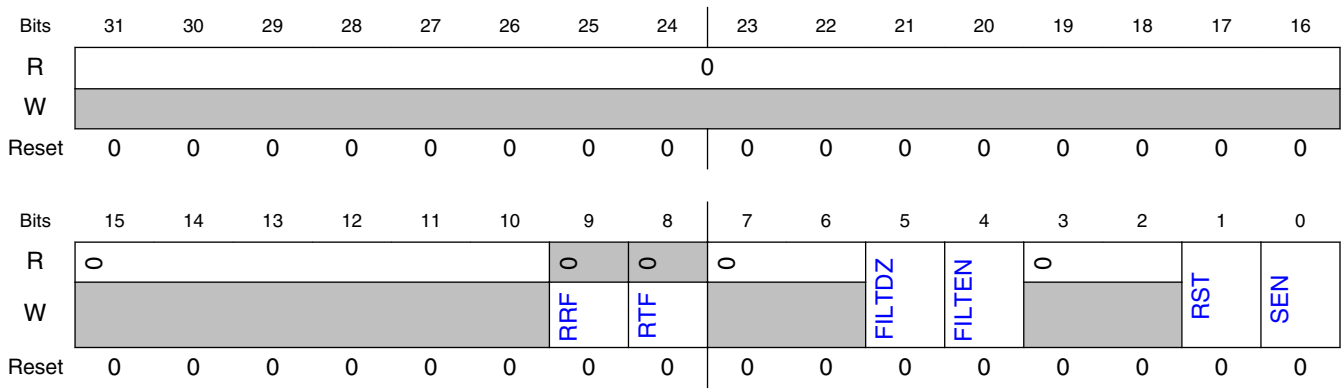
69.5.1.19 Slave Control (SCR)

SCR contains resets and other slave control settings.

69.5.1.19.1 Offset

Register	Offset
SCR	110h

69.5.1.19.2 Diagram



69.5.1.19.3 Fields

Field	Description
31-10 —	Reserved
9 RRF	Reset Receive FIFO Resets the receive FIFO to b be empty. 0 - No effect 1 - Receive Data Register is now empty
8 RTF	Reset Transmit FIFO Resets the transmit FIFO to be empty. 0 - No effect 1 - Transmit Data Register is now empty
7-6 —	Reserved
5 FILTDZ	Filter Doze Enable FILTDZ should only be updated when the I2C Slave is disabled. 0 - Filter remains enabled in Doze mode 1 - Filter is disabled in Doze mode
4	Filter Enable

Table continues on the next page...

Field	Description
FILTEN	FILTEN should only be updated when the I2C Slave is disabled. 0 - Disable digital filter and output delay counter for slave mode 1 - Enable digital filter and output delay counter for slave mode
3-2 —	Reserved
1 RST	Software Reset The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0 - Slave mode logic is not reset 1 - Slave mode logic is reset
0 SEN	Slave Enable Enables the I2C Slave mode. 0 - I2C Slave mode is disabled 1 - I2C Slave mode is enabled

69.5.1.20 Slave Status (SSR)

SSR contains status flags for transmit and receive data; error conditions; and bus and slave busy or idle status.

69.5.1.20.1 Offset

Register	Offset
SSR	114h

69.5.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	SBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SARF	GCF	AM1F	AM0F	FEF	BEF	SDF	RSF	0				TAF	AVF	RDF	TDF
W					W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.20.3 Fields

Field	Description
31-26 —	Reserved
25 BBF	Bus Busy Flag Indicates if an I2C bus is idle or busy. 0 - I2C Bus is idle 1 - I2C Bus is busy
24 SBF	Slave Busy Flag Indicates if an I2C slave is idle or busy. 0 - I2C Slave is idle 1 - I2C Slave is busy
23-16 —	Reserved
15 SARF	SMBus Alert Response Flag • SARF is cleared by reading the Slave Address Status (SASR) register • SARF cannot generate an asynchronous wakeup 0 - SMBus Alert Response is disabled or not detected 1 - SMBus Alert Response is enabled and detected
14 GCF	General Call Flag Indicates whether a slave has detected the General Call Address. • General Call Flag is cleared by reading the Slave Address Status (SASR) register • General Call Flag cannot generate an asynchronous wakeup 0 - Slave has not detected the General Call Address or the General Call Address is disabled 1 - Slave has detected the General Call Address
13 AM1F	Address Match 1 Flag Indicates that the received address has matched the ADDR1 field or ADDR0 to ADDR1 range as configured by SCFGR1[ADDRCFG]. • Address Match 1 Flag is cleared by reading the Slave Address Status (SASR) register • Address Match 1 Flag cannot generate an asynchronous wakeup 0 - Have not received an ADDR1 or ADDR0/ADDR1 range matching address 1 - Have received an ADDR1 or ADDR0/ADDR1 range matching address
12 AM0F	Address Match 0 Flag Indicates that the received address has matched the ADDR0 field as configured by SCFGR1[ADDRCFG]. • AM0F is cleared by reading the Slave Address Status (SASR) register • AM0F cannot generate an asynchronous wakeup 0 - Have not received an ADDR0 matching address 1 - Have received an ADDR0 matching address
11	FIFO Error Flag

Table continues on the next page...

Field	Description
FEF	FEF can only be set when clock stretching is disabled. 0 - FIFO underflow or overflow was not detected 1 - FIFO underflow or overflow was detected
10 BEF	Bit Error Flag BEF sets if the LPI2C slave transmits a logic one and detects a logic zero on the I2C bus. The slave ignores the rest of the transfer until the next (repeated) START condition. 0 - Slave has not detected a bit error 1 - Slave has detected a bit error
9 SDF	STOP Detect Flag SDF sets when the LPI2C slave detects a STOP condition and if the LPI2C slave matched the last address byte. 0 - Slave has not detected a STOP condition 1 - Slave has detected a STOP condition
8 RSF	Repeated Start Flag RSF sets when the LPI2C slave detects a repeated START condition and if the LPI2C slave matched the last address byte. The RSF does not set when the slave first detects a START condition. 0 - Slave has not detected a Repeated START condition 1 - Slave has detected a Repeated START condition
7-4 —	Reserved
3 TAF	Transmit ACK Flag TAF is cleared by writing to the Slave Transmit ACK (STAR) register. 0 - Transmit ACK/NACK is not required 1 - Transmit ACK/NACK is required
2 AVF	Address Valid Flag AVF is cleared by reading the Slave Address Status (SASR) register. When SCFGR1[RXCFG] = 1, SSR[AVF] is also cleared by reading the Slave Receive Data (SRDR) register. 0 - Address Status Register is not valid 1 - Address Status Register is valid
1 RDF	Receive Data Flag RDF is cleared by reading the Slave Receive Data (SRDR) register. When SCFGR1[RXCFG] = 1, RDF is not cleared when reading the Slave Receive Data (SRDR) register and if SSR[AVF] is set. 0 - Receive data is not ready 1 - Receive data is ready
0 TDF	Transmit Data Flag TDF is cleared by writing the to the Slave Transmit Data (STDR) register. When SCFGR1[TXCFG] = 0 and if a NACK or a Repeated START or STOP condition is detected, then TDF is also cleared. 0 - Transmit data not requested 1 - Transmit data is requested

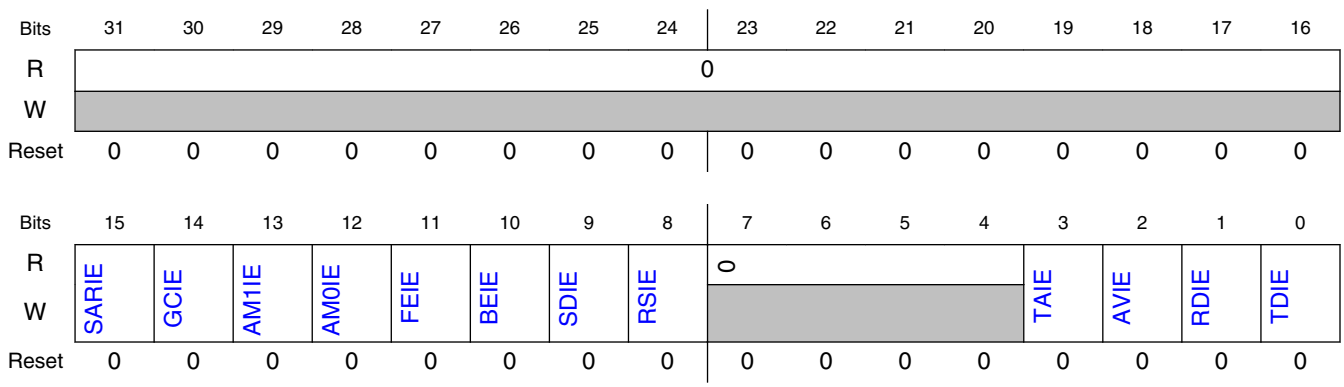
69.5.1.21 Slave Interrupt Enable (SIER)

SIER contains transmit and receive data interrupt enables; Start and Stop detect interrupt enables; and other slave interrupt enables.

69.5.1.21.1 Offset

Register	Offset
SIER	118h

69.5.1.21.2 Diagram



69.5.1.21.3 Fields

Field	Description
31-16 —	Reserved
15 SARIE	SMBus Alert Response Interrupt Enable Enables interrupt for SMBus alert response. 0 - Disabled 1 - Enabled
14 GCIE	General Call Interrupt Enable Enables interrupt for general call. 0 - Disabled 1 - Enabled
13 AM1IE	Address Match 1 Interrupt Enable Enables interrupt for address match 1.

Table continues on the next page...

Field	Description
	0 - Disabled 1 - Enabled
12 AM0IE	Address Match 0 Interrupt Enable Enables interrupt for address match 0. 0 - Disabled 1 - Enabled
11 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0 - Disabled 1 - Enabled
10 BEIE	Bit Error Interrupt Enable Enables interrupt for bit error. 0 - Disabled 1 - Enabled
9 SDIE	STOP Detect Interrupt Enable Enables interrupt for STOP detect. 0 - Disabled 1 - Enabled
8 RSIE	Repeated Start Interrupt Enable Enables interrupt for repeated start. 0 - Disabled 1 - Enabled
7-4 —	Reserved
3 TAIE	Transmit ACK Interrupt Enable Enables interrupt for transmit ACK. 0 - Disabled 1 - Enabled
2 AVIE	Address Valid Interrupt Enable Enables interrupt for valid address. 0 - Disabled 1 - Enabled
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0 - Disabled 1 - Enabled
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0 - Disabled

Field	Description
	1 - Enabled

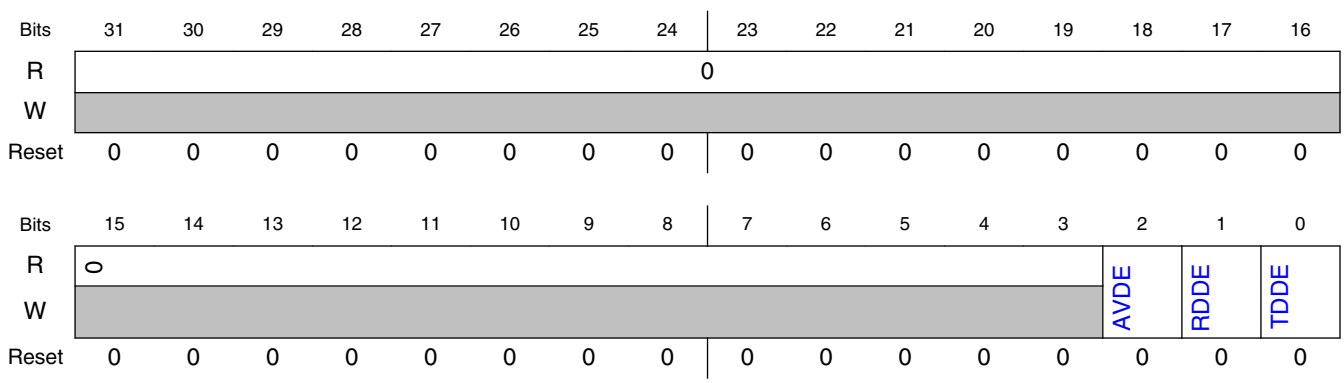
69.5.1.22 Slave DMA Enable (SDER)

SDER contains the transmit, request, and receive enables for DMA.

69.5.1.22.1 Offset

Register	Offset
SDER	11Ch

69.5.1.22.2 Diagram



69.5.1.22.3 Fields

Field	Description
31-3 —	Reserved
2 AVDE	Address Valid DMA Enable The Address Valid DMA request is shared with the Receive Data DMA request. If both Address Valid DMA request and Receive Data DMA request are enabled, then set SCFGR1[RXCFG] = 1, to allow the DMA to read the address from the Slave Receive Data (SRDR) register. 0 - DMA request is disabled 1 - DMA request is enabled
1 RDDE	Receive Data DMA Enable Enables receive data for DMA.

Table continues on the next page...

Field	Description
	0 - DMA request is disabled 1 - DMA request is enabled
0 TDDE	Transmit Data DMA Enable Enables transmit data for DMA. 0 - DMA request is disabled 1 - DMA request is enabled

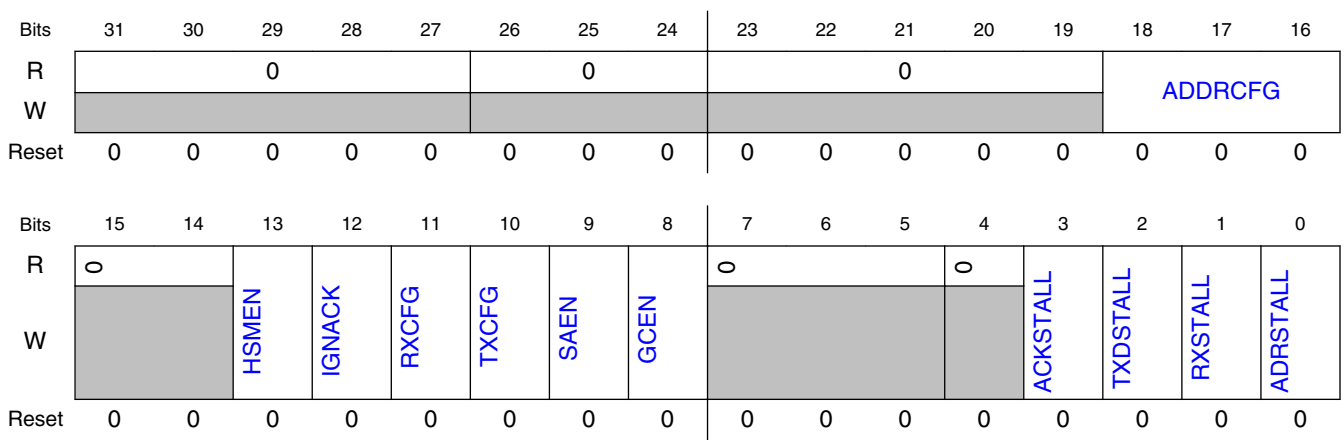
69.5.1.23 Slave Configuration 1 (SCFGR1)

The Slave Configuration Register 1 should only be written when the I2C Slave is disabled.

69.5.1.23.1 Offset

Register	Offset
SCFGR1	124h

69.5.1.23.2 Diagram



69.5.1.23.3 Fields

Field	Description
31-27 —	Reserved

Table continues on the next page...

Field	Description
26-24 —	Reserved
23-19 —	Reserved
18-16 ADDRCFG	Address Configuration Configures the condition that causes an address to match. 000 - Address match 0 (7-bit) 001 - Address match 0 (10-bit) 010 - Address match 0 (7-bit) or Address match 1 (7-bit) 011 - Address match 0 (10-bit) or Address match 1 (10-bit) 100 - Address match 0 (7-bit) or Address match 1 (10-bit) 101 - Address match 0 (10-bit) or Address match 1 (7-bit) 110 - From Address match 0 (7-bit) to Address match 1 (7-bit) 111 - From Address match 0 (10-bit) to Address match 1 (10-bit)
15-14 —	Reserved
13 HSMEN	High Speed Mode Enable Enables detection of the High-Speed Mode master code of slave address 0000_1XX, but does not cause an address match on this code. When set and any HS-mode master code is detected, SCR[FILTEN] and SCFGR1[ACKSTALL] bits are ignored until the next STOP condition is detected. 0 - Disables detection of HS-mode master code 1 - Enables detection of HS-mode master code
12 IGNACK	Ignore NACK When Ignore NACK is set, the LPI2C slave continues transfers after a NACK is detected. Ignore NACK bit is required to be set in Ultra-Fast mode. 0 - Slave ends transfer when NACK is detected 1 - Slave does not end transfer when NACK detected
11 RXCFG	Receive Data Configuration 0 - Reading the Receive Data register returns received data and clears the Receive Data flag (MSR[RDF]). 1 - Reading the Receive Data register when the Address Valid flag (SSR[AVF])is set, returns the Address Status register and clear the Address Valid flag. Reading the Receive Data register when the Address Valid flag is clear, returns received data and clears the Receive Data flag (MSR[RDF]).
10 TXCFG	Transmit Flag Configuration The transmit data flag always asserts before a NACK is detected at the end of a slave-transmit transfer. This can cause an extra word to be written to the transmit data FIFO. • When TXCFG = 0, the Transmit Data register is automatically emptied when a slave-transmit transfer is detected. This causes the transmit data flag to assert whenever a slave-transmit transfer is detected, and causes the transmit data flag to negate at the end of the slave-transmit transfer. • When TXCFG = 1, the Transmit Data flag asserts whenever the Transmit Data register is empty, and the Transmit Data flag negates when the Transmit Data register is full. This allows the Transmit Data register to be filled before a slave-transmit transfer is detected, but can cause the Transmit Data register to be written before a NACK is detected on the last byte of a slave transmit transfer.

Table continues on the next page...

Field	Description
	0 - Transmit Data Flag only asserts during a slave-transmit transfer when the Transmit Data register is empty 1 - Transmit Data Flag asserts whenever the Transmit Data register is empty
9 SAEN	SMBus Alert Enable Enables a match on an SMBus alert. 0 - Disables match on SMBus Alert 1 - Enables match on SMBus Alert
8 GCEN	General Call Enable Enables a general call address. 0 - General Call address is disabled 1 - General Call address is enabled
7-5 —	Reserved
4 —	Reserved
3 ACKSTALL	ACK SCL Stall Enables SCL clock stretching during slave-transmit address byte(s) and slave-receiver address and data byte(s), to allow software to write the Slave Transmit ACK (STAR) register before the ACK or NACK is transmitted. Clock stretching occurs when transmitting the 9th bit, and is therefore not compatible with high speed mode. When ACKSTALL is enabled, there is no need to set either RX SCL Stall (SCFGR1[RXSTALL]) or Address SCL Stall (SCFGR1[ADRSTALL]). 0 - Clock stretching is disabled 1 - Clock stretching is enabled
2 TXDSTALL	TX Data SCL Stall Enables SCL clock stretching when the SSR[TDF] = 1 during a slave-transmit transfer. Clock stretching occurs following the 9th bit, and is therefore compatible with high speed mode. 0 - Clock stretching is disabled 1 - Clock stretching is enabled
1 RXSTALL	RX SCL Stall Enables SCL clock stretching when SSR[RDF] = 1 during a slave-receive transfer. Clock stretching occurs following the 9th bit, and is therefore compatible with high speed mode. 0 - Clock stretching is disabled 1 - Clock stretching is enabled
0 ADRSTALL	Address SCL Stall Enables SCL clock stretching when SSR[AVF] = 1. Clock stretching only occurs following the 9th bit, and is therefore compatible with high speed mode. 0 - Clock stretching is disabled 1 - Clock stretching is enabled

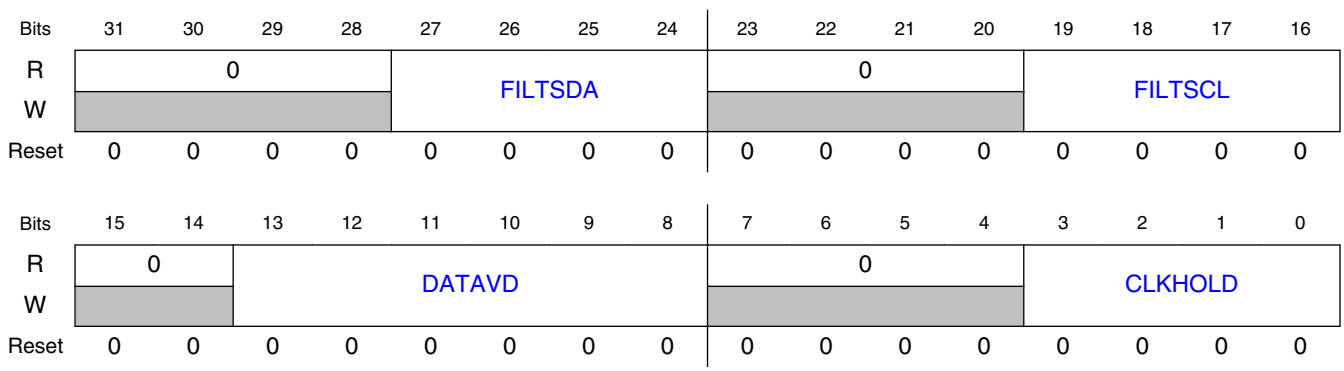
69.5.1.24 Slave Configuration 2 (SCFGR2)

The Slave Configuration Register 2 should only be written when the I2C Slave is disabled.

69.5.1.24.1 Offset

Register	Offset
SCFGR2	128h

69.5.1.24.2 Diagram



69.5.1.24.3 Fields

Field	Description
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C slave digital glitch filters for SDA input. • A configuration of 0 disables the glitch filter • Glitches equal to or less than FILTSDA cycles long are filtered out and ignored • The latency through the glitch filter is equal to FILTSDA+3 cycles, and must be configured to be less than the minimum SCL low or high period • The glitch filter cycle count is not affected by the PRESCALE configuration, and the glitch filter cycle count is disabled in high speed mode
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C slave digital glitch filters for SCL input. • A configuration of 0 disables the glitch filter • Glitches equal to or less than FILTSCL cycles long are filtered out and ignored

Table continues on the next page...

Field	Description
	• The latency through the glitch filter is equal to FILTSCL+3 cycles, and must be configured to be less than the minimum SCL low or high period • The glitch filter cycle count is not affected by the PRESCALE configuration, and the glitch filter cycle count is disabled in high speed mode
15-14 —	Reserved
13-8 DATAVD	Data Valid Delay Configures the SDA data valid delay time for the I2C slave, and is equal to FILTSCL+DATAVD+3 cycles. • The data valid delay must be configured to be less than the minimum SCL low period • The I2C slave data valid delay time is not affected by the PRESCALE configuration, and the I2C slave data valid delay time is disabled in high speed mode
7-4 —	Reserved
3-0 CLKHOLD	Clock Hold Time Configures the minimum clock hold time for the I2C slave, when clock stretching is enabled. • The minimum hold time is equal to CLKHOLD+3 cycles • The I2C slave clock hold time is not affected by the PRESCALE configuration, and the I2C slave clock hold time is disabled in high speed mode

69.5.1.25 Slave Address Match (SAMR)

The SAMR should only be written when the I2C Slave is disabled.

69.5.1.25.1 Offset

Register	Offset
SAMR	140h

69.5.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ADDR1										0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ADDR0										0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.25.3 Fields

Field	Description
31-27 —	Reserved
26-17 ADDR1	Address 1 Value Compared against the received address to detect the Slave Address. • In 10-bit mode, the first address byte is compared to { 11110, ADDR1[26:25] } and the second address byte is compared to ADDR1[24:17] • In 7-bit mode, the address is compared to ADDR1[23:17]
16-11 —	Reserved
10-1 ADDR0	Address 0 Value Compared against the received address to detect the Slave Address. • In 10-bit mode, the first address byte is compared to { 11110, ADDR0[10:9] } and the second address byte is compared to ADDR0[8:1] • In 7-bit mode, the address is compared to ADDR0[7:1] •
0 —	Reserved

69.5.1.26 Slave Address Status (SASR)

SASR contains the read-only RADDR and ANV fields.

69.5.1.26.1 Offset

Register	Offset
SASR	150h

69.5.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ANV	0			RADDR										
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

69.5.1.26.3 Fields

Field	Description
31-15 —	Reserved
14 ANV	Address Not Valid 0 - Received Address (RADDR) is valid 1 - Received Address (RADDR) is not valid
13-11 —	Reserved
10-0 RADDR	Received Address The Received Address updates whenever the AMF is set; the AMF is cleared by reading the Slave Address Status register. • In 7-bit mode, the address byte is stored in RADDR[7:0] • In 10-bit mode, the first address byte is { 11110, RADDR[10:9], RADDR[0] } and the second address byte is RADDR[8:1]. The R/W bit is therefore always stored in RADDR[0]

69.5.1.27 Slave Transmit ACK (STAR)

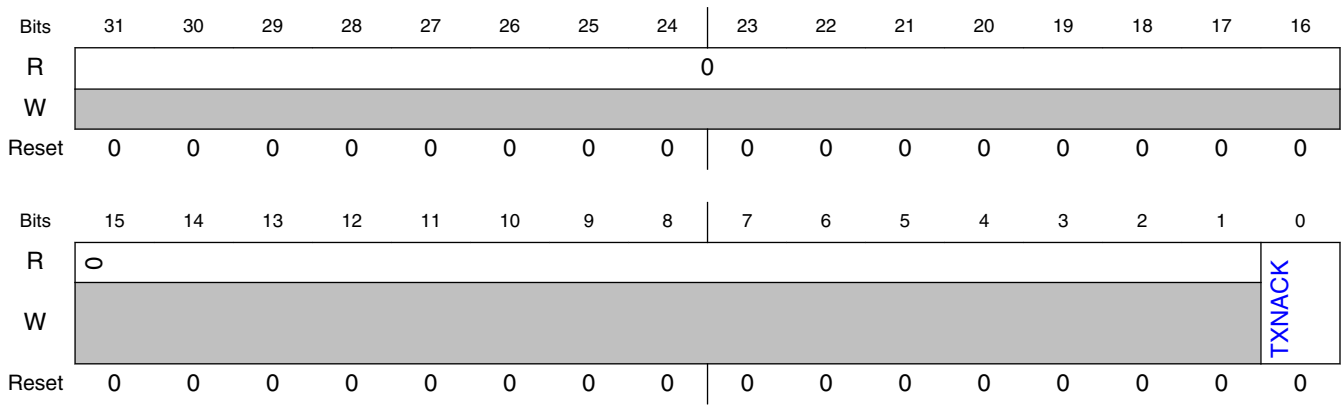
STAR can only be written when the ACK SCL Stall bit is set (SCFGR1[ACKSTALL]).

- The SCFGR1[ACKSTALL] bit enables clock stretching during the ACK/NACK bit slot, and during this time, the STAR register can be written by software.
- The logic ensures that the clock stretching continues for at least 1 bus clock cycle after the STAR register is updated.
- This clock stretching time can be extended more using the Clock Hold Time field (SCFGR2[CLKHOLD]).

69.5.1.27.1 Offset

Register	Offset
STAR	154h

69.5.1.27.2 Diagram



69.5.1.27.3 Fields

Field	Description
31-1 —	Reserved
0 TXNACK	Transmit NACK After receiving each word, software can transmit either an ACK (logic 0) or a NACK (logic 1); Transmit NACK selects which to use: ACK or NACK. • When SCFGR1[ACKSTALL] is set, a Transmit NACK must be written once for each matching address byte and each received word. ACKSTALL must be set, because that stalls the data transfer until software reads the received word (and decides whether to respond with an ACK or NACK). • To configure the default ACK/NACK, Transmit NACK can also be written when LPI2C Slave is disabled or idle. 0 - Write a Transmit ACK for each received word 1 - Write a Transmit NACK for each received word

69.5.1.28 Slave Transmit Data (STDR)

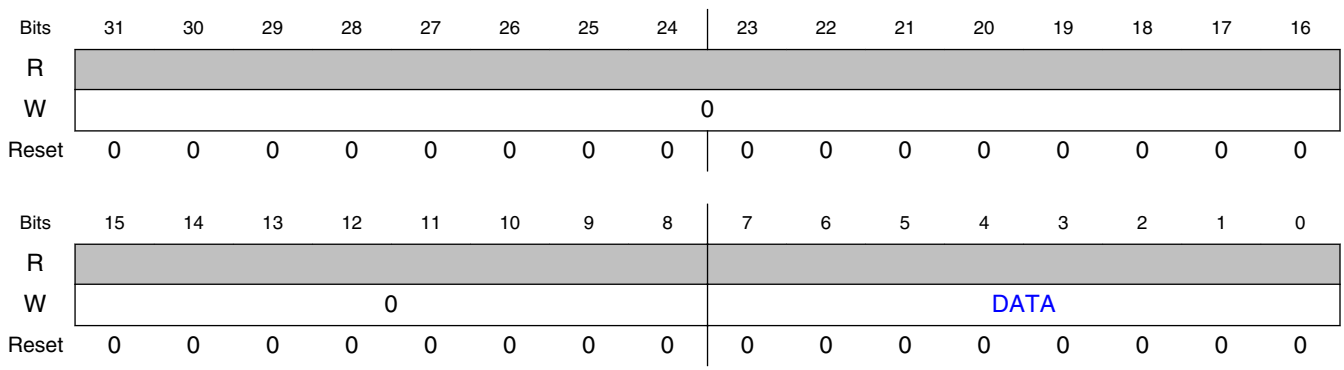
Clock stretching (enabled or disabled) affects when the transmit data is transferred. The SCFGR1[TXDSTALL] bit enables clock stretching during the 1st data bit of a slave-transmit transfer.

- **If clock stretching is enabled (SCFGR1[TXDSTALL] = 1)**, then the transmit data transfer is stalled until the STDR is updated. Clock stretching is extended by at least 1 bus clock cycle after STDR is updated, and clock stretching can be delayed even more using the Clock Hold Time field (SCFGR2[CLKHOLD].
- **If clock stretching is disabled (SCFGR1[TXDSTALL] = 0)**, then the transmit data should be written before the start of the slave-transmit transfer; otherwise (that is, if the transmit data is not written before the start of the slave-transmit transfer), the FIFO Error Flag (SSR[FEF]) sets.

69.5.1.28.1 Offset

Register	Offset
STDR	160h

69.5.1.28.2 Diagram



69.5.1.28.3 Fields

Field	Description
31-8 —	Reserved
7-0 DATA	Transmit Data Writing to the Slave Transmit Data Register (STDR) stores I2C slave transmit data *in* the Slave Transmit Data Register

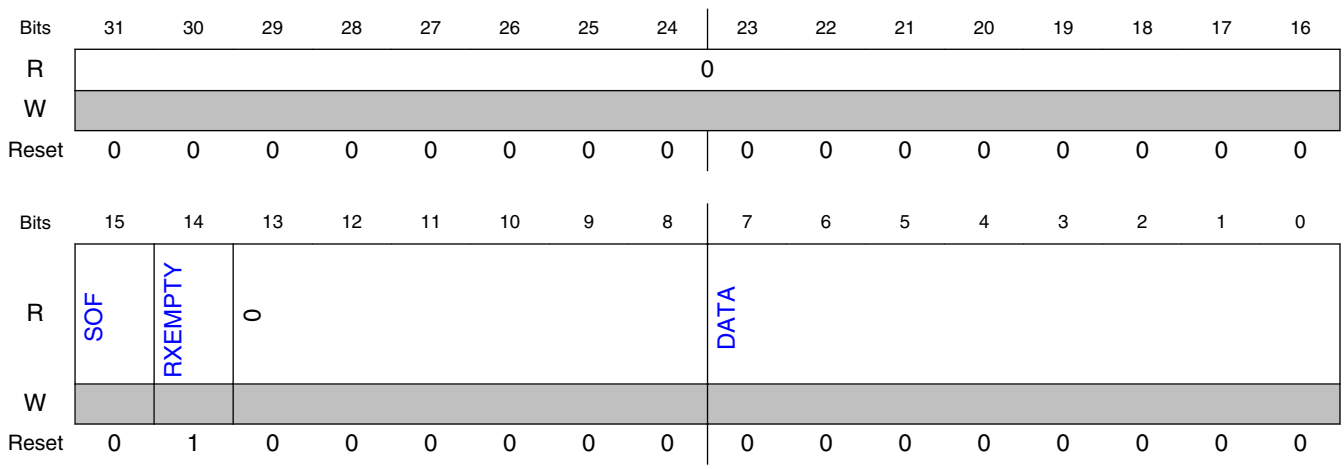
69.5.1.29 Slave Receive Data (SRDR)

Reading the Slave Receive Data Register returns the data received by the I2C slave.

69.5.1.29.1 Offset

Register	Offset
SRDR	170h

69.5.1.29.2 Diagram



69.5.1.29.3 Fields

Field	Description
31-16 —	Reserved
15 SOF	Start Of Frame Indicates whether this is the first data word. 0 - Indicates this is not the first data word since a (repeated) START or STOP condition 1 - Indicates this is the first data word since a (repeated) START or STOP condition
14 RXEMPTY	RX Empty Gives the empty status of the receive data register. 0 - The Receive Data Register is not empty 1 - The Receive Data Register is empty
13-8	Reserved

Table continues on the next page...

Field	Description
—	
7-0 DATA	Receive Data Contains data received by the I2C slave.

Chapter 70
Low Power Serial Peripheral Interface (LPSPI)

70.1 Chip-specific LPSPI information

Table 70-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX

NOTE

The output trigger in "Peripheral Triggers" section is not applicable for this device.

70.2 Overview

LPSPI provides an efficient interface to a SPI bus, either as a master or slave. A SPI bus is a synchronous serial communication interface used in embedded systems. It is typically used to perform short distance communications between microcontrollers and peripheral devices, on printed circuit boards. Typical applications include interfacing to Secure Digital cards and LCD displays.

70.2.1 Block diagram

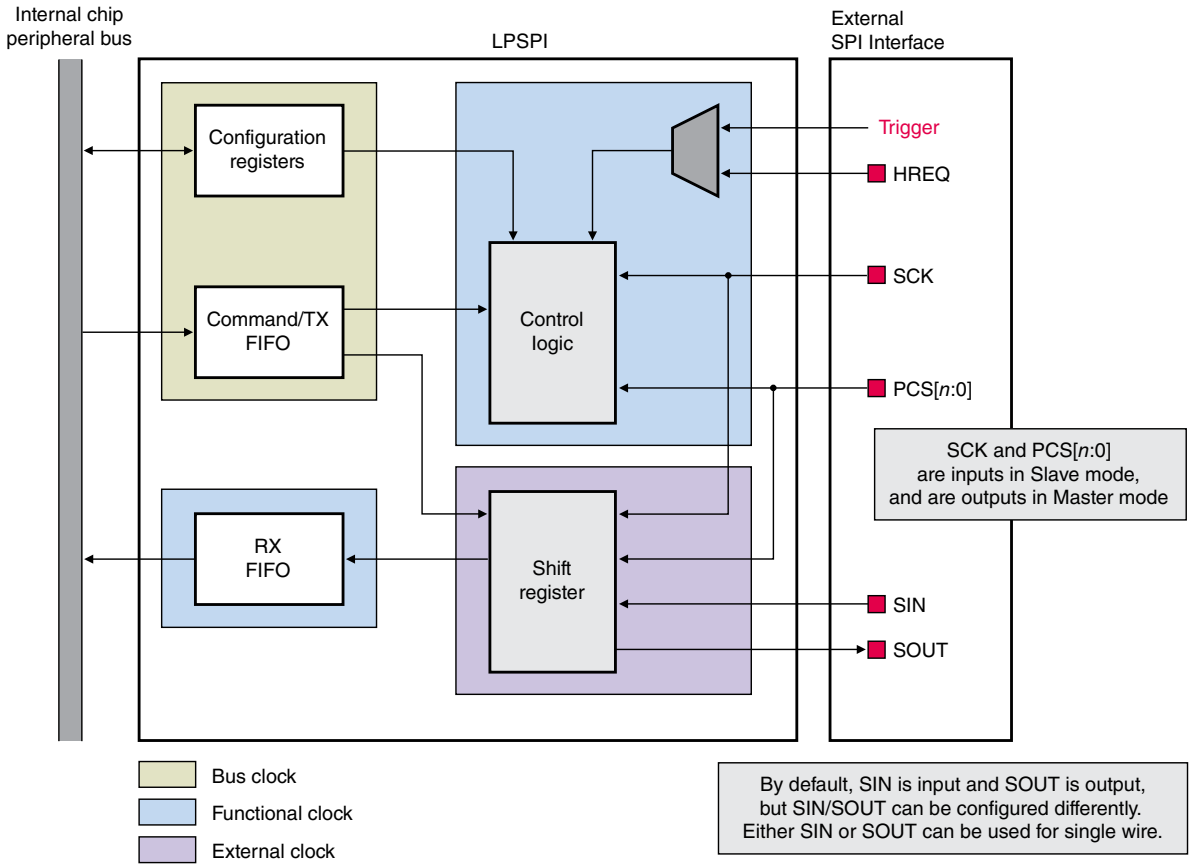


Figure 70-1. Block diagram

70.2.2 Features

- Requires minimal CPU overhead, with FIFO register access supported by DMA transmit and receive requests.
- Continues operating in Stop mode, if configured to do so and an appropriate clock is available
- 32-bit word size
- Configurable clock polarity and phase
- Master mode—supports 4 peripheral chip selects
- Slave mode
- 16-word transmit and command FIFO
- 16-word receive FIFO
- Flexible timing parameters in Master mode, including SCK frequency and delays between PCS and SCK edges
- Full-duplex transfers support 1-bit transmit and receive on each clock edge
- Half-duplex transfers support:

  - 1-bit transmit or receive on each clock edge
  - 2-bit transmit or receive on each clock edge (Master mode only)
  - 4-bit transmit or receive on each clock edge (Master mode only)
- Receive data match logic supports discard of non-matching data and interrupt on data match

70.3 Functional description

70.3.1 Master mode

70.3.1.1 Transmit and command FIFO commands

The transmit and command FIFO is a combined FIFO that includes both transmit data words and command words.

- You store transmit data words to the transmit and command FIFO, by writing to Transmit Data (TDR).
- You store command words to the transmit and command FIFO, by writing to Transmit Command (TCR).

When a command word is at the top of the transmit and command FIFO, the actions that can occur depend upon whether the LPSPI module is busy or between frames. See TCR[CONT] and TCR[CONTC].

Table 70-2. Possible actions when a command word is at the top of the transmit and command FIFO

Condition	Action
LPSPI is enabled and idle	The command word is pulled from the FIFO, and that command word controls all subsequent transfers.
LPSPI is busy and the Continuing Command field (TCR[CONTC]) is 0	The SPI frame completes at the end of the existing word, ignoring TCR[FRAMESZ]. The command word is then pulled from the FIFO and that command word controls all subsequent transfers (or until the next update to the command word). Note that a command word with TCR[CONTC] = 0 always terminates the existing transfer regardless of the previous TCR[CONT] value.
LPSPI is busy, the existing TCR[CONT] value is 1 and the new TCR[CONTC] value is 1	The command word must be updated at the frame boundary. The command word is pulled from the FIFO during the last SCK pulse of the existing frame (based on the FRAMESZ value), and the frame continues using the new command value for the rest of the frame (or until the next update to the command word). When TCR[CONTC] = 1, only the lower 24-bits of the command word are updated. If the command word is updated at a word boundary, then the transfer halts (stops) after that word. TCR[CONTC] is ignored when not at a frame boundary, so the frame ends prematurely.

About TCR[CONT] and TCR[CONTC] :

- TCR[CONT] = 1 keeps PCS asserted at end of frame, allowing the transfer to continue.
- TCR[CONTC] = 1 specifies that this command word should not terminate the existing frame, and the transfer can continue using the new command word.

TCR[CONTC] = 1 is restricted in the sense that the new command must load on a frame boundary, and the only way for a transfer to continue from a frame boundary is when the previous command has TCR[CONT] = 1.

You can read the current state of the existing command word from Transmit Command (TCR). It requires at least 3 LPSPI functional clock cycles for TCR to update after TCR is written (assuming an empty FIFO), and LPSPI must be enabled (CR[MEN] = 1).

Writing the TCR does not initiate a SPI bus transfer, unless the TCR[TXMSK] = 1. When the TCR[TXMSK] = 1, a new command word is not loaded until the end of the existing frame (based on the TCR[FRAMESZ] value); at the end of the transfer, TCR[TXMSK] transitions to 0.

In Master mode, the LPSPI command word in TCR controls SPI attributes based on selections in register fields.

Table 70-3. Command word in Master mode

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
CPOL	Clock polarity	Specifies the polarity of the SCK pin. Any change of CPOL value causes a transition on the SCK pin.	N
CPHA	Clock phase	Specifies the clock phase of the transfer.	N
PRESCALE	Prescaler value	Specifies a prescaler used to divide the LPSPI functional clock, to generate the timing parameters of the SPI bus transfer. Changing PRESCALE in conjunction with PCS enables the LPSPI module to connect to different slave devices at different frequencies.	N
PCS	Peripheral chip select	Specifies which PCS pin asserts for the transfer; the polarity of PCS is static and specified by CFGR1[PCSPOL]. If CFGR1[PCSCFG] = 1, do not select PCS[3:2].	N
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.	Y
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing to devices that organize data as big-endian.	Y
CONT	Continuous transfer	Configures LPSPI for a continuous transfer that keeps PCS asserted between frames (as specified by FRAMESZ). You must write a new	Y

Table continues on the next page...

Table 70-3. Command word in Master mode (continued)

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
		command word to cause PCS to negate. Also supports changing the command word at the frame size boundaries.	
CONTC	Continuing command	Indicates that this is a new command word for the existing continuous transfer. When CONTC = 1, the command word must only be written to the transmit and command FIFO on a frame boundary.	Y
RXMSK	Receive data mask	Masks the receive data and does not store the masked receive data to the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.	Y
TXMSK	Transmit data mask	Masks the transmit data; masked transmit data is not pulled from the transmit FIFO, and the output data pin is 3-stated (unless otherwise configured by CFGR1[OUTCFG]). This option is useful for half-duplex transfers.	Y
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse. • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. • 2-bit and 4-bit half-duplex transfers are useful for interfacing to QuadSPI memory devices, and either TXMSK or RXMSK must also be 1.	Y
FRAMESZ	Frame size	Configures the frame size in number of bits equal to (FRAMESZ + 1). • The minimum frame size is 8 bits. • The minimum word size is 2 bits; a frame size of 33 bits (or similar) is not supported. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remaining bits. For example, a 72-bit transfer consists of 3 words: the first and second words are 32 bits, and the third word is 8 bits.	Y

70.3.1.1.1 SPI bus transfers

LPSPI initiates a SPI bus transfer when all of the following conditions are true:

- Data is written to the transmit FIFO.
- LPSPI is enabled.

To perform the SPI bus transfer, LPSPI uses the attributes configured in Transmit Command (TCR) and uses the timing parameters in Clock Configuration (CCR).

The SPI bus transfer ends after the number of bits indicated by the FRAMESZ value have been transferred (provided CONT = 0), or at the end of a word when a new transmit command word is at the top of the transmit and command FIFO. When LPSPI is disabled, the SPI bus transfers end after the transmit FIFO is empty and LPSPI is idle.

70.3.1.1.2 Circular FIFO

The transmit and command FIFO supports a circular FIFO feature. This feature enables the LPSPI master to (periodically) repeat a short data transfer that fits within the transmit and command FIFO, without requiring additional FIFO accesses. When the circular FIFO is enabled (CFGR0[CIRFIFO] = 1), the current state of the FIFO read pointer is saved and the status flags do not update. After the FIFO is empty and LPSPI is idle, the FIFO read pointer is restored with the saved version, so the contents of the transmit and command FIFO are not permanently pulled from the FIFO while Circular FIFO mode is enabled.

70.3.1.2 Receive FIFO and data match

The receive FIFO stores received data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of being stored in the receive FIFO.

- Received data is written to the receive FIFO at the end of the frame.
- During a multiple-word or continuous transfer, the received data is also written to the receive FIFO at the same time the new transmit data is read from the transmit FIFO.
- During a continuous transfer, if the transmit FIFO is empty, then the received data is only written to the receive FIFO after the transmit FIFO is written or after TCR is written to end the frame.

LPSPI provides a receive data match function that can match received data against one of two words in DMR0 and DMR1, or against a masked data word. The received data match function can also be configured to compare only the first one or two received data words since the start of the frame.

- Received data that is already discarded due to the TCR[RXMSK] field cannot cause the data match flag to set, and delays the receive data match on the first received data word, until all discarded data is received.
- You can configure the receive data match function to discard all received data until a data match is detected, using CFGR0[RDMO].
- After a receive data match, to allow all subsequent data to be received, write 0 to the CFGR0[RDMO] field, then write 0 to SR[DMF].

70.3.1.3 Timing parameters

The timing parameters that are used for all SPI bus transfers are relative to the LPSPI functional clock divided by the TCR[PRESCALE] selection. Although you cannot change Clock Configuration (CCR) when the LPSPI module is busy, to support interfacing to different slave devices at different frequencies, you can change the TCR[PRESCALE] selection between SPI bus transfers using Transmit Command (TCR).

NOTE

The minimum value below is the minimum counter value, but the clock configuration register values must also satisfy the data sheet specs based on the LPSPI functional clock frequency and prescaler value.

Table 70-4. Timing parameters

Clock Configuration (CCR)		Description	Min	Max
Field	Name			
SCKDIV	SCK divider	Configures the SCK clock period to (SCKDIV + 2) cycles. When SCK Divider is configured to an odd number of cycles, the first half of the SCK cycle is one cycle longer than the second half of the SCK cycle.	0 (2 cycles)	255 (257 cycles)
DBT	Delay between transfers (PCS negation to PCS assertion)	Configures the minimum delay between PCS negation and the next PCS assertion to (DBT + 2) cycles. When the command word is updated between transfers, there is a minimum of (DBT ÷ 2) + 1 cycles between the command word update and any change on PCS pins.	0 (2 cycles)	255 (257 cycles)
DBT	Delay between transfers (last SCK edge to first SCK edge)	Configures the delay during a continuous transfer between the last SCK edge of a frame and the first SCK edge of the continuing frame to (DBT + 1) cycles. This is useful when the external slave requires a large delay between different words of an SPI bus transfer.	0 (1 cycle)	255 (256 cycles)
PCSSCK	PCS-to-SCK delay	Configures the minimum delay between PCS assertion and the first SCK edge to (PCSSCK + 1) cycles.	0 (1 cycle)	255 (256 cycles)
SCKPCS	SCK-to-PCS delay	Configures the minimum delay between the last SCK edge and the PCS negation to (SCKPCS + 1) cycles.	0 (1 cycle)	255 (256 cycles)

This figure shows the timing settings controlled by:

- TCR[CPHA]
- TCR[CPOL]
- CCR[SCKPCS]
- CCR[PCSSCK]

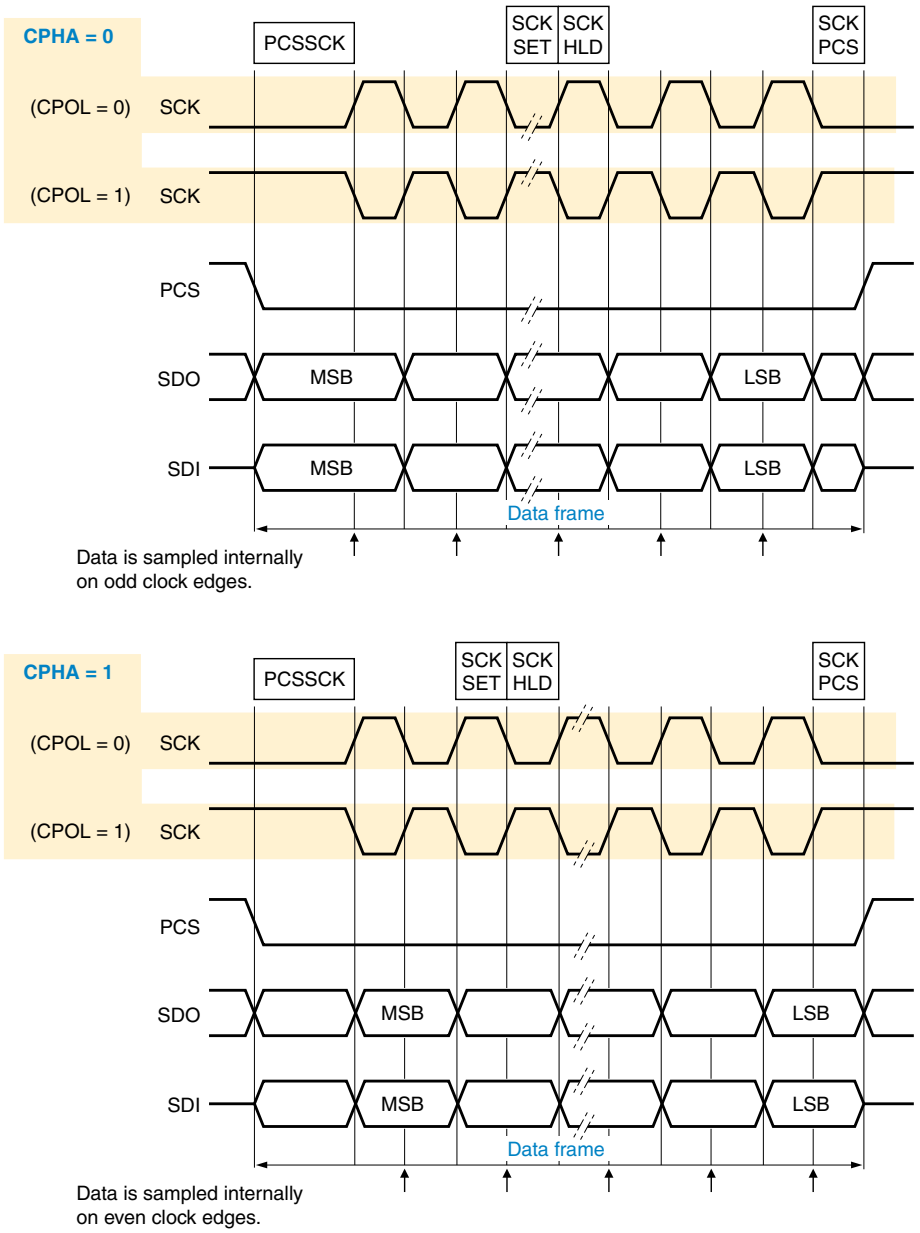


Figure 70-2. Clock phase (TCR[CPHA]) timing diagram example

70.3.1.4 Pin configuration

- To swap directions or support half-duplex transfers on the same pin, you can configure the SIN and SOUT pins using CFGR1[PINCFG].
- To specify whether an output data pin (SOUT, for example) 3-states when PCS is negated, or if the output data pin retains the last value, use CFGR1[OUTCFG].
- When configuring for half-duplex transfers, you must configure the output data pins to 3-state when PCS is negated (CFGR1[OUTCFG] = 1).

- When performing half-duplex 2-bit transfers, you can set CFGR1[PCSCFG] to any value.
- When performing half-duplex 4-bit transfers, you must set CFGR1[PCSCFG] to 1h.

70.3.1.5 Clock loopback

Configure the LPSPI master to use one of the following clocks to sample the input data:

- The SCK output clock
- A delayed version of the SCK output clock

The delayed version of the SCK is chosen by the SCK pin output delay, plus the SCK pin input delay, and is selected by writing 1 to CFGR1[SAMPLE]. Enabling the loopback version of the SCK can improve the setup time of the input data from the slave.

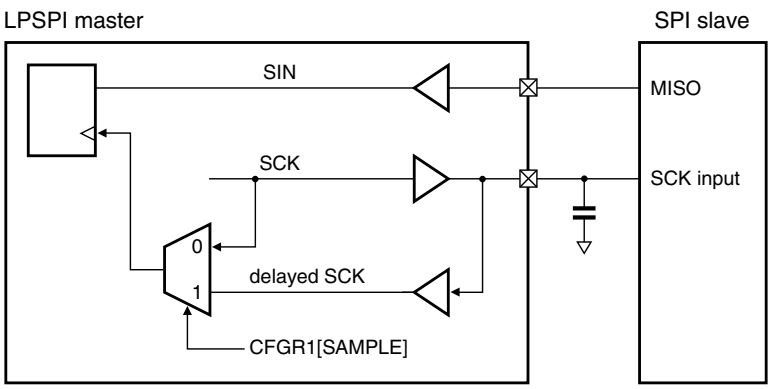


Figure 70-3. Clock loopback

See the chip data sheet for the specific input setup time in Master Loopback mode.

70.3.2 Slave mode

LPSPI slave mode:

- Uses the same shift register and logic that master mode uses.
- Does not use Clock Configuration Register (CCR).
- Requires that Transmit Command (TCR) remain static (unchanging) during SPI bus transfers.

70.3.2.1 Transmit and command FIFO commands

Before enabling LPSPI in Slave mode, initialize Transmit Command (TCR), although TCR does not update until after LPSPI is enabled. After LSPI is enabled, only change TCR when LPSPI is idle. In Slave mode, the LPSPI command word in TCR controls SPI attributes. Before the PCS input asserts, the transmit FIFO must be filled with transmit data, or the transmit error flag sets.

Table 70-5. Command word in Slave mode

Transmit Command (TCR)		Description
Field	Name	
CPOL	Clock polarity	Specifies the polarity of the external SCK input.
CPHA	Clock phase	Specifies the clock phase of transfer.
PRESCALE	Prescaler value	Specifies the LPSPI functional clock prescaler.
PCS	Peripheral chip select	Specifies which PCS is used. The polarity of PCS is static and configured by CFGR1[PCSPOL]. If CFGR1[PCSCFG] is not equal to zero, then do not select the PCS[3:2] pins.
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing to devices that organize data as big-endian.
CONT	Continuous transfer	When continuous transfer is selected in Slave mode, after the number of bits indicated by FRAMESZ are transferred, the LPSPI will passthrough and transmit the received data until the next PCS negation. Whatever is shifted in on the receive data is shifted out as transmit data as if there were a 32-bit shift register.
CONTC	Continuing command	This field is reserved in Slave mode.
RXMSK	Receive data mask	Masks the receive data; LPSPI does not store masked receive data to the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.
TXMSK	Transmit data mask	Masks the transmit data, so that the masked transmit data is not pulled from transmit FIFO, and the output data pin is 3-stated (unless otherwise specified by CFGR1[OUTCFG]). This option is useful for half-duplex transfers.
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse. • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats.
FRAMESZ	Frame size	Specifies the frame size in number of bits equal to (FRAMESZ + 1). • The minimum frame size is 8 bits. • The minimum word size is 2 bits; a frame size of 33 bits (or similar) is not supported. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contain the remainder bits. For example, a 72-bit transfer consists of 3 words: the first and second words are 32 bits, and the third word is 8 bits.

70.3.2.2 Receive FIFO and data match

The receive FIFO stores receive data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of storing the received data in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of two words in the DMR0 and DMR1 registers or against a masked data word. The data match function can also be configured to compare only the first one or two received data words since the start of the frame.

- Received data that is already discarded because TCR[RXMSK] = 1 cannot cause the data match to set, and delays the match on the first received data word, until all discarded data is received.
- The receiver match function can also be configured to discard all received data until a data match is detected, using the CFGR0[RDMO] bit.
- After a receive data match, to allow all subsequent data to be received, first clear the CFGR0[RDMO] bit, then clear the SR[DMF] bit.

70.3.2.3 Clocked interface

LPSPI supports interfacing to external masters that provide only clock and data pins (PCS is not required). This interface requires:

- Writing 1 to TCR[CPHA] (data is changed on the leading edge of SCK and captured on the following edge).
- Configuring the PCS input to be always asserted (CFGR1[PCSPOLn] = 1). For example, to configure PCS[0] to be always asserted, write 1 to PCSPOL[0], and do not configure PCS[0] in the pin muxing. The chip-level drives PCS to a certain value (ideally 1), CFGR1[PCSPOL*n*] could be used to invert that value.
- Writing 1 to CFGR1[AUTOPCS] to enable automatic PCS generation. When CFGR1[AUTOPCS] = 1, a minimum of four LPSPI functional clock cycles (divided by the TCR[PRESCALE] selection) is required between the last SCK edge of one word and the first SCK edge of the next word.

70.3.3 Low-power modes

Table 70-6. Low-power modes

Chip mode	LPSPI operation
Run	Normal operation
Stop	Can continue operating in Stop mode if CR[DOZEN] = 0 and LPSPI is using an external or internal clock source that remains operating during Stop mode

70.3.4 Debug mode

Table 70-7. Debug mode

Chip mode	LPSPI operation
Debug (the core is in Debug/Halted mode)	Can continue operating in Debug mode, if the CR[DBGEN] = 1.

70.3.5 Interrupts and DMA requests

The following table lists the Slave mode sources (status flags) that can generate LPSPI interrupts and LPSPI slave transmit and receive DMA requests.

Table 70-8. Interrupts and DMA requests

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
TDF	Transmit data flag	Data can be written to transmit FIFO, as configured by the transmit FIFO watermark FCR[TXWATER]	Y	TX	Y
RDF	Receive data flag	Data can be read from the receive FIFO, as configured by the receive FIFO watermark FCR[RXWATER]	Y	RX	Y
WCF	Word complete flag	Word is complete, the last bit of the word has been sampled	Y	N	Y
FCF	Frame complete flag	Frame is complete, and PCS has deasserted	Y	N	Y
TCF	Transfer complete flag	Transfer is complete, PCS has deasserted, and the transmit and command FIFO is empty	Y	N	Y
TEF	Transmit error flag	Indicates a transmit and command FIFO underrun. In Master mode when CFGR1[NOSTALL] = 0 (transfers stall when transmit FIFO is empty), the Transmit Error Flag cannot set.	Y	N	Y

Table continues on the next page...

Table 70-8. Interrupts and DMA requests (continued)

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
REF	Receive error flag	Indicates a receive FIFO overflow.	Y	N	Y
DMF	Data match flag	Indicates that the received data has matched the configured data match value	Y	N	Y
MBF	Module busy flag	LPSPI is busy performing a SPI bus transfer	N	N	N

70.3.6 Clocks

Table 70-9. LPSPI clocks

LPSPI functional clock	• Asynchronous to the bus clock. • If the LPSPI functional clock remains enabled in low-power modes, then LPSPI can perform SPI bus transfers and low-power wakeups, in both Master and Slave modes. • LPSPI divides the functional clock by a prescaler; the resulting frequency must be at least 2 times faster than the SPI external clock frequency (SCK).

External clock	• The LPSPI shift register is clocked directly by the SCK clock.
	• How the SCK clock is generated or supplied depends upon the mode (Master or Slave):
	  • In Master mode: the SCK clock is generated internally.
	  • In Slave mode: the SCK clock is supplied externally.

Bus clock	The bus clock is only used for bus accesses to the LPSPI control and configuration registers. The bus clock frequency must be high enough to support the data bandwidth requirements of the LPSPI registers, including the FIFOs.

See the chip-specific LPSPI information.

70.3.7 Resets

Table 70-10. Resets

Chip reset	Resets the LPSPI logic and registers to their default state.
Software reset	• Resets the LPSPI logic and registers to their default state, except for the Control Register. • The LPSPI software reset is controlled using CR[RST].
FIFO resets	• Resets the transmit and command FIFO and the receive FIFO. • The Reset Transmit FIFO field, CR[RTF], and the Reset Receive FIFO field, CR[RRF], are write only. • After being reset, a FIFO is empty.

70.3.8 Peripheral triggers

The connection of the LPSPI peripheral triggers to other peripherals depend upon the specific device being used.

Table 70-11. Peripheral triggers

Trigger	Description	Notes
Frame output trigger	The frame output trigger is an idle high signal, which: • Asserts at the end of each frame (when PCS deasserts) • Remains asserted until PCS next asserts	LPSPI generates two output triggers that can be connected to other peripherals on the chip.
Word output trigger	The word output trigger: • Asserts at the end of each received word • remains asserted for 1 SCK period	

70.4 Signals

Table 70-12. Signals

Signal	Name	Description	I/O
SCK	Serial clock	• Input in Slave mode • Output in Master mode	I/O
PCS[0]	Peripheral chip select	• Input in Slave mode • Output in Master mode	I/O
PCS[1]	Peripheral chip select	• Input in Slave mode • Output in Master mode	I/O
PCS[2] / DATA[2]	Peripheral chip select or data pin 2 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Slave mode • Output in Master mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers • Output in half-duplex parallel data transmit transfers	I/O
PCS[3] / DATA[3]	Peripheral chip select or data pin 3 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Slave mode • Output in Master mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers • Output in half-duplex parallel data transmit transfers	I/O
SOUT / DATA[0]	Serial data output	Can be configured as serial data input signal • Used as data pin 0 in half-duplex parallel data transfers	I/O
SIN / DATA[1]	Serial data input	Can be configured as serial data output signal	I/O

Table 70-12. Signals

Signal	Name	Description	I/O
		• Used as data pin 1 in half-duplex parallel data transfers	

70.5 Memory map and registers

NOTE

- Writing to a read-only register or reading a write-only register can cause bus errors.
- LPSPI does not check values programmed in registers for validity, so you must take care to write valid values only.

70.5.1 LPSPI register descriptions

70.5.1.1 LPSPI memory map

LPSPI1 base address: 4011_4000h

LPSPI2 base address: 4011_8000h

LPSPI3 base address: 4011_C000h

LPSPI4 base address: 4012_0000h

LPSPI5 base address: 40C2_C000h

LPSPI6 base address: 40C3_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	RO	0102_0004h
4h	Parameter (PARAM)	32	Table 70-12	0004_0404h
10h	Control (CR)	32	Table 70-12	0000_0000h
14h	Status (SR)	32	Table 70-12	0000_0001h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
18h	Interrupt Enable (IER)	32	Table 70-12	0000_0000h
1Ch	DMA Enable (DER)	32	Table 70-12	0000_0000h
20h	Configuration 0 (CFGR0)	32	Table 70-12	0000_0000h
24h	Configuration 1 (CFGR1)	32	Table 70-12	0000_0000h
30h	Data Match 0 (DMR0)	32	RW	0000_0000h
34h	Data Match 1 (DMR1)	32	RW	0000_0000h
40h	Clock Configuration (CCR)	32	RW	0000_0000h
58h	FIFO Control (FCR)	32	Table 70-12	0000_0000h
5Ch	FIFO Status (FSR)	32	Table 70-12	0000_0000h
60h	Transmit Command (TCR)	32	Table 70-12	0000_001Fh
64h	Transmit Data (TDR)	32	WO	0000_0000h
70h	Receive Status (RSR)	32	Table 70-12	0000_0002h
74h	Receive Data (RDR)	32	RO	0000_0000h

70.5.1.2 Version ID (VERID)

70.5.1.2.1 Offset

Register	Offset
VERID	0h

70.5.1.2.2 Function

Contains version numbers for the module design and feature set.

70.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

70.5.1.2.4 Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the module specification.
15-0 FEATURE	Module Identification Number Indicates the feature set number. 0000_0000_0000_0100b - Standard feature set supporting a 32-bit shift register. All other values are reserved.

70.5.1.3 Parameter (PARAM)

70.5.1.3.1 Offset

Register	Offset
PARAM	4h

70.5.1.3.2 Function

Contains:

- Number of PCS pins
- Receive FIFO size
- Transmit FIFO size

70.5.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PCSNUM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

70.5.1.3.4 Fields

Field	Function
31-24 —	Reserved
23-16 PCSNUM	PCS Number Indicates the number of PCS pins supported.
15-8 RXFIFO	Receive FIFO Size Indicates the maximum number of words in the receive FIFO. The maximum number of words is 2^{RXFIFO}.
7-0 TXFIFO	Transmit FIFO Size Indicates the maximum number of words in the transmit FIFO. The maximum number of words is 2^{TXFIFO}.

70.5.1.4 Control (CR)

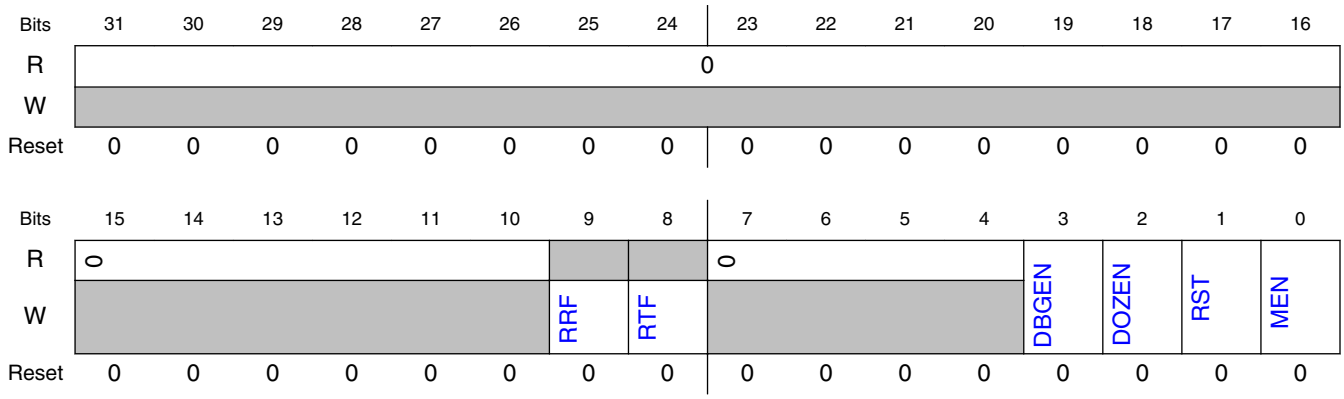
70.5.1.4.1 Offset

Register	Offset
CR	10h

70.5.1.4.2 Function

Contains fields that control module operation.

70.5.1.4.3 Diagram



70.5.1.4.4 Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Deletes all entries in the receive FIFO. This field always reads 0. 0b - No effect 1b - Reset
8 RTF	Reset Transmit FIFO Deletes all entries in the transmit FIFO. This field always reads 0. 0b - No effect 1b - Reset
7-4 —	Reserved
3 DBGEN	Debug Enable Enables LPSPI in when the CPU is in Debug mode. When this field is 0, LPSPI is disabled when the CPU is halted; the PCS pin is deasserted after the transmit FIFO is empty regardless of the state of Transmit Command (TCR). Update this field only when the LPSPI module is disabled (MEN = 0). 0b - Disable 1b - Enable
2 DOZEN	Doze Mode Enable Enables LPSPI when the chip is in Doze mode. Update this field only when the LPSPI module is disabled (MEN = 0). 0b - Enable 1b - Disable
1 RST	Software Reset Resets all internal logic and registers, except Control (CR). The reset takes effect immediately and remains asserted until you write 0 to it. There is no minimum delay required before clearing the software reset by writing 0.

Table continues on the next page...

Field	Function
	0b - Not reset 1b - Reset
0 MEN	Module Enable After writing 0, MEN remains set until the LPSPI has completed the current transfer and is idle. 0b - Disable 1b - Enable

70.5.1.5 Status (SR)

70.5.1.5.1 Offset

Register	Offset
SR	14h

70.5.1.5.2 Function

Contains data flow status.

70.5.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DMF	REF	TEF	TCF	FCF	WCF	0						RDF	TDF
W			W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

70.5.1.5.4 Fields

Field	Function
31-25 —	Reserved
24	Module Busy Flag

Table continues on the next page...

Field	Function
MBF	In Master mode, indicates that there is data to transmit and LPSPI is able to transmit (for example, HREQ is asserted). It deasserts after the PCS pin deasserts and the LPSPI master has waited half the CCR[DBT] time with no new data to transmit. Slave mode asserts this flag when LPSPI is enabled and PCS is asserted. 0b - LPSPI is idle 1b - LPSPI is busy
23-14 —	Reserved
13 DMF	Data Match Flag Indicates that received data matches DMR0[MATCH0] and/or DMR1[MATCH1] (as configured by CFGR1[MATCFG]). 0b - No match 1b - Match
12 REF	Receive Error Flag Indicates a receive FIFO overflow error. When this flag is set: 1. End the transfer. 2. Empty the Receive FIFO. 3. Clear this flag. 4. Restart the transfer from the beginning. 0b - No overflow 1b - Overflow
11 TEF	Transmit Error Flag Indicates a Transmit FIFO underrun error. When this flag is set: 1. End the transfer. 2. Clear this flag. 3. Restart the transfer from the beginning. 0b - No underrun 1b - Underrun
10 TCF	Transfer Complete Flag In Master mode, indicates that all transfers have completed and that LPSPI has returned to Idle state and the transmit FIFO is empty. 0b - Not complete 1b - Complete
9 FCF	Frame Complete Flag Indicates that a frame transfer has completed, when the PCS deasserts. 0b - Not complete 1b - Complete
8 WCF	Word Complete Flag Indicates that the last bit of a received word is sampled. 0b - Not complete 1b - Complete
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates that the number of words in the receive FIFO is greater than the value in FCR[RXWATER]. 0b - Receive data not ready

Table continues on the next page...

Field	Function
	1b - Receive data is ready
0 TDF	Transmit Data Flag Indicates that the number of words in the transmit FIFO is equal to or less than the value in FCR[TXWATER]. 0b - Transmit data not requested 1b - Transmit data is requested

70.5.1.6 Interrupt Enable (IER)

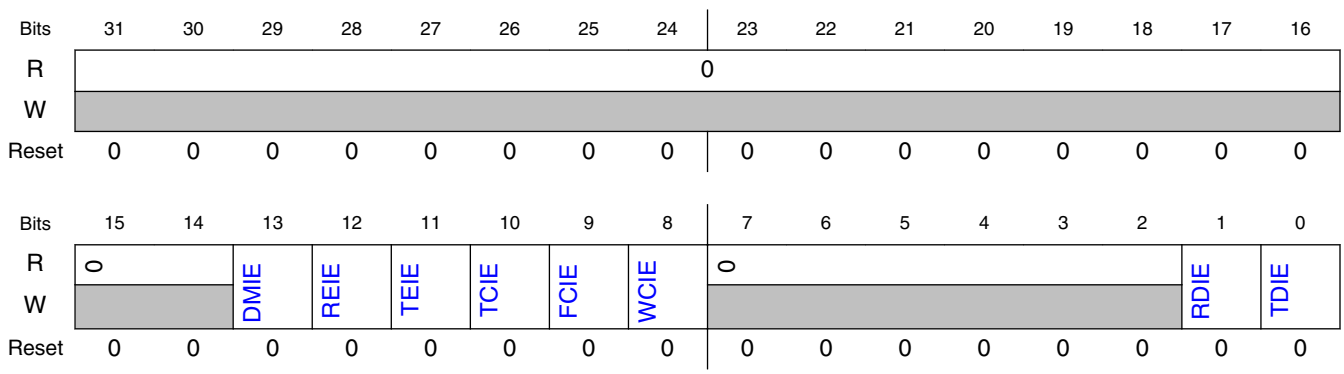
70.5.1.6.1 Offset

Register	Offset
IER	18h

70.5.1.6.2 Function

Enables interrupts based on data flow and errors.

70.5.1.6.3 Diagram



70.5.1.6.4 Fields

Field	Function
31-14 —	Reserved
13	Data Match Interrupt Enable 0b - Disable

Table continues on the next page...

Field	Function
DMIE	1b - Enable
12 REIE	Receive Error Interrupt Enable 0b - Disable 1b - Enable
11 TEIE	Transmit Error Interrupt Enable 0b - Disable 1b - Enable
10 TCIE	Transfer Complete Interrupt Enable 0b - Disable 1b - Enable
9 FCIE	Frame Complete Interrupt Enable 0b - Disable 1b - Enable
8 WCIE	Word Complete Interrupt Enable 0b - Disable 1b - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable 0b - Disable 1b - Enable

70.5.1.7 DMA Enable (DER)

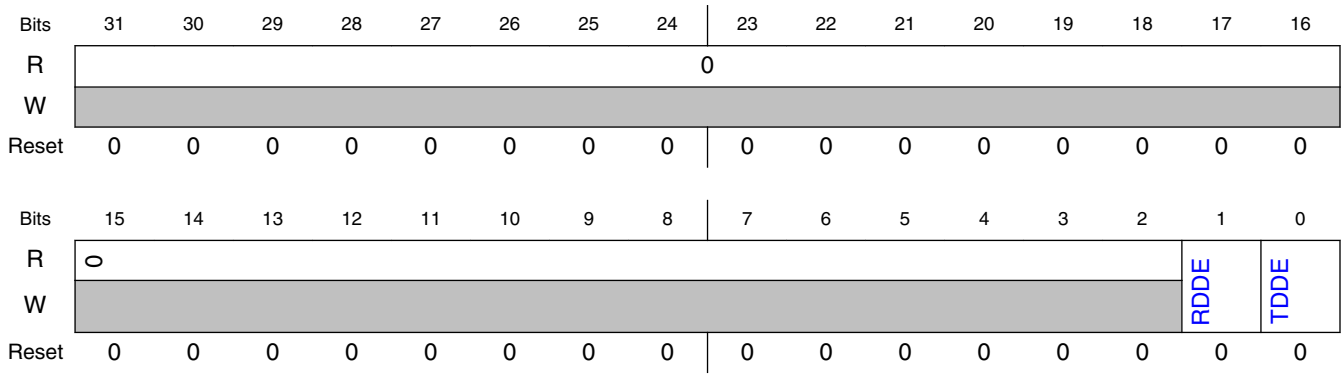
70.5.1.7.1 Offset

Register	Offset
DER	1Ch

70.5.1.7.2 Function

Enables DMA data flow.

70.5.1.7.3 Diagram



70.5.1.7.4 Fields

Field	Function
31-2 —	Reserved
1 RDDE	Receive Data DMA Enable 0b - Disable 1b - Enable
0 TDDE	Transmit Data DMA Enable 0b - Disable 1b - Enable

70.5.1.8 Configuration 0 (CFGR0)

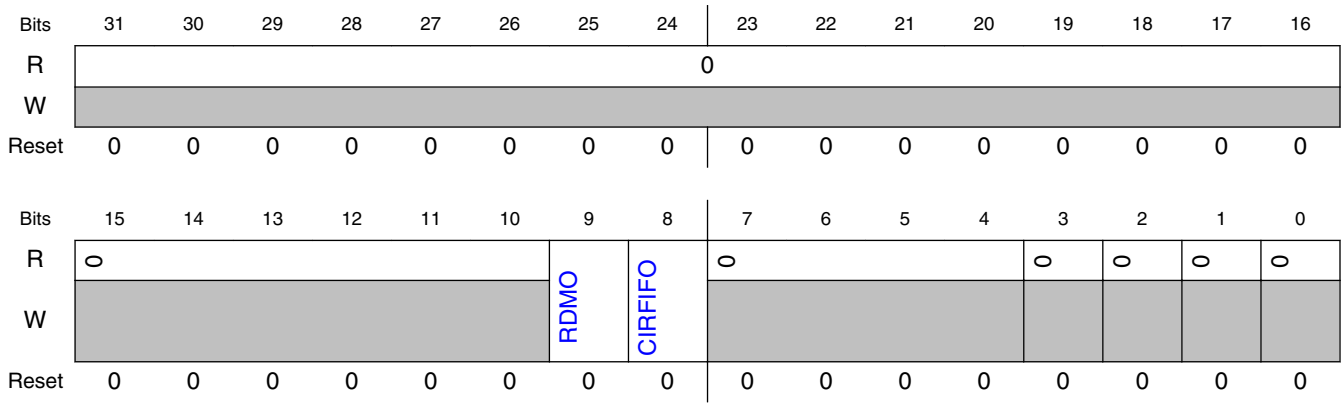
70.5.1.8.1 Offset

Register	Offset
CFGR0	20h

70.5.1.8.2 Function

Includes fields to configure LPSPI.

70.5.1.8.3 Diagram



70.5.1.8.4 Fields

Field	Function
31-10 —	Reserved
9 RDMO	Receive Data Match Only When enabled, all received data that does not cause the Data Match Flag (SR[DMF]) to assert is discarded. • Write 1 to this field when LPSPI is idle and SR[DMF] = 0. • After SR[DMF] = 1, this field is ignored. • To ensure that no receive data is lost when disabling RDMO, write 0 to this field before clearing SR[DMF]. See CFGR1[MATCFG] for the received data matching options. When disabled, all received data is stored in the receive FIFO. 0b - Disable 1b - Enable
8 CIRFIFO	Circular FIFO Enable When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO is emptied as in normal operation, but when LPSPI is idle and the transmit FIFO is empty, the read pointer value is restored from the temporary register. This restoring of the read pointer causes the contents of the transmit FIFO to be cycled through repeatedly. **NOTE:** The read pointer is restored for as long as this field is 1. Writing additional words to the FIFO when this field is 1 adds them to the end of the FIFO, up to the size of the transmit FIFO. 0b - Disable 1b - Enable
7-4 —	Reserved
3 —	Reserved
2	Reserved

Table continues on the next page...

Field	Function
—	
1 —	Reserved
0 —	Reserved

70.5.1.9 Configuration 1 (CFGR1)

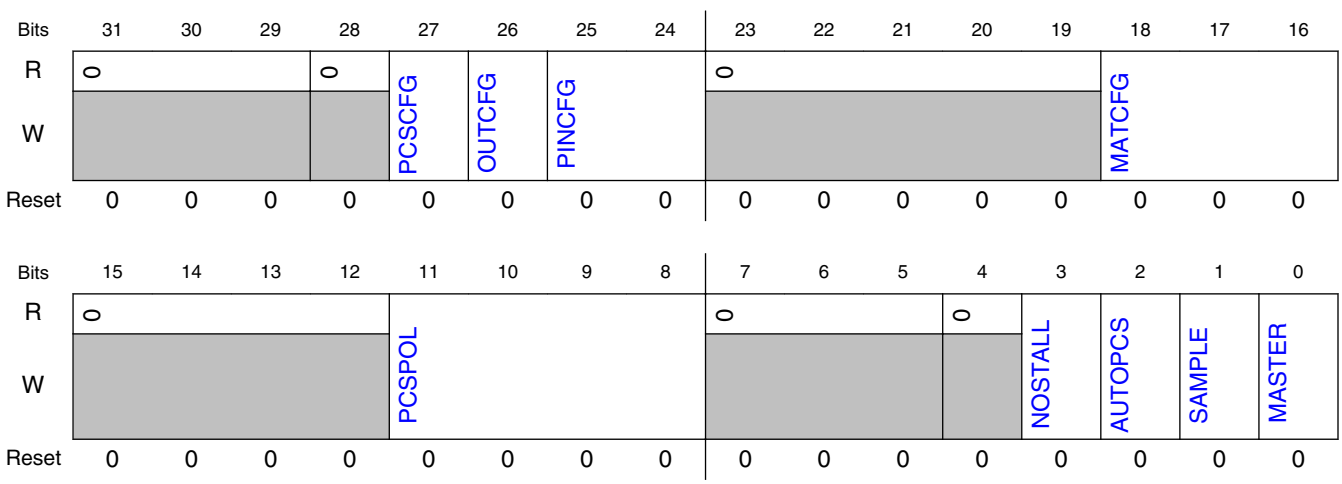
70.5.1.9.1 Offset

Register	Offset
CFGR1	24h

70.5.1.9.2 Function

Includes fields to configure LPSPI. Write to this register only when LPSPI is disabled.

70.5.1.9.3 Diagram



70.5.1.9.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>—</td><td></td></tr>
<tr><td>28
—</td><td>Reserved</td></tr>
<tr><td>27
PCSCFG</td><td>Peripheral Chip Select Configuration
Specifies the PCS pin configuration. When performing parallel transfers, this field must be configured to enable the desired transfer.
0b - PCS[3:2] are configured for chip select function
1b - PCS[3:2] are configured for half-duplex 4-bit transfers (PCS[3:2] = DATA[3:2])</td></tr>
<tr><td>26
OUTCFG</td><td>Output Configuration
Specifies whether the output data is 3-stated between accesses (when PCS is deasserted). When performing half-duplex transfers, this field must be 1.
0b - Output data retains last value.
1b - Output data is 3-stated.</td></tr>
<tr><td>25-24
PINCFG</td><td>Pin Configuration
Specifies the pins used for input and output data during serial transfers.
NOTE: When performing parallel transfers, this field must be 00b.
00b - SIN is used for input data; SOUT is used for output data.
01b - SIN is used for both input and output data. Only half-duplex serial transfers are supported.
10b - SOUT is used for both input and output data. Only half-duplex serial transfers are supported.
11b - SOUT is used for input data; SIN is used for output data.</td></tr>
<tr><td>23-19
—</td><td>Reserved</td></tr>
<tr><td>18-16
MATCFG</td><td>Match Configuration
Specifies the condition that causes SR[DMF] to assert.
<table><tr><th>Condition</th><th>Description</th></tr><tr><td>Match first data word with compare word</td><td>Match if first data word equals MATCH0 logically ORed with MATCH1
first_data_word == (MATCH0 || MATCH1)</td></tr><tr><td>Match any data word with compare word</td><td>Match if any data word equals MATCH0 logically ORed with MATCH1
any_data_word == (MATCH0 || MATCH1)</td></tr><tr><td>Sequential match, first data word</td><td>Match if first data word equals MATCH0, and second data word equals MATCH1
(first_data_word == MATCH0) && (second_data_word == MATCH1)</td></tr><tr><td>Sequential match, any data word</td><td>Match if any data word equals MATCH0, and the next data word equals MATCH1
(any_data_word == MATCH0) && (next_data_word == MATCH1)</td></tr><tr><td>Match first data word (masked) with compare word (masked)</td><td>Match if first data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1
(first_data_word && MATCH1) == (MATCH0 && MATCH1)</td></tr><tr><td>Match any data word (masked) with compare word (masked)</td><td>Match if any data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1
(any_data_word && MATCH1) == (MATCH0 && MATCH1)</td></tr></table>000b - Match is disabled</td></tr>
</table>

Table continues on the next page...

Field	Function
	001b - Reserved 010b - Match first data word with compare word 011b - Match any data word with compare word 100b - Sequential match, first data word 101b - Sequential match, any data word 110b - Match first data word (masked) with compare word (masked) 111b - Match any data word (masked) with compare word (masked)
15-12 —	Reserved
11-8 PCSPOL	Peripheral Chip Select Polarity Specifies the polarity of each PCS pin. Bit *n* in this field (the least-significant bit is bit 0) corresponds to PCS[*n*]. For each PCSPOL bit: 0b – PCS[*n*] pin is active low 1b – PCS[*n*] pin is active high **NOTE:** The entire PCSPOL field is not fully supported in every LPSPI module instance. See the LSPI chip-specific information.
7-5 —	Reserved
4 —	Reserved
3 NOSTALL	No Stall Disables a normal operating feature that causes LSPI, when in Master mode, to stall transfers when the transmit FIFO is empty. This feature prevents transmit FIFO underruns. Writing 1 to this field disables this functionality. 0b - Disable 1b - Enable
2 AUTOPCS	Automatic PCS Enables automatic PCS generation. For correct operation in Slave mode, LPSPI requires the PCS signal to deassert between frames. Writing 1 to this field generates an internal PCS signal at the end of each transfer word when TCR[CPHA] = 1. When this field is 1, SCK must remain idle for at least 4 LPSPI functional clock cycles, divided by the prescaler selected (see TCR[PRESCALE]) between each word, to ensure correct operation. This field is ignored in Master mode. 0b - Disable 1b - Enable
1 SAMPLE	Sample Point Specifies the SCK clock edge on which LSPI, when in Master mode, samples input data. Writing 1 to this field causes LSPI to sample input data on a delayed loopback SCK clock edge, which improves the setup time when sampling data (see Clock loopback). In this configuration. The input data setup time in Master mode is equal to the input data setup time in Slave mode. In Slave mode, this field is ignored. **NOTE:** When SAMPLE = 1 both the input buffer and output buffer must be enabled for the SCK pin. 0b - SCK edge 1b - Delayed SCK edge
0	Master Mode

Field	Function
MASTER	Specifies the LPSPI operating mode—either Master or Slave. This field directly controls the direction of the SCK and PCS pins. 0b - Slave mode 1b - Master mode

70.5.1.10 Data Match 0 (DMR0)

70.5.1.10.1 Offset

Register	Offset
DMR0	30h

70.5.1.10.2 Function

Specifies match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register while CFGR0[RDMO] = 1.

70.5.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

70.5.1.10.4 Fields

Field	Function
31-0 MATCH0	Match 0 Value MATCH0 value to be compared against received data.

70.5.1.11 Data Match 1 (DMR1)

70.5.1.11.1 Offset

Register	Offset
DMR1	34h

70.5.1.11.2 Function

Specifies match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register while CFGR0[RDMO] = 1.

70.5.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

70.5.1.11.4 Fields

Field	Function
31-0 MATCH1	Match 1 Value MATCH1 value to be compared against received data.

70.5.1.12 Clock Configuration (CCR)

70.5.1.12.1 Offset

Register	Offset
CCR	40h

70.5.1.12.2 Function

Contains clock configuration fields. These fields are only used in Master mode and you can only change them when LPSPI is disabled (CR[MEN] = 0).

70.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DBT								SCKDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

70.5.1.12.4 Fields

Field	Function
31-24 SCKPCS	SCK-to-PCS Delay In Master mode: configures the delay from the last SCK edge to the PCS negation. • The delay is equal to (SCKPCS + 1) cycles of the LPSPI functional clock divided by the prescaler selected (see TCR[PRESCALE]). • The minimum delay is 1 cycle. See Figure 70-2.
23-16 PCSSCK	PCS-to-SCK Delay In Master mode: configures the delay from the PCS assertion to the first SCK edge. • The delay is equal to (PCSSCK + 1) cycles of the LPSPI functional clock divided by the prescaler selected (see TCR[PRESCALE]). • The minimum delay is 1 cycle. See Figure 70-2.
15-8	Delay Between Transfers

Table continues on the next page...

Field	Function

DBT	In Master mode:
	• Configures the delay from the PCS negation to the next PCS assertion.
	  • The delay is equal to (DBT + 2) cycles of the LPSPI functional clock divided by the prescaler selected (see TCR[PRESCALE]).
	  • The minimum delay is 2 cycles.
	  • Half of the delay occurs before PCS assertion and the other half of the delay occurs after PCS negation. If the command word is updated between 2 transfers, then the command word is updated halfway between the PCS negation of the last transfer and PCS assertion of the next transfer.
	  • The command word sets which PCS signal is used (of PCS[3:0]}, the polarity/phase of the SCK signal, and the prescaler selected.
	• Configures the delay from the last SCK edge of a transfer word and the first SCK edge of the next transfer word, in a continuous transfer.
	  • The delay is equal to (DBT + 1) cycles of the LPSPI functional clock divided by the prescaler selected (see TCR[PRESCALE]).
	  • The minimum delay is 1 cycle.

7-0 SCKDIV	SCK Divider In Master mode, configures the divide ratio of the SCK pin. • The SCK period is equal to (SCKDIV + 2) cycles of LPSPI functional clock divided by the prescaler selected (see TCR[PRESCALE]). • The minimum SCK period is 2 cycles. • If the SCK period is an odd number of cycles, then the first half of the SCK period is 1 cycle longer than the second half of the SCK period. Baud rate = Function clock / (2^PRESCALE * (SCKDIV+2))

70.5.1.13 FIFO Control (FCR)

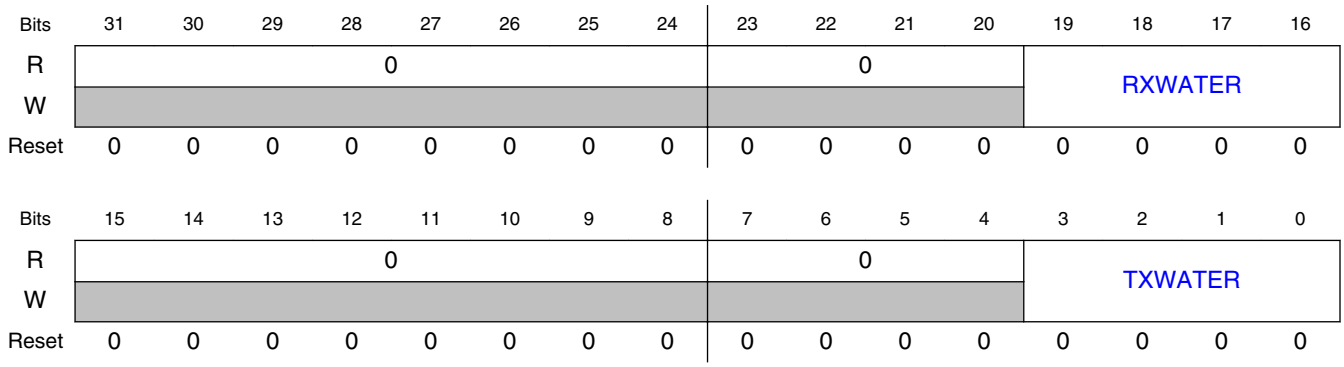
70.5.1.13.1 Offset

Register	Offset
FCR	58h

70.5.1.13.2 Function

Contains the receive FIFO and transmit FIFO watermark values.

70.5.1.13.3 Diagram



70.5.1.13.4 Fields

Field	Function
31-24 —	Reserved
23-20 —	Reserved
19-16 RXWATER	Receive FIFO Watermark Causes LPSPI to set the Receive Data Flag (SR[RDF]) when the number of words in the receive FIFO is greater than RXWATER. Writing a value equal or greater than the FIFO size truncates the written value.
15-8 —	Reserved
7-4 —	Reserved
3-0 TXWATER	Transmit FIFO Watermark Causes LPSPI to set the Transmit Data Flag (SR[TDF]) when the number of words in the transmit FIFO is equal or less than TXWATER. Writing a value equal or greater than the FIFO size truncates the written value.

70.5.1.14 FIFO Status (FSR)

70.5.1.14.1 Offset

Register	Offset
FSR	5Ch

70.5.1.14.2 Function

Contains fields that indicate the number of words currently stored in the receive FIFO and transmit FIFO.

70.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0			RXCOUNT				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			TXCOUNT				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

70.5.1.14.4 Fields

Field	Function
31-24 —	Reserved
23-21 —	Reserved
20-16 RXCOUNT	Receive FIFO Count Indicates the number of words currently stored in the receive FIFO.
15-8 —	Reserved
7-5 —	Reserved
4-0 TXCOUNT	Transmit FIFO Count Indicates the number of words currently stored in the transmit FIFO.

70.5.1.15 Transmit Command (TCR)

70.5.1.15.1 Offset

Register	Offset
TCR	60h

70.5.1.15.2 Function

Writes to either this register or Transmit Data (TDR) push the data into the transmit FIFO, in the order written. Only write to this register using 32-bit writes. Writes are tagged and cause the command register to update, after that entry reaches the top of the FIFO and the LPSPI is enabled. This allows changes to the command word and the transmit data itself to be interleaved. That is, the writes to the two registers can be interleaved (write command word, then data word, then command word, and so on). Changing the command word causes all subsequent SPI bus transfers to be performed using the new command word.

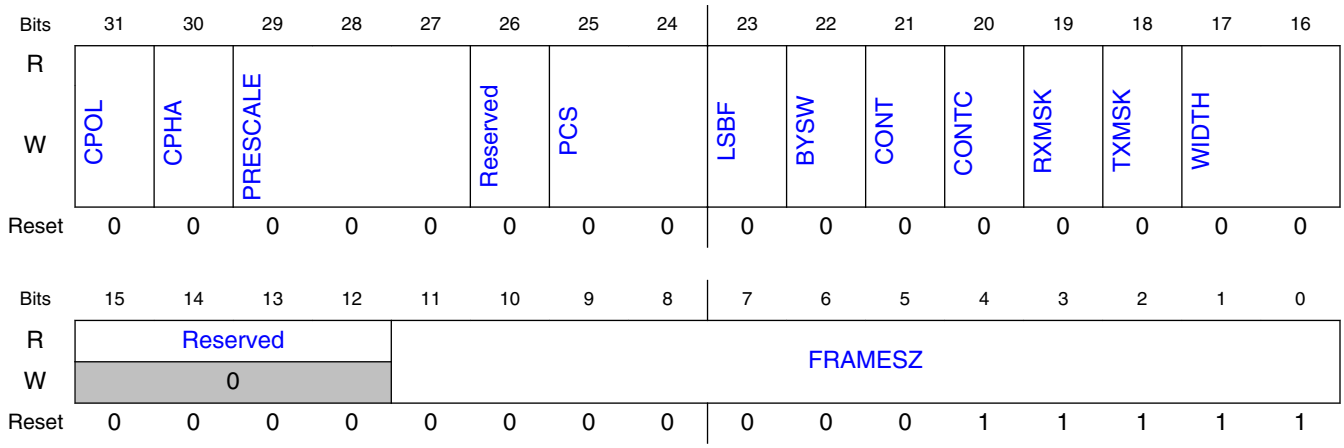
- **In Master mode,** writing a new command word does not initiate a new transfer, unless TXMSK is 1. Transfers are initiated by transmit data in the transmit FIFO, or by a new command word (with TXMSK = 1). Hardware writes 0 to TXMSK when the PCS deasserts.
- **In Master mode,** if the command word is changed before an existing frame has completed, then the existing frame terminates and the command word then updates. The command word can be changed during a continuous transfer, if CONTC of the new command word is 1 and the command word is written on a frame size boundary.
- **In Slave mode,** the command word should be changed only when LPSPI is idle and there is no SPI bus transfer.

Avoid resetting the Transmit FIFO after writing to the Transmit Command Register, wait for the command register to update from the FIFO first.

Avoid register reading problems: Reading the Transmit Command Register returns the current state of the command register. Reading the Transmit Command Register at the same time that the Transmit Command Register is loaded from the transmit FIFO, can return an incorrect Transmit Command Register value. It is recommended to:

- Read the Transmit Command Register when the transmit FIFO is empty,
- Read the Transmit Command Register more than once and then compare the returned values.

70.5.1.15.3 Diagram



70.5.1.15.4 Fields

Field	Function
31 CPOL	Clock Polarity Specifies the value of SCK when it is idle. This field can only be updated when PCS deasserted. See Figure 70-2. 0b - Inactive low 1b - Inactive high
30 CPHA	Clock Phase This field can only be updated when PCS deasserted. See Figure 70-2. 0b - Captured. Data is captured on the leading edge of SCK and changed on the following edge of SCK 1b - Changed. Data is changed on the leading edge of SCK and captured on the following edge of SCK
29-27 PRESCALE	Prescaler Value For all SPI bus transfers, this value is applied to Clock Configuration (CCR) clock. This field can only be updated when PCS deasserted. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32 110b - Divide by 64 111b - Divide by 128
26 —	Reserved
25-24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. The Peripheral Chip Select field is only updated when PCS deasserted.

Table continues on the next page...

Field	Function
	NOTE: The entire PCS field is not fully supported in every LPSPI module instance. See the chip-specific LPSPI information. 00b - Transfer using PCS[0] 01b - Transfer using PCS[1] 10b - Transfer using PCS[2] 11b - Transfer using PCS[3]
23 LSBF	LSB First 0b - Data is transferred MSB first 1b - Data is transferred LSB first
22 BYSW	Byte Swap Swaps the contents of [31:24] with [7:0] and [23:16] with [15:8] for each transmit data word read from the FIFO and for each received data word stored to the FIFO (or compared with match registers). 0b - Disabled 1b - Enabled
21 CONT	Continuous Transfer • In Master mode, CONT keeps the PCS asserted at the end of the frame size, until a command word is received that starts a new frame. • In Slave mode, when CONT is enabled, LPSPI only transmits the first FRAMESZ bits; after which LPSPI transmits received data (assuming a 32-bit shift register) until the next PCS negation. 0b - Continuous transfer is disabled 1b - Continuous transfer is enabled
20 CONTC	Continuing Command In Master mode, this field enables the command word to be changed within a continuous transfer. • The initial command word must enable continuous transfer (CONT = 1). • The continuing command must have CONTC = 1. • The continuing command word must be loaded on a frame size boundary. For example, if the continuous transfer has a frame size of 64-bits, then a continuing command word must be loaded on a 64-bit boundary. 0b - Command word for start of new transfer 1b - Command word for continuing transfer
19 RXMSK	Receive Data Mask Masks receive data (receive data is not stored in receive FIFO). 0b - Normal transfer 1b - Receive data is masked
18 TXMSK	Transmit Data Mask Masks transmit data (no data is loaded from transmit FIFO and the output pin is 3-stated). In Master mode, TXMSK initiates a new transfer which cannot be aborted by another command word. TXMSK automatically transitions to 0 at the end of the transfer. 0b - Normal transfer 1b - Mask transmit data
17-16 WIDTH	Transfer Width Configures serial (1-bit) or parallel transfers. For half-duplex parallel transfers, either Receive Data Mask (RXMSK) or Transmit Data Mask (TXMSK) must be 1. 00b - 1-bit transfer 01b - 2-bit transfer 10b - 4-bit transfer 11b - Reserved
15-12	Reserved

Table continues on the next page...

Field	Function
—	
11-0 FRAMESZ	Frame Size Configures the frame size in number of bits equal to (FRAMESZ + 1). • The minimum frame size is 8 bits • The minimum word size is 2 bits; a frame size of 33 bits is not supported. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remainder bits. For example, a 72-bit transfer consists of 3 words: the first and second words are 32 bits, and the third word is 8 bits.

70.5.1.16 Transmit Data (TDR)

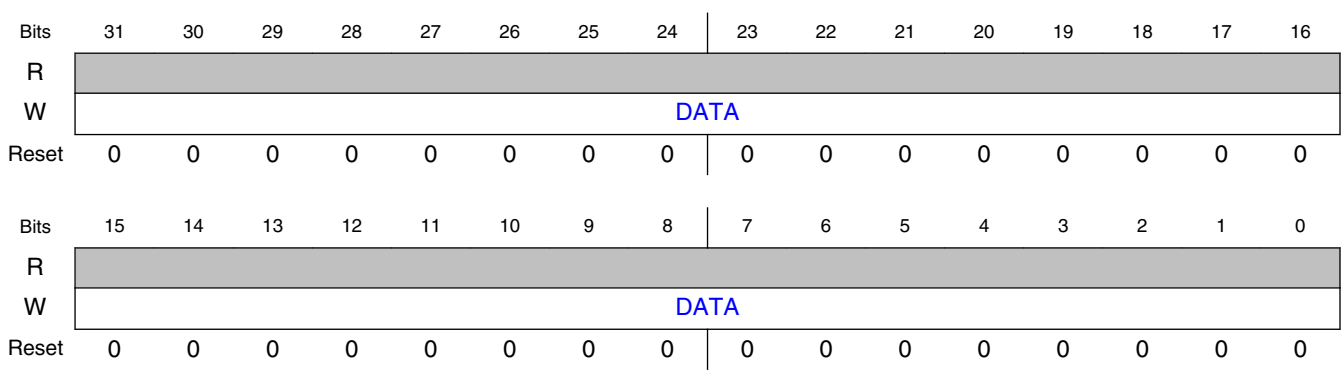
70.5.1.16.1 Offset

Register	Offset
TDR	64h

70.5.1.16.2 Function

Writes to either Transmit Command (TCR) or this register push the data into the transmit FIFO, in the order that the data is written. You can write to TDR using 32-, 16-, or 8-bit writes, but each write pushes data into the FIFO with zero pushed in unwritten bytes.

70.5.1.16.3 Diagram



70.5.1.16.4 Fields

Field	Function
31-0 DATA	Transmit Data Both 8- and 16-bit writes of transmit data zero-extend the data written and push the data into the transmit FIFO. To zero-extend 8- and 16-bit writes (to 32 bits), means that the higher order (most significant) empty parts of the 8-bit and 16-bit writes are filled with zeroes.

70.5.1.17 Receive Status (RSR)

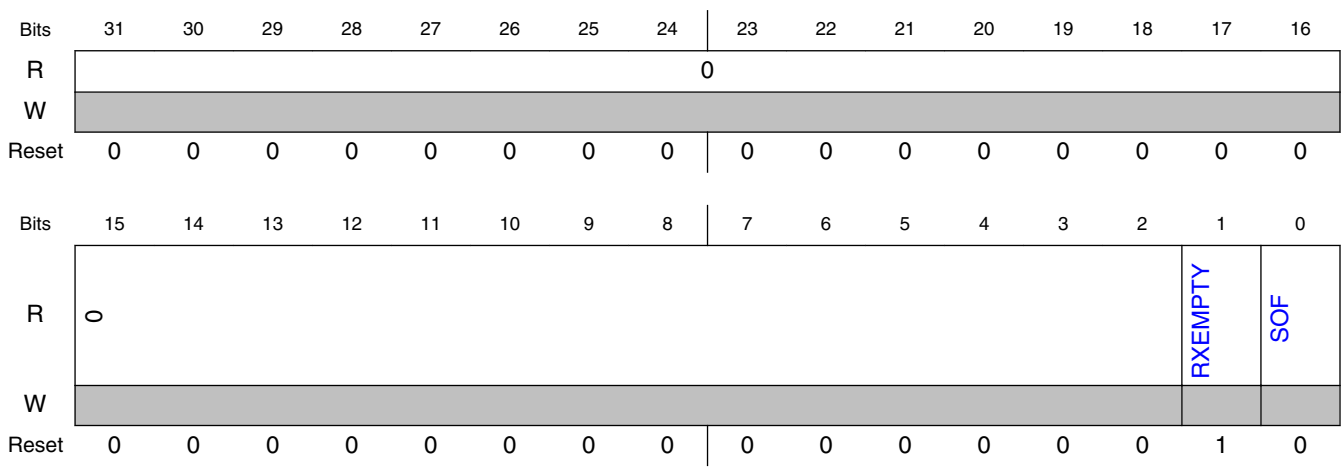
70.5.1.17.1 Offset

Register	Offset
RSR	70h

70.5.1.17.2 Function

Contains data flow status fields for receive FIFO.

70.5.1.17.3 Diagram



70.5.1.17.4 Fields

Field	Function
31-2	Reserved

Table continues on the next page...

Field	Function
—	
1 RXEMPTY	RX FIFO Empty 0b - Not empty 1b - Empty
0 SOF	Start Of Frame Indicates that this is the first data word received after PCS assertion. 0b - Subsequent data word 1b - First data word

70.5.1.18 Receive Data (RDR)

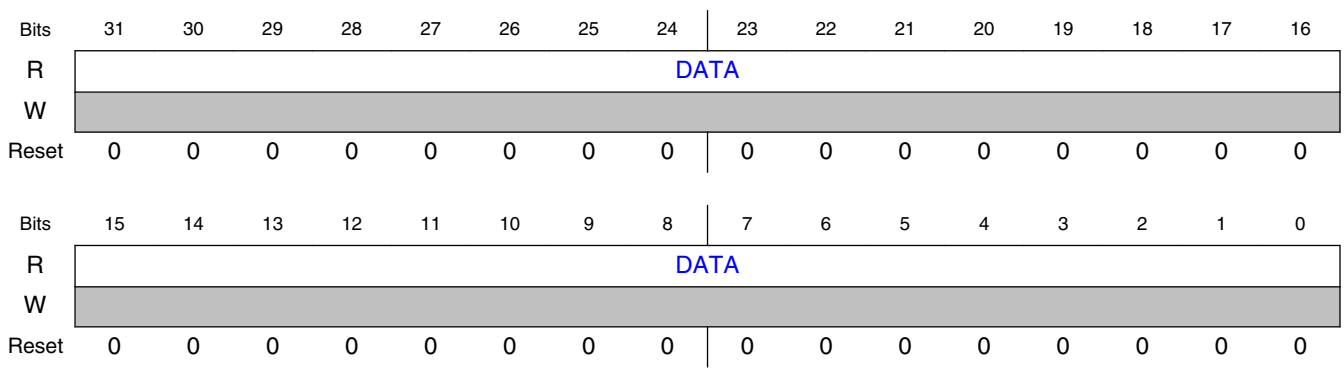
70.5.1.18.1 Offset

Register	Offset
RDR	74h

70.5.1.18.2 Function

Reading this register pulls the first entry from the receive FIFO.

70.5.1.18.3 Diagram



70.5.1.18.4 Fields

Field	Function
31-0 DATA	Receive Data

Chapter 71
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

71.1 Chip-specific LPUART information

Table 71-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

71.2 Overview

The Low Power Universal Asynchronous Receiver/Transmitter (LPUART) module provides asynchronous, serial communication capability with external devices. LPUART supports non-return-to-zero (NRZ) encoding format and IrDA compatible infrared (low-speed) SIR format. The LPUART can continue operating while the processor is in low-power mode if an appropriate peripheral clock is available.

71.2.1 Block diagram

The following figure shows the transmitter portion of the LPUART.

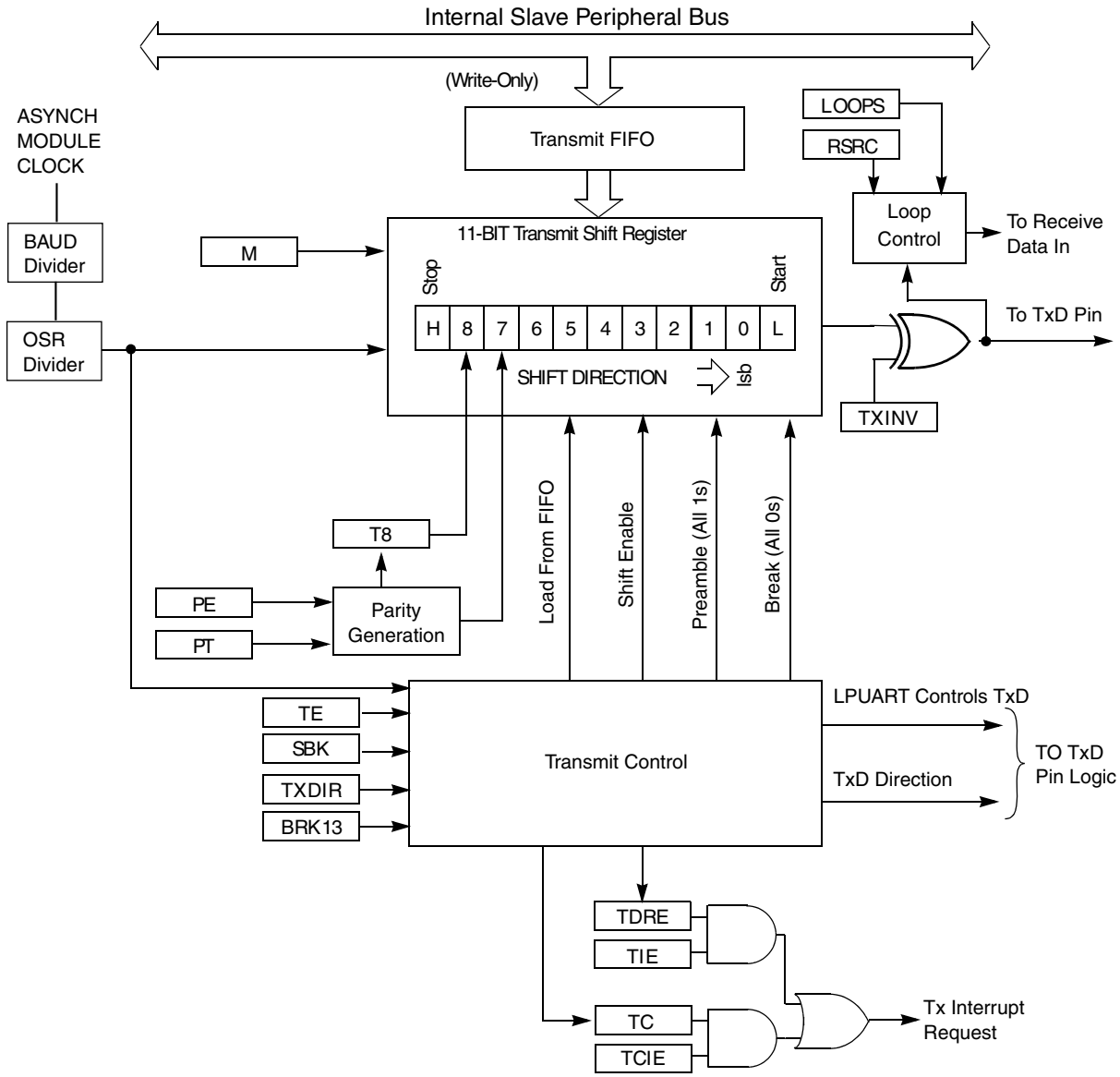


Figure 71-1. LPUART transmitter block diagram

The following figure shows the receiver portion of the LPUART.

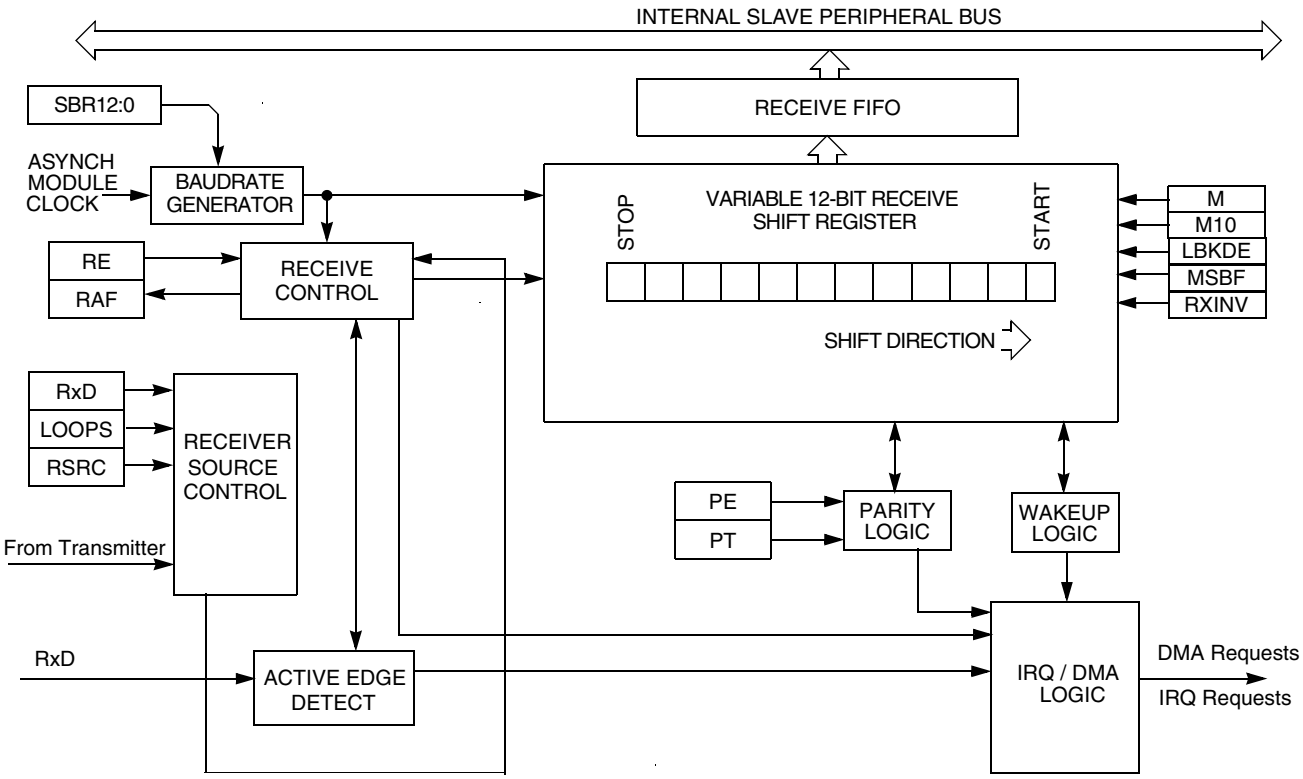


Figure 71-2. LPUART receiver block diagram

71.2.2 Features

Following are the features of the LPUART module.

- Full-duplex, standard non-return-to-zero (NRZ) format
- Programmable baud rates (13-bit modulo divider) with configurable oversampling ratio from 4x to 32x
- Asynchronous operation of transmit and receive baud rate with respect to the bus clock:
  - Baud rate can be configured independently of the bus clock frequency
  - Supports operation in Stop modes
- Interrupt, DMA or polled operation:
  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
  - Receive data match
- Hardware parity generation and checking
- Programmable 7-bit, 8-bit, 9-bit or 10-bit character length

- Programmable 1-bit or 2-bit stop bits
- Support for three receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
  - Receive data match
- Automatic address matching to reduce ISR overhead:
  - Address mark matching
  - Idle line address matching
  - Address match start, address match end
- Optional 13-bit/11-bit break character generation
- Configurable idle length detection supporting 1, 2, 4, 8, 16, 32, 64, or 128 idle characters
- Selectable transmitter output and receiver input polarity
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse width
- Independent FIFO structure for transmit and receive
  - Separate configurable watermark for receive and transmit requests
  - Option for receiver to assert request after a configurable number of idle characters if receive FIFO is not empty

71.3 Functional description

The LPUART supports full-duplex, asynchronous, NRZ serial communication and comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. The following describes each of the blocks of the LPUART.

71.3.1 Modes of operation

71.3.1.1 Stop mode

The LPUART remains functional during Stop mode, provided the CTRL[DOZEEN] bit is clear and the asynchronous transmit and receive clock remain enabled. The LPUART can generate an interrupt or DMA request to cause a wakeup from Stop mode.

If the LPUART is disabled in Stop mode, then it can generate a wakeup via the STAT[RXEDGIF] flag if the receiver detects an active edge.

71.3.1.2 Wait mode

The LPUART can be configured to Stop in Wait modes, when the CTRL[DOZEEN] bit is set. The transmitter and receiver finish transmitting/receiving the current word.

71.3.1.3 Debug mode

The LPUART remains functional in debug mode.

71.3.2 Clocking and Resets

Table 71-2. Clocks

LPUART Functional clock	The LPUART functional clock is asynchronous to the bus clock and can remain enabled in low power modes to support transmit and/or receive, including low power wakeups.
Bus clock	The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPUART transmit and receive registers, including the FIFOs.

Table 71-3. Resets

Chip reset	The logic and registers for the LPUART transmitter and receiver are reset to their default state on a chip reset.
Software reset	Resets the LPUART logic and registers to their default state, except for the Global Register. The LPUART software reset is in the Global Register GLOBAL[RST].
FIFO reset	The LPUART implements write-only control bits that reset the transmit FIFO (FIFO[TXFLUSH]) and receive FIFO (FIFO[RXFLUSH]). After a FIFO is reset, that FIFO is empty.

71.3.3 Baud rate generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value from 1 to 8191 written to BAUD[SBR] determines the baud clock divisor for the asynchronous LPUART baud clock. The baud rate clock drives the receiver, while the transmitter is driven by a bit clock which is generated from baud rate clock divided by the over sampling ratio (OSR). Depending on the over sampling ratio, the receiver has an acquisition rate of 4 to 32 samples per bit time.

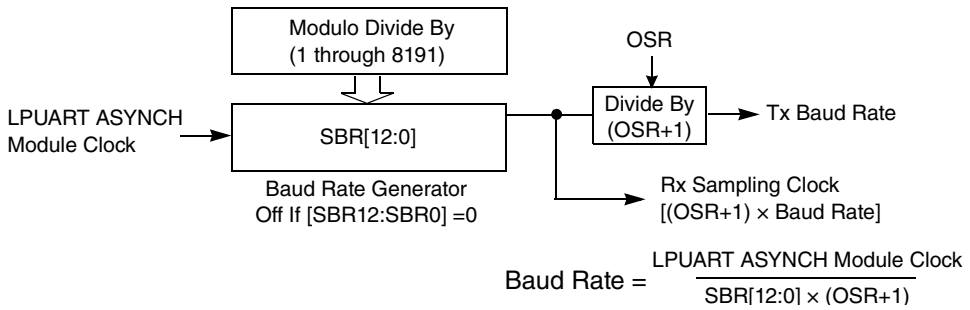


Figure 71-3. LPUART baud rate generation

Baud rate generation is subject to two sources of error:

- Integer division of the asynchronous LPUART baud clock may not give the exact target frequency.
- Synchronization with the asynchronous LPUART baud clock can cause phase shift.

The baud rate generation is a free-running counter that continues whenever the transmitter or receiver is enabled. The transmitter bit clock continues whenever the transmitter is enabled; each transmitted character aligns to the next edge of the transmit bit clock.

In general, configuring OSR for a higher oversampling ratio and/or sampling on both edges of the clock slightly improves the LPUART tolerance to baud rate mismatch between the received data and the LPUART configured baud rate. However, the three data samples in each bit will also be closer together which may impact noise sensitivity.

71.3.4 Transmitter functional description

This section describes the overall block diagram for the LPUART transmitter, as well as specialized functions for sending break and idle characters.

The transmitter output (TXD) idle state defaults to logic high; the transmitter output is inverted by setting CTRL[TXINV]. CTRL[TXINV] is cleared following reset. The transmitter is enabled by setting the CTRL[TE] bit. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit FIFO. Programs store data into the transmit FIFO by writing to the DATA register.

The central element of the LPUART transmitter is the transmit shift register that is 9-bit to 13-bit long depending on the setting in the CTRL[M], CTRL[M7], BAUD[M10] and BAUD[SBNS] control bits. For the remainder of this section, assume CTRL[M], CTRL[M7], BAUD[M10] and BAUD[SBNS] are cleared, selecting the normal 8-bit data

mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new character, the value waiting in the transmit data register is transferred to the transmit shift register, synchronized with the baud rate clock, and the transmit data register empty (STAT[TDRE]) status flag is set to indicate another character may be written to the transmit FIFO at the DATA register.

If no new character is waiting in the transmit FIFO after a stop bit is shifted out of the TXD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TXD high, waiting for more characters to transmit.

Writing 0 to CTRL[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed (that could include a data character, idle character or break character), although the transmitter does not start transmitting another character.

71.3.4.1 Send break and queued idle

The CTRL[SBK] bit sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 9-bit to 12-bit times including the start and stop bits. A longer break of 13-bit times can be enabled by setting STAT[BRK13]. Normally, a program waits for STAT[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to the CTRL[SBK] bit. This action queues a break character to be sent as soon as the shifter is available. If CTRL[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. When the LPUART is the receiving device, a break character is received as 0s in all data bits and a framing error (STAT[FE] = 1) is detected.

A break character can also be transmitted by writing to the DATA register with DATA[FRETSC] set and the data bits clear. This supports transmitting the break character as part of the normal data stream and also allows the DMA to transmit a break character.

When idle-line wakeup is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program waits for STAT[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 0, and then write 1 to the CTRL[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while CTRL[TE] is cleared, the LPUART transmitter never actually releases control of the TXD pin.

An idle character can also be transmitted by writing to the DATA register with DATA[FRETSC] and DATA[T9] set, and all other bits cleared. This supports transmitting the idle character as part of the normal data stream and also allows the DMA to transmit a idle character.

The length of the break character is affected by the STAT[BRK13], CTRL[M], CTRL[M7], BAUD[M10] and BAUD[SNBS] bits as shown below.

Table 71-4. Break character length

BRK13	M	M10	M7	SBNS	Break character length
0	0	0	0	0	10 bit times
0	0	0	0	1	11 bit times
0	0	0	1	0	9 bit times
0	0	0	1	1	10 bit times
0	1	0	X	0	11 bit times
0	1	0	X	1	12 bit times
0	X	1	X	0	12 bit times
0	X	1	X	1	13 bit times
1	0	0	0	0	13 bit times
1	0	0	0	1	13 bit times
1	0	0	1	0	12 bit times
1	0	0	1	1	12 bit times
1	1	0	X	0	14 bit times
1	1	0	X	1	14 bit times
1	X	1	X	0	15 bit times
1	X	1	X	1	15 bit times

71.3.4.2 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS_B. If the clear-to-send operation is enabled, the character is transmitted when CTS_B is asserted. If CTS_B is deasserted in the middle of a transmission with characters remaining in the transmitter FIFO, the character in the transmit shift register will be complete; any characters in the FIFO will wait for the CTS_B to assert again and TXD remains in the mark state (idle state) until CTS_B is reasserted. The CTS_B pin must assert for longer than one bit period to guarantee a new transmission is started when the transmitter is idle with data to send.

If the clear-to-send operation is disabled, the transmitter ignores the state of CTS_B.

The transmitter's CTS_B signal can be enabled even if the same LPUART receiver's RTS_B signal is disabled.

71.3.4.3 Transceiver driver enable

The transmitter can use RTS_B as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS_B for details. If the request-to-send operation is enabled, when a character is placed into an empty transmit shift register, RTS_B asserts one bit time before the start bit is transmitted. RTS_B remains asserted for the whole time that the transmit shift register has any characters. RTS_B deasserts one bit time after all characters in the transmit FIFO and shift register are completely sent, including the last stop bit. In other words, when RTS_B is used as a transceiver enable, the RTS_B asserts 1 bit time before the transmitter starts transmitting and negates 1 bit time after the transmitter goes idle.

Transmitting a break character also asserts RTS_B, with the same assertion and deassertion timing as having a character in the transmit shift register.

The transmitter's RTS_B signal asserts only when the transmitter is enabled. However, the transmitter's RTS_B signal is unaffected by its CTS_B signal. RTS_B remains asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

71.3.4.4 Transceiver driver enable using RTS_B

RS-485 is a multiple drop communication protocol in which the LPUART transceiver's driver is 3-stated unless the LPUART is driving. The RTS_B signal can be used by the transmitter to enable the driver of a transceiver. The polarity of RTS_B can be matched to the polarity of the transceiver's driver enable signal.

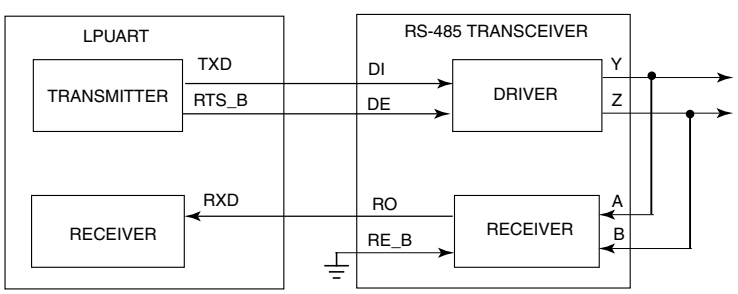


Figure 71-4. Transceiver driver enable using RTS_B

In the figure, the receiver enable signal is asserted. Another option for this connection is to connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled while driving. A pullup can pull RXD to a non-floating value during this time. This option can be refined further by operating the LPUART in single wire mode, freeing the RXD pin for other uses.

71.3.5 Receiver functional description

In this section, the receiver block diagram is a guide for the overall receiver functional description. Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, different variations of the receiver wakeup function are explained.

The receiver input is inverted by setting STAT[RXINV]. The receiver is enabled by setting the CTRL[RE] bit. Character frames consist of a start bit of logic 0, seven to ten data bits (msb or lsb first), and one or two stop bits of logic 1. For information about 7-bit, 9-bit or 10-bit data mode, refer to Data Modes. For the remainder of this discussion, assume the LPUART is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (STAT[RDRF]) status flag is set. If [RDRF] was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost.

Since the LPUART receiver is separate from the receive FIFO, the receive shift register can be receiving the next word when the receive FIFO is full and it is only at the end of the character that the next data is written into the receive FIFO, potentially triggering the overrun flag if FIFO is full.

When a program detects that the receive data register is full (STAT[RDRF] = 1), it gets the data from the receive data register by reading the DATA register. Refer to Interrupts and status flags for details about flag clearing.

71.3.5.1 Data sampling technique

The LPUART receiver supports a configurable oversampling rate of between 4× and 32× of the baud rate clock for sampling. The receiver starts by taking logic level samples at the oversampling rate times the baud rate to search for a falling edge on the RXD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is

located, three more samples are taken at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a received character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts the sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR/2), (OSR/2)+1, and (OSR/2)+2 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, the noise flag (STAT[NF]) is set when the received character is transferred to the receive FIFO.

When the LPUART receiver is configured to sample on both edges of the baud rate clock (that is, when BAUD[BOTHEDGE]=1), the number of segments in each received bit is effectively doubled (from 1 to OSR×2). The start and data bits are then sampled at OSR, OSR+1 and OSR+2. Sampling on both edges of the clock must be enabled for oversampling rates of 4× to 7× and is optional for higher oversampling rates.

The "synchronization" is a feature of the LPUART module to synchronize the internal oversampling counter with detected falling edge on Rx signal, and to adjust the data sampling window. The falling edge detection needs three consecutive '1' prior to '1'->'0' transition. After the initial falling edge detection for the start bit, the circuit continuously monitors the next falling edge, and resets the counter once another falling edge is detected. This is called "resynchronization".

- When LPUART_BAUD[RESYNCDIS] = 0, this falling edge detection and resynchronization is performed not only for the start bit but also for the rest of character reception after the start bit.
- When LPUART_BAUD[RESYNCDIS] = 1, the falling edge detection and resynchronization is performed only for the start bit. The use case for disabling the resynchronization is protocols that require this (for example, LIN 2.1 prohibits resynchronization within a byte).

  See the following table and figure.

Table 71-5. LPUART resynchronization

Resynchronization	LPUART_BAUD[RESYNCDIS]=0	LPUART_BAUD[RESYNCDIS]=1
For the starting bit falling edge	Yes	Yes
For every falling edges after the start bit	Yes	No

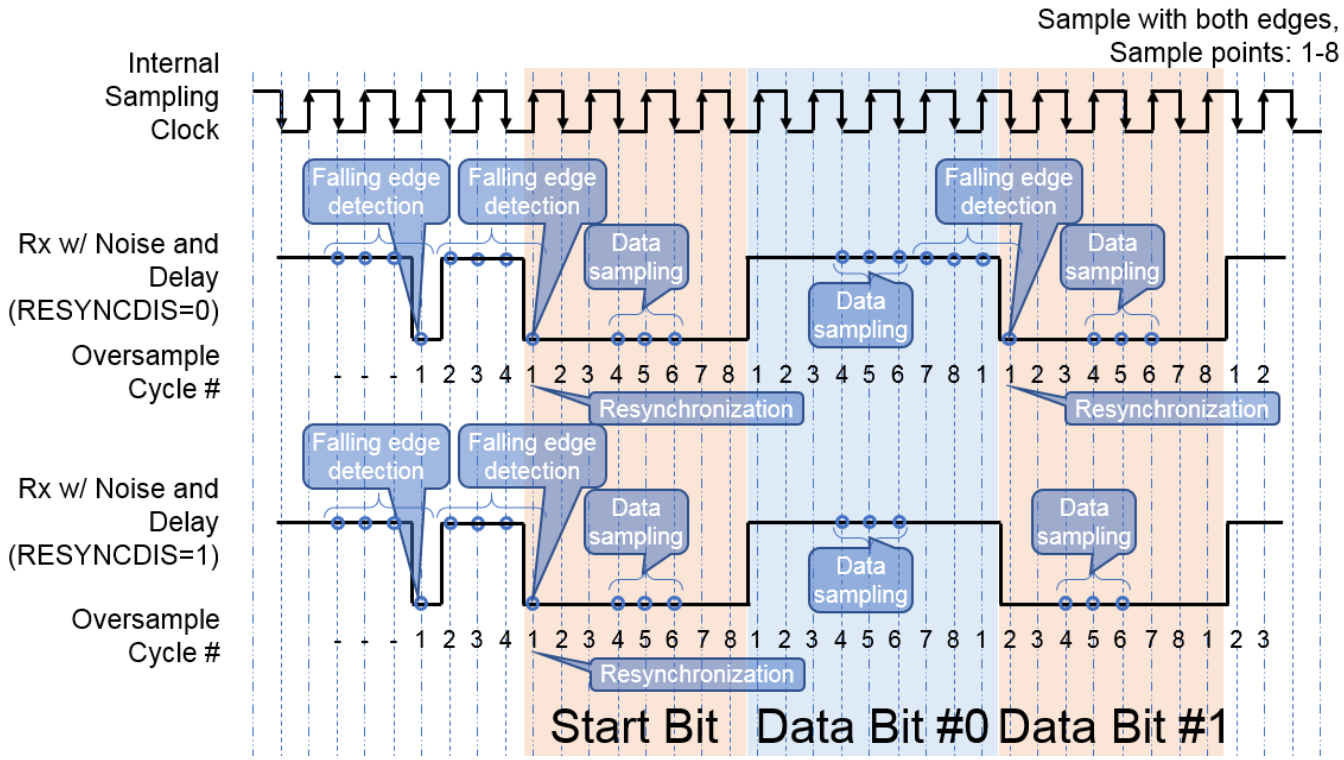


Figure 71-5. LPUART resynchronization

71.3.5.2 Receiver wakeup operation

Receiver wakeup and receiver address matching is a hardware mechanism that allows an LPUART receiver to ignore the characters in a message intended for a different receiver.

During receiver wakeup, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up control bit (CTRL[RWU]). When CTRL[RWU] and STAT[RWUID] bit are set, the status flags associated with the receiver, with the exception of the idle bit, STAT[IDLE], are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, all receivers automatically force CTRL[RWU] to 0 so all receivers wake up in time to look at the first character(s) of the next message.

During receiver address matching, the address matching is performed in hardware and the LPUART receiver ignores all characters that do not meet the address match requirements.

Table 71-6. Receiver Wakeup Options

RWU	MA1 | MA2	MATCFG	WAKE:RWUID	Receiver Wakeup
0	0	X	X	Normal operation

Table continues on the next page...

Table 71-6. Receiver Wakeup Options (continued)

RWU	MA1 | MA2	MATCFG	WAKE:RWUID	Receiver Wakeup
1	0	00	00	Receiver wakeup on idle line, IDLE flag not set
1	0	00	01	Receiver wakeup on idle line, IDLE flag set
1	0	00	10	Receiver wakeup on address mark
1	1	11	10	Receiver wakeup on data match
0	1	00	X0	Address mark address match, IDLE flag not set for discarded characters
0	1	00	X1	Address mark address match, IDLE flag set for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Match On and Match Off, IDLE flag not set for discarded characters
0	1	10	X1	Match On and Match Off, IDLE flag set for discarded characters

71.3.5.2.1 Idle-line wakeup

When CTRL[WAKE] is cleared, the receiver is configured for idle-line wakeup. In this mode, CTRL[RWU] is cleared automatically when the receiver detects a full character time of the idle-line level. The CTRL[M], CTRL[M7], and BAUD[M10] control bits select 7-bit to 10-bit data mode and the BAUD[SBNS] bit selects 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 9 to 13 bit times because of the start and stop bits.

When CTRL[RWU] is 1 and STAT[RWUID] is 0, the idle condition that wakes up the receiver does not set the STAT[IDLE] flag. The receiver wakes up and waits for the first data character of the next message that sets the STAT[RDRF] flag and generates an interrupt if enabled. When STAT[RWUID] is 1, any idle condition sets the STAT[IDLE] flag and generates an interrupt if enabled, regardless of whether CTRL[RWU] is 0 or 1.

The idle-line type (CTRL[ILT]) control bit selects one of two ways to detect an idle line.

- When CTRL[ILT] is cleared, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle.
- When CTRL[ILT] is set, the idle bit counter does not start until after the stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

71.3.5.2.2 Address-mark wakeup

When CTRL[WAKE] is set, the receiver is configured for address-mark wakeup. In this mode, CTRL[RWU] is cleared automatically when the receiver detects a logic 1 in the most significant bit of the received character. When parity is enabled, the second most significant bit is used for address-mark wakeup.

Address-mark wakeup allows messages to contain idle characters, but requires one bit be reserved for use in address frames. The logic 1 in the most significant bit (or second most significant bit when parity is enabled) of an address frame clears the CTRL[RWU] bit and sets the STAT[RDRF] flag. In this case, the character with the address-mark bit is received even though the receiver was sleeping during most of this character time.

71.3.5.2.3 Data match wakeup

When CTRL[RWU] is set, CTRL[WAKE] is set and BAUD[MATCFG] equals 11, the receiver is configured for data match wakeup. In this mode, CTRL[RWU] is cleared automatically when the receiver detects a character that matches MATCH[MA1] field when BAUD[MAEN1] is set, or that matches MATCH[MA2] when BAUD[MAEN2] is set.

71.3.5.2.4 Address Match operation

Address match operation is enabled when the BAUD[MAEN1] or BAUD[MAEN2] bit is set and BAUD[MATCFG] is equal to 00. In this function, a character received by the RXD pin with a logic 1 in the most significant bit (or second most significant bit when parity is enabled) is considered an address and is compared with the associated MATCH[MA1] or MATCH[MA2] field. The character is only transferred to the receive buffer, and STAT[RDRF] is set, if the comparison matches. All subsequent characters received with a logic 0 in the most significant bit (or second most significant bit when parity is enabled) are considered to be data associated with the address and are transferred to the receive FIFO. If no marked address match occurs then no transfer is made to the receive FIFO, and all following characters with logic zero in the most significant bit (or

second most significant bit when parity is enabled) are also discarded. If both the BAUD[MAEN1] and BAUD[MAEN2] bits are negated, the receiver operates normally and all data received is transferred to the receive FIFO.

Address match operation functions in the same way for both MATCH[MA1] and MATCH[MA2] fields.

- If only one of BAUD[MAEN1] and BAUD[MAEN2] is asserted, a marked address is compared only with the associated match register field and data is transferred to the receive FIFO only on a match.
- If BAUD[MAEN1] and BAUD[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either of the MATCH[MA1] or MATCH[MA2] fields.

71.3.5.2.5 Idle Match operation

Idle match operation is enabled when the BAUD[MAEN1] or BAUD[MAEN2] bit is set and BAUD[MATCFG] is equal to 01. In this function, the first character received by the RXD pin after an idle line condition is considered an address and is compared with the associated MATCH[MA1] or MATCH[MA2] field. The character is transferred only to the receive buffer, and STAT[RDRF] is set, if the comparison matches. All subsequent characters are considered to be data associated with the address and are transferred to the receive FIFO until the next idle line condition is detected. If no address match occurs then no transfer is made to the receive FIFO, and all following frames until the next idle condition are also discarded. If both the BAUD[MAEN1] and BAUD[MAEN2] bits are negated, the receiver operates normally and all data received is transferred to the receive FIFO.

Idle match operation functions in the same way for both MATCH[MA1] or MATCH[MA2] fields.

- If only one of BAUD[MAEN1] and BAUD[MAEN2] is asserted, the first character after an idle line is compared only with the associated match register and data is transferred to the receive FIFO only on a match.
- If BAUD[MAEN1] and BAUD[MAEN2] are asserted, the first character after an idle line is compared with both MATCH[MA1] or MATCH[MA2] fields and data is transferred only on a match with either of the fields.

71.3.5.2.6 Match On Match Off operation

Match on, match off operation is enabled when both BAUD[MAEN1] and BAUD[MAEN2] are set and BAUD[MATCFG] is equal to 10. In this function, a character received by the RXD pin that matches MATCH[MA1] is received and transferred to the receive buffer, and STAT[RDRF] is set. All subsequent characters are considered to be data and are also transferred to the receive FIFO, until a character is received that matches MATCH[MA2] field. The character that matches MATCH[MA2] and all following characters are discarded; this continues until another character that matches MATCH[MA1] is received. If both the BAUD[MAEN1] and BAUD[MAEN2] bits are negated, the receiver operates normally and all data received is transferred to the receive FIFO.

NOTE

Match on, match off operation requires both BAUD[MAEN1] and BAUD[MAEN2] to be asserted.

71.3.5.3 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert RTS_B.

- RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS_B for more details.
- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts RTS_B if the number of characters in the receiver data register is full or a start bit is detected that causes the receiver data register to be full.
- The receiver asserts RTS_B when the number of characters in the receiver data register is not full and has not detected a start bit that causes the receiver data register to be full. It is not affected if STAT[RDRF] is asserted.
- Even if RTS_B is deasserted, the receiver continues to receive characters until the receive FIFO is overrun.
- If the receiver request-to-send functionality is disabled, the receiver RTS_B remains deasserted.

71.3.6 Additional LPUART functions

The following sections describe additional LPUART functions.

71.3.6.1 Data Modes

The LPUART transmitter and receiver can be configured to operate in 7-bit data mode by setting CTRL[M7], 9-bit data mode by setting the CTRL[M] or 10-bit data mode by setting BAUD[M10]. In 9-bit mode, there is a ninth data bit and in 10-bit mode, there is a tenth data bit.

When performing 8-bit writes to the transmit FIFO, the 9th and 10th bit are pushed into the FIFO from CTRL[T8] and CTRL[T9]. For coherent 8-bit writes, write to CTRL[T8] and CTRL[T9] before writing to DATA[7:0], but if values in CTRL[T8] or CTRL[T9] do not need to change then it is not necessary to update CTRL[T8] and CTRL[T9] before every 8-bit write to the DATA register.

When performing 16-bit or 32-bit writes to the transmit FIFO, all 10-bits are pushed into the transmit FIFO from the write data.

When performing 8-bit reads of the receive FIFO, the 9th and 10th bit are held in CTRL[R8] and CTRL[R9] but should be read before reading the DATA register. A 16-bit or 32-bit read of the receive FIFO will return all 10-bits in the DATA register.

The 9-bit data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with address-mark wakeup so the ninth data bit can serve as the wakeup bit. The 10-bit data mode is typically used with parity and address-mark wakeup so the ninth data bit can serve as the wakeup bit and the tenth bit as the parity bit. In custom protocols, the ninth and/or tenth bits can also serve as software-controlled markers.

71.3.6.2 Idle length

An idle character is a character where the start bit, all data bits and stop bits are in the mark position (idle state, generally logic 1). The CTRL[ILT] bit can be configured to start detecting an idle character from the previous start bit (any data bits and stop bits count towards the idle character detection) or from the previous stop bit.

The number of idle characters that must be received before an idle line condition is detected can also be configured using the CTRL[IDLECFG] field. This field configures the number of idle characters that must be received before the STAT[IDLE] flag is set, the STAT[RAF] flag is cleared and the DATA[IDLINE] flag is set with the next received character.

Idle-line wakeup and idle match operation are also affected by the CTRL[IDLECFG] field. When address match or match on/off operation is enabled, setting the STAT[RWUID] bit causes any discarded characters to be treated as if they were idle characters.

71.3.6.3 Loop mode

When CTRL[LOOPS] is set, the CTRL[RSRC] bit chooses between loop mode (CTRL[RSRC] = 0) or single-wire mode (CTRL[RSRC] = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RXD pin is not used by the LPUART.

71.3.6.4 Single-wire operation

When CTRL[LOOPS] is set, the CTRL[RSRC] bit chooses between loop mode (CTRL[RSRC] = 0) or single-wire mode (CTRL[RSRC] = 1). Single-wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TXD pin (the RXD pin is not used).

In single-wire mode, the CTRL[TXDIR] bit controls the direction of serial data on the TXD pin. When CTRL[TXDIR] is cleared, the TXD pin is an input to the receiver and the transmitter is temporarily disconnected from the TXD pin so an external device can send serial data to the receiver. When CTRL[TXDIR] is set, the TXD pin is an output driven by the transmitter, the internal loop back connection is disabled, and as a result the receiver cannot receive characters that are sent out by the transmitter.

71.3.7 Infrared interface

The LPUART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the LPUART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. The LPUART IrDA support is limited to SIR mode which supports data rates only between 2.4 kbits/s and 115.2 kbits/s.

The LPUART has an infrared transmit encoder and receive decoder. The infrared decoder converts the received character from the IrDA format to the NRZ format used by the receiver. It also has a OSR oversampling baud rate clock counter that filters noise and indicates when a 1 is received. The LPUART transmits serial bits of data that are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode (external to LPUART) and transformed to CMOS levels by the IR receive decoder. The narrow pulses are then stretched by the infrared receive decoder to get back

to a serial bit stream to be received by the LPUART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active high pulses.

The infrared submodule receives its clock sources from the LPUART. One of these two clocks are selected in the infrared submodule to generate either 1/OSR, 2/OSR, 3/OSR, or 4/OSR narrow pulses during transmission.

71.3.7.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD signal. A narrow pulse is transmitted for a 0 bit and no pulse for a 1 bit. The narrow pulse is sent at the start of the bit with a duration of 1/OSR, 2/OSR, 3/OSR, or 4/OSR of a bit time. A narrow low pulse is transmitted for a 0 bit when CTRL[TXINV] is cleared, while a narrow high pulse is transmitted for a 0 bit when CTRL[TXINV] is set.

71.3.7.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each 0 received and no pulse is expected for each 1 received. A narrow low pulse is expected for a 0 bit when STAT[RXINV] is cleared, while a narrow high pulse is expected for a 0 bit when STAT[RXINV] is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

71.3.7.3 Start bit detection

When STAT[RXINV] is cleared, the first falling edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently from each other.

71.3.7.4 Noise filtering

Any further rising edges detected during the first half of the infrared decoder counter are ignored by the decoder. Any pulses less than one oversampling baud clock can be undetected by it regardless of whether it is seen in the first or second half of the count.

71.3.7.5 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as a 0, which is sent to the receiver. The decoder counter is also reset.

71.3.7.6 High-bit detection

At OSR oversampling baud rate clocks after the previous rising edge, if a rising edge is not seen, then the decoder sends a 1 to the receiver.

If the next bit is a 0, which arrives late, then a low-bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to a 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, then the delay of a 0 is not recorded as noise.

71.3.8 Interrupts and status flags

The LPUART transmitter has two status flags that can optionally generate hardware interrupt requests. Transmit Data Register Empty (STAT[TDRE]) indicates when there is room in the transmit FIFO to write another transmit character to the DATA register. If the Transmit Interrupt Enable (CTRL[TIE]) bit is set, a hardware interrupt is requested when STAT[TDRE] is set. Transmit complete (STAT[TC]) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TXD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (CTRL[TCIE]) bit is set, a hardware interrupt is requested when STAT[TC] is set. Instead of hardware interrupts, software polling may be used to monitor the STAT[TDRE] and STAT[TC] status flags if the corresponding CTRL[TIE] or CTRL[TCIE] local interrupt masks are cleared.

When a program detects that the receive data register is full (STAT[RDRF] = 1), it gets the data from the receive data register by reading the DATA register. The STAT[RDRF] flag is cleared by reading the DATA register.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RXD line remains idle for an extended period of time. IDLE is cleared by writing 1 to the STAT[IDLE] flag. After STAT[IDLE] has been cleared, it cannot become set again until the receiver has received at least one new character and has set STAT[RDRF].

If the associated error is detected in the received character that caused STAT[RDRF] to be set, the error flags — noise flag (STAT[NF]), framing error (STAT[FE]), and parity error flag (STAT[PF]) — are set at the same time as STAT[RDRF]. These flags are not set in overrun cases.

If STAT[RDRF] is already set when a new character is ready to be transferred from the receive shifter to the receive FIFO, the overrun (STAT[OR]) flag is set instead of the data along with any associated STAT[NF], STAT[FE], or STAT[PF] condition is lost.

If the received character matches the contents of MATCH[MA1] and/or MATCH[MA2] then the STAT[MA1F] and/or STAT[MA2F] flags are set at the same time that STAT[RDRF] is set.

At any time, an active edge on the RXD serial data input pin causes the STAT[RXEDGIF] flag to set. The STAT[RXEDGIF] flag is cleared by writing a 1 to it. This function depends on the receiver being enabled (CTRL[RE] = 1).

71.3.9 Peripheral Triggers

The connection of the LPUART peripheral triggers with other peripherals are device specific.

71.3.9.1 Output Triggers

The LPUART generates the following output triggers that can be connected to other peripherals on the device.

- The transmit word trigger asserts at the end of each transmitted word, it negates after one bit period.
- The receive word trigger asserts at the end of each received word that is written to the receive FIFO, for one oversampling clock period.
- The receive idle trigger asserts when the idle flag would set, for one oversampling clock period.

71.3.9.2 Input Trigger

The LPUART supports one peripheral input trigger, that can be configured in one of the following ways.

- The CTS function must be enabled. The input trigger can be connected in place of the CTS_B pin input. The input trigger must assert for longer than one bit clock period when the transmitter is idle with data to send to guarantee a new transmission.
- The input trigger can modulate the transmit data output (trigger is logically ANDed with the TXD output). The input trigger is expected to be a free running clock (carrier signal) generated from a timer or PWM source with a frequency that is greater than the bit clock frequency. The carrier signal must not toggle faster than the maximum supported bit time.
- The input trigger can be connected in place of the RXD pin input. The input trigger is expected to be generated from a receive data source, such as analog comparator or external pin.

71.4 External signals

Signal	Description	I/O
TXD	Transmit data. This pin is normally an output, but is an input (tristated) in single wire mode whenever the transmitter is disabled or transmit direction is configured for receive data.	I/O
RXD	Receive data.	I
CTS_B	Clear to send.	I
RTS_B	Request to send.	O

71.5 Register definition

The LPUART includes registers to control baud rate, select options, report status, and store transmit/receive data. Access to an address outside the valid memory map generates a bus error.

NOTE

Writing a Read-Only (RO) register or reading a Write-Only (WO) register can cause bus errors. This module does not check if programmed values in the registers are correct; the application software must ensure that valid programmed values are being written.

71.5.1 LPUART register descriptions

71.5.1.1 LPUART memory map

LPUART1 base address: 4007_C000h

LPUART2 base address: 4008_0000h

LPUART3 base address: 4008_4000h

LPUART4 base address: 4008_8000h

LPUART5 base address: 4008_C000h

LPUART6 base address: 4009_0000h

LPUART7 base address: 4009_4000h

LPUART8 base address: 4009_8000h

LPUART9 base address: 4009_C000h

LPUART10 base address: 400A_0000h

LPUART11 base address: 40C2_4000h

LPUART12 base address: 40C2_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID Register (VERID)	32	RO	0401_0003
4	Parameter Register (PARAM)	32	RO	0000_0202
8	LPUART Global Register (GLOBAL)	32	RW	0000_0000
C	LPUART Pin Configuration Register (PINCFG)	32	RW	0000_0000
10	LPUART Baud Rate Register (BAUD)	32	RW	0F00_0004
14	LPUART Status Register (STAT)	32	RW	00C0_0000
18	LPUART Control Register (CTRL)	32	RW	0000_0000
1C	LPUART Data Register (DATA)	32	RW	0000_1000
20	LPUART Match Address Register (MATCH)	32	RW	0000_0000
24	LPUART Modem IrDA Register (MODIR)	32	RW	0000_0000
28	LPUART FIFO Register (FIFO)	32	RW	00C0_0011
2C	LPUART Watermark Register (WATER)	32	RW	0000_0000

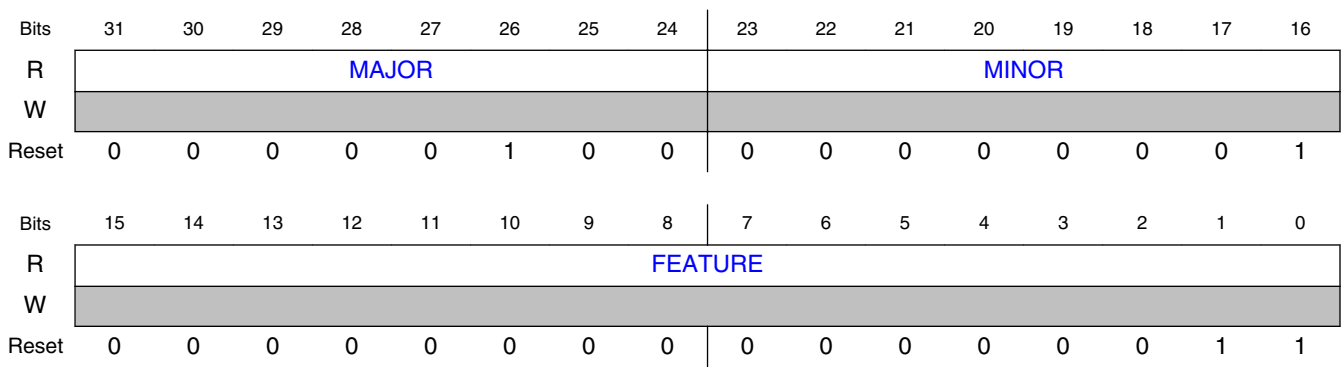
71.5.1.2 Version ID Register (VERID)

The Version ID register indicates the version integrated for this instance on the device and also indicates inclusion/exclusion of several optional features.

71.5.1.2.1 Offset

Register	Offset
VERID	0h

71.5.1.2.2 Diagram



71.5.1.2.3 Fields

Field	Description
31-24	Major Version Number
MAJOR	This read-only field returns the major version number for the module specification.
23-16	Minor Version Number
MINOR	This read-only field returns the minor version number for the module specification.
15-0	Feature Identification Number
FEATURE	This read-only field returns the feature set number. 0000000000000001 - Standard feature set. 0000000000000011 - Standard feature set with MODEM/IrDA support.

71.5.1.3 Parameter Register (PARAM)

The Parameter register indicates the parameter configuration for this instance on the device.

71.5.1.3.1 Offset

Register	Offset
PARAM	4h

71.5.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

71.5.1.3.3 Fields

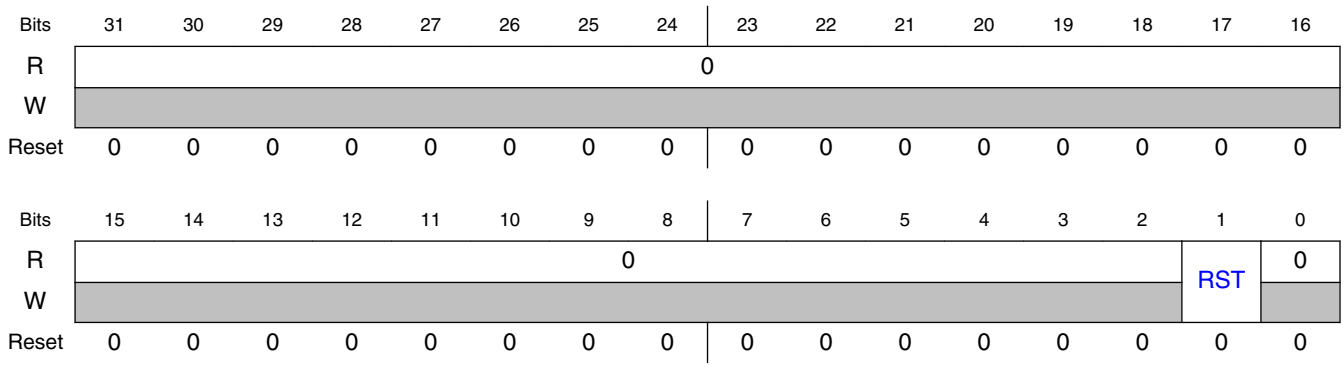
Field	Description
31-16 —	Reserved
15-8 RXFIFO	Receive FIFO Size The number of characters in the receive FIFO is 2^RXFIFO.
7-0 TXFIFO	Transmit FIFO Size The number of characters in the transmit FIFO is 2^TXFIFO.

71.5.1.4 LPUART Global Register (GLOBAL)

71.5.1.4.1 Offset

Register	Offset
GLOBAL	8h

71.5.1.4.2 Diagram



71.5.1.4.3 Fields

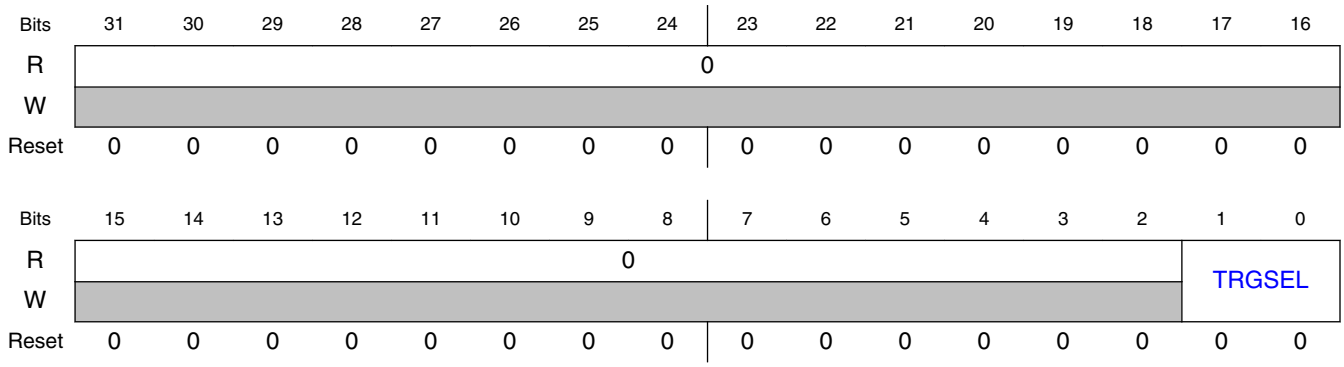
Field	Description
31-2 —	Reserved
1 RST	Software Reset Resets all internal logic and registers, except the Global Register. The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0 - Module is not reset. 1 - Module is reset.
0 —	Reserved

71.5.1.5 LPUART Pin Configuration Register (PINCFG)

71.5.1.5.1 Offset

Register	Offset
PINCFG	Ch

71.5.1.5.2 Diagram



71.5.1.5.3 Fields

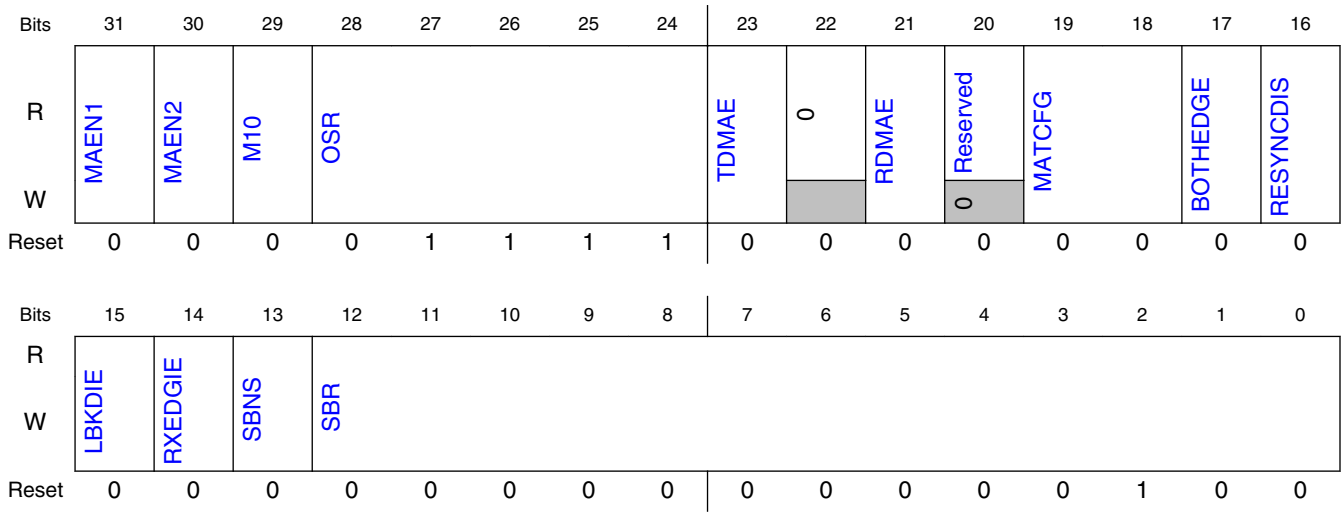
Field	Description
31-2 —	Reserved
1-0 TRGSEL	Trigger Select Configures the input trigger usage. This field should be changed only when the transmitter and receiver are both disabled. 00 - Input trigger is disabled. 01 - Input trigger is used instead of RXD pin input. 10 - Input trigger is used instead of CTS_B pin input. 11 - Input trigger is used to modulate the TXD pin output. The TXD pin output (after TXINV configuration) is internally ANDed with the input trigger.

71.5.1.6 LPUART Baud Rate Register (BAUD)

71.5.1.6.1 Offset

Register	Offset
BAUD	10h

71.5.1.6.2 Diagram



71.5.1.6.3 Fields

Field	Description
31 MAEN1	Match Address Mode Enable 1 0 - Normal operation. 1 - Enables automatic address matching or data matching mode for MATCH[MA1].
30 MAEN2	Match Address Mode Enable 2 0 - Normal operation. 1 - Enables automatic address matching or data matching mode for MATCH[MA2].
29 M10	10-bit Mode select The M10 bit causes a tenth bit to be part of the serial transmission. This bit should be changed only when the transmitter and receiver are both disabled. 0 - Receiver and transmitter use 7-bit to 9-bit data characters. 1 - Receiver and transmitter use 10-bit data characters.
28-24 OSR	Oversampling Ratio This field configures the oversampling ratio for the receiver. This field should be changed only when the transmitter and receiver are both disabled. 00000 - Writing 0 to this field results in an oversampling ratio of 16 00001 - Reserved 00010 - Reserved 00011 - Oversampling ratio of 4, requires BOTHEDGE to be set. 00100 - Oversampling ratio of 5, requires BOTHEDGE to be set. 00101 - Oversampling ratio of 6, requires BOTHEDGE to be set. 00110 - Oversampling ratio of 7, requires BOTHEDGE to be set. 00111 - Oversampling ratio of 8.

Table continues on the next page...

Field	Description
	01000 - Oversampling ratio of 9. 01001 - Oversampling ratio of 10. 01010 - Oversampling ratio of 11. 01011 - Oversampling ratio of 12. 01100 - Oversampling ratio of 13. 01101 - Oversampling ratio of 14. 01110 - Oversampling ratio of 15. 01111 - Oversampling ratio of 16. 10000 - Oversampling ratio of 17. 10001 - Oversampling ratio of 18. 10010 - Oversampling ratio of 19. 10011 - Oversampling ratio of 20. 10100 - Oversampling ratio of 21. 10101 - Oversampling ratio of 22. 10110 - Oversampling ratio of 23. 10111 - Oversampling ratio of 24. 11000 - Oversampling ratio of 25. 11001 - Oversampling ratio of 26. 11010 - Oversampling ratio of 27. 11011 - Oversampling ratio of 28. 11100 - Oversampling ratio of 29. 11101 - Oversampling ratio of 30. 11110 - Oversampling ratio of 31. 11111 - Oversampling ratio of 32.
23 TDMAE	Transmitter DMA Enable TDMAE configures the transmit data register empty flag, STAT[TDRE], to generate a DMA request. 0 - DMA request disabled. 1 - DMA request enabled.
22 —	Reserved
21 RDMAE	Receiver Full DMA Enable RDMAE configures the receiver data register full flag, STAT[RDRF], to generate a DMA request. 0 - DMA request disabled. 1 - DMA request enabled.
20 —	Reserved
19-18 MATCFG	Match Configuration Configures the match addressing mode used. This field should be changed only when the transmitter and receiver are both disabled.

Table continues on the next page...

Field	Description
	00 - Address Match Wakeup 01 - Idle Match Wakeup 10 - Match On and Match Off 11 - Enables RWU on Data Match and Match On/Off for transmitter CTS input
17 BOTHEDGE	Both Edge Sampling Enables sampling of the received data on both edges of the baud rate clock, effectively doubling the number of times the receiver samples the input data for a given oversampling ratio. This bit must be set for oversampling ratios between x4 and x7 and is optional for higher oversampling ratios. This bit should be changed only when the receiver is disabled. 0 - Receiver samples input data using the rising edge of the baud rate clock. 1 - Receiver samples input data using the rising and falling edge of the baud rate clock.
16 RESYNCDIS	Resynchronization Disable When set, this field disables the resynchronization of the received data word when a data one followed by data zero transition is detected. This bit should be changed only when the receiver is disabled. 0 - Resynchronization during received data word is supported. 1 - Resynchronization during received data word is disabled.
15 LBKDIE	LIN Break Detect Interrupt Enable LBKDIE enables the LIN break detect flag, STAT[LBKDIF], to generate interrupt requests. 0 - Hardware interrupts from STAT[LBKDIF] flag are disabled (use polling). 1 - Hardware interrupt is requested when STAT[LBKDIF] flag is 1.
14 RXEDGIE	RX Input Active Edge Interrupt Enable Enables the receive input active edge, STAT[RXEDGIF], to generate interrupt requests. Changing CTRL[LOOPS] or CTRL[RSRC] when RXEDGIE is set can cause the STAT[RXEDGIF] flag to set. 0 - Hardware interrupts from STAT[RXEDGIF] are disabled. 1 - Hardware interrupt is requested when STAT[RXEDGIF] flag is 1.
13 SBNS	Stop Bit Number Select SBNS determines whether data characters have one or two stop bits. This bit should be changed only when the transmitter and receiver are both disabled. 0 - One stop bit. 1 - Two stop bits.
12-0 SBR	Baud Rate Modulo Divisor. The 13 bits in SBR[12:0] set the modulo divide rate for the baud rate generator. When SBR is 1 - 8191, the baud rate equals "baud clock / ((OSR+1) * SBR)". The 13-bit baud rate setting [SBR12:SBR0] must be updated only when the transmitter and receiver are both disabled (CTRL[RE] and CTRL[TE] are both 0).

71.5.1.7 LPUART Status Register (STAT)

71.5.1.7.1 Offset

Register	Offset
STAT	14h

71.5.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	W1C	W1C										W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0				0		0						0	
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

71.5.1.7.3 Fields

Field	Description
31 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. Writing a 1 clears the LBKDIF field. 0 - No LIN break character has been detected. 1 - LIN break character has been detected.
30 RXEDGIF	RXD Pin Active Edge Interrupt Flag RXEDGIF is set whenever the receiver is enabled and an active edge (falling if RXINV = 0, rising if RXINV=1,) on the RXD pin occurs. To clear RXEDGIF, write a 1 to the RXEDGIF field. 0 - No active edge on the receive pin has occurred. 1 - An active edge on the receive pin has occurred.
29 MSBF	MSB First Setting this bit reverses the order of the bits that are transmitted and received on the wire. This bit does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. This bit should be changed only when the transmitter and receiver are both disabled. 0 - LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0.

Table continues on the next page...

Field	Description
	1 - MSB (identified as bit9, bit8, bit7 or bit6) is the first bit that is transmitted following the start bit depending on the setting of CTRL[M], CTRL[PE] and BAUD[M10]. .
28 RXINV	Receive Data Inversion Setting this bit reverses the polarity of the received data input. This bit should be changed only when the receiver is disabled. **NOTE:** Setting RXINV inverts the RXD input for all cases: data bits, start and stop bits, break, and idle. 0 - Receive data not inverted. 1 - Receive data inverted.
27 RWUID	Receive Wake Up Idle Detect For RWU on idle character, RWUID controls whether the idle character that wakes up the receiver sets the IDLE bit. For address match wakeup, RWUID controls if the IDLE bit is set when the address does not match. This bit should be changed only when the receiver is disabled. 0 - During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character. During address match wakeup, the IDLE bit does not set when an address does not match. 1 - During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character. During address match wakeup, the IDLE bit does set when an address does not match.
26 BRK13	Break Character Generation Length BRK13 selects a longer transmitted break character length. The state of this bit does not affect the detection of a framing error. This bit should be changed only when the transmitter is disabled. A break character can be sent by setting CTRL[SBK] or by writing the transmit FIFO with DATA[FRETSC] set and DATA[R9T9] clear. 0 - Break character is transmitted with length of 9 to 13 bit times. 1 - Break character is transmitted with length of 12 to 15 bit times.
25 LBKDE	LIN Break Detection Enable LBKDE selects a longer break character detection length. While LBKDE is set, receive data is not stored in the receive FIFO. **NOTE:** The LBKDE bit enables the LIN break detect circuit and disables writing receive data to FIFO. So it essentially ignores all characters except a LIN break. 0 - LIN break detect is disabled, normal break character can be detected. 1 - LIN break detect is enabled. LIN break character is detected at length of 11 bit times (if M = 0) or 12 (if M = 1) or 13 (M10 = 1).
24 RAF	Receiver Active Flag RAF is set when the receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. 0 - LPUART receiver idle waiting for a start bit. 1 - LPUART receiver active (RXD input not idle).
23 TDRE	Transmit Data Register Empty Flag When the transmit FIFO is enabled, TDRE is set when the number of datawords in the transmit FIFO is equal to or less than the number indicated by WATER[TXWATER]. To clear TDRE, write to the DATA register until the number of words in the transmit FIFO is greater than the number indicated by WATER[TXWATER]. When the transmit FIFO is disabled, TDRE is set when the transmit DATA register is empty. To clear TDRE, write to the DATA register. TDRE is not affected by a character that is in the process of being transmitted; it is updated at the start of each transmitted character.

Table continues on the next page...

Field	Description
	0 - Transmit FIFO level is greater than watermark. 1 - Transmit FIFO level is equal or less than watermark.
22 TC	Transmission Complete Flag TC is cleared when there is a transmission in progress or when a preamble or break character is loaded. TC is set when the transmit buffer is empty and no data, preamble, or break character is being transmitted. When TC is set, the transmit data output signal becomes idle (logic 1). TC is cleared by writing to the DATA register to transmit new data, queuing a preamble by clearing and then setting CTRL[TE], queuing a break character by writing 1 to CTRL[SBK]. 0 - Transmitter active (sending data, a preamble, or a break). 1 - Transmitter idle (transmission activity complete).
21 RDRF	Receive Data Register Full Flag When the receive FIFO is enabled, RDRF is set when the number of datawords in the receive buffer is greater than the number indicated by WATER[RXWATER]. To clear RDRF, read the DATA register until the number of datawords in the receive FIFO is equal to or less than the number indicated by WATER[RXWATER]. When the receive FIFO is disabled, RDRF is set when the receive buffer (the DATA register) is full. To clear RDRF, read the DATA register. A character that is in the process of being received does not cause a change in RDRF until the entire character is received. Even if RDRF is set, the character continues to be received until an overrun condition occurs once the entire character is received. 0 - Receive FIFO level is less than watermark. 1 - Receive FIFO level is equal or greater than watermark.
20 IDLE	Idle Line Flag IDLE is set when the LPUART receive line becomes idle for a full character time after a period of activity. When CTRL[ILT] is cleared, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bits time count toward the full character time of logic high, 10 to 13 bit times, needed for the receiver to detect an idle line. When CTRL[ILT] is set, the receiver doesn't start counting idle bit times until after the stop bits. The stop bits and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, write logic 1 to the IDLE flag. After IDLE has been cleared, it cannot be set again until after a new character is stored in the receive buffer or a LIN break character has set the LBKDIF flag . IDLE is set only once even if the receive line remains idle for an extended period. 0 - No idle line detected. 1 - Idle line is detected.
19 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If LBKDE is enabled and a LIN Break is detected, the OR field asserts if LBKDIF is not cleared before the next data character is received. While the OR flag is set, no additional data is stored in the receive FIFO even if sufficient room exists. To clear OR, write logic 1 to the OR flag. 0 - No overrun. 1 - Receive overrun (new LPUART data lost).
18 NF	Noise Flag

Table continues on the next page...

Field	Description
	The advanced sampling technique used in the receiver takes three samples in each of the received bits. If any of these samples disagrees with the rest of the samples within any bit time in the frame then noise is detected for that character. NF is set whenever the next character to be read from the DATA register is received with noise detected within the character. To clear NF, write logic 1 to the NF field. 0 - No noise detected. 1 - Noise detected in the received character in the DATA register.
17 FE	Framing Error Flag FE is set whenever the next character to be read from the DATA register is received with logic 0 detected where a stop bit was expected. To clear FE, write logic 1 to the FE field. 0 - No framing error detected. This does not guarantee the framing is correct. 1 - Framing error.
16 PF	Parity Error Flag PF is set whenever the next character to be read from the DATA register is received when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, write a logic 1 to the PF field. 0 - No parity error. 1 - Parity error.
15 MA1F	Match 1 Flag MA1F is set whenever the next character to be read from the DATA register matches MA1. To clear MA1F, write a logic 1 to the MA1F field. 0 - Received data is not equal to MA1 1 - Received data is equal to MA1
14 MA2F	Match 2 Flag MA2F is set whenever the next character to be read from the DATA register matches MA2. To clear MA2F, write a logic 1 to the MA2F field. 0 - Received data is not equal to MA2 1 - Received data is equal to MA2
13-10 —	Reserved
9-8 —	Reserved
7-2 —	Reserved
1-0 —	Reserved

71.5.1.8 LPUART Control Register (CTRL)

This read/write register controls various optional features of the LPUART system. This register should only be altered when the transmitter and receiver are both disabled.

71.5.1.8.1 Offset

Register	Offset
CTRL	18h

71.5.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1IE	MA2IE	0		M7	IDLECFG			LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

71.5.1.8.3 Fields

Field	Description
31 R8T9	Receive Bit 8 / Transmit Bit 9 R8 is the ninth data bit received when the LPUART is configured for 9-bit or 10-bit data formats. When reading 9-bit or 10-bit data, read R8 before reading the DATA register. T9 is the tenth data bit transmitted when the LPUART is configured for 10-bit data formats. When writing 10-bit data, write T9 before writing the DATA register. If T9 does not need to change from its previous value, such as when it is used to generate address mark or parity, then it need not be written each time the DATA register is written. **NOTE:** R8 is a read-only bit and T9 is a write-only bit, the value read is different from the value written.
30 R9T8	Receive Bit 9 / Transmit Bit 8 R9 is the tenth data bit received when the LPUART is configured for 10-bit data formats. When reading 10-bit data, read R9 before reading the DATA register T8 is the ninth data bit transmitted when the LPUART is configured for 9-bit or 10-bit data formats. When writing 9-bit or 10-bit data, write T8 before writing the DATA register. If T8 does not need to change from its previous value, such as when it is used to generate address mark or parity, then it need not be written each time the DATA register is written. **NOTE:** R9 is a read-only bit and T8 is a write-only bit, the value read is different from the value written.

Table continues on the next page...

Field	Description
29 TXDIR	TXD Pin Direction in Single-Wire Mode When the LPUART is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the TXD pin. When clearing TXDIR, the transmitter finishes receiving the current character (if any) before the receiver starts receiving data from the TXD pin. 0 - TXD pin is an input in single-wire mode. 1 - TXD pin is an output in single-wire mode.
28 TXINV	Transmit Data Inversion Setting this bit reverses the polarity of the transmitted data output. **NOTE:** Setting TXINV inverts the TXD output for all cases: data bits, start and stop bits, break, and idle. 0 - Transmit data not inverted. 1 - Transmit data inverted.
27 ORIE	Overrun Interrupt Enable This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 - OR interrupts disabled; use polling. 1 - Hardware interrupt is requested when OR is set.
26 NEIE	Noise Error Interrupt Enable This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 - NF interrupts disabled; use polling. 1 - Hardware interrupt is requested when NF is set.
25 FEIE	Framing Error Interrupt Enable This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 - FE interrupts disabled; use polling. 1 - Hardware interrupt is requested when FE is set.
24 PEIE	Parity Error Interrupt Enable This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 - PF interrupts disabled; use polling). 1 - Hardware interrupt is requested when PF is set.
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests. 0 - Hardware interrupts from TDRE disabled. 1 - Hardware interrupt is requested when TDRE flag is 1.
22 TCIE	Transmission Complete Interrupt Enable for TCIE enables the transmission complete flag, TC, to generate interrupt requests. 0 - Hardware interrupts from TC disabled. 1 - Hardware interrupt is requested when TC flag is 1.
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate interrupt requests. 0 - Hardware interrupts from RDRF disabled. 1 - Hardware interrupt is requested when RDRF flag is 1.

Table continues on the next page...

Field	Description
20 ILIE	Idle Line Interrupt Enable ILIE enables the idle line flag, STAT[IDLE], to generate interrupt requests. 0 - Hardware interrupts from IDLE disabled; use polling. 1 - Hardware interrupt is requested when IDLE flag is 1.
19 TE	Transmitter Enable Enables the LPUART transmitter. TE can also be used to queue an idle preamble by clearing and then setting TE. When TE is cleared, this register bit reads as 1 until the transmitter has completed the current character and the TXD pin is tristated. A single idle character can also be queued by writing to the transmit FIFO with DATA[FRETSC] set and DATA[R9T9] set. 0 - Transmitter disabled. 1 - Transmitter enabled.
18 RE	Receiver Enable Enables the LPUART receiver. When RE is written to 0, this register bit reads as 1 until the receiver finishes receiving the current character (if any). 0 - Receiver disabled. 1 - Receiver enabled.
17 RWU	Receiver Wakeup Control This field can be set to place the LPUART receiver in a standby state. RWU automatically clears when an RWU event occurs, that is, an IDLE event when CTRL[WAKE] is clear or an address match when CTRL[WAKE] is set and STAT[RWUID] is clear. **NOTE:** RWU must be set only with CTRL[WAKE] = 0 (wakeup on idle) if the channel is currently not idle. This can be determined by STAT[RAF]. If the flag is set to wake up an IDLE event and the channel is already idle, it is possible that the LPUART will discard data. This is because the data must be received or a LIN break detected after an IDLE is detected before IDLE is allowed to be reasserted. 0 - Normal receiver operation. 1 - LPUART receiver in standby waiting for wakeup condition.
16 SBK	Send Break Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 9 to 13 bits, or 12 to 15 bits if STAT[BRK13] is set, bit times of logic 0 are queued as long as SBK is set. Depending on the timing of the set and clear of SBK relative to the character currently being transmitted, a second break character may be queued before software clears SBK. If software takes too long to clear SBK (for example, it is not cleared by the end of the 1st break character), then a 2nd break character is sent. This is compared to queuing a break character through the transmit FIFO which guarantees only one is sent. A single break character can also be queued by writing to the transmit FIFO with DATA[FRETSC] set and DATA[R9T9] clear. 0 - Normal transmitter operation. 1 - Queue break character(s) to be sent.
15 MA1IE	Match 1 Interrupt Enable 0 - MA1F interrupt disabled 1 - MA1F interrupt enabled
14	Match 2 Interrupt Enable

Table continues on the next page...

Field	Description
MA2IE	0 - MA2F interrupt disabled 1 - MA2F interrupt enabled
13-12 —	Reserved
11 M7	7-Bit Mode Select This bit should be changed only when the transmitter and receiver are both disabled. 0 - Receiver and transmitter use 8-bit to 10-bit data characters. 1 - Receiver and transmitter use 7-bit data characters.
10-8 IDLECFG	Idle Configuration Configures the number of idle characters that must be received before the IDLE flag is set. 000 - 1 idle character 001 - 2 idle characters 010 - 4 idle characters 011 - 8 idle characters 100 - 16 idle characters 101 - 32 idle characters 110 - 64 idle characters 111 - 128 idle characters
7 LOOPS	Loop Mode Select When LOOPS is set, the RXD pin is disconnected from the LPUART and the transmitter output is internally connected to the receiver input. The transmitter and the receiver must be enabled to use the loop function. 0 - Normal operation - RXD and TXD use separate pins. 1 - Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input (see RSRC bit).
6 DOZEEN	Doze Enable 0 - LPUART is enabled in Doze mode. 1 - LPUART is disabled in Doze mode .
5 RSRC	Receiver Source Select This field has no meaning or effect unless the LOOPS field is set. When LOOPS is set, the RSRC field determines the source for the receiver shift register input. 0 - Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the LPUART does not use the RXD pin. 1 - Single-wire LPUART mode where the TXD pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select 0 - Receiver and transmitter use 8-bit data characters. 1 - Receiver and transmitter use 9-bit data characters.
3 WAKE	Receiver Wakeup Method Select Determines which condition wakes the LPUART when RWU=1:

Table continues on the next page...

Field	Description
	• Address mark in the bit preceding the stop bit (or bit preceding the parity bit when parity is enabled) of the received data character. • An idle condition on the receive pin input signal. 0 - Configures RWU for idle-line wakeup. 1 - Configures RWU with address-mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** In case the LPUART is programmed with ILT = 1, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0 - Idle character bit count starts after start bit. 1 - Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables hardware parity generation and checking. When parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0 - No hardware parity generation or checking. 1 - Parity enabled.
0 PT	Parity Type Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of logic 1 bits in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 - Even parity. 1 - Odd parity.

71.5.1.9 LPUART Data Register (DATA)

NOTE

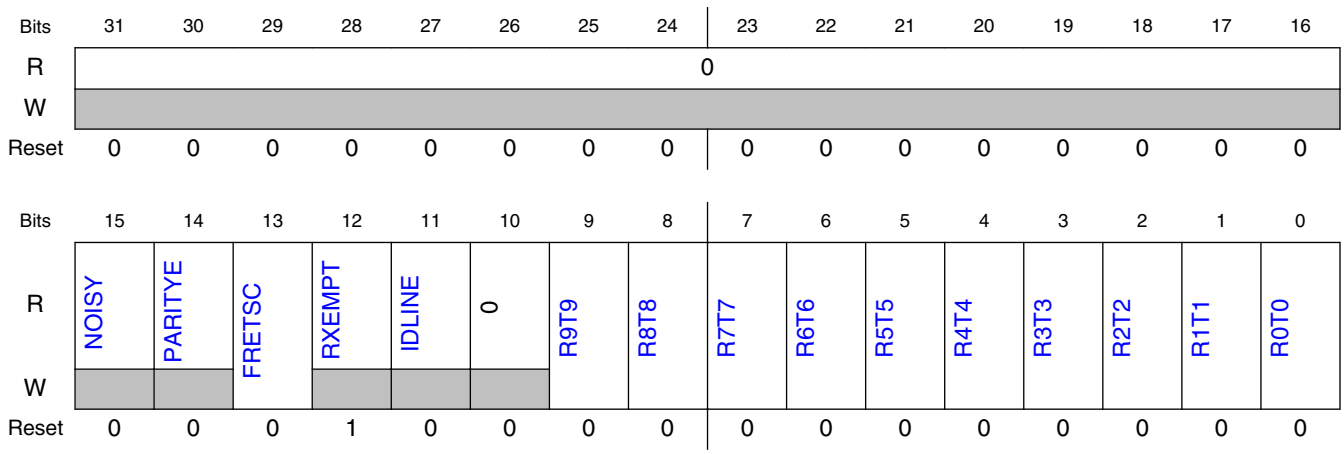
This register is two separate registers. Reads return the contents of the read-only receive FIFO and writes go to the write-only transmit FIFO.

Reads and writes of this register are also involved in the automatic flag clearing mechanisms for some of the LPUART status flags. This register can be written using 8-bit, 16-bit or 32-bit writes. An 8-bit write to DATA[7:0] will push { R8T9, R9T8, DATA[7:0] } the transmit FIFO with TSC clear. A 16-bit or 32-bit write will push the data written into the FIFO and does not update the value in R8T9 or R9T8.

71.5.1.9.1 Offset

Register	Offset
DATA	1Ch

71.5.1.9.2 Diagram



71.5.1.9.3 Fields

Field	Description
31-16 —	Reserved
15 NOISY	Noisy Data Received The current received dataword contained in DATA[R9:R0] is received with noise. 0 - The dataword is received without noise. 1 - The data is received with noise.
14 PARITYE	Parity Error The current received dataword contained in DATA[R9:R0] is received with a parity error. 0 - The dataword is received without a parity error. 1 - The dataword is received with a parity error.
13 FRETSC	Frame Error / Transmit Special Character For reads, indicates the current received dataword contained in DATA[R9:R0] was received with a frame error. For writes, indicates a break or idle character is to be transmitted instead of the contents in DATA[T9:T0]. T9 is used to indicate a break character when 0 and a idle character when 1, the contents of DATA[T8:T0] should be zero. 0 - The dataword is received without a frame error on read, or transmit a normal character on write. 1 - The dataword is received with a frame error, or transmit an idle or break character on transmit.
12	Receive Buffer Empty

Table continues on the next page...

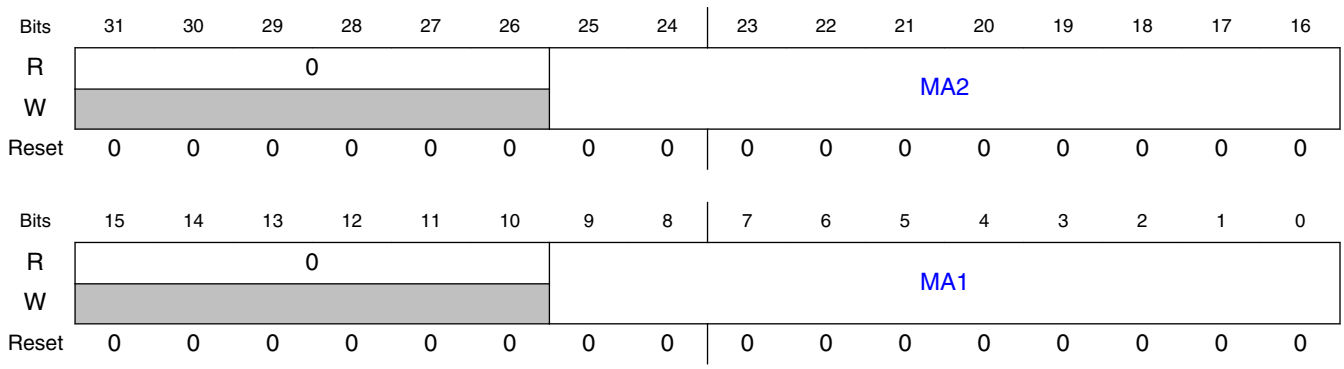
Field	Description
RXEMPT	Asserts when there is no data in the receive buffer. This field does not take into account data that is in the receive shift register. 0 - Receive buffer contains valid data. 1 - Receive buffer is empty, data returned on read is not valid.
11 IDLINE	Idle Line Indicates the receiver line was idle before receiving the character in DATA[9:0]. Unlike the IDLE flag, this bit can set for the first character received when the receiver is first enabled. 0 - Receiver was not idle before receiving this character. 1 - Receiver was idle before receiving this character.
10 —	Reserved
9 R9T9	R9T9 Read receive FIFO bit 9 or write transmit FIFO bit 9.
8 R8T8	R8T8 Read receive FIFO bit 8 or write transmit FIFO bit 8.
7 R7T7	R7T7 Read receive FIFO bit 7 or write transmit FIFO bit 7.
6 R6T6	R6T6 Read receive FIFO bit 6 or write transmit FIFO bit 6.
5 R5T5	R5T5 Read receive FIFO bit 5 or write transmit FIFO bit 5.
4 R4T4	R4T4 Read receive FIFO bit 4 or write transmit FIFO bit 4.
3 R3T3	R3T3 Read receive FIFO bit 3 or write transmit FIFO bit 3.
2 R2T2	R2T2 Read receive FIFO bit 2 or write transmit FIFO bit 2.
1 R1T1	R1T1 Read receive FIFO bit 1 or write transmit FIFO bit 1.
0 R0T0	R0T0 Read receive FIFO bit 0 or write transmit FIFO bit 0.

71.5.1.10 LPUART Match Address Register (MATCH)

71.5.1.10.1 Offset

Register	Offset
MATCH	20h

71.5.1.10.2 Diagram



71.5.1.10.3 Fields

Field	Description
31-26 —	Reserved
25-16 MA2	Match Address 2 The MA1 and MA2 fields are compared to input data addresses when the most significant bit is set and the associated BAUD[MAEN] bit is set. If a match occurs, the following data is transferred to the DATA register. If a match fails, the following data is discarded. Software should only write a MAx field when the associated BAUD[MAEN] bit is clear.
15-10 —	Reserved
9-0 MA1	Match Address 1 The MA1 and MA2 fields are compared to input data addresses when the most significant bit is set and the associated BAUD[MAEN] bit is set. If a match occurs, the following data is transferred to the DATA register. If a match fails, the following data is discarded. Software should only write a MAx field when the associated BAUD[MAEN] bit is clear.

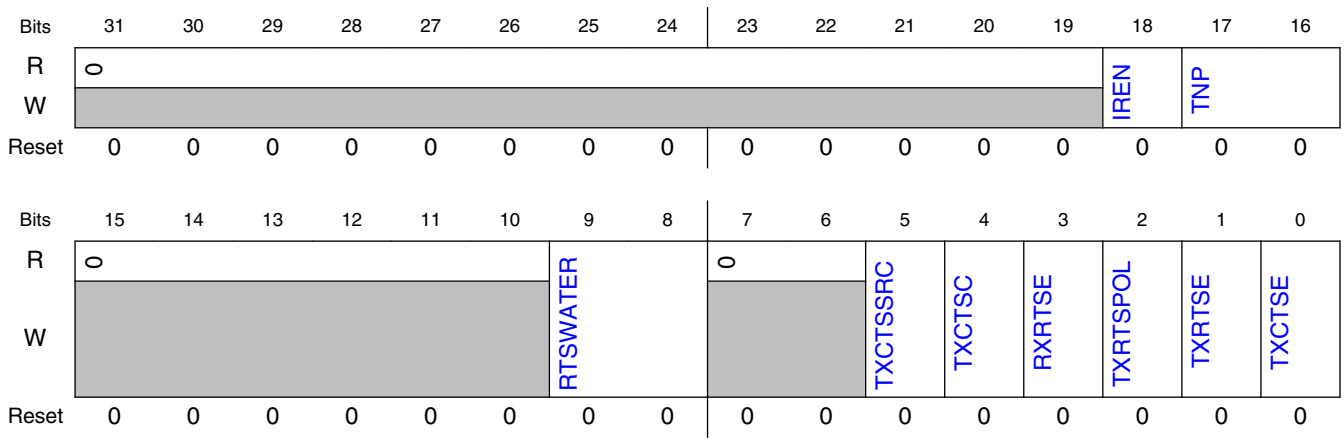
71.5.1.11 LPUART Modem IrDA Register (MODIR)

The MODIR register controls options for setting the modem configuration.

71.5.1.11.1 Offset

Register	Offset
MODIR	24h

71.5.1.11.2 Diagram



71.5.1.11.3 Fields

Field	Description
31-19 —	Reserved
18 IREN	Infrared enable Enables/disables the infrared modulation/demodulation. This bit should be changed only when the transmitter and receiver are both disabled. 0 - IR disabled. 1 - IR enabled.
17-16 TNP	Transmitter narrow pulse Configures whether the LPUART transmits a 1/OSR, 2/OSR, 3/OSR or 4/OSR narrow pulse when IR is enabled. This bit should be changed only when the transmitter and receiver are both disabled. The IR pulse width should be configured to less than half of the oversampling ratio. Common pulse widths are 3/16, 1/16, 1/32 or 1/4 of the bit length. These can be configured by selecting the appropriate oversample ratio and pulse width. 00 - 1/OSR. 01 - 2/OSR. 10 - 3/OSR. 11 - 4/OSR.
15-10	Reserved

Table continues on the next page...

Register definition

Field	Description
—	
9-8 RTSWATER	Receive RTS Configuration Configures the assertion and negation of the RX RTS_B output. The RX RTS_B output negates when the number of empty words in the receive FIFO is greater or equal to RTSWATER. If RTSWATER is configured to zero, RTS_B negates when the receive FIFO is full. For the purpose of RX RTS_B generation, the number of words in the receive FIFO updates when a start bit is detected. This supports additional latency between RTS_B negation and the external transmitter ceasing transmission. If both RX RTS_B and address/data matching is enabled, then RTS_B could assert at the end of a character if there is not a match. This field should be changed only when the receiver is disabled.
7-6 —	Reserved
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0 - CTS input is the CTS_B pin. 1 - CTS input is an internal connection to the receiver address match result.
4 TXCTSC	Transmit CTS Configuration Configures if the CTS state is checked at the start of each character or only when the transmitter is idle. 0 - CTS input is sampled at the start of each character. 1 - CTS input is sampled when the transmitter is idle.
3 RXRTSE	Receiver request-to-send enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. This bit should be changed only when the receiver is disabled. **NOTE:** Do not set both RXRTSE and TXRTSE. 0 - The receiver has no effect on RTS. 1 - RTS is deasserted if the receiver data register is full or a start bit has been detected that would cause the receiver data register to become full. RTS is asserted if the receiver data register is not full and has not detected a start bit that would cause the receiver data register to become full.
2 TXRTSPOL	Transmitter request-to-send polarity Controls the polarity of the transmitter RTS. TXRTSPOL does not affect the polarity of the receiver RTS. RTS remains negated in the active low state unless TXRTSE is set. This bit should be changed only when the transmitter is disabled. 0 - Transmitter RTS is active low. 1 - Transmitter RTS is active high.
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission. This bit should be changed only when the transmitter is disabled. 0 - The transmitter has no effect on RTS. 1 - When a character is placed into an empty transmit shift register, RTS asserts one bit time before the start bit is transmitted. RTS deasserts one bit time after all characters in the transmitter FIFO and shift register are completely sent, including the last stop bit.
0 TXCTSE	Transmitter clear-to-send enable TXCTSE controls the operation of the transmitter. TXCTSE can be set independently from the state of TXRTSE and RXRTSE.

Field	Description
	0 - CTS has no effect on the transmitter. 1 - Enables clear-to-send operation. The transmitter checks the state of CTS each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the signal TXD remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS as a character is being sent do not affect its transmission.

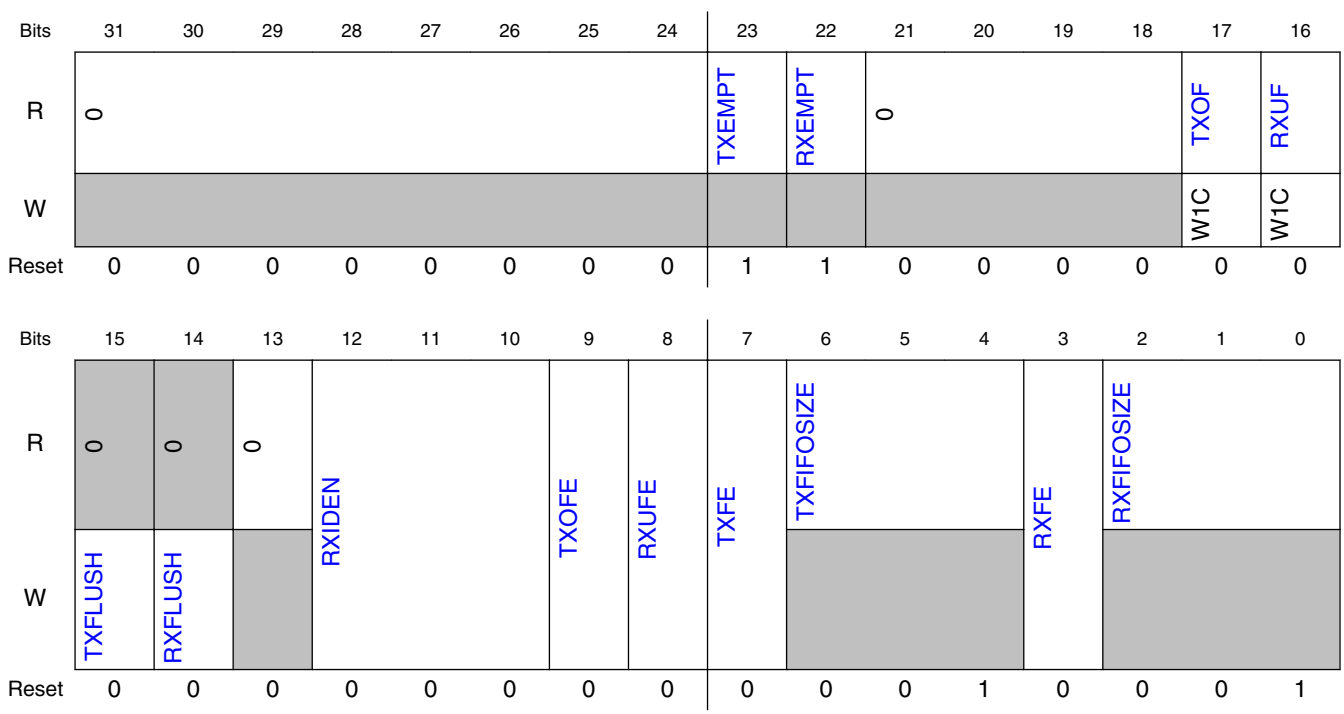
71.5.1.12 LPUART FIFO Register (FIFO)

This register provides the ability for the programmer to turn on and off FIFO functionality. It also provides the size of the FIFO that has been implemented. This register may be read at any time. This register must be written only when CTRL[RE] and CTRL[TE] are cleared/not set and when the FIFO is empty.

71.5.1.12.1 Offset

Register	Offset
FIFO	28h

71.5.1.12.2 Diagram



71.5.1.12.3 Fields

Field	Description
31-24 —	Reserved
23 TXEMPT	Transmit FIFO/Buffer Empty Asserts when there is no data in the Transmit FIFO/Buffer. This field does not take into account data that is in the transmit shift register. 0 - Transmit buffer is not empty. 1 - Transmit buffer is empty.
22 RXEMPT	Receive FIFO/Buffer Empty Asserts when there is no data in the Receive FIFO/Buffer. This field does not take into account data that is in the receive shift register. 0 - Receive buffer is not empty. 1 - Receive buffer is empty.
21-18 —	Reserved
17 TXOF	Transmitter FIFO Overflow Flag Indicates that more data has been written to the transmit FIFO than it can hold. This field asserts regardless of the value of TXOFE. However, an interrupt is issued to the host only if TXOFE is set. Write 1 to clear this flag. 0 - No transmit FIFO overflow has occurred since the last time the flag was cleared. 1 - At least one transmit FIFO overflow has occurred since the last time the flag was cleared.
16 RXUF	Receiver FIFO Underflow Flag Indicates that more data has been read from the receive FIFO than was present. This field asserts regardless of the value of RXUFE. However, an interrupt is issued to the host only if RXUFE is set. Write a logic 1 to clear RXUF. 0 - No receive FIFO underflow has occurred since the last time the flag was cleared. 1 - At least one receive FIFO underflow has occurred since the last time the flag was cleared.
15 TXFLUSH	Transmit FIFO Flush Writing to this field causes all data that is stored in the transmit FIFO to be flushed. This does not affect data that is in the transmit shift register. 0 - No flush operation occurs. 1 - All data in the transmit FIFO is cleared out.
14 RXFLUSH	Receive FIFO Flush Writing to this field causes all data that is stored in the receive FIFO to be flushed. This does not affect data that is in the receive shift register. 0 - No flush operation occurs. 1 - All data in the receive FIFO/buffer is cleared out.
13 —	Reserved
12-10	Receiver Idle Empty Enable

Table continues on the next page...

Field	Description
RXIDEN	When set, enables the assertion of RDRF when the receiver is idle for a number of idle characters and the FIFO is not empty. 000 - Disable RDRF assertion due to partially filled FIFO when receiver is idle. 001 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 1 character. 010 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 2 characters. 011 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 4 characters. 100 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 8 characters. 101 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 16 characters. 110 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 32 characters. 111 - Enable RDRF assertion due to partially filled FIFO when receiver is idle for 64 characters.
9 TXOFE	Transmit FIFO Overflow Interrupt Enable When this field is set, the TXOF flag generates an interrupt to the host. 0 - TXOF flag does not generate an interrupt to the host. 1 - TXOF flag generates an interrupt to the host.
8 RXUFE	Receive FIFO Underflow Interrupt Enable When this field is set, the RXUF flag generates an interrupt to the host. 0 - RXUF flag does not generate an interrupt to the host. 1 - RXUF flag generates an interrupt to the host.
7 TXFE	Transmit FIFO Enable When this field is set, the built-in FIFO structure for the transmit buffer is enabled. The size of the FIFO structure is indicated by TXFIFOSIZE. If this field is not set, the transmit buffer operates as a FIFO of depth one dataword regardless of the value in TXFIFOSIZE. Both CTRL[TE] and CTRL[RE] must be cleared prior to changing this field. 0 - Transmit FIFO is not enabled. Buffer depth is 1. 1 - Transmit FIFO is enabled. Buffer depth is indicated by TXFIFOSIZE.
6-4 TXFIFOSIZE	Transmit FIFO Buffer Depth The maximum number of transmit datawords that can be stored in the transmit buffer. This field is read-only. 000 - Transmit FIFO/Buffer depth = 1 dataword. 001 - Transmit FIFO/Buffer depth = 4 datawords. 010 - Transmit FIFO/Buffer depth = 8 datawords. 011 - Transmit FIFO/Buffer depth = 16 datawords. 100 - Transmit FIFO/Buffer depth = 32 datawords. 101 - Transmit FIFO/Buffer depth = 64 datawords. 110 - Transmit FIFO/Buffer depth = 128 datawords. 111 - Transmit FIFO/Buffer depth = 256 datawords
3 RXFE	Receive FIFO Enable When this field is set, the built-in FIFO structure for the receive buffer is enabled. The size of the FIFO structure is indicated by the RXFIFOSIZE field. If this field is not set, the receive buffer operates as a FIFO of depth one dataword regardless of the value in RXFIFOSIZE. Both CTRL[TE] and CTRL[RE] must be cleared prior to changing this field.

Table continues on the next page...

Field	Description
	0 - Receive FIFO is not enabled. Buffer depth is 1. 1 - Receive FIFO is enabled. Buffer depth is indicted by RXFIFOSIZE.
2-0 RXFIFOSIZE	Receive FIFO Buffer Depth The maximum number of receive datawords that can be stored in the receive buffer before an overrun occurs. This field is read-only. 000 - Receive FIFO/Buffer depth = 1 dataword. 001 - Receive FIFO/Buffer depth = 4 datawords. 010 - Receive FIFO/Buffer depth = 8 datawords. 011 - Receive FIFO/Buffer depth = 16 datawords. 100 - Receive FIFO/Buffer depth = 32 datawords. 101 - Receive FIFO/Buffer depth = 64 datawords. 110 - Receive FIFO/Buffer depth = 128 datawords. 111 - Receive FIFO/Buffer depth = 256 datawords.

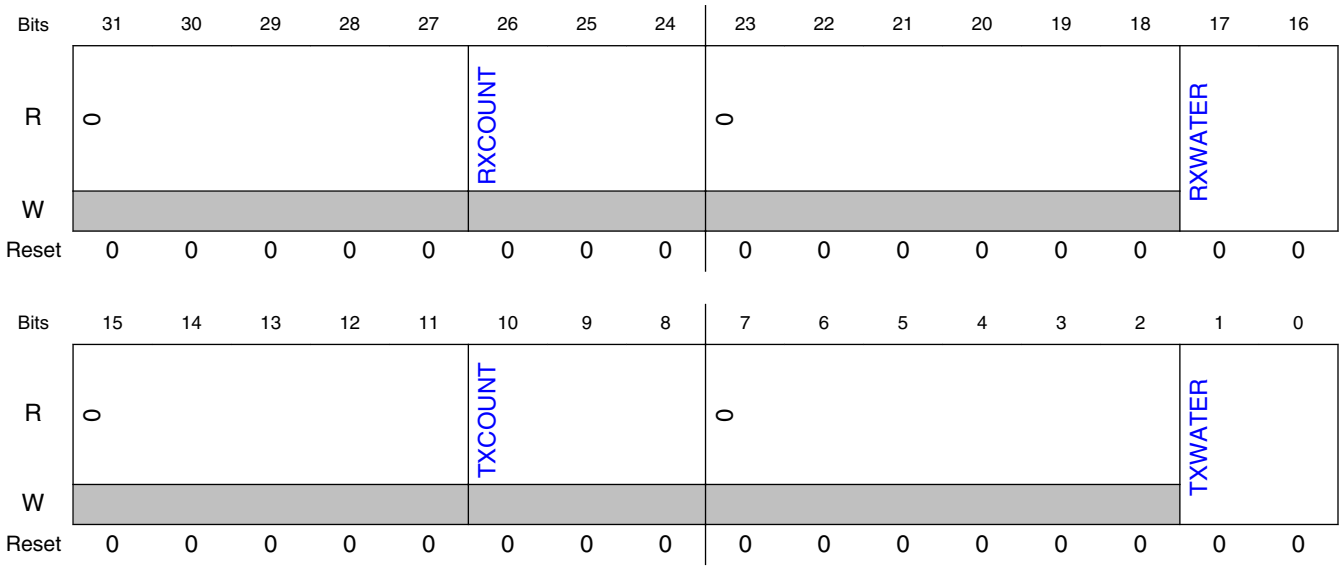
71.5.1.13 LPUART Watermark Register (WATER)

This register provides the ability to set a programmable threshold for notification of needing additional transmit data. This register may be read at any time but must be written only when CTRL[TE] is not set.

71.5.1.13.1 Offset

Register	Offset
WATER	2Ch

71.5.1.13.2 Diagram



71.5.1.13.3 Fields

Field	Description
31-27 —	Reserved
26-24 RXCOUNT	Receive Counter The value in this register indicates the number of datawords that are in the receive FIFO/buffer. If a dataword is being received, that is, in the receive shift register, it is not included in the count. This value may be used in conjunction with FIFO[RXFIFOSIZE] to calculate how much room is left in the receive FIFO/buffer.
23-18 —	Reserved
17-16 RXWATER	Receive Watermark When the number of datawords in the receive FIFO/buffer is greater than the value in this register field, an interrupt or a DMA request is generated. For proper operation, the value in RXWATER must be set to be less than the receive FIFO/buffer size as indicated by FIFO[RXFIFOSIZE] and FIFO[RXFE].
15-11 —	Reserved
10-8 TXCOUNT	Transmit Counter The value in this register indicates the number of datawords that are in the transmit FIFO/buffer. If a dataword is being transmitted, that is, in the transmit shift register, it is not included in the count. This value may be used in conjunction with FIFO[TXFIFOSIZE] to calculate how much room is left in the transmit FIFO/buffer.
7-2 —	Reserved
1-0	Transmit Watermark

Register definition

Field	Description
TXWATER	When the number of datawords in the transmit FIFO/buffer is equal to or less than the value in this register field, an interrupt or a DMA request is generated. For proper operation, the value in TXWATER must be set to be less than the size of the transmit buffer/FIFO size as indicated by FIFO[TXFIFOSIZE] and FIFO[TXFE].

Chapter 72
Timers Overview

72.1 Overview

The following timers are supported in this chip:

- General Purpose Timer (GPT): A 32-bit up-counter with 12-bit pre-scaler
- Periodic Interrupt Timer (PIT): A 32-bit counter timer that features programmable count modulus, clock division features etc.
- Quad Timer (TMR): It provides four timer channels with variety of controls for individual and multi-channel features
- Quadrature Decoder (QDC): It provides interfacing capability to position/speed sensors
- Enhanced FlexPWM: It contains PWM submodules each of which is set up to control a single half bridge power stage
- Watchdog Timer (WDOG1,2): The WDOG1 and WDOG2 protect against system failures by providing a method by which to escape from unexpected events or programming errors
- Watchdog timer (WDOG3,4): It is a high reliability independent timer that is available for system use
- External Watchdog Monitor (EWM): It is designed to monitor external circuits, as well as the MCU software flow

72.1.1 General Purpose Timer (GPT)

The GPT has a 32-bit up-counter. The timer counter value can be captured in a register using an event on an external pin. The capture trigger can be programmed to be a rising or/and falling edge. The GPT also features output compare event generation, when the timer matches a preset value, GPT can generate an interrupt for software and a compare event signal on output pins for external hardware.

The GPT has a 12-bit pre-scaler, which provides a programmable clock frequency derived from multiple clock sources.

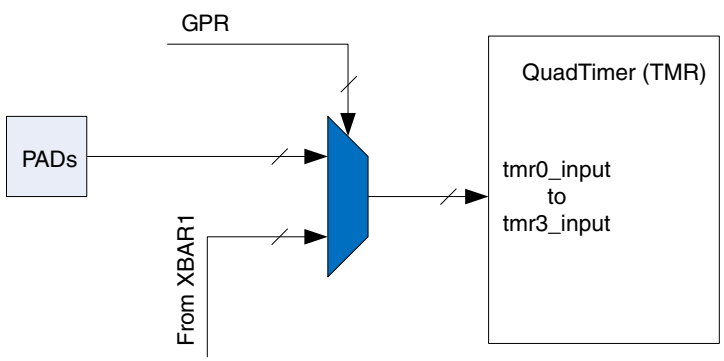
72.1.2 Periodic Interrupt Timer (PIT)

The PIT is a basic 32 bit counter timer. It features 32-bit counter timer, programmable count modulus, clock division features, interrupt generation, and ability to chain adjacent timers to achieve longer interval

72.1.3 Quad Timer (TMR)

The quad-timer provides four timer channels. Specific features include up/down count, cascading of counters, programmable modulo, count once/repeated, counter preload, compare registers with preload, shared use of input signals, prescaler controls, independent capture/compare, fault input control, programmable input filters, and multi-channel synchronization.

As shown in diagram below, input of QTimer can be sourced from PADs or XBAR1:



72.1.4 Enhanced Flex Pulse Width Modulator (eFlexPWM)

This module can generate various switching patterns, including highly sophisticated waveforms. Each module shall be configured with 4 channels supporting A, B, and X PWM output. All channels are configured for standard edge placement and have their capture function enabled.

72.1.5 Quadrature Decoder (QDC)

The enhanced Quadrature Decoder module provides interfacing capability to position/ speed sensors used in industrial motor control applications. It has five input signals: PHASEA, PHASEB, INDEX, TRIGGER, and HOME. This module is used to decode shaft position, revolution count and speed.

72.1.6 Watchdog Timer (WDOG1,2)

The WDOG1 and WDOG2 protect against system failures by providing a method by which to escape from unexpected events or programming errors.

After WDOG1 and WDOG2 are activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG1 asserts the internal system reset signal, WDOG_RESET_B_DEB to the System Reset Controller (SRC); and WDOG2 asserts interrupt to SNVS to report the security violation condition.

Main features of the WDOG1 and WDOG2 module are:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode
- Programmable interrupt generation prior to timeout

72.1.7 Watchdog Timer (WDOG3,4)

The WDOG module is an high reliability independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the CPU (WDOG3 for M7 and WDOG4 for M4).

Main features of the WDOG module are:

- Configurable independent clock source input from bus clock
- Programmable time-out period

- Robust write sequence for counter refresh
- Windowed refresh option
- Optional timeout interrupt to allow post-processing diagnostics
- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered
- Robust write sequence for unlocking write-once control/configuration bits

The WDOG3/WDOG4 clock can be sourced from:

- 1MHz RC OSC
- 32KHz clock generated by 32 KHZ XTALOSC which could be automatically switched to 32KHz RC OSC upon XTAL clock loss
- IP Bus Clock

72.1.8 External Watchdog Monitor (EWM)

The External Watchdog Monitor provides a back-up mechanism to the internal WDOG that can reset the system. The EWM differs from the internal WDOG in that it does not reset the system.

The EWM, if allowed to time-out, provides an independent trigger pin that when asserted resets or places an external circuit into a safe mode.

The EWM clock can be sourced from:

- lpo_clk_0: 16MHz RC Oscillator
- lpo_clk_1: 1MHz RC Oscillator
- lpo_clk_2: 32KHz clock generated by 32 KHZ XTALOSC which could be automatically switched to 32KHz RC OSC upon XTAL clock loss
- lpo_clk_3: IP Bus Clock

Chapter 73
Enhanced Flex Pulse Width Modulator (eFlexPWM)

73.1 Chip-specific FlexPWM information

Table 73-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

In this device, PWM*x*_EXTB is not applicable. It also does not support external ADC input. Also there is **no** NanoEdge placement block in this device.

73.2 Overview

The Pulse Width Modulator (PWM) module contains PWM submodules, each of which can be used to control a single half-bridge power stage. Fault channel support is provided.

This module generates various switching patterns, including highly sophisticated waveforms. It can be used to control all known motor types and is ideal for controlling different Switched Mode Power Supplies (SMPS) topologies.

73.2.1 Block diagram

The following figure shows the PWM block diagram.

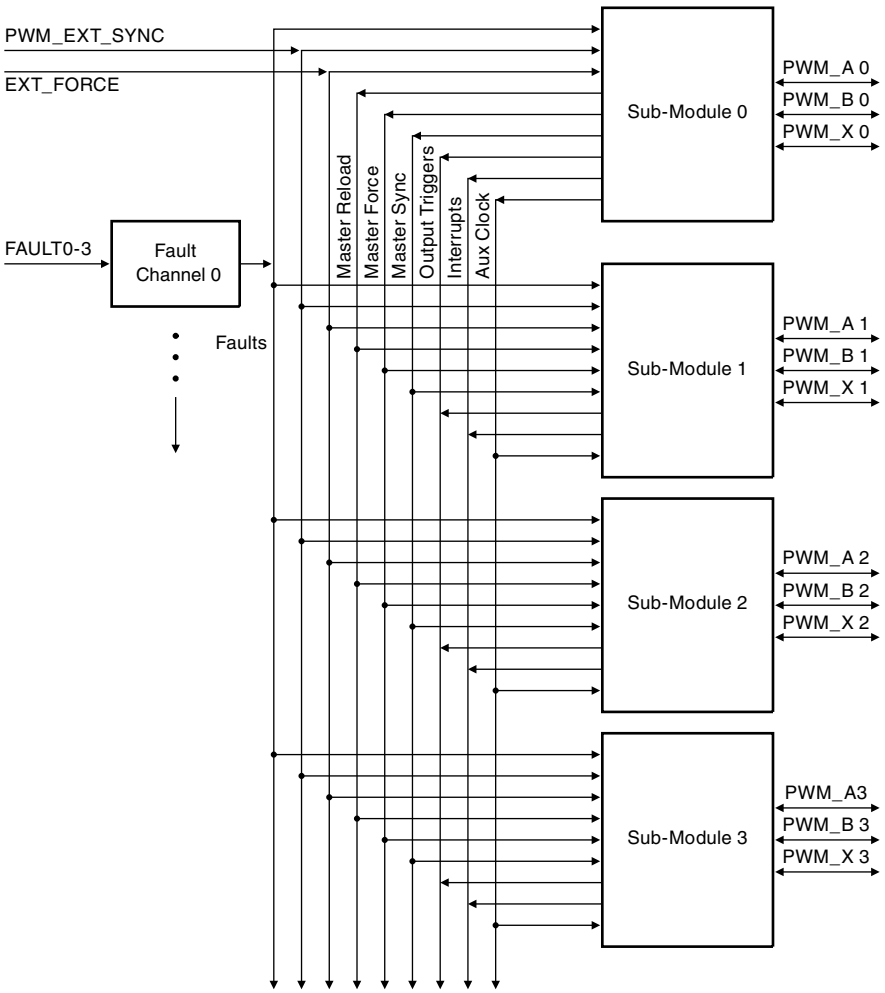


Figure 73-1. PWM block diagram

73.2.2 Features

Following are the features of PWM:

- 16-bit resolution for center, edge-aligned, and asymmetrical PWMs
- Dithering to simulate enhanced resolution when fine edge placement is not available
- PWM outputs that can operate as complementary pairs or independent channels
- Ability to accept signed numbers for PWM generation
- Independent control of both edges of each PWM output
- Support for synchronization to external hardware or other PWM
- Double buffered PWM registers

    - Integral reload rates from 1 to 16
    - Half cycle reload capability
- Multiple output trigger events can be generated per PWM cycle via hardware
- Support for double switching PWM outputs
- Fault inputs can be assigned to control multiple PWM outputs
- Programmable filters for fault inputs
- Independently programmable PWM output polarity
- Independent top and bottom deadtime insertion
- Each complementary pair can operate with its own PWM frequency and deadtime values
- Individual software control for each PWM output
- All outputs can be programmed to change simultaneously via a FORCE_OUT event
- PWM_X pin can optionally output a third PWM signal from each submodule
- Channels not used for PWM generation can be used for buffered output compare functions and for input capture functions
- Enhanced dual edge capture functionality

73.3 Functional description

73.3.1 PWM submodule

The following figure shows the PWM Submodule Block Diagram.

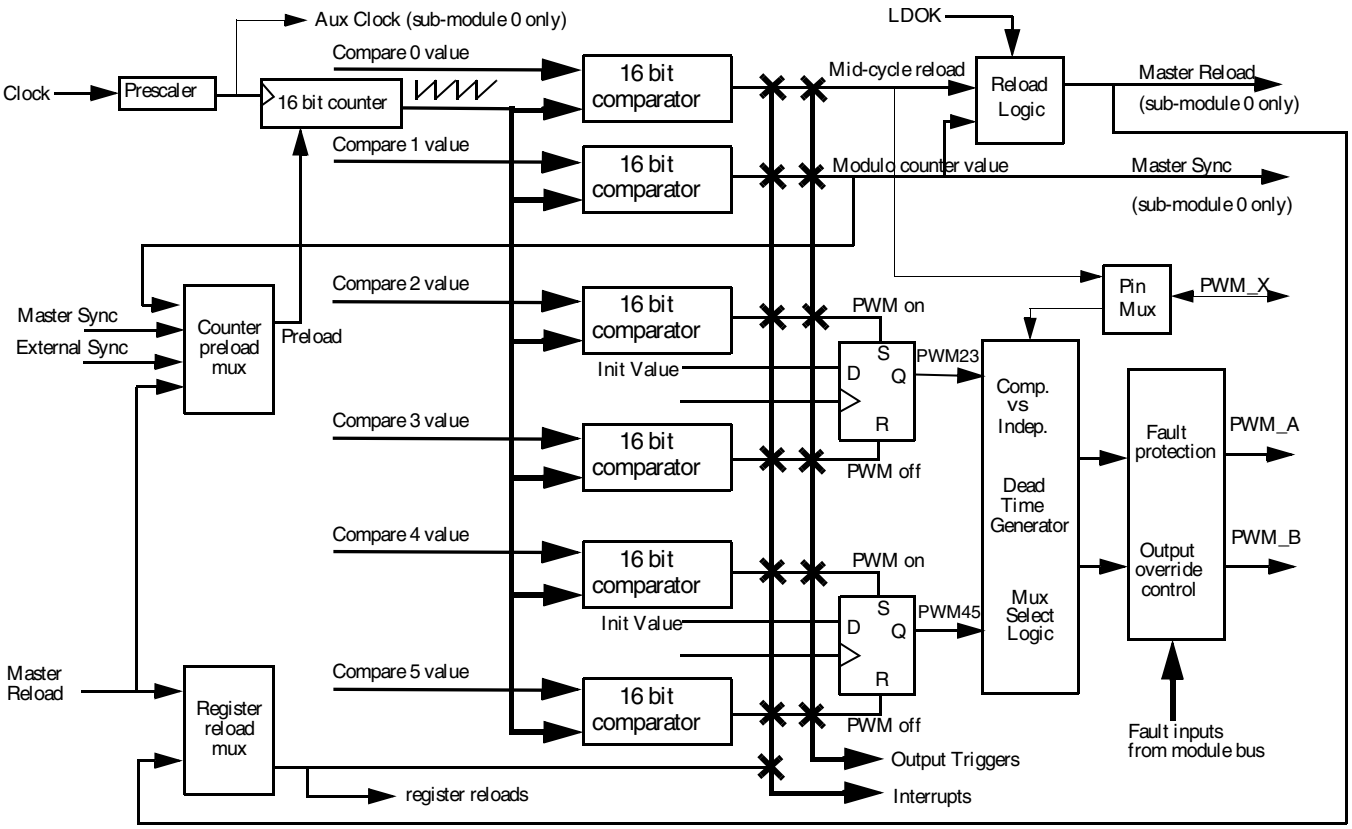


Figure 73-2. PWM submodule block diagram

73.3.2 PWM capabilities

This section describes some capabilities of the PWM module.

73.3.2.1 Center aligned PWMs

Each submodule has its own timer that is capable of generating PWM signals on two output pins. The edges of each of these signals are controlled independently as shown in Figure 73-3.

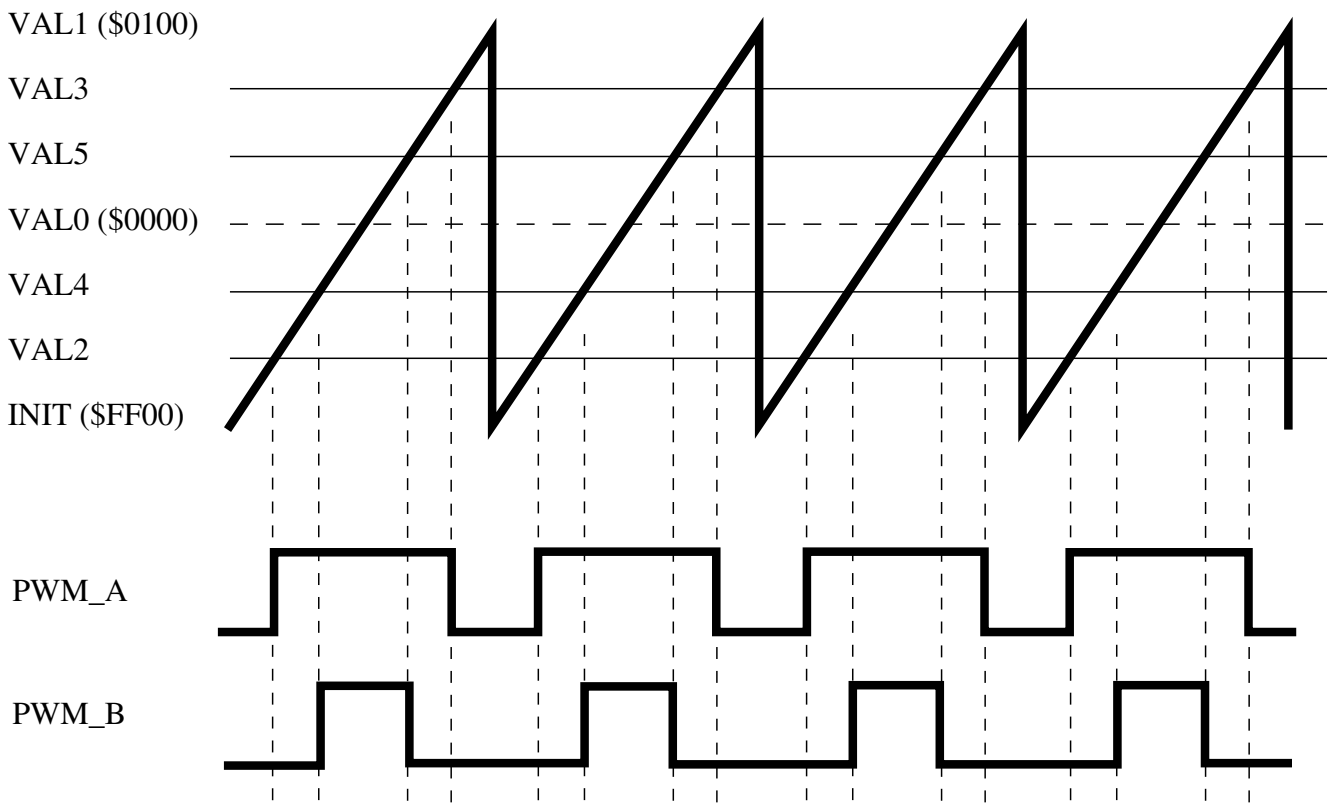


Figure 73-3. Center aligned PWM example

The submodule timers only count in the up direction and then reset to the INIT value. Instead of having a single value that determines pulse width, there are two values that must be specified: the turn-on edge and the turn-off edge. This double-action edge generation provides the user control over the pulse width and also the relative alignment of the signal. As a result, there is no need to support separate PWM alignment modes since the PWM alignment mode is inherently a function of the turn-on and turn-off edge values.

Figure 73-3 also illustrates an additional enhancement to the PWM generation process. When the counter resets, it is reloaded with a user-specified value, which may or may not be zero. If the value chosen happens to be the 2's complement of the modulus value, then the PWM generator operates in "signed" mode. This means if each PWM's turn-on and turn-off edge values are same in numbers but different in their sign, the "on" portion of the output signal is centered around a count value of zero. Therefore, only one PWM value is calculated in software and then this value and its negative are provided to the submodule as the turn-off and turn-on edges respectively. This technique results in a pulse width consists of an odd number of timer counts. If all PWM signal edge calculations follow this convention, then the signals will be center aligned with each other, which is the goal. The center alignment between the signals is not restricted to

symmetry around the zero count value, as any other number would also work. However, centering on zero provides the greatest range in signed mode and also simplifies the calculations.

73.3.2.2 Edge aligned PWMs

Figure 73-4 shows the results of edge aligned operation when the turn-on edge for each pulse is specified to be the INIT value. Therefore, only the turn-off edge value needs to be periodically updated to change the pulse width.

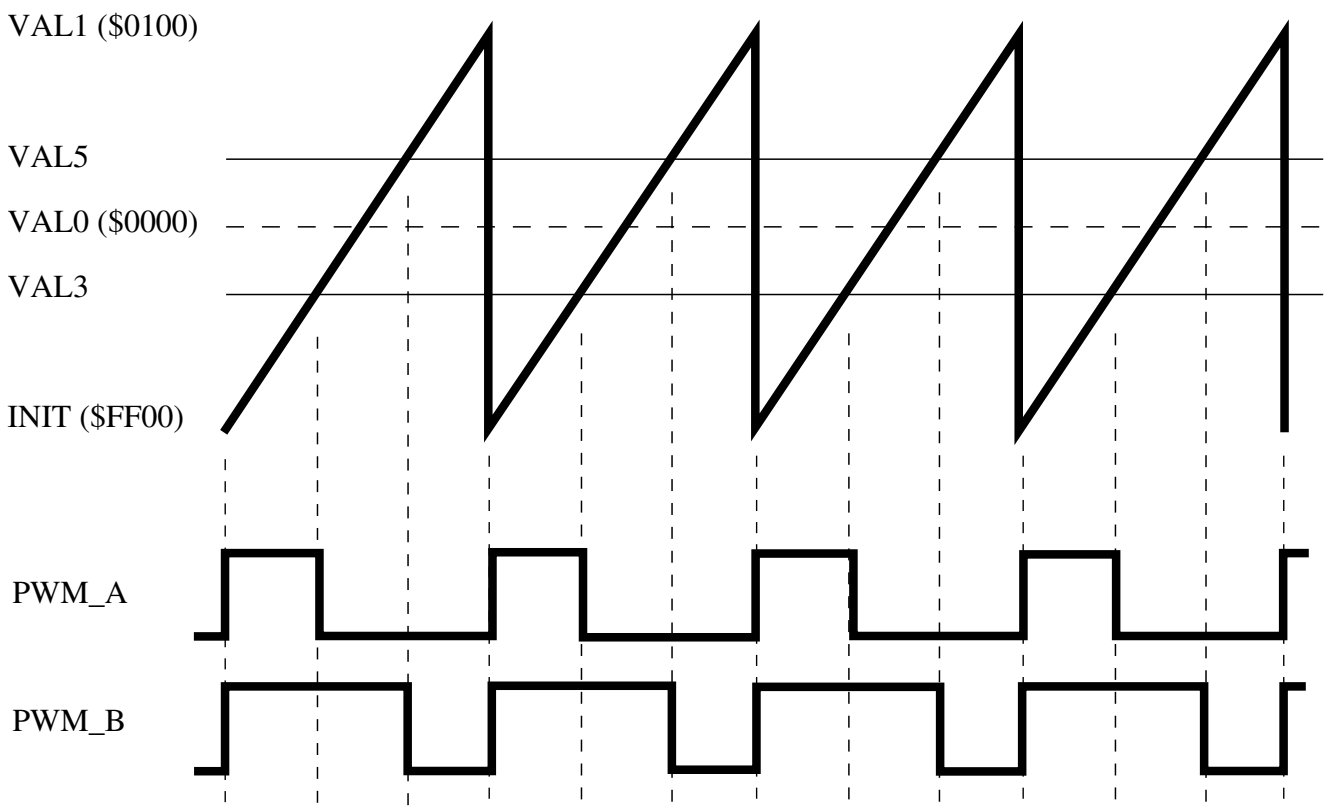


Figure 73-4. Edge aligned example (INIT=VAL2=VAL4)

With edge aligned PWMs, another example of the benefits of signed mode can be seen. Use "bipolar" PWMs to drive an H-bridge, where a 50% duty cycle results in zero volts on the load. Duty cycles less than 50% generate negative load voltages and duty cycles greater than 50% generate positive load voltages. If the module is set to signed mode operation (the INIT and VAL1 values are the same number with opposite signs), then there is a direct proportionality between the PWM turn-off edge value and the motor inverter voltage, including the sign. Therefore, signed mode of operation simplifies the software interface to the PWM module since no offset calculations are required to translate the output variable control algorithm to the voltage on an H-Bridge load.

73.3.2.3 Phase shifted PWMs

In the previous sections, the benefits of the signed mode of operation were discussed in the context of simplifying the required software calculations by eliminating the requirement to bias up signed variables before applying them to the module. However, if numerical biases are applied to the turn-on and turn-off edges of different PWM signals, the signals will be phase shifted to each other, as shown in Figure 73-5. This results in certain advantages when applied to a power stage. For example, when operating a multi-phase inverter at a low modulation index, all of the PWM switching edges from the different phases occur at the same time. This can be troublesome from a noise standpoint, especially if ADC readings of the inverter must be scheduled near those times. Phase shifting the PWM signals can open up timing windows between the switching edges to allow a signal to be sampled by the ADC. However, phase shifting does not affect the duty cycle so average load voltage is not affected.

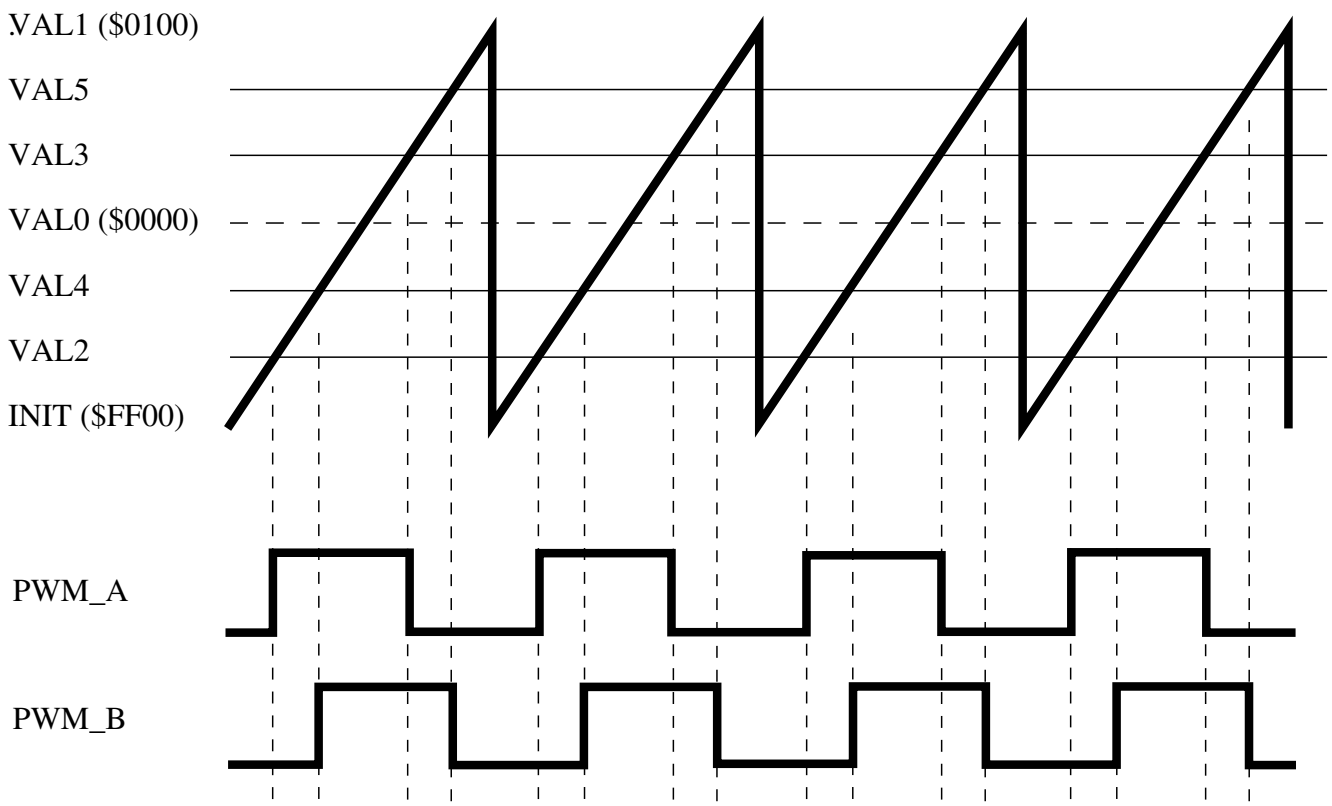


Figure 73-5. Phase shifted outputs - example for 1 submodule

An additional benefit of phase shifted PWMs is shown in Figure 73-6. In this case, an H-Bridge circuit is driven by 4 PWM signals to control the voltage waveform on the primary of a transformer. Both left and right side PWMs are configured to generate a square wave with 50% duty cycle. This works for the H-Bridge since no narrow pulse widths are generated reducing the high frequency switching requirements of the

transistors. The RMS value of this waveform is directly controlled by the amount of phase shift of the square waves. Regardless of the phase shift, no DC component appears in the load voltage as long as the duty cycle of each square wave remains at 50% suitable for transformer loads. As a result, this topology is frequently used in industrial welders to adjust the amount of energy delivered to the weld arc.

Note

The square wave on the right side of the H-Bridge is phase shifted compared to the left side of the H-Bridge. As a result, the transformer primary sees the bottom waveform across its terminals.

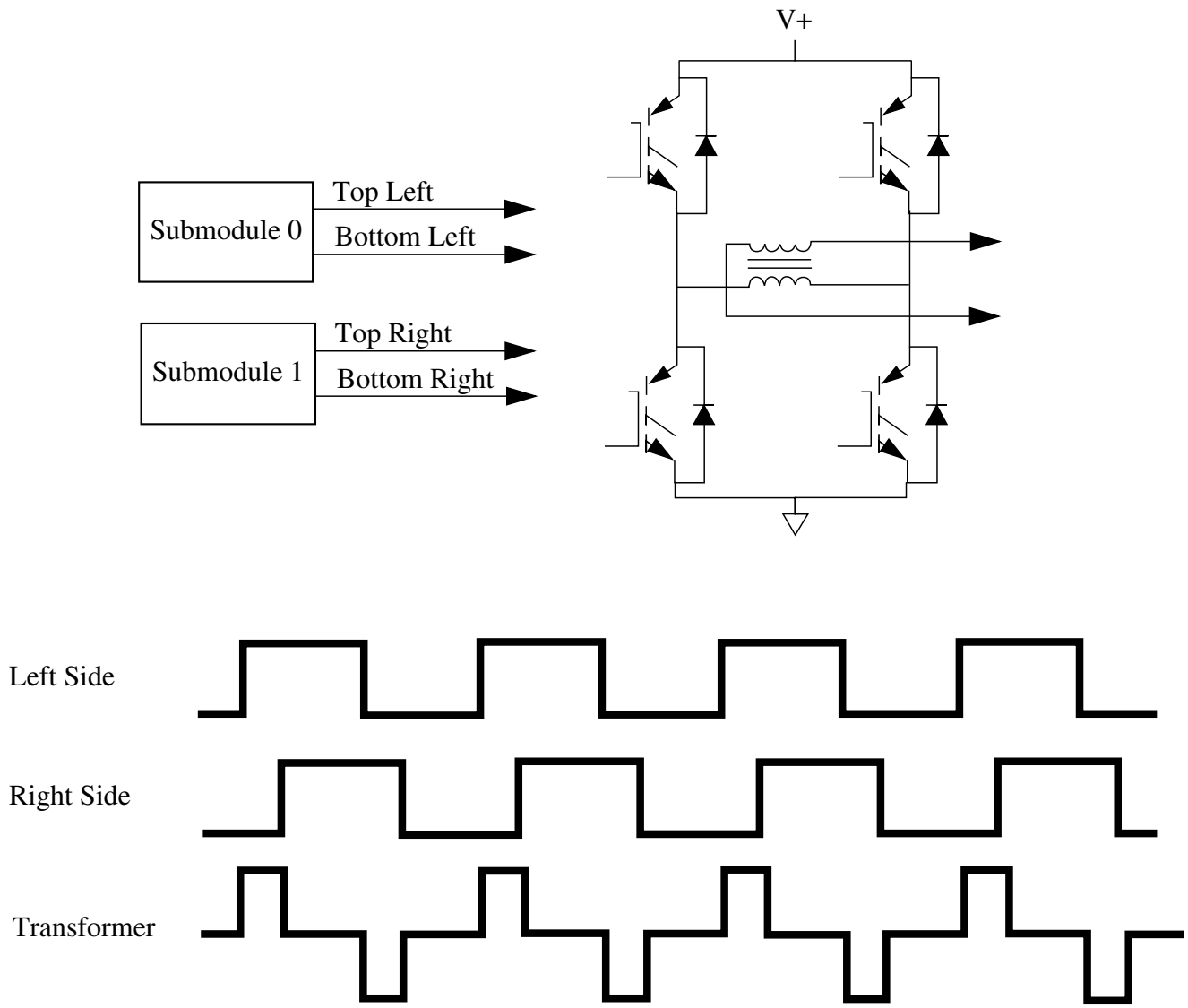


Figure 73-6. Phase Shifted PWMs Applied to a Transformer Primary

73.3.2.4 Double switching PWMs

Double switching PWM output is supported to aid in single shunt current measurement and three-phase reconstruction. This method supports two independent rising edges and two independent falling edges per PWM cycle. The VAL2 and VAL3 registers are used to generate the even channel (labeled as PWM_A in Figure 73-7) while VAL4 and VAL5 are used to generate the odd channel. The two channels (PWM23 or PWM_A and PWM45 or PWM_B from force out logic) are combined using XOR logic (force out logic) as the following figure shows. The DBLPWM signal can be run through the deadtime insertion logic.

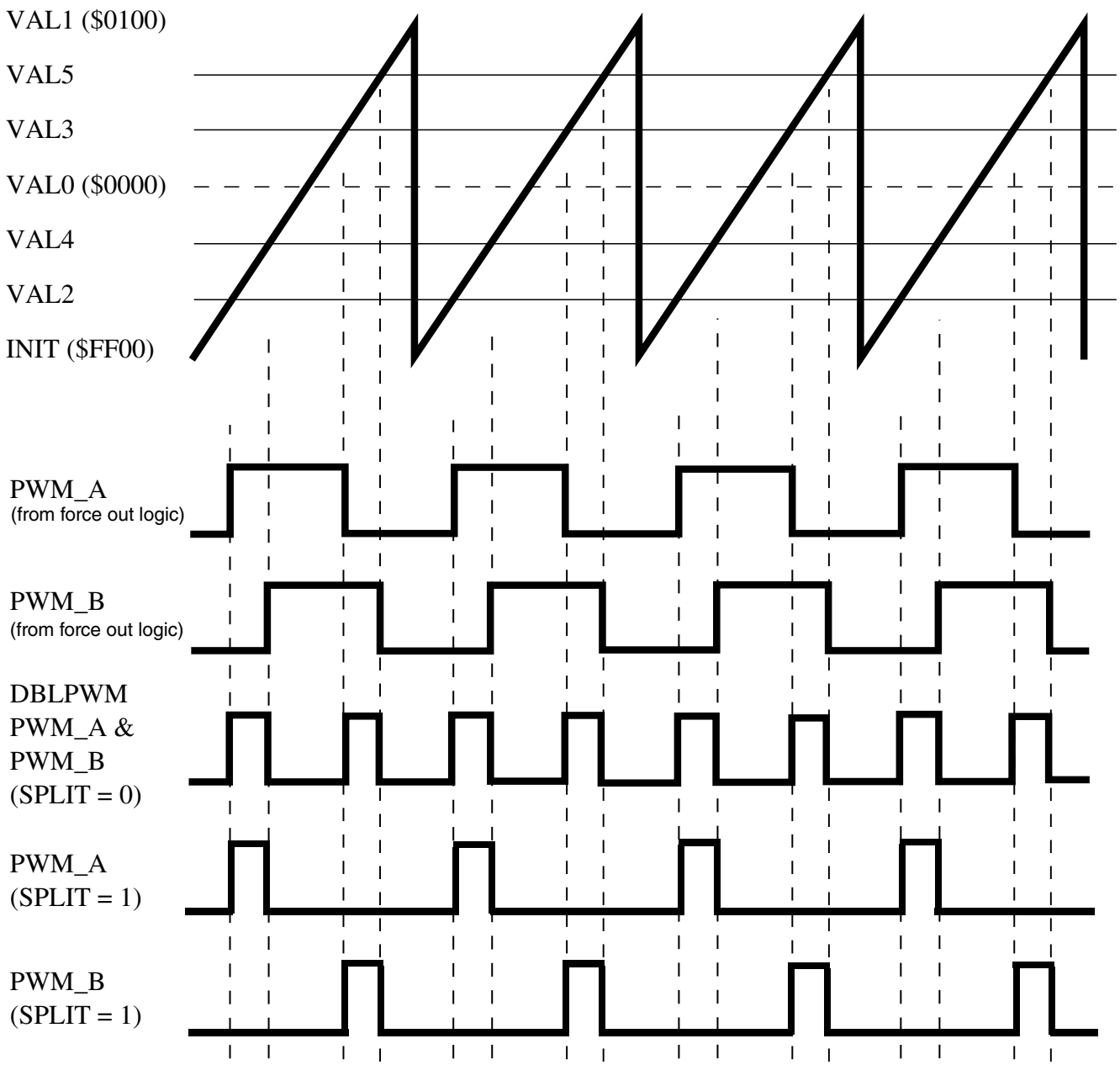


Figure 73-7. Double switching output example

73.3.2.5 ADC triggering

In cases where the timing of the ADC triggering is critical, it must be scheduled as a hardware event instead of software activated. With this PWM module, multiple ADC triggers can be generated in hardware per PWM cycle without the requirement of another timer module. Figure 73-8 shows how this is accomplished. When specifying a complementary mode of operation, only two edge comparators are required to generate the output PWM signals for a given submodule. This means the other comparators are free to perform other functions. In this example, the software does not need to quickly respond after the first conversion to set up other conversions that must occur in the same PWM cycle.

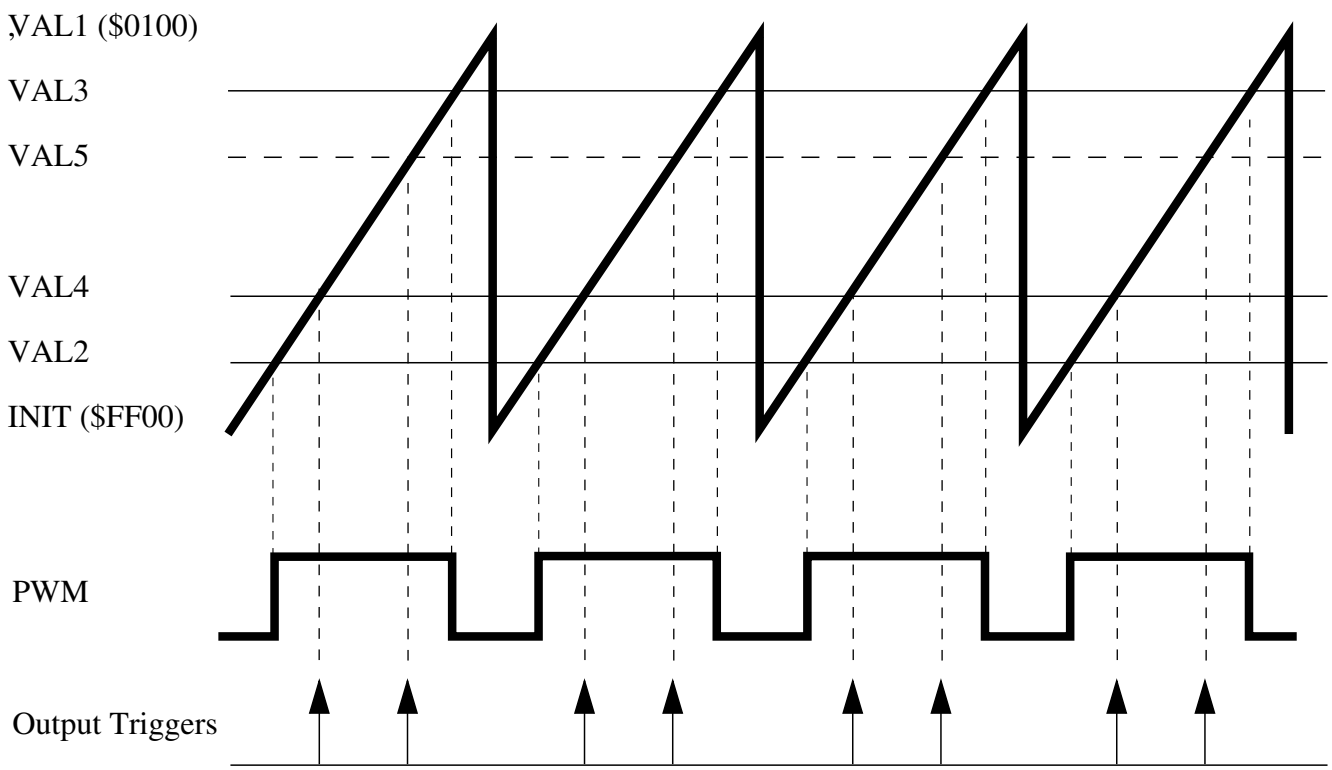


Figure 73-8. Multiple output trigger generation in hardware

Because each submodule has its own timer, it is possible for each submodule to run at a different frequency. One of the options possible with this PWM module is to have one or more submodules running at a lower frequency, but still synchronized to the timer in submodule0. Figure 73-9 shows how this feature can be used to schedule ADC triggers over multiple PWM cycles. You can use the lower-frequency submodule to control the sampling frequency of the software control algorithm where multiple ADC triggers can now be scheduled over the entire sampling period. In Figure 73-9, *all* submodule comparators are shown being used for ADC trigger generation.

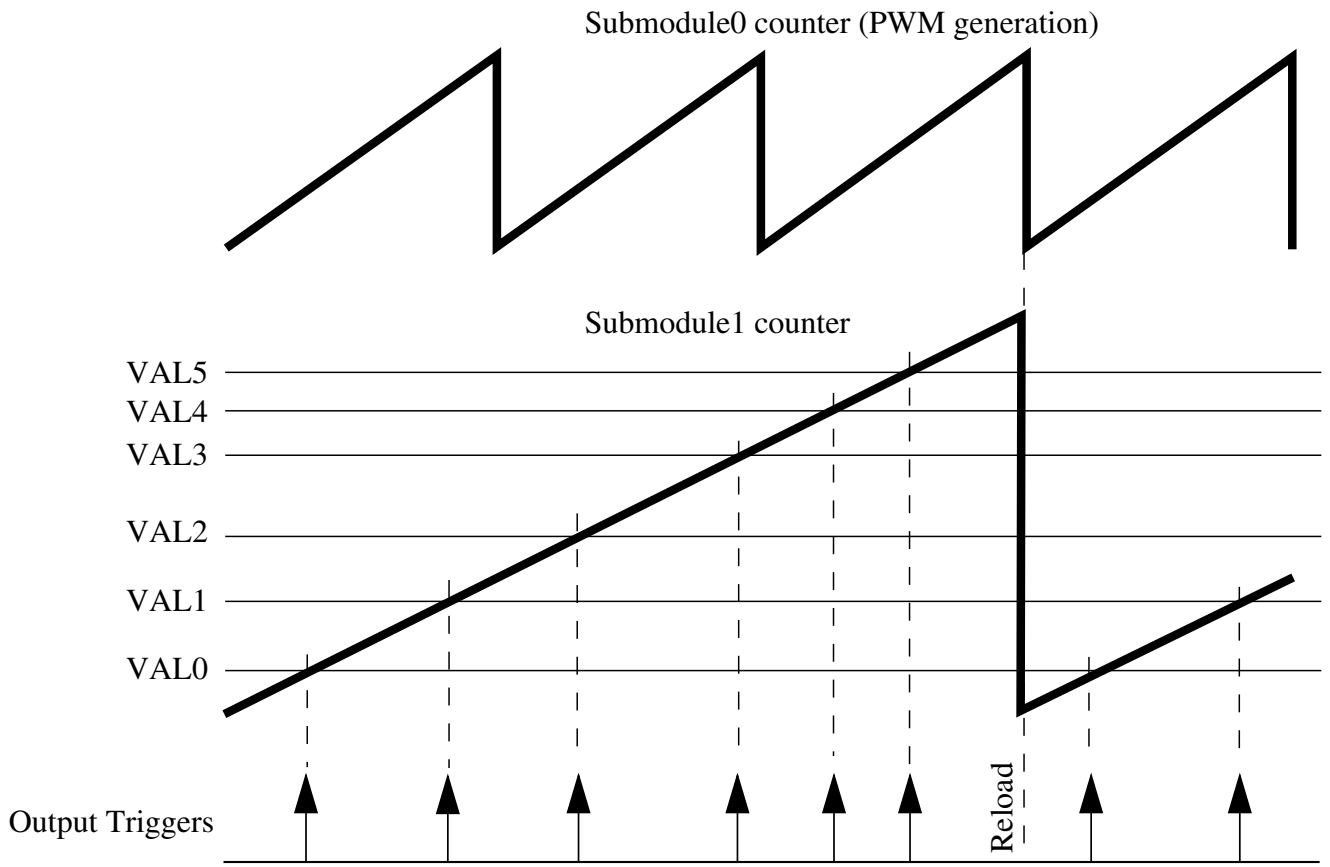


Figure 73-9. Multiple output triggers over several PWM cycles

73.3.2.6 Enhanced capture capabilities (E-Capture)

When a PWM pin is not used for PWM generation, it can be used to perform input captures. For PWM generation, both edges of the PWM signals are specified via separate compare register values. When programmed for input capture, both of these registers work on the same pin to capture multiple edges, toggling from one to the other in either a free running or one-shot fashion. Programming the desired edge of each capture circuit, period, and pulse width of an input signal can easily be measured without the requirement to re-arm the circuit. In addition, each edge of the input signal can clock an 8-bit counter where the counter output is compared to a user specified value (EDGCMP). When the counter output equals EDGCMP, the value of the submodule timer is captured and the counter is automatically reset. This feature counts a specified number of edge events and then performs a capture and interrupt. The following figure illustrates some of the functionality of the E-Capture circuit.

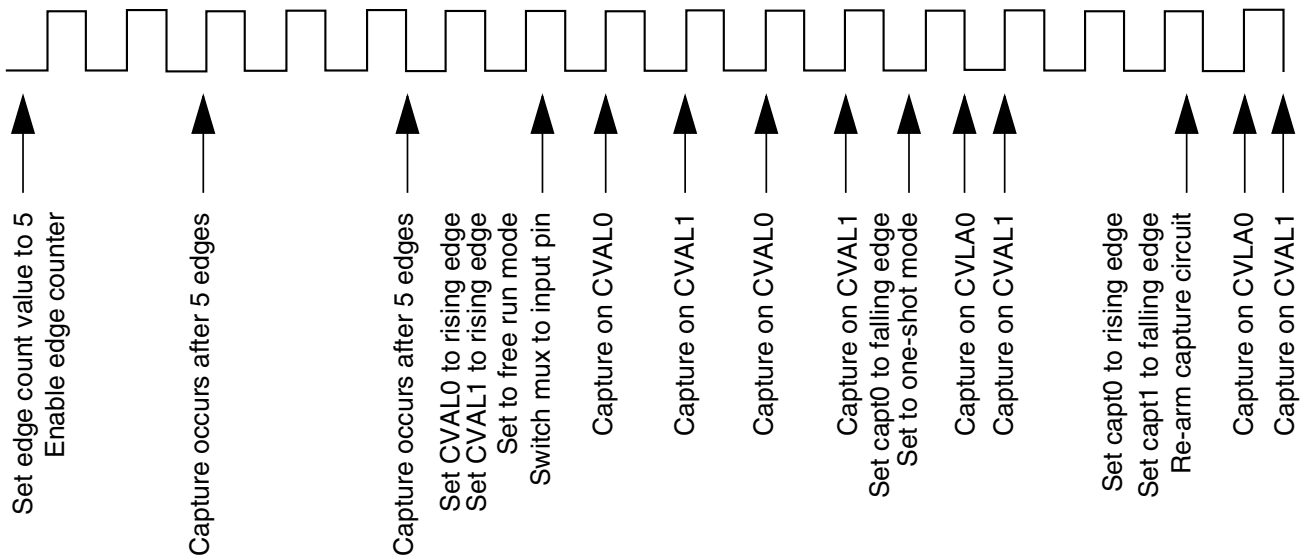


Figure 73-10. Capture capabilities of the E-Capture circuit

When a submodule is used for PWM generation, its timer counts up to the modulus value used to specify the PWM frequency and then is re-initialized. Therefore, using this timer for input captures on one of the other pins (for example, PWM_X) has limited utility since it does not count through all of the numbers and the timer reset represents a discontinuity in the 16-bit number range. However, when measuring a signal that is synchronous to the PWM frequency, the timer modulus range is suited for the application. As shown in Figure 73-11, the output of a PWM power stage is connected to the PWM_X pin that is configured for free running input captures. Specifically, the CVAL0 capture circuitry is programmed for rising edges and the CVAL1 capture circuitry is set for falling edges. This results in new load pulse width data acquired every PWM cycle. To calculate the pulse width, subtract the CVAL0 register value from the CVAL1 register value. This measurement is beneficial when performing deadtime distortion correction on a half bridge circuit driving an inductive load. Also, these values can be directly compared to the VALx registers responsible for generating the PWM outputs to obtain a measurement of system propagation delays. For details, refer to the separate discussion of deadtime distortion correction.

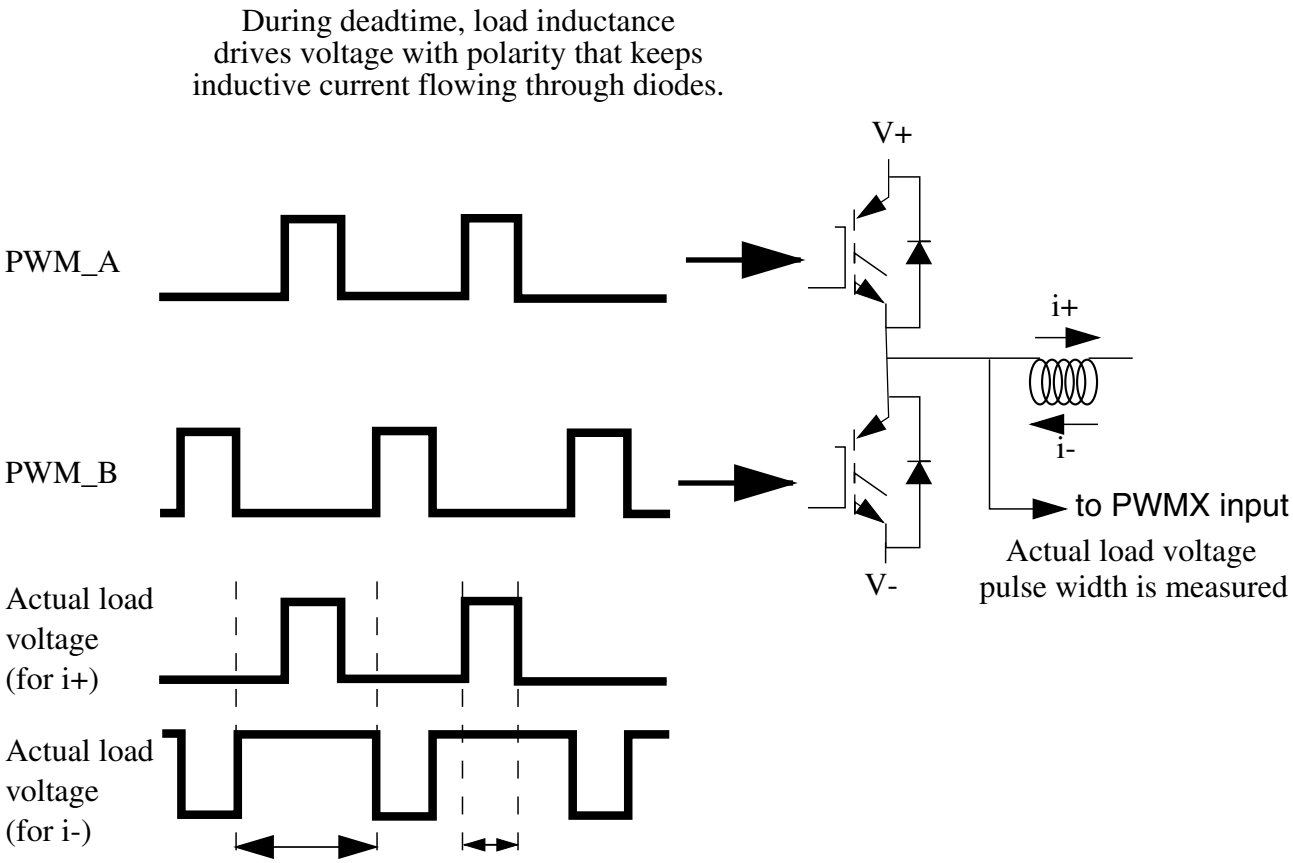


Figure 73-11. Output pulse width measurement possible with the E-capture circuit

73.3.2.7 Synchronous switching of multiple outputs

Before the PWM signals are routed to the output pins, they are processed by a hardware block that permits all submodule outputs to be switched synchronously. This feature is useful in commutated motor applications where the next commutation state can be laid in ahead of time and then immediately switched to the outputs when the appropriate condition or time is reached. All the changes occur immediately after the trigger event occurs eliminating any interrupt latency and also the changes occurs synchronously on all submodule outputs.

The synchronous output switching is accomplished via a signal called FORCE_OUT. This signal originates from the local FORCE bit within the submodule, from submodule0, or from external to the PWM module, and in most cases, is supplied from an external timer channel configured for output compare. In a typical application, software sets up the desired states of the output pins in preparation for the next FORCE_OUT event. This selection lays dormant until the FORCE_OUT signal transitions and then all outputs are

switched simultaneously. The signal switching is performed upstream from the deadtime generator so that any abrupt changes that occur do not violate deadtime on the power stage when in complementary mode.

Figure 73-12 shows an application that can benefit from this feature. On a brushless DC motor in many cases, it is required to spin the motor without need of hall-effect sensor feedback. Instead, the back EMF of the motor phases is monitored and this information is used to schedule the next commutation event. The top waveforms of Figure 73-12 represent these back EMF signals. Timer compare events (represented by the long vertical lines in the diagram) are scheduled based on the zero crossings of the back-EMF waveforms. The PWM module is configured via software ahead of time with the next state of the PWM pins in anticipation of the compare event. When it happens, the output compare of the timer drives the FORCE_OUT signal which immediately changes the state of the PWM pins to the next commutation state with no software latency.

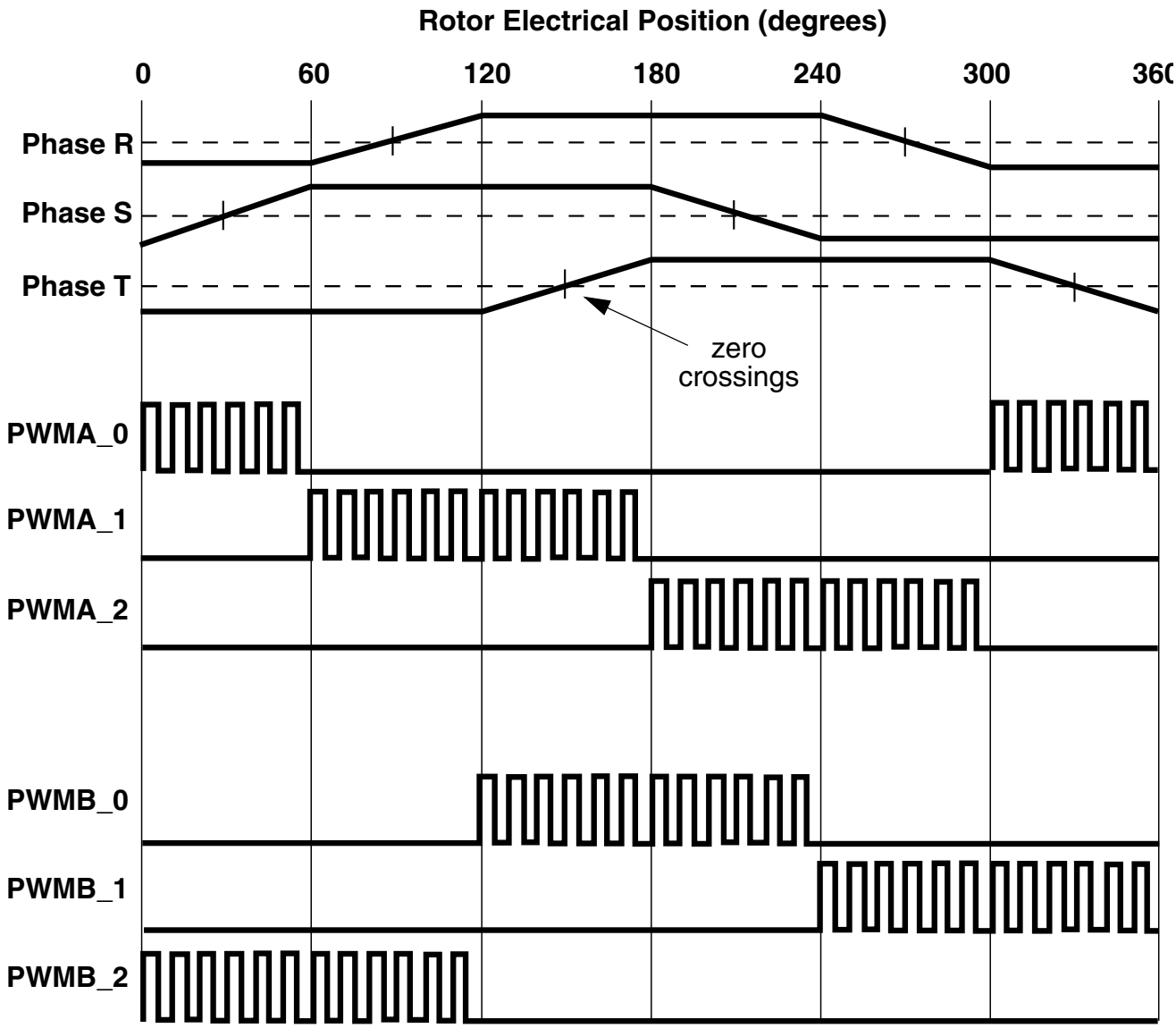


Figure 73-12. Sensorless BLDC commutation using the force out function

73.3.3 Operation

This section describes the implementation of various sections of the PWM in detail.

The following figure is a high-level block diagram of output PWM generation.

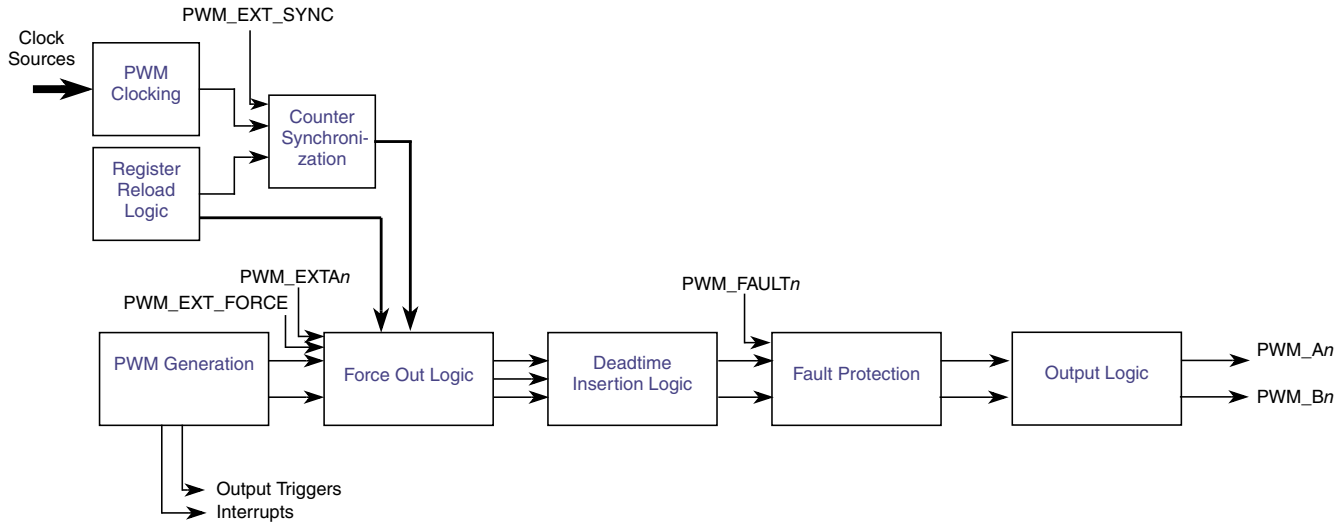


Figure 73-13. High-level output PWM generation block diagram

73.3.3.1 Register reload logic

The register reload logic is used to determine when the outer set of registers for all double buffered register pairs will be transferred to the inner set of registers. The register reload event can be scheduled to occur every "n" PWM cycles using CTRL[LDFQ] and CTRL[FULL], which is defined by VAL1 register. A half cycle reload option is also supported (CTRL[HALF]) where the reload can take place in the middle of a PWM cycle. The half cycle point is defined by the VAL0 register and does not have to be exactly in the middle of the PWM cycle.

As shown in Figure 73-14 the reload signal from submodule0 can be broadcast as the Master Reload signal allowing the reload logic from submodule0 to control the reload of registers in other submodules.

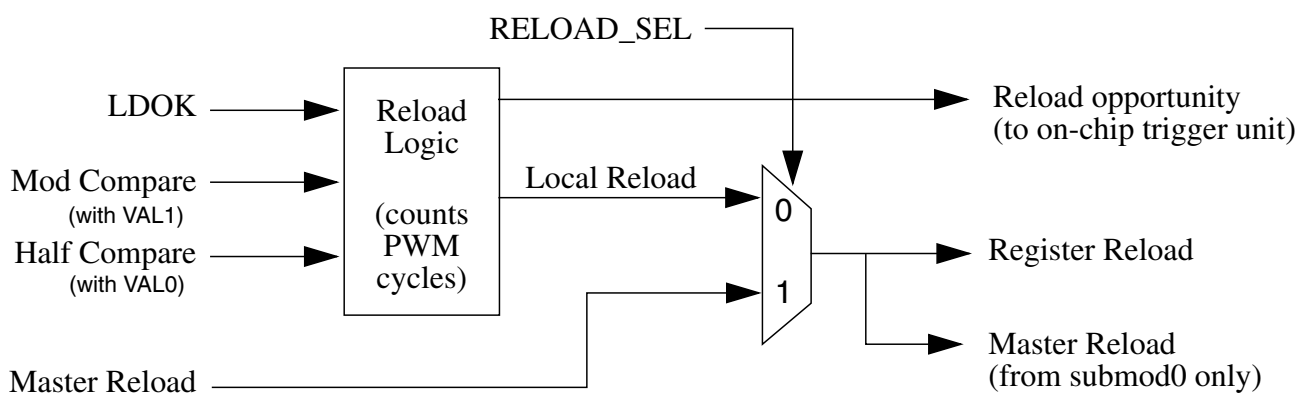


Figure 73-14. Register reload logic of 1 submodule

73.3.3.2 Counter synchronization

As shown in Figure 73-15, the 16-bit counter counts up until its output equals VAL1 which is used to specify the counter modulus value. The resulting compare causes a rising edge to occur on the Local Synchronization signal which is one of four possible sources used to cause the 16-bit counter to be initialized with INIT. If Local Synchronization is selected as the counter initialization signal, VAL1 within the submodule effectively controls the timer period (and then the PWM frequency generated by that submodule), and everything operates or functions at a local level.

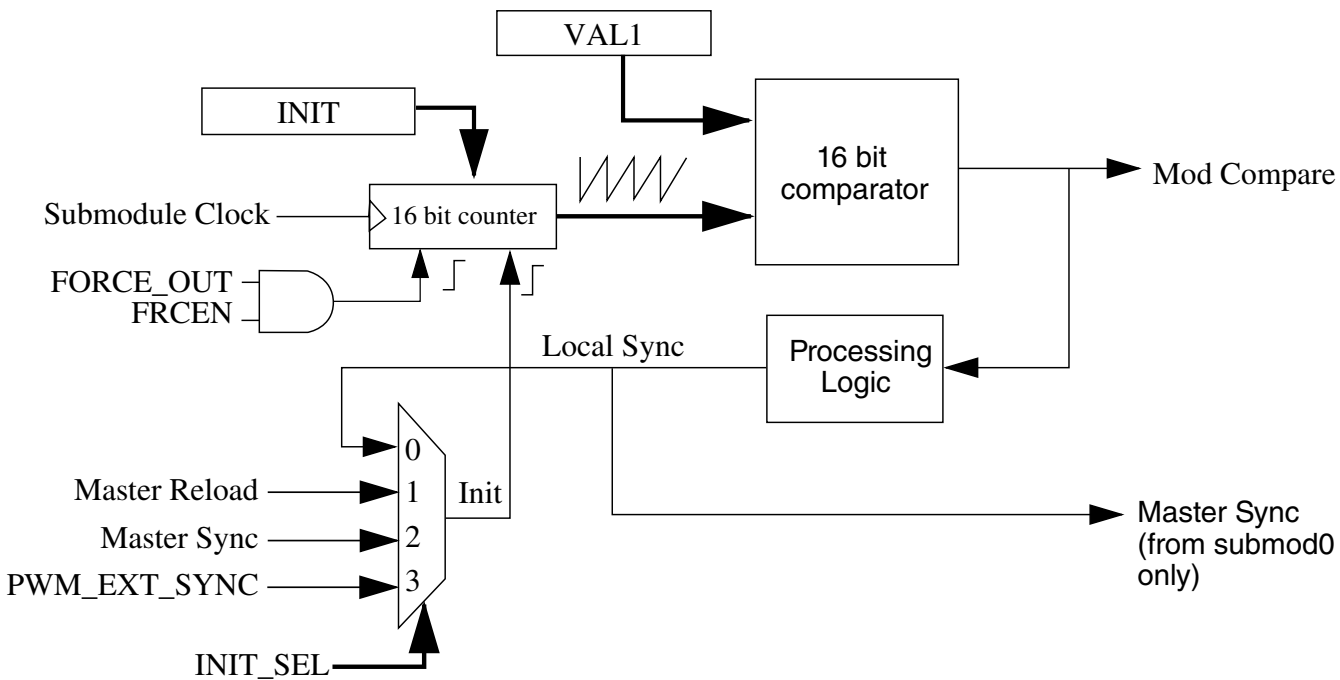


Figure 73-15. Submodule timer synchronization for 1 submodule

The Master Synchronization signal originates as the Local Sync from submodule0. If configured to do so, the timer period of any submodule can be locked to the period of the timer in submodule0.

The PWM_EXT_SYNC signal originates on chip or off chip depending on the system architecture. This signal may be selected as the source for counter initialization so that an external source can control the period of all submodules.

If the Master Reload signal is selected as the source for counter initialization, then the period of the counter will be locked to the register reload frequency of submodule0. Since the reload frequency is commensurate to the sampling frequency of the software control algorithm, the submodule counter period is equal the sampling period. As a result, this

timer can be used to generate output compares or output triggers over the entire sampling period which may consist of several PWM cycles. The Master Reload signal can only originate from submodule0.

The counter can optionally initialize upon the assertion of the FORCE_OUT signal assuming that CTRL2[FRCEN] is set. As shown in Figure 73-15, this constitutes a second initialization input into the counter, which causes the counter to initialize regardless of which signal is selected as the counter initialization signal. A forced initialization causes a register reload if MCTRL[LDOK] is set.

The counter can be initialized by FORCE_OUT signal only when MCTRL[RUN] = 0 or CTLR2[CLK_SEL] = 2 which chooses auxiliary clock from SM0.

The FORCE_OUT signal is provided mainly for commutated applications. When PWM signals are commutated on an inverter controlling a brushless DC motor, it is necessary to restart the PWM cycle at the beginning of the commutation interval. This action effectively resynchronizes the PWM waveform to the commutation timing. Otherwise, the average voltage applied to a motor winding integrated over the entire commutation interval will be a function of the timing between the asynchronous commutation event for the PWM cycle. The effect is more critical at higher motor speeds where each commutation interval may consist of only a few PWM cycles. If the counter is not initialized at the start of each commutation interval, the result is an oscillation caused by the beating between the PWM frequency and the commutation frequency.

73.3.3.3 PWM generation

Figure 73-16 illustrates how PWM generation is accomplished in each submodule. In each case, two comparators and associated VALx registers are utilized for each PWM output signal. One comparator and VALx register are used to control the turn-on edge, while a second comparator and VALx register control the turn-off edge.

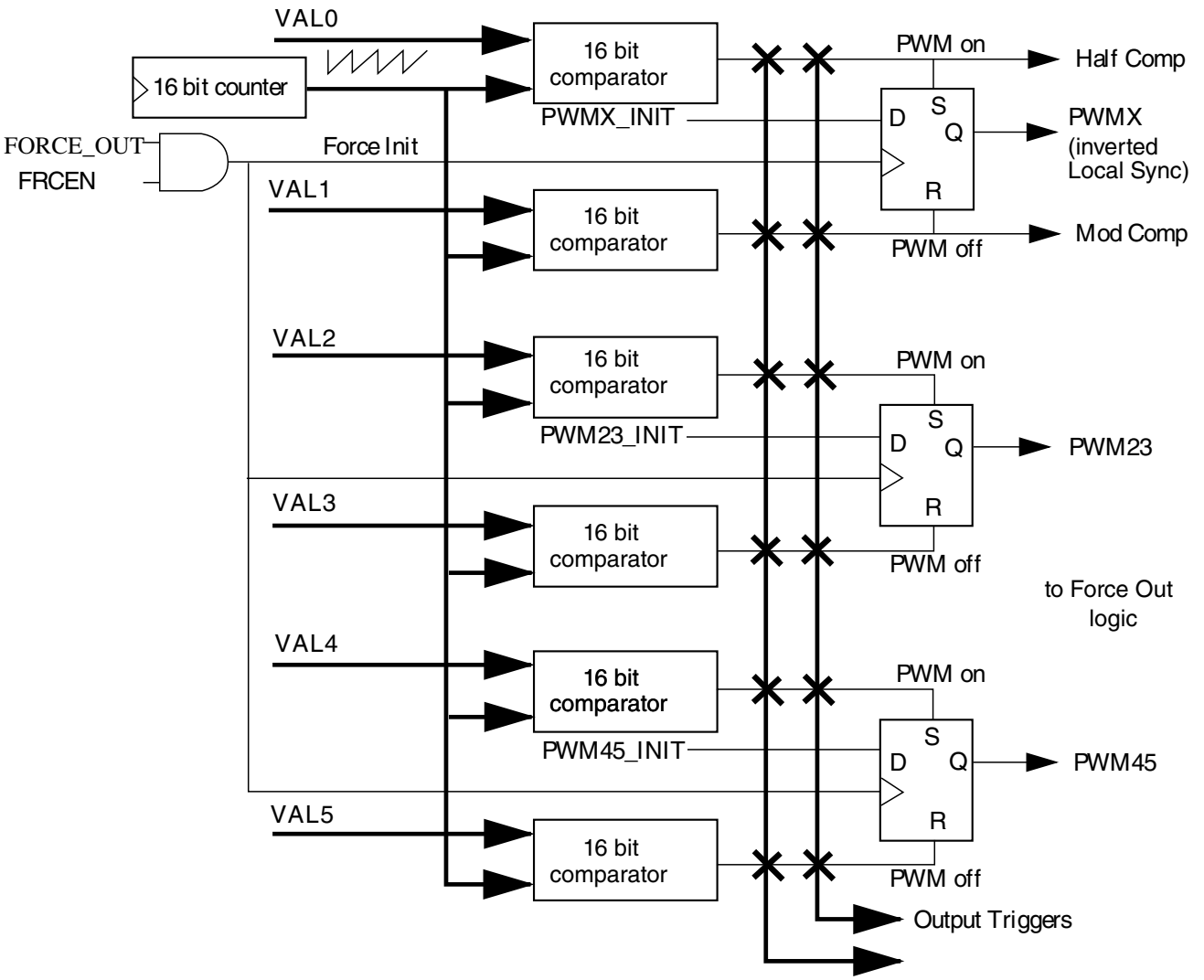


Figure 73-16. PWM Generation Hardware of 1 submodule

The generation of the Local Synchronization signal is performed exactly the same way as the other PWM signals in the submodule. While comparator 0 causes a falling edge of the Local Synchronization signal, comparator 1 generates a rising edge. Comparator 1 is also hardwired to the reload logic to generate the full cycle reload indicator.

If VAL1 is controlling the modulus of the counter and VAL0 is half of the VAL1 register minus the INIT value, then the half cycle reload pulse occurs exactly half way through the timer count period and the Local Synchronization will have a 50% duty cycle. On the other hand, if the VAL1 and VAL0 registers are not required for register reloading or counter initialization, they can be used to modulate the duty cycle of the Local Synchronization signal, effectively turning it into an auxiliary PWM signal (PWM_X) assuming that the PWM_X pin is not being used for another function such as input capture or deadtime distortion correction. Including the Local Synchronization signal, each submodule is capable of generating three PWM signals where software has complete control over each edge of each of the signals.

If the comparators and edge value registers are not required for PWM generation, they can be used for other functions such as output compares, generating output triggers, or generating interrupts at timed intervals.

The 16-bit comparators shown in Figure 73-16 are "equal to" comparators. In addition, if both the set and reset of the flip-flop are asserted, then the flop output goes to 0.

73.3.3.4 Output compare capabilities

By using the VALx registers in conjunction with the submodule timer and 16-bit comparators, buffered output compare functionality can be achieved with no additional hardware required. Specifically, the following output compare functions are possible:

- An output compare sets the output high
- An output compare sets the output low
- An output compare generates an interrupt
- An output compare generates an output trigger

In PWM generation, an output compare is initiated by programming a VALx register for a timer compare, which in turn causes the output of the D flip-flop to either set or reset. For example, if an output compare is desired on the PWM_A signal that sets it high, VAL2 would be programmed with the counter value where the output compare should take place. However, to prevent the D flip-flop from being reset again after the compare has occurred, the VAL3 register must be programmed to a value out of the modulus range of the counter. Therefore, a comparison that would result in resetting the D flip-flop output would never occur. Conversely, if an output compare is desired on the PWM_A signal that sets it low, the VAL3 register is programmed with the appropriate count value and the VAL2 register is programmed with a value out of the counter modulus range. Regardless of whether a high compare or low compare is programmed, an interrupt, or output trigger can be generated when the compare event occurs.

73.3.3.5 Force out logic

For each submodule, the software can select between eight signal sources for the FORCE_OUT signal depending on the chip architecture:

1. Local CTRL2[FORCE]
2. Master Force signal from submodule0
3. Local Reload signal
4. Master Reload signal from submodule0

5. Local Synchronization signal
6. Master Synchronization signal from submodule0
7. EXT_SYNC signal from on or off chip
8. EXT_FORCE signal from on or off chip

The local signals are used to change the signals on the output pins of the submodule without regard for synchronization with other submodules. However, if it is required that all signals on all submodule outputs change at the same time, the Master, EXT_SYNC, or EXT_FORCE signals must be selected.

Figure 73-17 illustrates the Force logic. The SEL23 and SEL45 fields each choose from one of four signals that can be supplied to the submodule outputs: the PWM signal, the inverted PWM signal, a binary level specified by software via the OUT23 and OUT45 bits, or the PWM_EXTA or PWM_EXTB alternate external control signals. The selection can be determined ahead of time, and when a FORCE_OUT event occurs, these values are presented to the signal selection mux that immediately switches the requested signal to the output of the mux for further processing downstream.

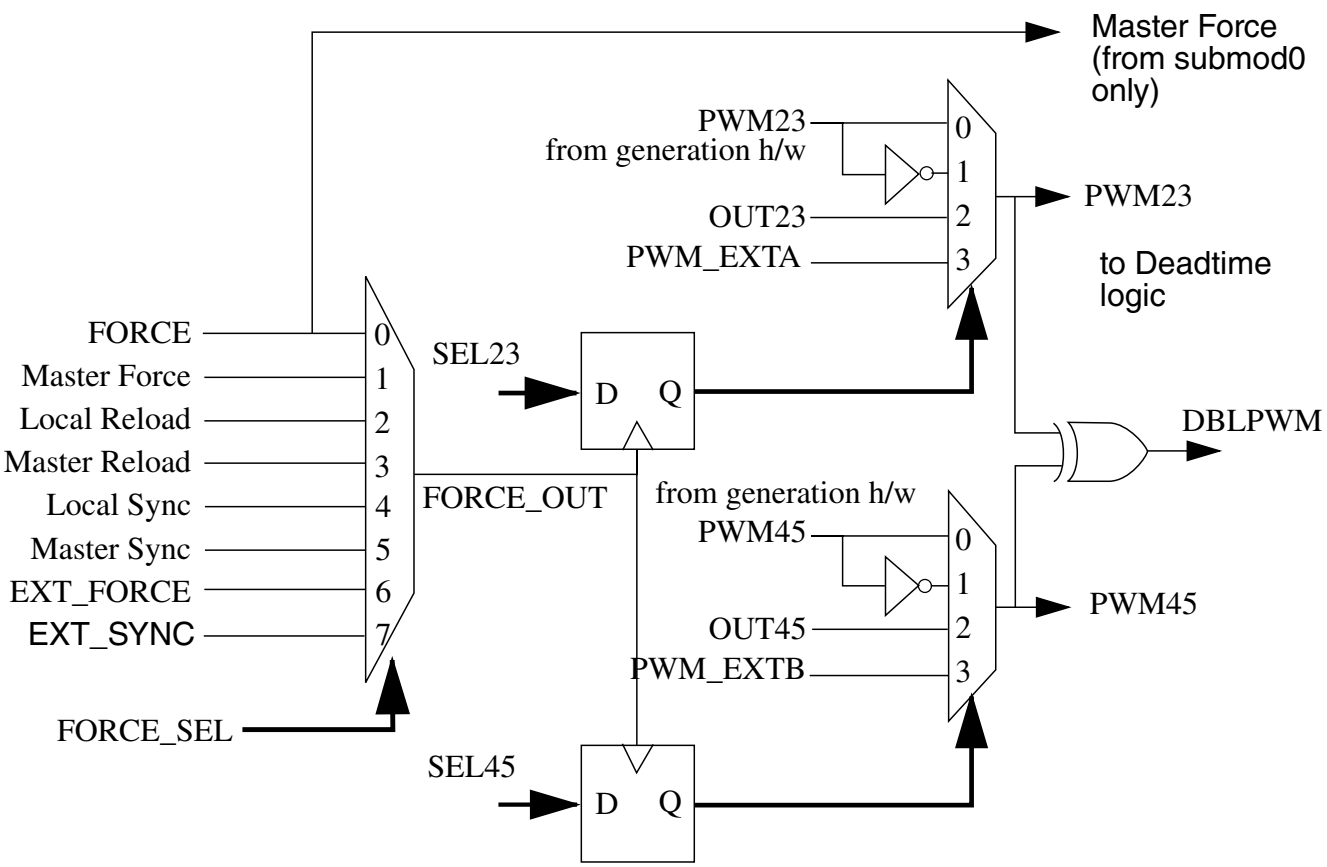


Figure 73-17. Force out logic for 1 submodule

The local CTRL2[FORCE] signal of submodule0 can be broadcast as the Master Force signal to other submodules. This feature allows the CTRL2[FORCE] of submodule0 to synchronously update all of the submodule outputs at the same time. The EXT_FORCE signal originates from outside the PWM module from a source such as a timer or digital comparators in the Analog-to-Digital Converter.

73.3.3.6 Independent or complementary channel operation

Writing a logic one to CTRL2[INDEP] configures the pair of PWM outputs as two independent PWM channels. Each PWM output is controlled by its own VALx pair operating independently of the other output.

Writing a logic zero to CTRL2[INDEP] configures the PWM output as a pair of complementary channels. The PWM pins are paired as shown in Figure 73-18 in complementary channel operation. The signal that is connected to the output pin (PWM23 or PWM45) is determined by MCTRL[IPOL].

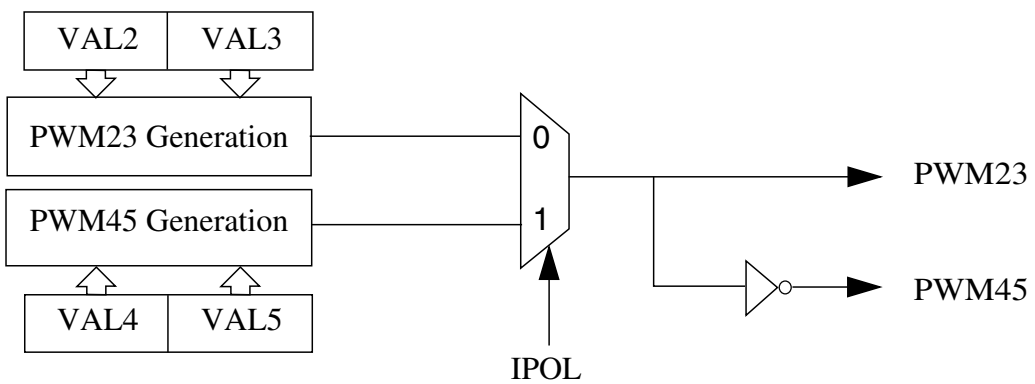


Figure 73-18. Complementary channel pair for 1 submodule

The complementary channel operation is for driving top and bottom transistors in a motor drive circuit, as shown in Figure 73-19.

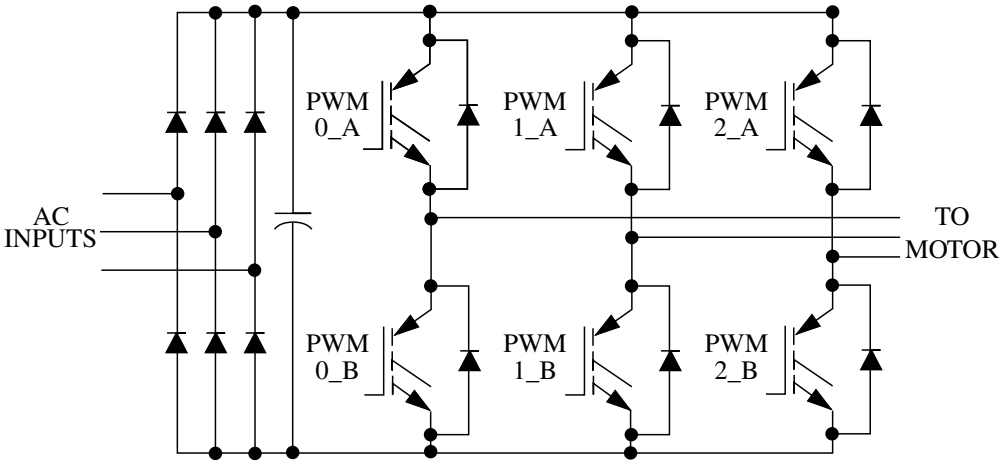


Figure 73-19. Typical 3 phase AC motor drive

The complementary operation allows the use of the deadtime insertion feature.

73.3.3.7 Deadtime insertion logic

The following figure shows the deadtime insertion logic of each submodule which is used to create non-overlapping complementary signals when not in independent mode.

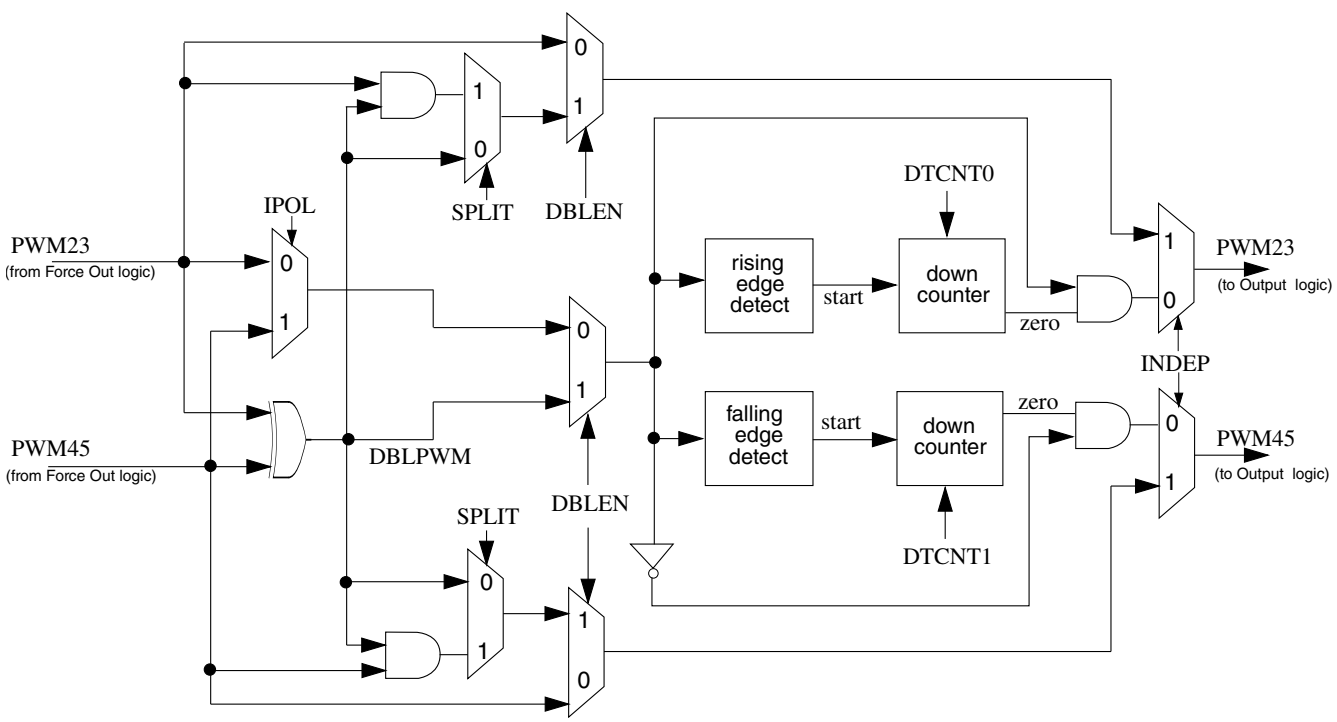


Figure 73-20. Deadtime Insertion Logic for 1 submodule

While in the complementary mode, a PWM pair can be used to drive top/bottom transistors, as shown in the figure. When the top PWM channel is active, the bottom PWM channel is inactive, and vice versa.

Note

To avoid short-circuiting the DC bus and endangering the transistor, there must be no overlap of conducting intervals between the top and bottom transistors. But the transistor's characteristics make its switching-off time longer than switching-on time. To avoid the conducting overlap of top and bottom transistors, deadtime needs to be inserted in the switching period, as shown in Figure 73-21.

The deadtime generators automatically insert software-selectable activation delays into the pair of PWM outputs. The deadtime registers (DTCNT0 and DTCNT1) specify the number of IPBus clock cycles to use for deadtime delay. Every time the deadtime generator inputs change state, deadtime is inserted. Deadtime forces both PWM outputs in the pair to the inactive state.

When deadtime is inserted in complementary PWM signals connected to an inverter driving an inductive load, the PWM waveform on the inverter output will have a different duty cycle than what appears on the output pins of the PWM module. This results in a distortion in the voltage applied to the load. A method of correcting this, adding to or subtracting from the PWM value used, is discussed next.

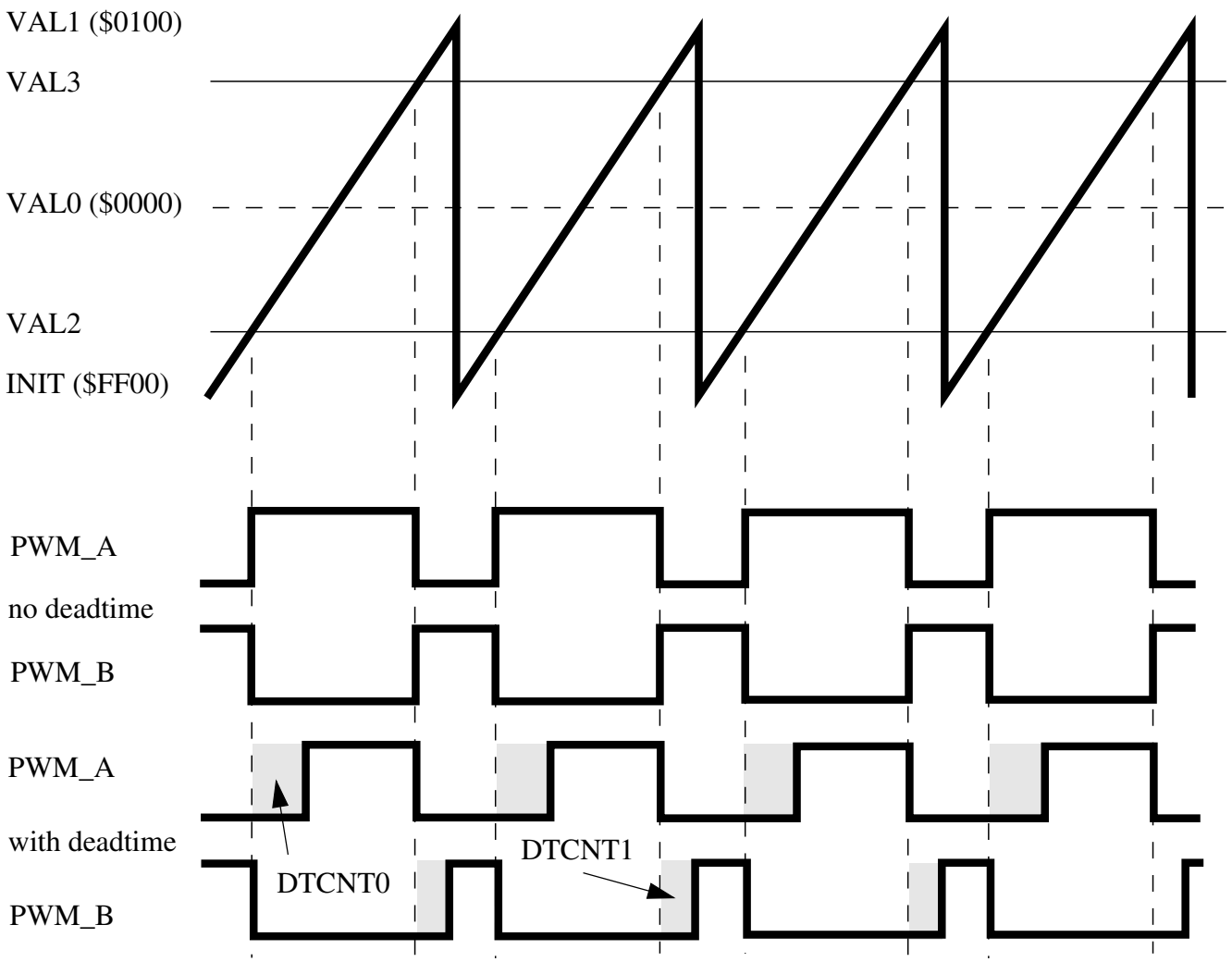


Figure 73-21. Deadtime insertion

73.3.3.7.1 Top/Bottom correction

In complementary mode, either the top or the bottom transistor controls the output voltage. However, deadtime has to be inserted to avoid overlap of conducting interval between the top and bottom transistor. Both transistors in complementary mode are off during deadtime, allowing the output voltage to be determined by the current status of load and introducing distortion in the output voltage, as shown in Figure 73-22. On AC induction motors running open-loop, the distortion typically manifests itself as poor low-speed performance, such as torque ripple and rough operation.

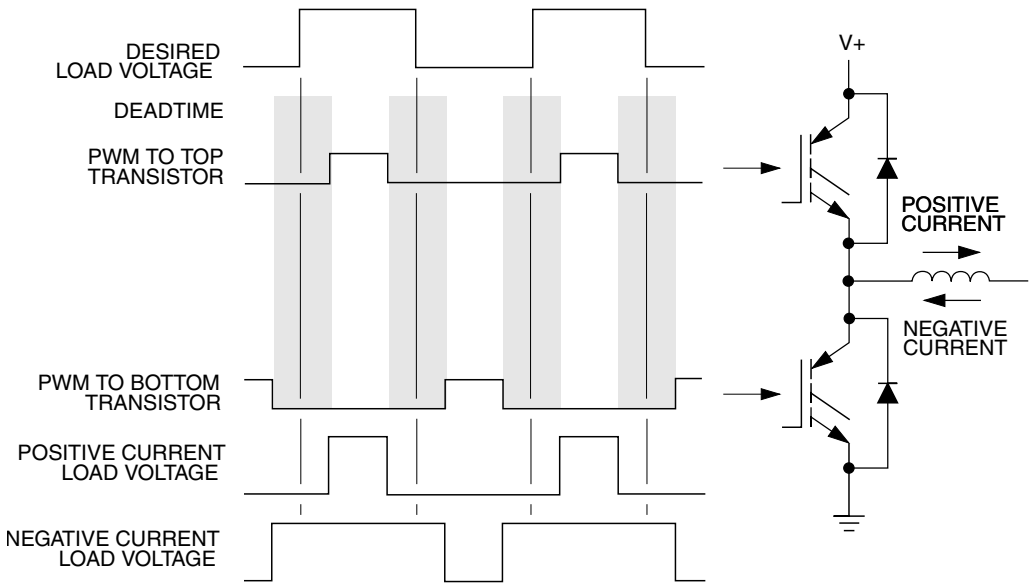


Figure 73-22. Deadtime distortion

During deadtime, load inductance distorts output voltage by keeping current flowing through the diodes. This deadtime current flow creates a load voltage that varies with the current direction. With a positive current flow, the load voltage during deadtime is equal to the bottom supply, putting the top transistor in control. With a negative current flow, the load voltage during deadtime is equal to the top supply putting the bottom transistor in control.

Remembering that the original PWM pulse widths were shortened by deadtime insertion, the averaged sinusoidal output is less than the desired value. However, when deadtime is inserted, it distorts in the motor current waveform inverter outputs. This distortion is aggravated by different turn-on and turn-off delays of each of the transistors. By giving the PWM module information on which transistor is controlling at a given time this distortion can be corrected.

For a typical circuit in complementary channel operation, only one of the transistors is effective in controlling the output voltage at any given time. This depends on the direction of the motor inverter current for that pair, as shown in Figure 73-22. To correct distortion one of two different factors must be added to the desired PWM value, depending on whether the top or bottom transistor is controlling the output voltage. Therefore, the software is responsible for calculating both compensated PWM values prior to placing them in the VALx registers. Either the VAL2/VAL3 or the VAL4/VAL5 register pair controls the pulse width at any given time. For a given PWM pair, whether the VAL2/VAL3 or VAL4/VAL5 pair is active depends on either:

- The state of the current status pin, PWMX, for that driver
- The state of the odd/even correction bit, MCTRL[IPOL], for that driver

To correct deadtime distortion, the software can decrease or increase the value in the appropriate VALx register.

- In edge-aligned operation, decreasing or increasing the PWM value by a correction value equal to the deadtime typically compensates for deadtime distortion.
- In center-aligned operation, decreasing or increasing the PWM value by a correction value equal to one-half the deadtime typically compensates for deadtime distortion.

73.3.3.7.2 Manual correction

To detect the current status, the voltage on each PWMX pin is sampled twice in a PWM period, at the end of each deadtime. The value is stored in CTRL[DT]. CTRL[DT] is a timing marker indicating when to toggle between PWM value registers. The software can then set MCTRL[IPOL] to switch between VAL2/VAL3 and VAL4/VAL5 register pairs according to CTRL[DT] values.

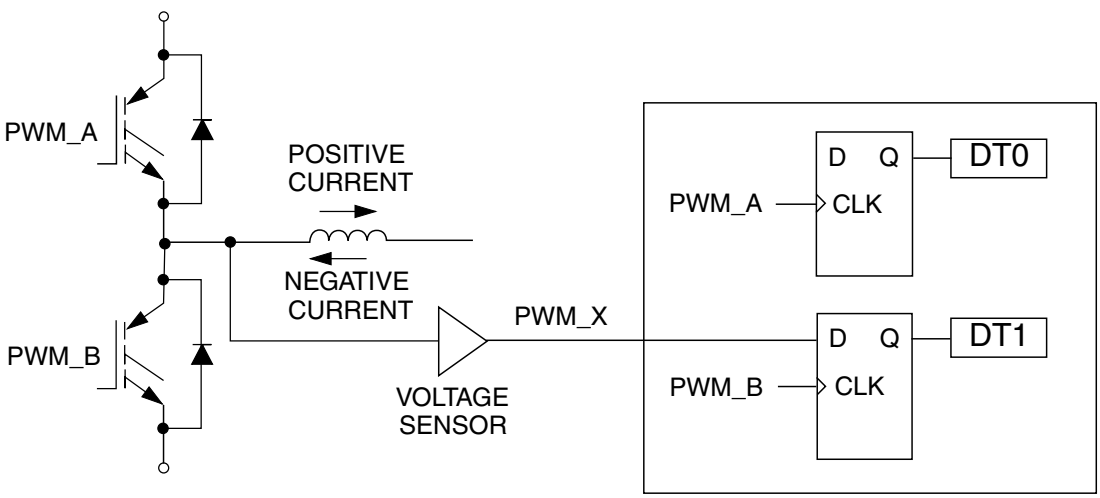


Figure 73-23. Current-status sense scheme for deadtime correction

Both D flip-flops latch low, CTRL[DT] = 00, during deadtime periods if the current is large and flowing out of the complementary circuit. See the preceding figure. Both D flip-flops latch the high, CTRL[DT] = 11, during deadtime periods if the current is also large and flowing into the complementary circuit.

However, under low-current, the output voltage of the complementary circuit during deadtime is between the high and low levels. The current cannot free-wheel through the opposition anti-body diode, regardless of polarity, giving additional distortion when the current crosses zero. Sampled results are CTRL[DT] = b10. Thus, the best time to change one PWM value register to another is just before the current zero crossing.

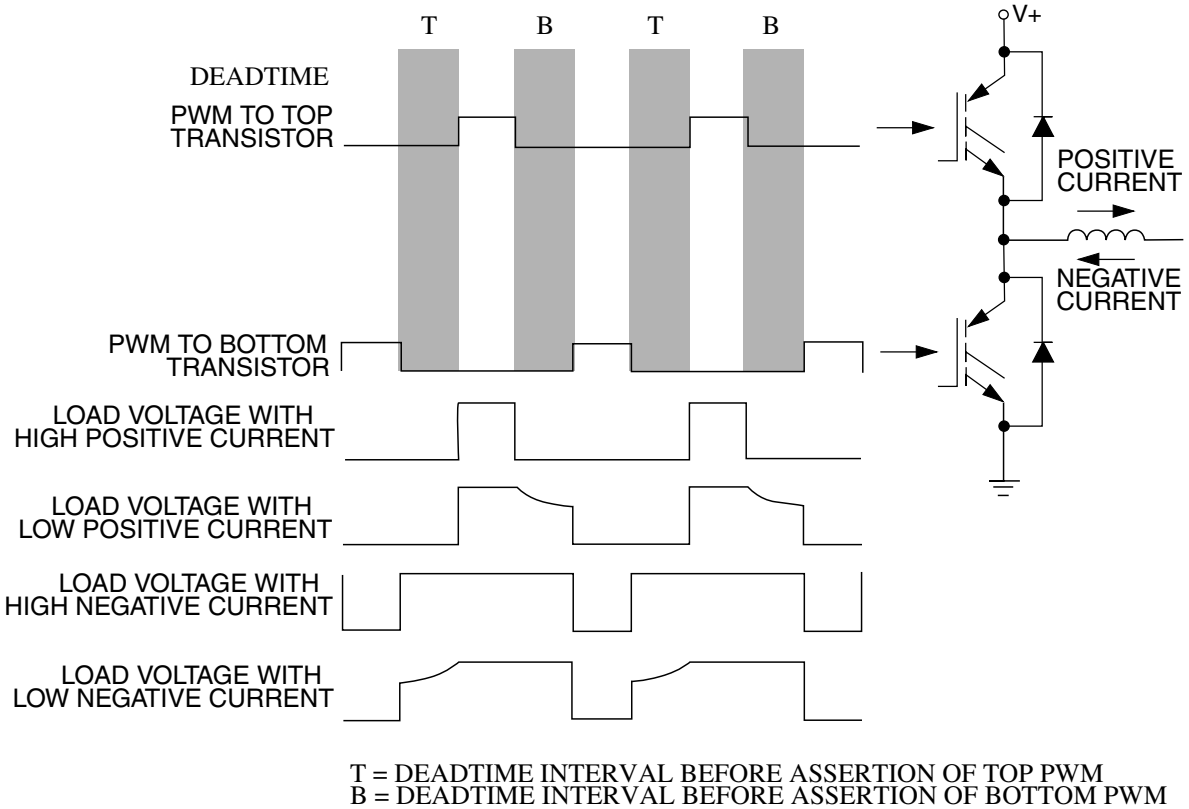


Figure 73-24. Output voltage waveforms

73.3.3.8 Fractional delay logic

For applications where more resolution than a single IPBus clock period is needed, the fractional delay logic can be used to achieve fine resolution on the rising and falling edges of the PWM_A and PWM_B outputs. Enable the use of the fractional delay logic by setting FRCTRL[FRACx_EN]. The FRACVALx registers act as a fractional clock cycle addition to the turn-on and turn-off count specified by the VAL2, VAL3, VAL4, or VAL5 registers. The FRACVAL1 register acts as a fractional increase in the PWM period as defined by VAL1. If FRACVAL1 is programmed to a non-zero value, then the largest value for the VAL1 register is 0xFFFE for unsigned usage or 0x7FFE for signed usage. This limit is required to avoid counter rollovers when accumulating the fractional additional period.

Both the fractional enables (1, 23, 45) and the fractional values (1-5) are double buffered and reloaded at the same time as the value registers.

Each PWM cycle, the value compare point is increased by 1 when there is overflow on the double buffered value of the fractional register plus the 5-bit accumulated fractional value. The accumulated fractional value starts at zero, so it is impossible to overflow the first PWM cycle.

At the end of each PWM cycle, if the corresponding fractional enable is set then the accumulated fractional value increments by the fractional value (double buffered value). The accumulated fractional value is 5-bits, so in the case of overflow only the remainder is kept. If the corresponding fractional enable is clear, then the accumulated fractional value is reset. This is the only way to reset the accumulated fractional value.

The accumulated fractional values are not accessible to software.

As an example, assume the following conditions:

- INIT = 0x0000
- VAL1 = 0x000F
- VAL2 = 0x0000
- VAL3 = 0x0007
- FRACVAL3 = 0x00

This would cause the PWM output to have a 50% duty cycle. It would be high from a count value of 0x0 to the count value of 0x7, at which point it would go low until the counter reaches the maximum value of 0xF. If FRACVAL3 was changed to a value of 0x17, then the time the PWM output is active would be 8 cycles + (23/32) cycle = 8.719 cycles, and for a high duty cycle of (8.719 cycles / 16 cycles) x 100% = 54.49%.

Another case involves fine tuning of the PWM period using the FRACVAL1 register. If you want a period of 100.25 clock cycles, the program VAL1 with 0x0064 and FRACVAL1 with 0x4000. The fractional value will accumulate so that every 4 PWM cycles will be 1 clock cycle longer (101 instead of 100). The rising and falling edges of the PWM outputs will also use the accumulated fraction to delay their edges and maintain a consistent 100.25 cycles spacing between corresponding edges from one cycle to the next.

73.3.3.8.1 Fractional delay logic without NanoEdge placement block

For submodules that are not supported by the NanoEdge placer, the PWM can use dithering to simulate fine edge control. Enable this feature by setting the FRCTRL[FRAC1_EN], FRCTRL[FRAC23_EN], and FRCTRL[FRAC45_EN] bits. The PWM period or the PWM edges will dither from the nearest whole number values to achieve an average value that is equivalent to the programmed fractional value. The added cycles are based on the accumulation of the fractional component. For example, if you want the PWM period to be 50.25 clock cycles, then program VAL1 with 0x0032 and FRACVAL1 with 0x4000. The PWM period will be 50 cycles long most of the time, but will occasionally be 51 cycles long to achieve a long-term average of 50.25 cycles.

In submodules that are not supported by a NanoEdge placer, the clock frequency is not required to be any specific value to achieve proper operation.

73.3.3.9 Output logic

The following figure shows the output logic of each submodule including how each PWM output has individual fault disabling, polarity control, and output enable. This allows for maximum flexibility when interfacing with the external circuitry.

The PWM23 and PWM45 signals which are output from the deadtime logic (as shown in Figure 73-25) are positive true signals. In other words, a high level on these signals should result in the corresponding transistor in the PWM inverter being turned ON. The voltage level required at the PWM output pin to turn the transistor ON or OFF is a function of the logic between the pin and the transistor. Therefore, it is imperative that the user program OCTRL[POLA] and OCTRL[POLB] before enabling the output pins. A fault condition can result in the PWM output being put in a high-impedance state, forced to a logic 1 state, or forced to a logic 0 state, depending on the values programmed into the OCTRL[PWMxFS] fields.

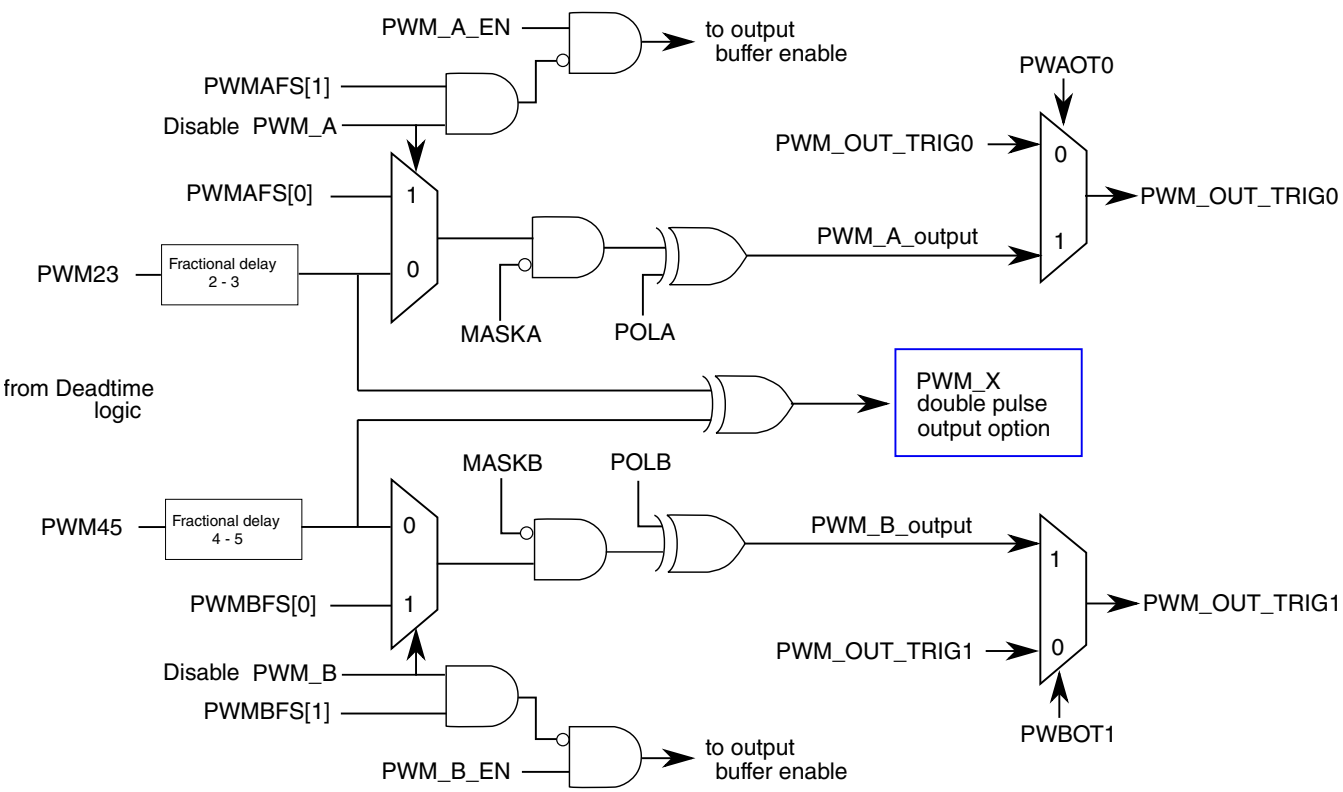


Figure 73-25. Output logic for 1 submodule

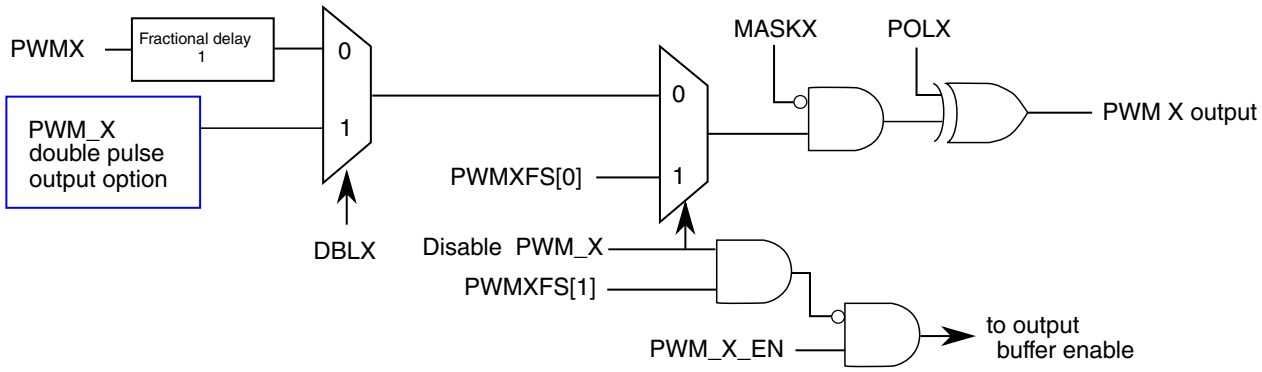


Figure 73-26. Output logic continued

73.3.3.10 E-Capture

The Enhanced Capture (E-Capture) logic is designed to measure both edges of an input signal. As a result, when a submodule pin is configured for input capture, the CVALx registers associated with that pin are used to record the edge values.

The following figure shows block diagram of the E-Capture circuit. Upon entering the pin input, the signal is split into two paths. One goes straight to a mux input where software can select to pass the signal directly to the capture logic for processing. The other path connects the signal to an 8-bit counter which counts both the rising and falling edges of the signal. The output of this counter is compared to an 8-bit value that is specified by the user (EDGCMPx) and when the two values are equal, the comparator generates a pulse that resets the counter. This pulse is also supplied to the mux input where software can select it to be processed by the capture logic. This feature permits the E-Capture circuit to count up to 256 edge events before initiating a capture event. This feature is useful for dividing down high frequency signals for capture processing so that capture interrupts do not overwhelm the CPU. Also, this feature can be used to generate an interrupt after "n" events have been counted.

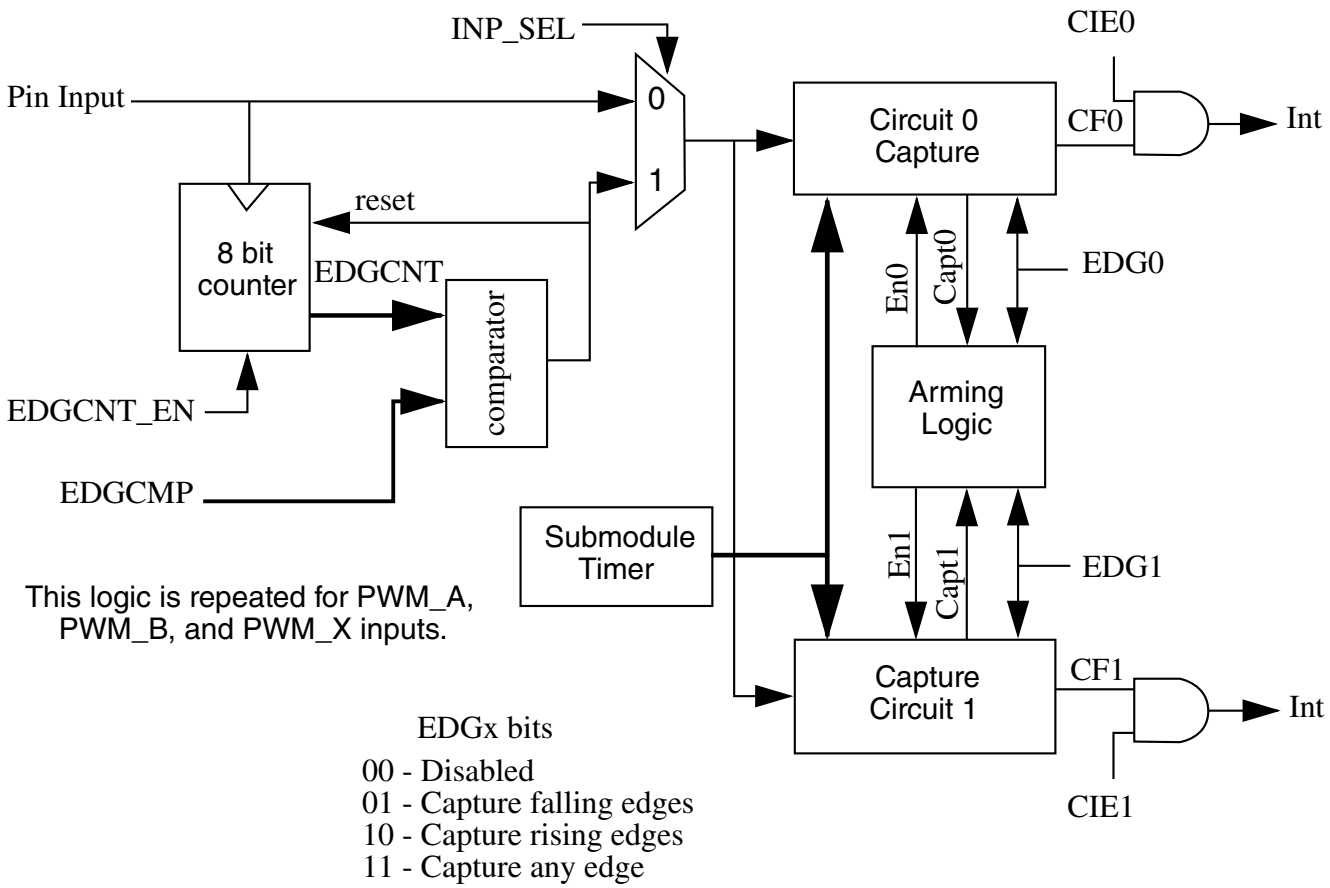


Figure 73-27. Enhanced capture (E-Capture) logic

Based on the mode selection, the mux selects either the pin input or the compare output from the count/compare circuit to be processed by the capture logic. The selected signal is routed to two separate capture circuits which work in tandem to capture sequential edges of the signal. The type of edge to be captured by each circuit is determined by CAPTCTRLx[EDGx1] and CAPTCTRLx[EDGx0], whose functionality is listed in the above figure. Also, controlling the operation of the capture circuits is the arming logic which allows captures to be performed in a free running (continuous) or one shot mode. In free running mode, the capture sequences will be performed indefinitely. If both capture circuits are enabled, they will work together in a ping-pong style where a capture event from one circuit leads to the arming of the other and vice versa. In one shot mode, only one capture sequence will be performed. If both capture circuits are enabled, capture circuit 0 is first armed and when a capture event occurs, capture circuit 1 is armed. Once the second capture occurs, further captures are disabled until another capture sequence is initiated. Both capture circuits are also capable of generating an interrupt to the CPU.

73.3.3.11 Fault protection

Fault protection can control any combination of PWM output pins. Faults are generated by a logic one on any of the FAULTx pins. This polarity can be changed via FCTRL[FLVL]. Each FAULTx pin can be mapped arbitrarily to any of the PWM outputs. When fault protection hardware disables PWM outputs, the PWM generator continues to run, only the output pins are forced to logic 0, logic 1, or high impedance depending on the values of OCTRL[PWMxFS].

The fault decoder disables PWM pins selected by the fault logic and the disable mapping (DISMAPn) registers. The following figure shows an example of the fault disable logic. Each bank of bits in DISMAPn control the mapping for a single PWM pin. See the following table.

The fault protection is enabled even when the PWM module is not enabled. Therefore, a fault is latched in and must be cleared to prevent an interrupt when the PWM is enabled.

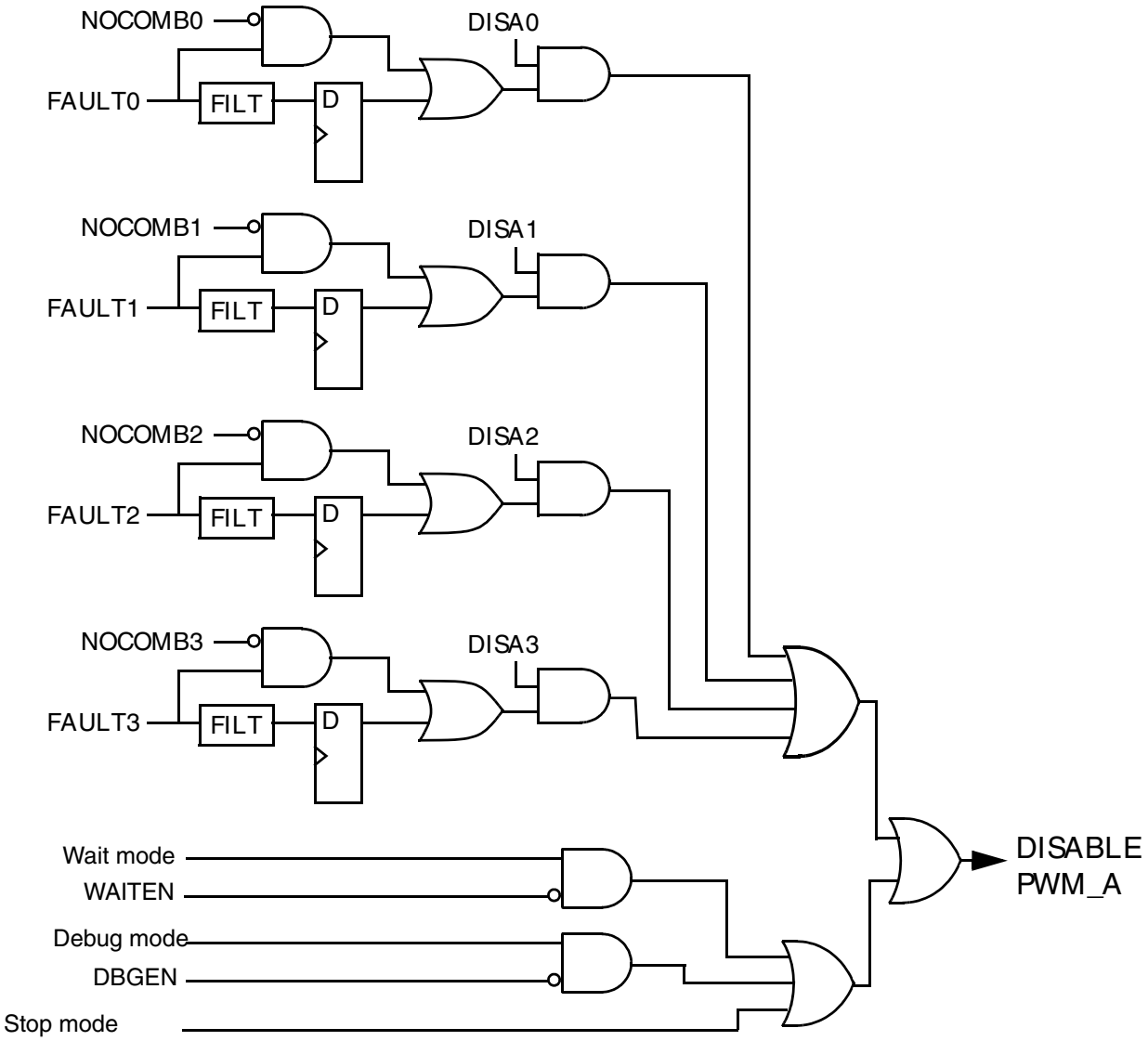


Figure 73-28. Fault decoder for PWM_A

Table 73-2. Fault Mapping

PWM Pin for 1 submodule	Controlling Register Bits
PWM_A	DISMAP0[DIS0A]
PWM_B	DISMAP0[DIS0B]
PWM_X	DISMAP0[DIS0X]

73.3.3.11.1 Fault pin filter

Each fault pin has a programmable filter that can be bypassed. The sampling period of the filter can be adjusted with FFILT[FILT_PER]. The number of consecutive samples that must agree before an input transition is recognized can be adjusted using FFILT[FILT_CNT]. Setting FFILT[FILT_PER] to all 0 disables the input filter for a given FAULTx pin.

Upon detecting a logic 0 on the filtered FAULTx pin (or a logic 1 if FCTRL[FLVLx] is set), the corresponding FSTS[FFPINx] and FSTS[FFLAGx] bits are set. FSTS[FFPINx] remains set as long as the filtered FAULTx pin is zero. Clear FSTS[FFLAGx] by writing a logic 1 to FSTS[FFLAGx].

If the FIEx, FAULTx pin interrupt enable bit is set, FSTS[FFLAGx] generates a CPU interrupt request. The interrupt request latch remains set until:

- Software clears FSTS[FFLAGx] by writing a logic one to the bit
- Software clears the FIEx bit by writing a logic zero to it
- A reset occurs

Even with the filter enabled, there is a combinational path from the FAULTx inputs to the PWM pins, which in turn bypasses the filter when FCTRL20[NOCOMBx] = 0. This logic is also capable of holding a fault condition in the event of loss of clock to the PWM module.

73.3.3.11.2 Automatic fault clearing

Setting an automatic clearing mode bit, FCTRL[FAUTOx] configures faults from the FAULTx pin for automatic clearing.

When FCTRL[FAUTOx] is set, disabled PWM pins are enabled when the FAULTx pin returns to logic one and a new PWM full or half cycle begins. See the following figure. If FSTS[FFULLx] is set and the fault condition on FAULTx disappears, then the disabled PWM pins are enabled at the start of next full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle. Clearing FSTS[FFLAGx] does not affect disabled PWM pins when FCTRL[FAUTOx] is set.

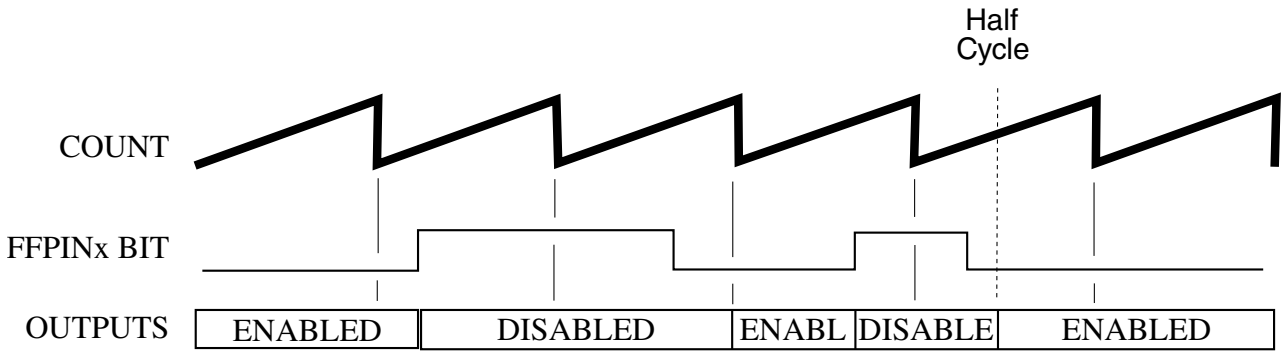


Figure 73-29. Automatic fault clearing

73.3.3.11.3 Manual fault clearing

Clearing the automatic clearing mode bit, FCTRL[FAUTOx], configures faults from the FAULTx pin for manual clearing:

- If the fault safety mode bits FCTRL[FSAFEx] are clear, then PWM pins disabled by the FAULTx pins are enabled when:
  - Software clears the corresponding FSTS[FFLAGx] flag
  - The pins are enabled when the next PWM full or half cycle begins regardless of the logic level detected by the filter at the FAULTx pin, as shown in Figure 73-30. If FSTS[FFULLx] is set, then the disabled PWM pins are enabled at the start of a full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle.
- If the fault safety mode bits FCTRL[FSAFEx] are set, then PWM pins disabled by the FAULTx pins are enabled when:
  - Software clears the corresponding FSTS[FFLAGx] flag
  - The filter detects a logic zero on the FAULTx pin at the start of the next PWM full or half cycle boundary, as shown in Figure 73-31. If FSTS[FFULLx] is set, then the disabled PWM pins are enabled at the start of a full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle.

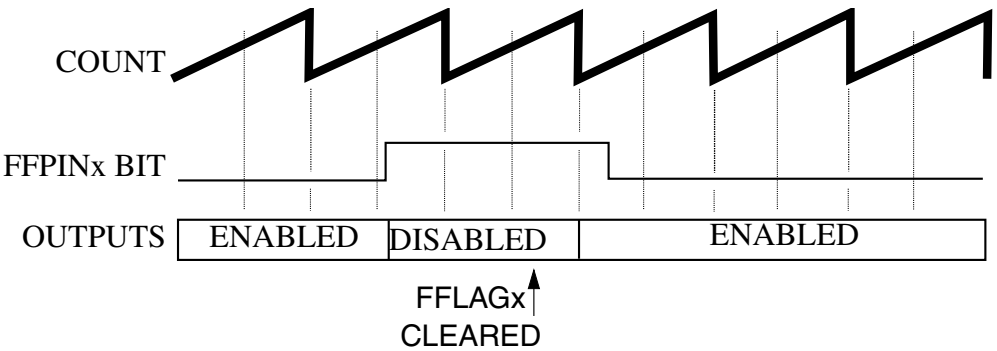


Figure 73-30. Manual fault clearing (FCTRL[FSAFEx] = 0, FSTS[FFULLx] = 1)

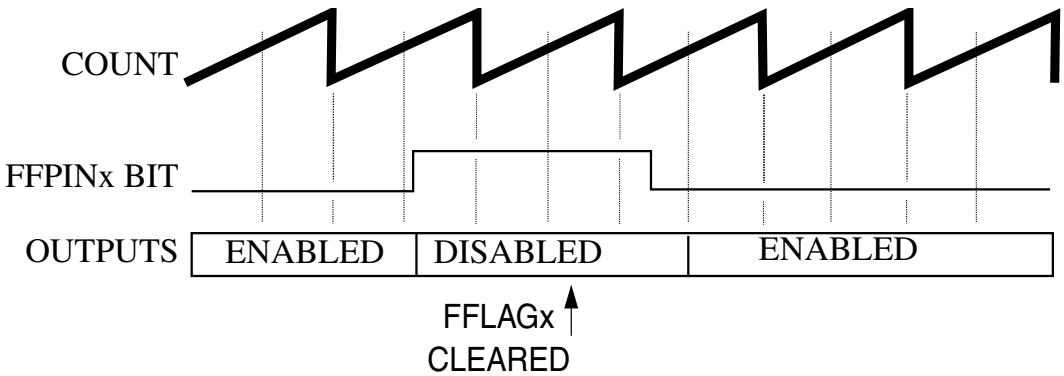


Figure 73-31. Manual fault clearing (FCTRL[FSAFEx] = 1, FSTS[FHALFx] = 1)

Note

Fault protection also applies during software output control when the SEL23 and SEL45 fields are set to select OUT23 and OUT45 bits or PWM_EXTA and PWM_EXTB. Fault clearing still occurs at half or full PWM cycle boundaries while the PWM generator is engaged, MCTRL[RUN] equals one. But the OUTx bits can control the PWM pins while the PWM generator is off, MCTRL[RUN] equals zero. Thus, fault clearing occurs at IPBus cycles while the PWM generator is off and at the start of PWM cycles when the generator is engaged.

73.3.3.11.4 Fault testing

FTST[FTEST] is used to simulate a fault condition on each of the fault inputs within that fault channel.

73.3.3.12 PWM generator loading

73.3.3.12.1 Load enable

MCTRL[LDOK] enables loading of the following PWM generator parameters.

- The prescaler divisor: from CTRL[PRSC]
- The PWM period and pulse width: from the INIT, FRACVALx, and VALx registers

MCTRL[LDOK] allows the software to finish calculating all of these PWM parameters so they can be synchronously updated. The CTRL[PRSC], INIT, and VALx registers are loaded by software into a set of outer buffers. When MCTRL[LDOK] is set, these values are transferred to an inner set of registers at the beginning of the next PWM reload cycle to be used by the PWM generator when CTRL[LDMOD] is cleared. These values can be transferred to the inner set of registers immediately upon setting MCTRL[LDOK] if CTRL[LDMOD] is set. Set MCTRL[LDOK] by reading it when it is a logic zero and then writing a logic one to it. After loading, MCTRL[LDOK] is automatically cleared.

73.3.3.12.2 Load frequency

CTRL[LDFQ] selects an integral loading frequency of one to 16 PWM reload opportunities. CTRL[LDFQ] takes effect at every PWM reload opportunity, regardless of the state of MCTRL[LDOK]. CTRL[HALF] and CTRL[FULL] control reload timing. If CTRL[FULL] is set, a reload opportunity occurs at the end of every PWM cycle when the count equals VAL1. If CTRL[HALF] is set, a reload opportunity occurs at the half cycle when the count equals VAL0. If both CTRL[HALF] and CTRL[FULL] are set, a reload opportunity occurs twice per PWM cycle when the count equals VAL1 and when it equals VAL0.

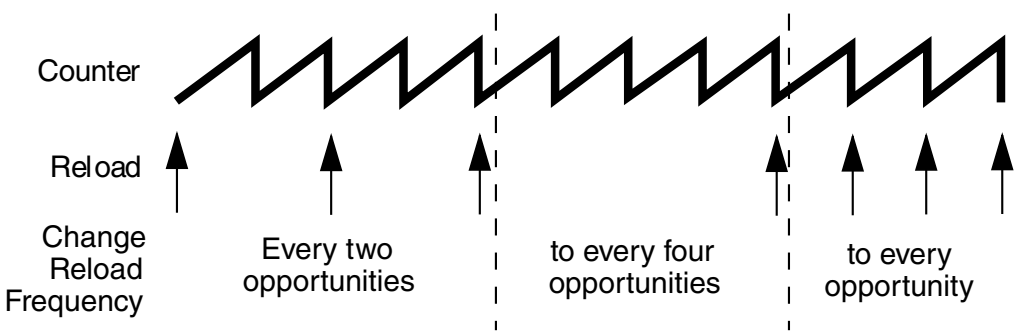


Figure 73-32. Full cycle reload frequency change

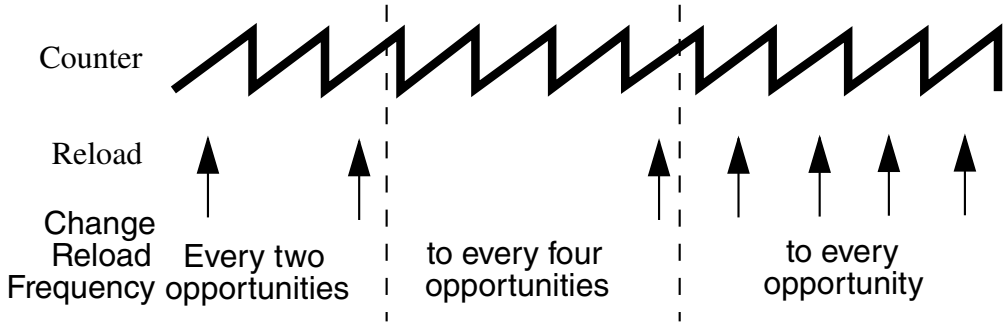


Figure 73-33. Half cycle reload frequency change

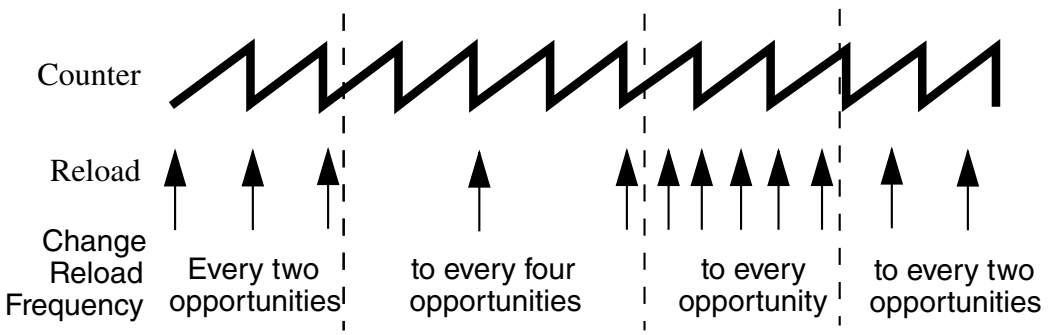


Figure 73-34. Full and half cycle reload frequency change

73.3.3.12.3 Reload flag

At every reload opportunity the PWM Reload Flag (STS[RF]) is set. Setting STS[RF] happens even if an actual reload is prevented by MCTRL[LDOK]. If the PWM reload interrupt enable bit INTEN[RIE] is set, the STS[RF] flag generates CPU interrupt requests allowing software to calculate new PWM parameters in real time. When INTEN[RIE] is not set, reloads still occur at the selected reload rate without generating CPU interrupt requests.

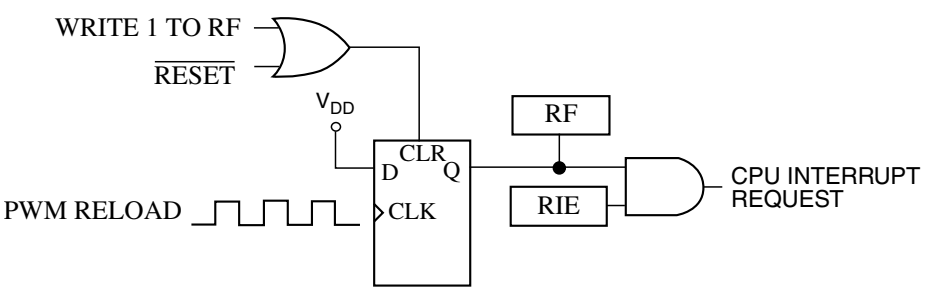


Figure 73-35. PWMF reload interrupt request

73.3.3.12.4 Reload errors

When one of the INIT, VALx, FRACVALx, or CTRL[PRSC] registers is updated (written by software), the STS[RUF] flag is set to indicate the data in the set of double buffered registers is not coherent. STS[RUF] is cleared by a successful reload which consists of the reload signal while MCTRL[LDOK] is set. If STS[RUF] is set and MCTRL[LDOK] is clear when the reload signal occurs, a reload error takes place and STS[REF] is set. If STS[RUF] is clear when a reload signal asserts, then the data is coherent and no error is flagged.

73.3.3.12.5 Initialization

Initialize all registers and then set MCTRL[LDOK] before setting MCTRL[RUN].

Note

If MCTRL[LDOK] is not set, setting MCTRL[RUN] also sets the STS[RF] flag. To prevent a CPU interrupt request, clear INTEN[RIE] before setting MCTRL[RUN].

The PWM generator uses the last values loaded if MCTRL[RUN] is cleared and then set while MCTRL[LDOK] equals zero.

When MCTRL[RUN] is cleared:

- The STS[RF] flag and pending CPU interrupt requests are not cleared
- All fault circuitry remains active
- Software/external output control remains active
- Deadtime insertion continues during software/external output control

73.3.4 Power modes

Be careful when using this module in Stop, Wait, and Debug operating modes.

CAUTION

Some applications (such as three-phase AC motors) require regular software updates for proper operation. Failure to provide regular software updates could result in destroying the hardware setup.

To accommodate this situation, PWM outputs are placed in their inactive states in Stop mode, and they can optionally be placed in inactive states in Wait and Debug modes. PWM outputs are reactivated (assuming they were active beforehand) when these modes are exited.

Table 73-3. Modes when PWM operation is restricted

Mode	Description
Stop	PWM outputs are inactive.
Wait	PWM outputs are driven or inactive as a function of CTRL2[WAITEN].
Debug	PWM outputs are driven or inactive as a function of CTRL2[DBGEN].

73.3.5 Clocking

Figure 73-36 shows the logic used to generate the main counter clock. Each submodule can select between three clock signals: the IPBus clock, EXT_CLK, and AUX_CLK. The EXT_CLK goes to all of the submodules. The AUX_CLK signal is broadcast from submodule0 and can be selected as the clock source by other submodules so that the 8-bit prescaler and MCTRL[RUN] from submodule0 can control all of the submodules.

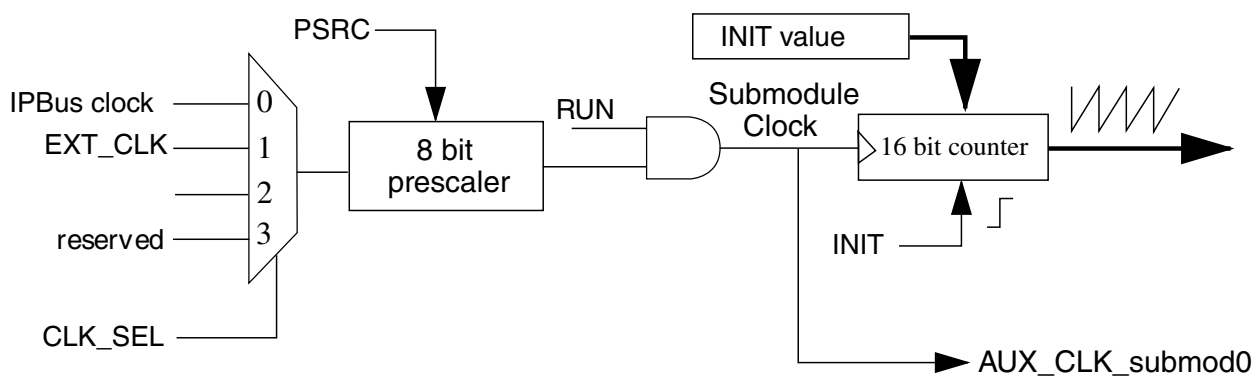


Figure 73-36. Clocking Block Diagram for PWM Submodule 0

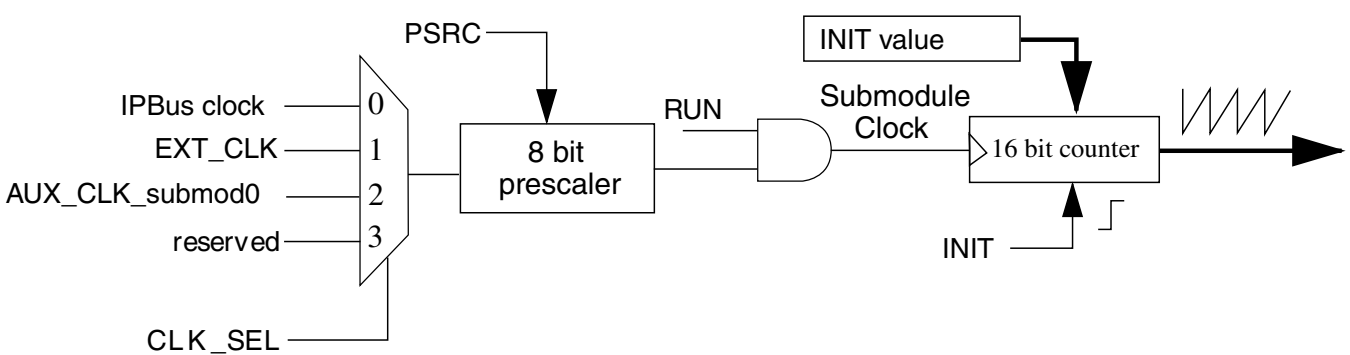


Figure 73-37. Clocking Block Diagram for all other PWM Submodules

To permit lower PWM frequencies, the prescaler produces the PWM clock frequency by dividing the IPBus clock frequency by 1-128. The prescaler bits, CTRL[PRSC], select the prescaler divisor. This prescaler is buffered and will not be used by the PWM generator until MCTRL[LDOK] is set and a new PWM reload cycle begins or CTRL[LDMOD] is set.

73.3.6 Resets

All PWM registers are reset to their default values upon any system reset.

The reset forces all registers to their reset states and tristates the PWM outputs.

73.3.7 Interrupts

Each of the submodules within the eFlexPWM module can generate an interrupt from several sources. The fault logic can also generate interrupts. The interrupt service routine (ISR) must check the related interrupt enables and interrupt flags to determine the actual cause of the interrupt.

Table 73-4. Interrupt Summary

Core Interrupt	Interrupt Flag	Interrupt Enable	Name	Description
PWM_CMP0	SM0STS[CMPF]	SM0INTEN[CMPIE]	Submodule 0 compare interrupt	Compare event has occurred
PWM_CAP0	SM0STS[CFA1], SM0STS[CFA0], SM0STS[CFB1], SM0STS[CFB0], SM0STS[CFX1], SM0STS[CFX0]	SM0INTEN[CFA1IE], SM0INTEN[CFA0IE], SM0INTEN[CFB1IE], SM0INTEN[CFB0IE], SM0INTEN[CFX1IE], SM0INTEN[CFX0IE]	Submodule 0 input capture interrupt	Input capture event has occurred
PWM_RELOAD0	SM0STS[RF]	SM0INTEN[RIE]	Submodule 0 reload interrupt	Reload event has occurred
PWM_CMP1	SM1STS[CMPF]	SM1INTEN[CMPIE]	Submodule 1 compare interrupt	Compare event has occurred
PWM_CAP1	SM1STS[CFA1], SM1STS[CFA0], SM1STS[CFB1], SM1STS[CFB0], SM1STS[CFX1], SM1STS[CFX0]	SM1INTEN[CFA1IE], SM1INTEN[CFA0IE], SM1INTEN[CFB1IE], SM1INTEN[CFB0IE], SM1INTEN[CFX1IE], SM1INTEN[CFX0IE]	Submodule 1 input capture interrupt	Input capture event has occurred

Table continues on the next page...

Table 73-4. Interrupt Summary (continued)

Core Interrupt	Interrupt Flag	Interrupt Enable	Name	Description
PWM_RELOAD1	SM1STS[RF]	SM1INTEN[RIE]	Submodule 1 reload interrupt	Reload event has occurred
PWM_CMP2	SM2STS[CMPF]	SM2INTEN[CMPIE]	Submodule 2 compare interrupt	Compare event has occurred
PWM_CAP2	SM2STS[CFA1], SM2STS[CFA0], SM2STS[CFB1], SM2STS[CFB0], SM2STS[CFX1], SM2STS[CFX0]	SM2INTEN[CFA1IE], SM2INTEN[CFA0IE], SM2INTEN[CFB1IE], SM2INTEN[CFB0IE], SM2INTEN[CFX1IE], SM2INTEN[CFX0IE	Submodule 2 input capture interrupt	Input capture event has occurred
PWM_RELOAD2	SM2STS[RF]	SM2INTEN[RIE]	Submodule 2 reload interrupt	Reload event has occurred
PWM_CMP3	SM3STS[CMPF]	SM3INTEN[CMPIE]	Submodule 3 compare interrupt	Compare event has occurred
PWM_CAP3	SM3STS[CFA1], SM3STS[CFA0], SM3STS[CFB1], SM3STS[CFB0], SM3STS[CFX1], SM3STS[CFX0]	SM3INTEN[CFA1IE], SM3INTEN[CFA0IE], SM3INTEN[CFB1IE], SM3INTEN[CFB0IE], SM3INTEN[CFX1IE], SM3INTEN[CFX0IE	Submodule 3 input capture interrupt	Input capture event has occurred
PWM_RELOAD3	SM3STS[RF]	SM3INTEN[RIE]	Submodule 3 reload interrupt	Reload event has occurred
PWM_RERR	SM0STS[REF]	SM0INTEN[REIE]	Submodule 0 reload error interrupt	Reload error has occurred
	SM1STS[REF]	SM1INTEN[REIE]	Submodule 1 reload error interrupt	
	SM2STS[REF]	SM2INTEN[REIE]	Submodule 2 reload error interrupt	
	SM3STS[REF]	SM3INTEN[REIE]	Submodule 3 reload error interrupt	
PWM_FAULT	FSTS[FFLAG]	FCTRL[FIE]	Fault input interrupt	Fault condition has been detected

73.3.8 DMA

Each submodule can request a DMA read access for its capture FIFOs and a DMA write request for its double buffered registers.

Table 73-5. DMA summary

DMA request	DMA enable	Name	Description
Submodule 0 read request	SM0DMAEN[CX0DE]	SM0 Capture FIFO X0 read request	SM0CVAL0 contains a value to be read
	SM0DMAEN[CX1DE]	SM0 Capture FIFO X1 read request	SM0CVAL1 contains a value to be read
	SM0DMAEN[CA0DE]	SM0 Capture FIFO A0 read request	SM0CVAL2 contains a value to be read
	SM0DMAEN[CA1DE]	SM0 Capture FIFO A1 read request	SM0CVAL3 contains a value to be read
	SM0DMAEN[CB0DE]	SM0 Capture FIFO B0 read request	SM0CVAL4 contains a value to be read
	SM0DMAEN[CB1DE]	SM0 Capture FIFO B1 read request	SM0CVAL5 contains a value to be read
	SM0DMAEN[CAPTDE]	SM0 Capture FIFO read request source select	Selects source of submodule0 read DMA request
Submodule 0 write request	SM0DMAEN[VALDE]	SM0VALx write request	SM0VALx and SM0FRACVALx registers need to be updated
Submodule 1 read request	SM1DMAEN[CX0DE]	SM1 Capture FIFO X0 read request	SM1CVAL0 contains a value to be read
	SM1DMAEN[CX1DE]	SM1 Capture FIFO X1 read request	SM1CVAL1 contains a value to be read
	SM1DMAEN[CA0DE]	SM1Capture FIFO A0 read request	SM1CVAL2 contains a value to be read
	SM1DMAEN[CA1DE]	SM1 Capture FIFO A1 read request	SM1CVAL3 contains a value to be read
	SM1DMAEN[CB0DE]	SM1 Capture FIFO B0 read request	SM1CVAL4 contains a value to be read
	SM1DMAEN[CB1DE]	SM1 Capture FIFO B1 read request	SM1CVAL5 contains a value to be read
	SM1DMAEN[CAPTDE]	SM1 Capture FIFO read request source select	Selects source of submodule1 read DMA request
Submodule 1 write request	SM1DMAEN[VALDE]	SM1VALx write request	SM1VALx and SM1FRACVALx registers need to be updated
Submodule 2 read request	SM2DMAEN[CX0DE]	SM2 Capture FIFO X0 read request	SM2CVAL0 contains a value to be read
	SM2DMAEN[CX1DE]	SM2 Capture FIFO X1 read request	SM2CVAL1 contains a value to be read
	SM2DMAEN[CA0DE]	SM2 Capture FIFO A0 read request	SM2CVAL2 contains a value to be read
	SM2DMAEN[CA1DE]	SM2 Capture FIFO A1 read request	SM2CVAL3 contains a value to be read
	SM2DMAEN[CB0DE]	SM2 Capture FIFO B0 read request	SM2CVAL4 contains a value to be read
	SM2DMAEN[CB1DE]	SM2 Capture FIFO B1 read request	SM2CVAL5 contains a value to be read

Table continues on the next page...

Table 73-5. DMA summary (continued)

DMA request	DMA enable	Name	Description
	SM2DMAEN[CAPTDE]	SM2 Capture FIFO read request source select	Selects source of submodule2 read DMA request
Submodule 2 write request	SM2DMAEN[VALDE]	SM2VALx write request	SM2VALx and SM2FRACVALx registers need to be updated
Submodule 3 read request	SM3DMAEN[CX0DE]	SM3 Capture FIFO X0 read request	SM3CVAL0 contains a value to be read
	SM3DMAEN[CX1DE]	SM3 Capture FIFO X1 read request	SM3CVAL1 contains a value to be read
	SM3DMAEN[CA0DE]	SM3 Capture FIFO A0 read request	SM3CVAL2 contains a value to be read
	SM3DMAEN[CA1DE]	SM3 Capture FIFO A1 read request	SM3CVAL3 contains a value to be read
	SM3DMAEN[CB0DE]	SM3 Capture FIFO B0 read request	SM3CVAL4 contains a value to be read
	SM3DMAEN[CB1DE]	SM3 Capture FIFO B1 read request	SM3CVAL5 contains a value to be read
	SM3DMAEN[CAPTDE]	SM3 Capture FIFO read request source select	Selects source of submodule3 read DMA request
Submodule 3 write request	SM3DMAEN[VALDE]	SM3VALx write request	SM3VALx and SM3FRACVALx registers need to be updated

73.4 External signals

The PWM has pins named PWM_An, PWM_Bn, PWM_Xn, FAULTn, EXT_SYNC, EXT_FORCE, and PWMn_EXTA. The PWM also has an on-chip input called EXT_CLK and an output signal called PWMn_OUT_TRIGx.

73.4.1 PWM_An and PWM_Bn - External PWM output pair

These pins are the output pins of the PWM channels. These pins can be independent PWM signals or a complementary pair. When not needed as an output, they can be used as inputs to the input capture circuitry.

73.4.2 PWM_Xn - Auxiliary PWM output signal

These pins are the auxiliary output pins of the PWM channels. They can be independent PWM signals. When not needed as an output, they can be used as inputs to the input capture circuitry or used to detect the polarity of the current flowing through the complementary circuit at deadtime correction.

73.4.3 FAULTn - Fault Inputs

These are input pins for disabling selected PWM outputs.

73.4.4 EXT_SYNC - External synchronization signal

These input signals allow a source external to the PWM to initialize the PWM counter. Therefore, the PWM can be synchronized to external circuitry.

73.4.5 EXT_FORCE - External output force signal

This input signal allows a source external to the PWM to force an update of the PWM outputs. Therefore, the PWM can be synchronized to external circuitry.

73.4.6 PWMn_EXTA - Alternate PWM control signals

These pins allow an alternate source to control the PWM_An and PWM_Bn outputs. Typically, the PWMn_EXTA input (depending on the state of MCTRL[IPOL]) is used for the generation of a complementary pair. Typical control signals include ADC conversion high/low limits, TMR outputs, GPIO inputs, and comparator outputs.

For pin input details, see chip-specific eFlexPWM information.

73.4.7 PWMn_OUT_TRIG0 and PWMn_OUT_TRIG1 - Output triggers

These outputs allow the PWM submodules to control timing of ADC conversions. See the description of the SMnTCTRL[OUT_TRIG_EN] for information about how to enable these outputs and how the compare registers match up to the output triggers.

73.4.8 EXT_CLK - External clock signal

This signal allows a source external to the PWM (typically a timer or an off-chip source) to control the PWM clocking. Therefore, the PWM can be synchronized to the timer, or multiple chips can be synchronized to each other.

73.5 PWM register descriptions

The address of a register is the sum of a base address and an address offset. The base address is defined at the core level, and the address offset is defined at the module level. The PWM module has a set of registers for each PWM submodule, for the configuration logic, and for each fault channel.

Note

While the registers are 16-bit wide, they can be accessed in pairs as 32-bit registers, if the pairs are word-aligned.

Submodule registers are repeated for each PWM submodule. To designate which submodule that they are in, register names are prefixed with SMx(x=0,1...). The base address of submodule 0 is the same as the base address for the PWM module as a whole. The base address of submodule 1 is offset 0x60 from the base address for the PWM module as a whole. This 0x60 offset is based on the number of registers in a submodule.The base address of submodule 2 is equal to the base address of submodule 1 plus this same 0x60 offset.The pattern repeats for the base address of submodule 3.

The base address of the configuration registers is equal to the base address of the PWM module as a whole plus an offset of 0x180.

73.5.1 PWM memory map

PWM1 base address: 4018_C000h

PWM2 base address: 4019_0000h

PWM3 base address: 4019_4000h

PWM4 base address: 4019_8000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Counter Register (SM0CNT)	16	RO	0000h
2h	Initial Count Register (SM0INIT)	16	RW	0000h
4h	Control 2 Register (SM0CTRL2)	16	Table 73-5	0000h
6h	Control Register (SM0CTRL)	16	Table 73-5	0400h
Ah	Value Register 0 (SM0VAL0)	16	RW	0000h
Ch	Fractional Value Register 1 (SM0FRACVAL1)	16	Table 73-5	0000h
Eh	Value Register 1 (SM0VAL1)	16	RW	0000h
10h	Fractional Value Register 2 (SM0FRACVAL2)	16	Table 73-5	0000h
12h	Value Register 2 (SM0VAL2)	16	RW	0000h
14h	Fractional Value Register 3 (SM0FRACVAL3)	16	Table 73-5	0000h
16h	Value Register 3 (SM0VAL3)	16	RW	0000h
18h	Fractional Value Register 4 (SM0FRACVAL4)	16	Table 73-5	0000h
1Ah	Value Register 4 (SM0VAL4)	16	RW	0000h
1Ch	Fractional Value Register 5 (SM0FRACVAL5)	16	Table 73-5	0000h
1Eh	Value Register 5 (SM0VAL5)	16	RW	0000h
20h	Fractional Control Register (SM0FRCTRL)	16	Table 73-5	0000h
22h	Output Control Register (SM0OCTRL)	16	Table 73-5	0000h
24h	Status Register (SM0STS)	16	Table 73-5	0000h
26h	Interrupt Enable Register (SM0INTEN)	16	Table 73-5	0000h
28h	DMA Enable Register (SM0DMAEN)	16	Table 73-5	0000h
2Ah	Output Trigger Control Register (SM0TCTRL)	16	Table 73-5	0000h
2Ch	Fault Disable Mapping Register 0 (SM0DISMAP0)	16	Table 73-5	FFFFh
30h	Deadtime Count Register 0 (SM0DTCNT0)	16	RW	07FFh
32h	Deadtime Count Register 1 (SM0DTCNT1)	16	RW	07FFh
34h	Capture Control A Register (SM0CAPTCTRLA)	16	Table 73-5	0000h
36h	Capture Compare A Register (SM0CAPTCOMPA)	16	Table 73-5	0000h
38h	Capture Control B Register (SM0CAPTCTRLB)	16	Table 73-5	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3Ah	Capture Compare B Register (SM0CAPTCOMPB)	16	Table 73-5	0000h
3Ch	Capture Control X Register (SM0CAPTCTRLX)	16	Table 73-5	0000h
3Eh	Capture Compare X Register (SM0CAPTCOMPX)	16	Table 73-5	0000h
40h	Capture Value 0 Register (SM0CVAL0)	16	RO	0000h
42h	Capture Value 0 Cycle Register (SM0CVAL0CYC)	16	Table 73-5	0000h
44h	Capture Value 1 Register (SM0CVAL1)	16	RO	0000h
46h	Capture Value 1 Cycle Register (SM0CVAL1CYC)	16	Table 73-5	0000h
48h	Capture Value 2 Register (SM0CVAL2)	16	RO	0000h
4Ah	Capture Value 2 Cycle Register (SM0CVAL2CYC)	16	Table 73-5	0000h
4Ch	Capture Value 3 Register (SM0CVAL3)	16	RO	0000h
4Eh	Capture Value 3 Cycle Register (SM0CVAL3CYC)	16	Table 73-5	0000h
50h	Capture Value 4 Register (SM0CVAL4)	16	RO	0000h
52h	Capture Value 4 Cycle Register (SM0CVAL4CYC)	16	Table 73-5	0000h
54h	Capture Value 5 Register (SM0CVAL5)	16	RO	0000h
56h	Capture Value 5 Cycle Register (SM0CVAL5CYC)	16	Table 73-5	0000h
60h	Counter Register (SM1CNT)	16	RO	0000h
62h	Initial Count Register (SM1INIT)	16	RW	0000h
64h	Control 2 Register (SM1CTRL2)	16	Table 73-5	0000h
66h	Control Register (SM1CTRL)	16	Table 73-5	0400h
6Ah	Value Register 0 (SM1VAL0)	16	RW	0000h
6Ch	Fractional Value Register 1 (SM1FRACVAL1)	16	Table 73-5	0000h
6Eh	Value Register 1 (SM1VAL1)	16	RW	0000h
70h	Fractional Value Register 2 (SM1FRACVAL2)	16	Table 73-5	0000h
72h	Value Register 2 (SM1VAL2)	16	RW	0000h
74h	Fractional Value Register 3 (SM1FRACVAL3)	16	Table 73-5	0000h
76h	Value Register 3 (SM1VAL3)	16	RW	0000h
78h	Fractional Value Register 4 (SM1FRACVAL4)	16	Table 73-5	0000h
7Ah	Value Register 4 (SM1VAL4)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7Ch	Fractional Value Register 5 (SM1FRACVAL5)	16	Table 73-5	0000h
7Eh	Value Register 5 (SM1VAL5)	16	RW	0000h
80h	Fractional Control Register (SM1FRCTRL)	16	Table 73-5	0000h
82h	Output Control Register (SM1OCTRL)	16	Table 73-5	0000h
84h	Status Register (SM1STS)	16	Table 73-5	0000h
86h	Interrupt Enable Register (SM1INTEN)	16	Table 73-5	0000h
88h	DMA Enable Register (SM1DMAEN)	16	Table 73-5	0000h
8Ah	Output Trigger Control Register (SM1TCTRL)	16	Table 73-5	0000h
8Ch	Fault Disable Mapping Register 0 (SM1DISMAP0)	16	Table 73-5	FFFFh
90h	Deadtime Count Register 0 (SM1DTCNT0)	16	RW	07FFh
92h	Deadtime Count Register 1 (SM1DTCNT1)	16	RW	07FFh
94h	Capture Control A Register (SM1CAPTCTRLA)	16	Table 73-5	0000h
96h	Capture Compare A Register (SM1CAPTCOMPA)	16	Table 73-5	0000h
98h	Capture Control B Register (SM1CAPTCTRLB)	16	Table 73-5	0000h
9Ah	Capture Compare B Register (SM1CAPTCOMPB)	16	Table 73-5	0000h
9Ch	Capture Control X Register (SM1CAPTCTRLX)	16	Table 73-5	0000h
9Eh	Capture Compare X Register (SM1CAPTCOMPX)	16	Table 73-5	0000h
A0h	Capture Value 0 Register (SM1CVAL0)	16	RO	0000h
A2h	Capture Value 0 Cycle Register (SM1CVAL0CYC)	16	Table 73-5	0000h
A4h	Capture Value 1 Register (SM1CVAL1)	16	RO	0000h
A6h	Capture Value 1 Cycle Register (SM1CVAL1CYC)	16	Table 73-5	0000h
A8h	Capture Value 2 Register (SM1CVAL2)	16	RO	0000h
AAh	Capture Value 2 Cycle Register (SM1CVAL2CYC)	16	Table 73-5	0000h
ACh	Capture Value 3 Register (SM1CVAL3)	16	RO	0000h
AEh	Capture Value 3 Cycle Register (SM1CVAL3CYC)	16	Table 73-5	0000h
B0h	Capture Value 4 Register (SM1CVAL4)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B2h	Capture Value 4 Cycle Register (SM1CVAL4CYC)	16	Table 73-5	0000h
B4h	Capture Value 5 Register (SM1CVAL5)	16	RO	0000h
B6h	Capture Value 5 Cycle Register (SM1CVAL5CYC)	16	Table 73-5	0000h
C0h	Counter Register (SM2CNT)	16	RO	0000h
C2h	Initial Count Register (SM2INIT)	16	RW	0000h
C4h	Control 2 Register (SM2CTRL2)	16	Table 73-5	0000h
C6h	Control Register (SM2CTRL)	16	Table 73-5	0400h
CAh	Value Register 0 (SM2VAL0)	16	RW	0000h
CCh	Fractional Value Register 1 (SM2FRACVAL1)	16	Table 73-5	0000h
CEh	Value Register 1 (SM2VAL1)	16	RW	0000h
D0h	Fractional Value Register 2 (SM2FRACVAL2)	16	Table 73-5	0000h
D2h	Value Register 2 (SM2VAL2)	16	RW	0000h
D4h	Fractional Value Register 3 (SM2FRACVAL3)	16	Table 73-5	0000h
D6h	Value Register 3 (SM2VAL3)	16	RW	0000h
D8h	Fractional Value Register 4 (SM2FRACVAL4)	16	Table 73-5	0000h
DAh	Value Register 4 (SM2VAL4)	16	RW	0000h
DCh	Fractional Value Register 5 (SM2FRACVAL5)	16	Table 73-5	0000h
DEh	Value Register 5 (SM2VAL5)	16	RW	0000h
E0h	Fractional Control Register (SM2FRCTRL)	16	Table 73-5	0000h
E2h	Output Control Register (SM2OCTRL)	16	Table 73-5	0000h
E4h	Status Register (SM2STS)	16	Table 73-5	0000h
E6h	Interrupt Enable Register (SM2INTEN)	16	Table 73-5	0000h
E8h	DMA Enable Register (SM2DMAEN)	16	Table 73-5	0000h
EAh	Output Trigger Control Register (SM2TCTRL)	16	Table 73-5	0000h
ECh	Fault Disable Mapping Register 0 (SM2DISMAP0)	16	Table 73-5	FFFFh
F0h	Deadtime Count Register 0 (SM2DTCNT0)	16	RW	07FFh
F2h	Deadtime Count Register 1 (SM2DTCNT1)	16	RW	07FFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
F4h	Capture Control A Register (SM2CAPTCTRLA)	16	Table 73-5	0000h
F6h	Capture Compare A Register (SM2CAPTCOMPA)	16	Table 73-5	0000h
F8h	Capture Control B Register (SM2CAPTCTRLB)	16	Table 73-5	0000h
FAh	Capture Compare B Register (SM2CAPTCOMPB)	16	Table 73-5	0000h
FCh	Capture Control X Register (SM2CAPTCTRLX)	16	Table 73-5	0000h
FEh	Capture Compare X Register (SM2CAPTCOMPX)	16	Table 73-5	0000h
100h	Capture Value 0 Register (SM2CVAL0)	16	RO	0000h
102h	Capture Value 0 Cycle Register (SM2CVAL0CYC)	16	Table 73-5	0000h
104h	Capture Value 1 Register (SM2CVAL1)	16	RO	0000h
106h	Capture Value 1 Cycle Register (SM2CVAL1CYC)	16	Table 73-5	0000h
108h	Capture Value 2 Register (SM2CVAL2)	16	RO	0000h
10Ah	Capture Value 2 Cycle Register (SM2CVAL2CYC)	16	Table 73-5	0000h
10Ch	Capture Value 3 Register (SM2CVAL3)	16	RO	0000h
10Eh	Capture Value 3 Cycle Register (SM2CVAL3CYC)	16	Table 73-5	0000h
110h	Capture Value 4 Register (SM2CVAL4)	16	RO	0000h
112h	Capture Value 4 Cycle Register (SM2CVAL4CYC)	16	Table 73-5	0000h
114h	Capture Value 5 Register (SM2CVAL5)	16	RO	0000h
116h	Capture Value 5 Cycle Register (SM2CVAL5CYC)	16	Table 73-5	0000h
120h	Counter Register (SM3CNT)	16	RO	0000h
122h	Initial Count Register (SM3INIT)	16	RW	0000h
124h	Control 2 Register (SM3CTRL2)	16	Table 73-5	0000h
126h	Control Register (SM3CTRL)	16	Table 73-5	0400h
12Ah	Value Register 0 (SM3VAL0)	16	RW	0000h
12Ch	Fractional Value Register 1 (SM3FRACVAL1)	16	Table 73-5	0000h
12Eh	Value Register 1 (SM3VAL1)	16	RW	0000h
130h	Fractional Value Register 2 (SM3FRACVAL2)	16	Table 73-5	0000h
132h	Value Register 2 (SM3VAL2)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
134h	Fractional Value Register 3 (SM3FRACVAL3)	16	Table 73-5	0000h
136h	Value Register 3 (SM3VAL3)	16	RW	0000h
138h	Fractional Value Register 4 (SM3FRACVAL4)	16	Table 73-5	0000h
13Ah	Value Register 4 (SM3VAL4)	16	RW	0000h
13Ch	Fractional Value Register 5 (SM3FRACVAL5)	16	Table 73-5	0000h
13Eh	Value Register 5 (SM3VAL5)	16	RW	0000h
140h	Fractional Control Register (SM3FRCTRL)	16	Table 73-5	0000h
142h	Output Control Register (SM3OCTRL)	16	Table 73-5	0000h
144h	Status Register (SM3STS)	16	Table 73-5	0000h
146h	Interrupt Enable Register (SM3INTEN)	16	Table 73-5	0000h
148h	DMA Enable Register (SM3DMAEN)	16	Table 73-5	0000h
14Ah	Output Trigger Control Register (SM3TCTRL)	16	Table 73-5	0000h
14Ch	Fault Disable Mapping Register 0 (SM3DISMAP0)	16	Table 73-5	FFFFh
150h	Deadtime Count Register 0 (SM3DTCNT0)	16	RW	07FFh
152h	Deadtime Count Register 1 (SM3DTCNT1)	16	RW	07FFh
154h	Capture Control A Register (SM3CAPTCTRLA)	16	Table 73-5	0000h
156h	Capture Compare A Register (SM3CAPTCOMPA)	16	Table 73-5	0000h
158h	Capture Control B Register (SM3CAPTCTRLB)	16	Table 73-5	0000h
15Ah	Capture Compare B Register (SM3CAPTCOMPB)	16	Table 73-5	0000h
15Ch	Capture Control X Register (SM3CAPTCTRLX)	16	Table 73-5	0000h
15Eh	Capture Compare X Register (SM3CAPTCOMPX)	16	Table 73-5	0000h
160h	Capture Value 0 Register (SM3CVAL0)	16	RO	0000h
162h	Capture Value 0 Cycle Register (SM3CVAL0CYC)	16	Table 73-5	0000h
164h	Capture Value 1 Register (SM3CVAL1)	16	RO	0000h
166h	Capture Value 1 Cycle Register (SM3CVAL1CYC)	16	Table 73-5	0000h
168h	Capture Value 2 Register (SM3CVAL2)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
16Ah	Capture Value 2 Cycle Register (SM3CVAL2CYC)	16	Table 73-5	0000h
16Ch	Capture Value 3 Register (SM3CVAL3)	16	RO	0000h
16Eh	Capture Value 3 Cycle Register (SM3CVAL3CYC)	16	Table 73-5	0000h
170h	Capture Value 4 Register (SM3CVAL4)	16	RO	0000h
172h	Capture Value 4 Cycle Register (SM3CVAL4CYC)	16	Table 73-5	0000h
174h	Capture Value 5 Register (SM3CVAL5)	16	RO	0000h
176h	Capture Value 5 Cycle Register (SM3CVAL5CYC)	16	Table 73-5	0000h
180h	Output Enable Register (OUTEN)	16	Table 73-5	0000h
182h	Mask Register (MASK)	16	Table 73-5	0000h
184h	Software Controlled Output Register (SWCOUT)	16	Table 73-5	0000h
186h	PWM Source Select Register (DTSRCSEL)	16	RW	0000h
188h	Master Control Register (MCTRL)	16	Table 73-5	0000h
18Ch	Fault Control Register (FCTRL0)	16	RW	0000h
18Eh	Fault Status Register (FSTS0)	16	Table 73-5	0000h
190h	Fault Filter Register (FFILT0)	16	Table 73-5	0000h
192h	Fault Test Register (FTST0)	16	Table 73-5	0000h
194h	Fault Control 2 Register (FCTRL20)	16	Table 73-5	0000h

73.5.2 Counter Register (SM0CNT - SM3CNT)

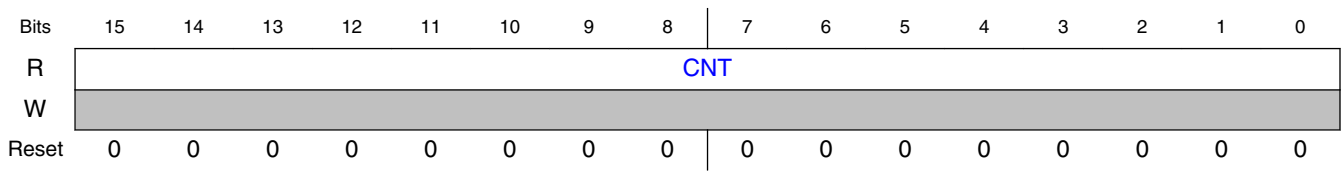
73.5.2.1 Offset

Register	Offset
SM0CNT	0h
SM1CNT	60h
SM2CNT	C0h
SM3CNT	120h

73.5.2.2 Function

This field displays the state of the signed 16-bit submodule counter. This register is not byte accessible.

73.5.2.3 Diagram



73.5.2.4 Fields

Field	Function
15-0 CNT	Counter Register Bits

73.5.3 Initial Count Register (SM0INIT - SM3INIT)

73.5.3.1 Offset

Register	Offset
SM0INIT	2h
SM1INIT	62h
SM2INIT	C2h
SM3INIT	122h

73.5.3.2 Function

The 16-bit signed value in this buffered register defines the initial count value for the PWM in PWM clock periods. This is the value loaded into the submodule counter when local sync, master sync, or master reload is asserted (based on the value of CTRL2[INIT_SEL]) or when CTRL2[FORCE] is asserted and force init is enabled. For PWM operation, the buffered contents of this register are loaded into the counter at the start of every PWM cycle. This register is not byte accessible.

Note

The INIT register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. This register cannot be written when MCTRL[LDOK] is set. Reading INIT reads the value in a buffer and not necessarily the value the PWM generator is currently using.

73.5.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.3.4 Fields

Field	Function
15-0 INIT	Initial Count Register Bits

73.5.4 Control 2 Register (SM0CTRL2 - SM3CTRL2)

73.5.4.1 Offset

Register	Offset
SM0CTRL2	4h
SM1CTRL2	64h
SM2CTRL2	C4h
SM3CTRL2	124h

73.5.4.2 Function

Contains control fields for clock select and forcing output and initialization.

73.5.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DBGEN	WAITEN	INDEP	PWM23_INIT	PWM45_INIT	PWMX_INIT	INIT_SEL		FRCEN	0	FORCE_SEL			RELOAD_SEL	CLK_SEL	
W										FORCE						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.4.4 Fields

Field	Function
15 DBGEN	Debug Enable When set to one, the PWM continues to run while the chip is in Debug mode. If the device enters Debug mode and this bit is zero, then the PWM outputs are disabled until Debug mode is exited. At that point, the PWM pins resume operation as programmed in the PWM registers. For certain types of motors (such as 3-phase AC), it is imperative that this bit be left in its default state (in which the PWM is disabled in Debug mode). Failure to do so could result in damage to the motor or inverter. For other types of motors (example: DC motors), this bit might safely be set to one, enabling the PWM in Debug mode. The key point is PWM parameter updates will not occur in Debug mode. Any motors requiring such updates should be disabled during Debug mode. If in doubt, leave this bit set to zero.
14 WAITEN	Wait Enable

Table continues on the next page...

Field	Function
	When set to one, the PWM continues to run while the chip is in Wait mode. In this mode, the peripheral clock continues to run but the CPU clock does not. If the device enters Wait mode and this bit is zero, then the PWM outputs are disabled until Wait mode is exited. At that point the PWM pins resume operation as programmed in the PWM registers. For certain types of motors (such as 3-phase AC), it is imperative that this bit be left in its default state (in which the PWM is disabled in Wait mode). Failure to do so could result in damage to the motor or inverter. For other types of motors (example: DC motors), this bit might safely be set to one, enabling the PWM in Wait mode. The key point is PWM parameter updates will not occur in this mode. Any motors requiring such updates should be disabled during Wait mode. If in doubt, leave this bit set to zero.
13 INDEP	Independent or Complementary Pair Operation This bit determines if the PWM_A and PWM_B channels are independent PWMs or a complementary PWM pair. 0b - PWM_A and PWM_B form a complementary PWM pair. 1b - PWM_A and PWM_B outputs are independent PWMs.
12 PWM23_INIT	PWM23 Initial Value This bit determines the initial value for PWM23 and the value to which it is forced when FORCE_INIT is asserted.
11 PWM45_INIT	PWM45 Initial Value This bit determines the initial value for PWM45 and the value to which it is forced when FORCE_INIT is asserted.
10 PWMX_INIT	PWM_X Initial Value This bit determines the initial value for PWM_X and the value to which it is forced when FORCE_INIT is asserted.
9-8 INIT_SEL	Initialization Control Select These bits control the source of the INIT signal which goes to the counter. 00b - Local sync (PWM_X) causes initialization. 01b - Master reload from submodule 0 causes initialization. This setting should not be used in submodule 0 as it forces the INIT signal to logic 0. The submodule counter will only re-initialize when a master reload occurs. 10b - Master sync from submodule 0 causes initialization. This setting should not be used in submodule 0 as it forces the INIT signal to logic 0. 11b - EXT_SYNC causes initialization.
7 FRCEN	Force Enable This bit allows the CTRL2[FORCE] signal to initialize the counter without regard to the signal selected by CTRL2[INIT_SEL]. This is a software controlled initialization. A forced initialization will also assert the register reload if MCTRL[LDOK] is set. 0b - Initialization from a FORCE_OUT is disabled. 1b - Initialization from a FORCE_OUT is enabled.
6 FORCE	Force Initialization If CTRL2[FORCE_SEL] is set to 000, writing a 1 to this bit results in a FORCE_OUT event. This causes the following actions to be taken: • The PWM_A and PWM_B output pins assume values based on DTSRCSEL[SMxSEL23] and DTSRCSEL[SMxSEL45]. • If CTRL2[FRCEN] is set, the counter value is initialized with the INIT register value only when the submodule MCTRL[RUN] = 1 or CTRL2[CLK_SEL] = 2.
5-3 FORCE_SEL	Force Select This bit determines the source of the FORCE OUTPUT signal for this submodule. 000b - The local force signal, CTRL2[FORCE], from this submodule is used to force updates.

Table continues on the next page...

Field	Function
	001b - The master force signal from submodule 0 is used to force updates. This setting should not be used in submodule 0 as it holds the FORCE OUTPUT signal to logic 0. 010b - The local reload signal from this submodule is used to force updates without regard to the state of LDOK. 011b - The master reload signal from submodule0 is used to force updates if LDOK is set. This setting should not be used in submodule0 as it holds the FORCE OUTPUT signal to logic 0. 100b - The local sync signal from this submodule is used to force updates. 101b - The master sync signal from submodule0 is used to force updates. This setting should not be used in submodule0 as it holds the FORCE OUTPUT signal to logic 0. 110b - The external force signal, EXT_FORCE, from outside the PWM module causes updates. 111b - The external sync signal, EXT_SYNC, from outside the PWM module causes updates.
2 RELOAD_SEL	Reload Source Select This bit determines the source of the RELOAD signal for this submodule. When this bit is set, MCTRL[LDOK[0]] for submodule 0 should be used since the local MCTRL[LDOK] will be ignored. 0b - The local RELOAD signal is used to reload registers. 1b - The master RELOAD signal (from submodule 0) is used to reload registers. This setting should not be used in submodule 0 as it forces the RELOAD signal to logic 0.
1-0 CLK_SEL	Clock Source Select These bits determine the source of the clock signal for this submodule. 00b - The IPBus clock is used as the clock for the local prescaler and counter. 01b - EXT_CLK is used as the clock for the local prescaler and counter. 10b - Submodule 0's clock (AUX_CLK) is used as the source clock for the local prescaler and counter. This setting should not be used in submodule 0 as it forces the clock to logic 0. 11b - Reserved

73.5.5 Control Register (SM0CTRL - SM3CTRL)

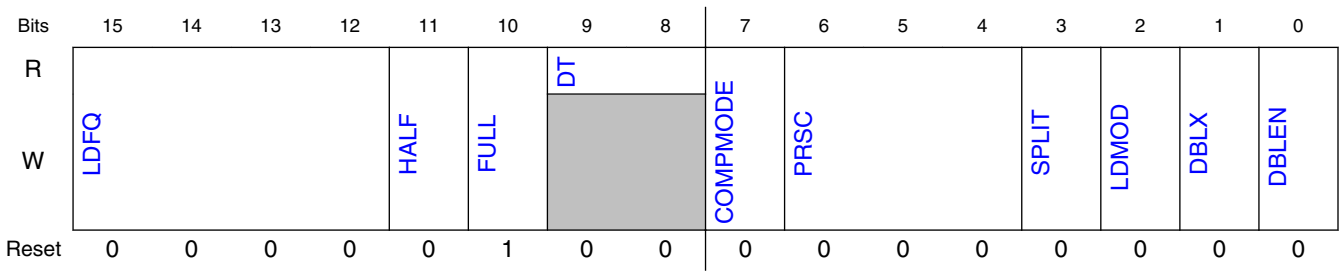
73.5.5.1 Offset

Register	Offset
SM0CTRL	6h
SM1CTRL	66h
SM2CTRL	C6h
SM3CTRL	126h

73.5.5.2 Function

Includes control settings for timing, loading, and buffering.

73.5.5.3 Diagram



73.5.5.4 Fields

Field	Function
15-12 LDFQ	Load Frequency These buffered bits select the PWM load frequency. Reset clears LDFQ, selecting loading every PWM opportunity. A PWM opportunity is determined by HALF and FULL. **Note:** LDFQ takes effect when the current load cycle is complete, regardless of the state of MCTRL[LDOK]. Reading LDFQ reads the buffered values and not necessarily the values currently in effect. 0000b - Every PWM opportunity 0001b - Every 2 PWM opportunities 0010b - Every 3 PWM opportunities 0011b - Every 4 PWM opportunities 0100b - Every 5 PWM opportunities 0101b - Every 6 PWM opportunities 0110b - Every 7 PWM opportunities 0111b - Every 8 PWM opportunities 1000b - Every 9 PWM opportunities 1001b - Every 10 PWM opportunities 1010b - Every 11 PWM opportunities 1011b - Every 12 PWM opportunities 1100b - Every 13 PWM opportunities 1101b - Every 14 PWM opportunities 1110b - Every 15 PWM opportunities 1111b - Every 16 PWM opportunities
11 HALF	Half Cycle Reload This bit enables half-cycle reloads. A half cycle is defined by when the submodule counter matches the VAL0 register and does not have to be halfway through the PWM cycle. 0b - Half-cycle reloads disabled. 1b - Half-cycle reloads enabled.
10 FULL	Full Cycle Reload This bit enables full-cycle reloads. A full cycle is defined by when the submodule counter matches the VAL1 register. Either CTRL[HALF] or CTRL[FULL] must be set to move the buffered data into the registers used by the PWM generators or CTRL[LDMOD] must be set. If both CTRL[HALF] and CTRL[FULL] are set, then reloads can occur twice per cycle. 0b - Full-cycle reloads disabled. 1b - Full-cycle reloads enabled.

Table continues on the next page...

Field	Function
9-8 DT	Deadtime These bits reflect the sampled values of the PWM_X input at the end of each deadtime. Sampling occurs at the end of deadtime 0 for DT[0] and the end of deadtime 1 for DT[1]. Reset clears these bits.
7 COMPMODE	Compare Mode This bit controls how comparisons are made between the VAL* registers and the PWM submodule counter. This bit can be written one time after which it requires a reset to release the bit for writing again. 0b - The VAL* registers and the PWM counter are compared using an "equal to" method. This means that PWM edges are only produced when the counter is equal to one of the VAL* register values. This implies that a PWM_A output that is high at the end of a period maintains this state until a match with VAL3 clears the output in the following period. 1b - The VAL* registers and the PWM counter are compared using an "equal to or greater than" method. This means that PWM edges are produced when the counter is equal to or greater than one of the VAL* register values. This implies that a PWM_A output that is high at the end of a period could go low at the start of the next period if the starting counter value is greater than (but not necessarily equal to) the new VAL3 value.
6-4 PRSC	Prescaler These buffered bits select the divide ratio of the PWM clock frequency selected by CTRL2[CLK_SEL]. **Note:** Reading CTRL[PRSC] reads the buffered values and not necessarily the values currently in effect. CTRL[PRSC] takes effect at the beginning of the next PWM cycle and only when the load okay bit MCTRL[LDOK] is set, or CTRL[LDMOD] is set. This field cannot be written when MCTRL[LDOK] is set. 000b - Prescaler 1. PWM clock frequency = f_{clk} 001b - Prescaler 2. PWM clock frequency = f_{clk} /2 010b - Prescaler 4. PWM clock frequency = f_{clk} /4 011b - Prescaler 8. PWM clock frequency = f_{clk} /8 100b - Prescaler 16. PWM clock frequency = f_{clk} /16 101b - Prescaler 32. PWM clock frequency = f_{clk} /32 110b - Prescaler 64. PWM clock frequency = f_{clk} /64 111b - Prescaler 128. PWM clock frequency = f_{clk} /128
3 SPLIT	Split the DBLPWM signal to PWM_A and PWM_B This bit is used in independent mode when DBLEN is set. This bit allows the two PWM pulses generated by DBLEN to be split with one pulse on PWM_A and one on PWM_B. The two pulses within the same PWM period are created by an XOR function of the PWM_A and PWM_B sources. The splitting function causes PWM_A to output the pulse that occurs when the PWM_A source is 1 and the PWM_B source is 0. The PWM_B output occurs when the PWM_B source is 1 and the PWM_A source is 0. (See Double switching PWMs .) 0b - DBLPWM is not split. PWM_A and PWM_B each have double pulses. 1b - DBLPWM is split to PWM_A and PWM_B.
2 LDMOD	Load Mode Select This bit selects the timing of loading the buffered registers for this submodule. 0b - Buffered registers of this submodule are loaded and take effect at the next PWM reload if MCTRL[LDOK] is set. 1b - Buffered registers of this submodule are loaded and take effect immediately upon MCTRL[LDOK] being set. In this case, it is not necessary to set CTRL[FULL] or CTRL[HALF].
1 DBLX	PWM_X Double Switching Enable This bit enables the double switching behavior on PWM_X. When this bit is set, the PWM_X output shall be the exclusive OR combination of PWM_A and PWM_B prior to polarity and masking considerations. 0b - PWM_X double pulse disabled. 1b - PWM_X double pulse enabled.

Table continues on the next page...

Field	Function
0 DBLEN	Double Switching Enable This bit enables the double switching PWM behavior (See Double switching PWMs). 0b - Double switching disabled. 1b - Double switching enabled.

73.5.6 Value Register 0 (SM0VAL0 - SM3VAL0)

73.5.6.1 Offset

Register	Offset
SM0VAL0	Ah
SM1VAL0	6Ah
SM2VAL0	CAh
SM3VAL0	12Ah

73.5.6.2 Function

Note

The VAL0 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL0 cannot be written when MCTRL[LDOK] is set. Reading VAL0 reads the value in a buffer. It is not necessarily the value that the PWM generator is currently using.

73.5.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.6.4 Fields

Field	Function
15-0 VAL0	Value 0 The 16-bit signed value in this buffered register defines the mid-cycle reload point for the PWM in PWM clock periods. This value also defines when the PWM_X signal is set and the local sync signal is reset. This register is not byte accessible. **Note:** The actual behavior takes effect when counter equal VAL0+1.

73.5.7 Fractional Value Register 1 (SM0FRACVAL1 - SM3FRACVAL1)

73.5.7.1 Offset

Register	Offset
SM0FRACVAL1	Ch
SM1FRACVAL1	6Ch
SM2FRACVAL1	CCh
SM3FRACVAL1	12Ch

73.5.7.2 Function

Note

The FRACVAL1 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRACVAL1 cannot be written when MCTRL[LDOK] is set. Reading FRACVAL1 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FRACVAL1					0										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.7.4 Fields

Field	Function
15-11 FRACVAL1	Fractional Value 1 These bits act as a fractional addition to the value in the VAL1 register which controls the PWM period width. The PWM period is computed in terms of IPBus clock cycles. This fractional portion is accumulated at the end of every cycle until an additional whole IPBus cycle is reached. At this time, the value being used for VAL1 is temporarily incremented and the PWM cycle is extended by one clock period to compensate for the accumulated fractional values.
10-0 —	Reserved

73.5.8 Value Register 1 (SM0VAL1 - SM3VAL1)

73.5.8.1 Offset

Register	Offset
SM0VAL1	Eh
SM1VAL1	6Eh
SM2VAL1	CEh
SM3VAL1	12Eh

73.5.8.2 Function

Note

The VAL1 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL1 cannot be written when MCTRL[LDOK] is set. Reading VAL1 reads the value in a buffer. It is not necessarily the value that the PWM generator is currently using.

When using FRACVAL1, limit the maximum value of VAL1 to 0xFFFE for unsigned applications or to 0x7FFE for signed applications, to avoid counter rollovers caused by accumulating the fractional period defined by FRACVAL1.

If the VAL1 register defines the timer period (Local Sync is selected as the counter initialization signal), a 100% duty cycle cannot be achieved on the PWM_X output. After the count reaches VAL1, the PWM_X output is low for a minimum of one count every cycle. When the Master Sync signal (only originated by the Local Sync from submodule 0) is used to control the timer period, the VAL1 register can be free for other functions such as PWM generation without the duty cycle limitation.

73.5.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.8.4 Fields

Field	Function
15-0 VAL1	Value 1

Field	Function
	The 16-bit signed value written to this buffered register defines the modulo count value (maximum count) for the submodule counter. When reaching this count value, the counter reloads itself with the contents of the INIT register and asserts the local sync signal while resetting PWM_X. This register is not byte accessible. **Note:** The actual behavior takes effect when the counter equals VAL1+1.

73.5.9 Fractional Value Register 2 (SM0FRACVAL2 - SM3FRACVAL2)

73.5.9.1 Offset

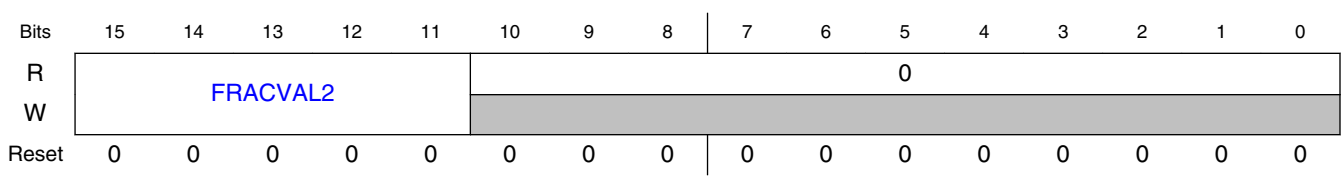
Register	Offset
SM0FRACVAL2	10h
SM1FRACVAL2	70h
SM2FRACVAL2	D0h
SM3FRACVAL2	130h

73.5.9.2 Function

Note

The FRACVAL2 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRACVAL2 cannot be written when MCTRL[LDOK] is set. Reading FRACVAL2 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.9.3 Diagram



73.5.9.4 Fields

Field	Function
15-11 FRACVAL2	Fractional Value 2 These bits act as a fractional addition to the value in the VAL2 register which controls the PWM_A turn on timing. It is also used to control the fractional addition to the turn off delay of PWM_B when MCTRL[IPOLx]=0 in complementary mode and CTRL2[INDEP]=0. FRCTRL[FRAC23_EN] must be set to 0 when the values of VAL2 and VAL3 cause the high or low time of the PWM output is 3 cycles or less.
10-0 —	Reserved

73.5.10 Value Register 2 (SM0VAL2 - SM3VAL2)

73.5.10.1 Offset

Register	Offset
SM0VAL2	12h
SM1VAL2	72h
SM2VAL2	D2h
SM3VAL2	132h

73.5.10.2 Function

Note

The VAL2 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL2 cannot be written when MCTRL[LDOK] is set. Reading VAL2 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.10.4 Fields

Field	Function
15-0 VAL2	Value 2 The 16-bit signed value in this buffered register defines the count value to set PWM23 high. This register is not byte accessible. **Note:** The actual behavior takes effect when the counter equals VAL2+1.

73.5.11 Fractional Value Register 3 (SM0FRACVAL3 - SM3FRACVAL3)

73.5.11.1 Offset

Register	Offset
SM0FRACVAL3	14h
SM1FRACVAL3	74h
SM2FRACVAL3	D4h
SM3FRACVAL3	134h

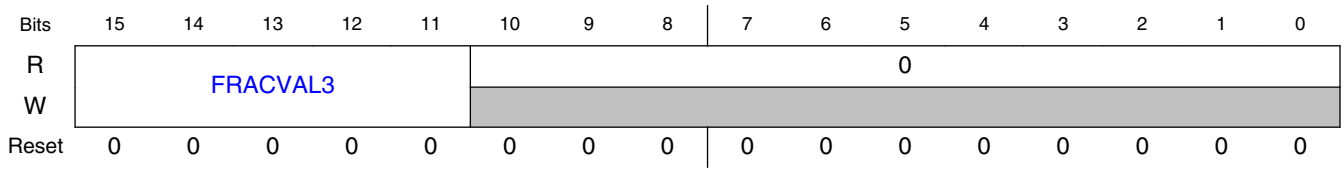
73.5.11.2 Function

Note

The FRACVAL3 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRACVAL3 cannot be written when MCTRL[LDOK] is set. Reading

FRACVAL3 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.11.3 Diagram



73.5.11.4 Fields

Field	Function
15-11 FRACVAL3	Fractional Value 3 These bits act as a fractional addition to the value in the VAL3 register which controls the PWM_A turn off timing. It is also used to control the fractional addition to the turn on delay of PWM_B when MCTRL[IPOLx]=0 in complementary mode and CTRL2[INDEP]=0. **Note:** FRCTRL[FRAC23_EN] must be set to 0 when the values of VAL2 and VAL3 cause the high or low time of the PWM output is 3 cycles or less.
10-0 —	Reserved

73.5.12 Value Register 3 (SM0VAL3 - SM3VAL3)

73.5.12.1 Offset

Register	Offset
SM0VAL3	16h
SM1VAL3	76h
SM2VAL3	D6h
SM3VAL3	136h

73.5.12.2 Function

Note

The VAL3 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL3 cannot be written when MCTRL[LDOK] is set. Reading VAL3 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.12.4 Fields

Field	Function
15-0 VAL3	Value 3 The 16-bit signed value in this buffered register defines the count value to set PWM23 low. This register is not byte accessible. **Note:** The actual behavior takes effect when the counter equals VAL3+1.

73.5.13 Fractional Value Register 4 (SM0FRACVAL4 - SM3FRACVAL4)

73.5.13.1 Offset

Register	Offset
SM0FRACVAL4	18h
SM1FRACVAL4	78h
SM2FRACVAL4	D8h
SM3FRACVAL4	138h

73.5.13.2 Function

Note

The FRACVAL4 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRACVAL4 cannot be written when MCTRL[LDOK] is set. Reading FRACVAL4 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FRACVAL4								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.13.4 Fields

Field	Function
15-11 FRACVAL4	Fractional Value 4 These bits act as a fractional addition to the value in the VAL4 register which controls the PWM_B turn on timing. It is also used to control the fractional addition to the turn off delay of PWM_A when MCTRL[IPOLx]=1 in complementary mode and CTRL2[INDEP]=0. **Note:** FRCTRL[FRAC45_EN] must be set to 0 when the values of VAL4 and VAL5 cause the high or low time of the PWM output is 3 cycles or less.
10-0 —	Reserved

73.5.14 Value Register 4 (SM0VAL4 - SM3VAL4)

73.5.14.1 Offset

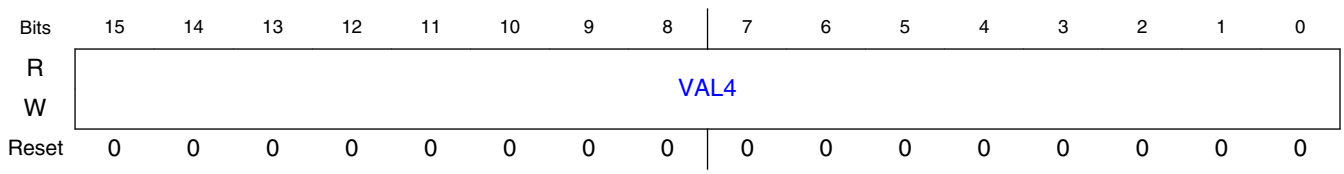
Register	Offset
SM0VAL4	1Ah
SM1VAL4	7Ah
SM2VAL4	DAh
SM3VAL4	13Ah

73.5.14.2 Function

Note

The VAL4 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL4 cannot be written when MCTRL[LDOK] is set. Reading VAL4 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.14.3 Diagram



73.5.14.4 Fields

Field	Function
15-0 VAL4	Value 4 The 16-bit signed value in this buffered register defines the count value to set PWM45 high. This register is not byte accessible. **Note:** The actual behavior takes effect when the counter equals VAL4+1.

73.5.15 Fractional Value Register 5 (SM0FRACVAL5 - SM3FRACVAL5)

73.5.15.1 Offset

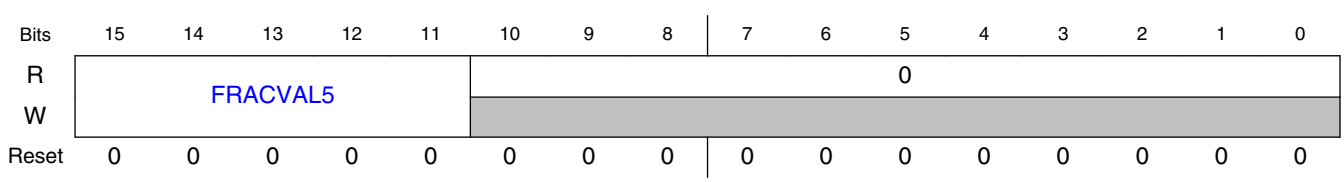
Register	Offset
SM0FRACVAL5	1Ch
SM1FRACVAL5	7Ch
SM2FRACVAL5	DCh
SM3FRACVAL5	13Ch

73.5.15.2 Function

Note

The FRACVAL5 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRACVAL5 cannot be written when MCTRL[LDOK] is set. Reading FRACVAL5 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.15.3 Diagram



73.5.15.4 Fields

Field	Function
15-11 FRACVAL5	Fractional Value 5 These bits act as a fractional addition to the value in the VAL5 register which controls the PWM_B turn off timing. It is also used to control the fractional addition to the turn on delay of PWM_A when MCTRL[IPOLx]=1 in complementary mode and CTRL2[INDEP]=0.

Table continues on the next page...

Field	Function
	Note: FRCTRL[FRAC45_EN] must be set to 0 when the values of VAL4 and VAL5 cause the high or low time of the PWM output is 3 cycles or less.
10-0 —	Reserved

73.5.16 Value Register 5 (SM0VAL5 - SM3VAL5)

73.5.16.1 Offset

Register	Offset
SM0VAL5	1Eh
SM1VAL5	7Eh
SM2VAL5	DEh
SM3VAL5	13Eh

73.5.16.2 Function

Note

The VAL5 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL5 cannot be written when MCTRL[LDOK] is set. Reading VAL5 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.16.4 Fields

Field	Function
15-0 VAL5	Value 5 The 16-bit signed value in this buffered register defines the count value to set PWM45 low. This register is not byte accessible. **Note:** The actual behavior takes effect when the counter equals VAL5+1.

73.5.17 Fractional Control Register (SM0FRCTRL - SM3FRCTRL)

73.5.17.1 Offset

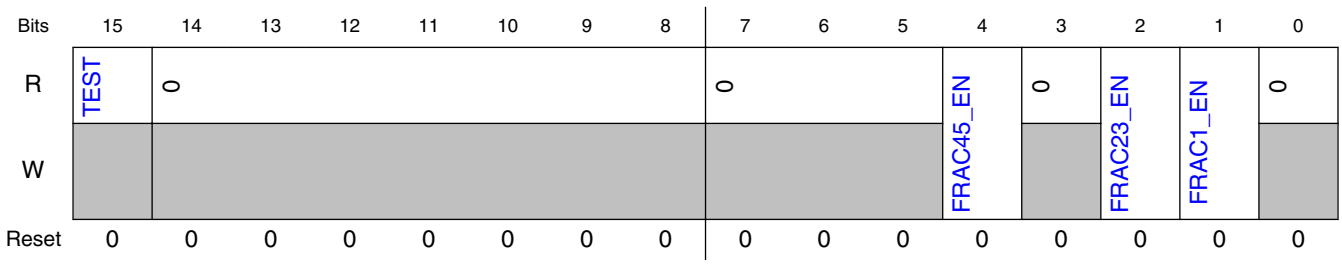
Register	Offset
SM0FRCTRL	20h
SM1FRCTRL	80h
SM2FRCTRL	E0h
SM3FRCTRL	140h

73.5.17.2 Function

Note

The FRAC1_EN bit is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRAC1_EN cannot be written when MCTRL[LDOK] is set. Reading FRAC1_EN reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

73.5.17.3 Diagram



73.5.17.4 Fields

Field	Function
15 TEST	Test Status Bit This is a test bit for factory use. This bit resets to 0 but may be either 0 or 1 during PWM operation.
14-8 —	Reserved
7-5 —	Reserved
4 FRAC45_EN	Fractional Cycle Placement Enable for PWM_B This bit is used to enable the fractional cycle edge placement of PWM_B using the FRACVAL4 and FRACVAL5 registers. When disabled, bypass the fractional cycle edge placement of PWM_B. **NOTE:** The FRAC45_EN bit is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRAC45_EN cannot be written when MCTRL[LDOK] is set. Reading FRAC45_EN reads the value in a buffer and not necessarily the value that the PWM generator is currently using. 0b - Disable fractional cycle placement for PWM_B. 1b - Enable fractional cycle placement for PWM_B.
3 —	Reserved
2 FRAC23_EN	Fractional Cycle Placement Enable for PWM_A This bit is used to enable the fractional cycle edge placement of PWM_A using the FRACVAL2 and FRACVAL3 registers. When disabled, bypass the fractional cycle edge placement of PWM_A. **NOTE:** The FRAC23_EN bit is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRAC23_EN cannot be written when MCTRL[LDOK] is set. Reading FRAC23_EN reads the value in a buffer and not necessarily the value that the PWM generator is currently using. 0b - Disable fractional cycle placement for PWM_A. 1b - Enable fractional cycle placement for PWM_A.
1 FRAC1_EN	Fractional Cycle PWM Period Enable This bit is used to enable the fractional cycle length of the PWM period using the FRACVAL1 register. When disabled, bypass the fractional cycle length of the PWM period.

Table continues on the next page...

Field	Function
	NOTE: The FRAC1_EN bit is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. FRAC1_EN cannot be written when MCTRL[LDOK] is set. Reading FRAC1_EN reads the value in a buffer and not necessarily the value that the PWM generator is currently using. 0b - Disable fractional cycle length for the PWM period. 1b - Enable fractional cycle length for the PWM period.
0 —	Reserved

73.5.18 Output Control Register (SM0OCTRL - SM3OCTRL)

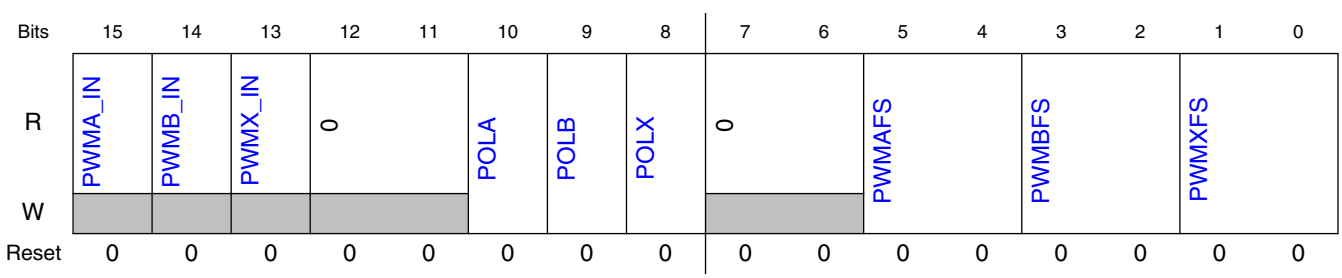
73.5.18.1 Offset

Register	Offset
SM0OCTRL	22h
SM1OCTRL	82h
SM2OCTRL	E2h
SM3OCTRL	142h

73.5.18.2 Function

Contains output controls for fault states.

73.5.18.3 Diagram



73.5.18.4 Fields

Field	Function
15 PWMA_IN	PWM_A Input This bit shows the logic value currently being driven into the PWM_A input. The reset state is undefined.
14 PWMB_IN	PWM_B Input This bit shows the logic value currently being driven into the PWM_B input. The reset state is undefined.
13 PWMX_IN	PWM_X Input This bit shows the logic value currently being driven into the PWM_X input. The reset state is undefined.
12-11 —	Reserved
10 POLA	PWM_A Output Polarity This bit inverts the PWM_A output polarity. 0b - PWM_A output not inverted. A high level on the PWM_A pin represents the "on" or "active" state. 1b - PWM_A output inverted. A low level on the PWM_A pin represents the "on" or "active" state.
9 POLB	PWM_B Output Polarity This bit inverts the PWM_B output polarity. 0b - PWM_B output not inverted. A high level on the PWM_B pin represents the "on" or "active" state. 1b - PWM_B output inverted. A low level on the PWM_B pin represents the "on" or "active" state.
8 POLX	PWM_X Output Polarity This bit inverts the PWM_X output polarity. 0b - PWM_X output not inverted. A high level on the PWM_X pin represents the "on" or "active" state. 1b - PWM_X output inverted. A low level on the PWM_X pin represents the "on" or "active" state.
7-6 —	Reserved
5-4 PWMAFS	PWM_A Fault State These bits determine the fault state for the PWM_A output during fault conditions and Stop mode. It may also define the output state during Wait and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.
3-2 PWMBFS	PWM_B Fault State These bits determine the fault state for the PWM_B output during fault conditions and Stop mode. It may also define the output state during Wait and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.
1-0 PWMXFS	PWM_X Fault State

Field	Function
	These bits determine the fault state for the PWM_X output during fault conditions and Stop mode. It may also define the output state during Wait and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.

73.5.19 Status Register (SM0STS - SM3STS)

73.5.19.1 Offset

Register	Offset
SM0STS	24h
SM1STS	84h
SM2STS	E4h
SM3STS	144h

73.5.19.2 Function

Contains Compare and Capture flag status.

73.5.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RUF	REF	RF	CFA1	CFA0	CFB1	CFB0	CFX1	CFX0	CMPF					
W			W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.19.4 Fields

Field	Function
15	Reserved

Table continues on the next page...

Field	Function
—	
14 RUF	Registers Updated Flag This bit is set when one of the INIT, VALx, FRACVALx, or CTRL[PRSC] registers has been written, which indicates potentially non-coherent data in the set of double buffered registers. Clear this bit by a proper reload sequence consisting of a reload signal while MCTRL[LDOK] = 1. Reset clears this bit. 0b - No register update has occurred since last reload. 1b - At least one of the double buffered registers has been updated since the last reload.
13 REF	Reload Error Flag This bit is set when a reload cycle occurs while MCTRL[LDOK] is 0 and the double buffered registers are in a non-coherent state (STS[RUF] = 1). Clear this bit by writing a logic one to this location. Reset clears this bit. 0b - No reload error occurred. 1b - Reload signal occurred with non-coherent data and MCTRL[LDOK] = 0.
12 RF	Reload Flag This bit is set at the beginning of every reload cycle regardless of the state of MCTRL[LDOK]. Clear this bit by writing a logic one to this location when DMAEN[VALDE] is clear (non-DMA mode). This bit can also be cleared by the DMA done signal when DMAEN[VALDE] is set (DMA mode) . Reset clears this bit. 0b - No new reload cycle since last STS[RF] clearing 1b - New reload cycle since last STS[RF] clearing
11 CFA1	Capture Flag A1 This bit is set when a capture event occurs on the Capture A1 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CA1DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CA1DE] is set (DMA mode) . Reset clears this bit.
10 CFA0	Capture Flag A0 This bit is set when a capture event occurs on the Capture A0 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CA0DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CA0DE] is set (DMA mode) . Reset clears this bit.
9 CFB1	Capture Flag B1 This bit is set when a capture event occurs on the Capture B1 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CB1DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CB1DE] is set (DMA mode) . Reset clears this bit.
8 CFB0	Capture Flag B0 This bit is set when a capture event occurs on the Capture B0 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CB0DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CB0DE] is set (DMA mode) . Reset clears this bit.
7 CFX1	Capture Flag X1 This bit is set when a capture event occurs on the Capture X1 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CX1DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CX1DE] is set (DMA mode) . Reset clears this bit.
6 CFX0	Capture Flag X0 This bit is set when a capture event occurs on the Capture X0 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CX0DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CX0DE] is set (DMA mode) . Reset clears this bit.
5-0 CMPF	Compare Flags These bits are set when the submodule counter value matches the value of one of the VALx registers. Clear these bits by writing a 1 to a bit position.

Field	Function
	00_0000b - No compare event has occurred for a particular VALx value. 00_0001b - A compare event has occurred for a particular VALx value.

73.5.20 Interrupt Enable Register (SM0INTEN - SM3INTEN)

73.5.20.1 Offset

Register	Offset
SM0INTEN	26h
SM1INTEN	86h
SM2INTEN	E6h
SM3INTEN	146h

73.5.20.2 Function

Contains Compare and Capture interrupt enables.

73.5.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		REIE	RIE	CA1IE	CA0IE	CB1IE	CB0IE	CX1IE	CX0IE	CMPIE					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.20.4 Fields

Field	Function
15-14 —	Reserved
13 REIE	Reload Error Interrupt Enable This bit enables the reload error flag, STS[REF], to generate CPU interrupt requests. Reset clears this bit.

Table continues on the next page...

Field	Function
	0b - STS[REF] CPU interrupt requests disabled 1b - STS[REF] CPU interrupt requests enabled
12 RIE	Reload Interrupt Enable This bit enables the reload flag, STS[RF], to generate CPU interrupt requests. Reset clears this bit. 0b - STS[RF] CPU interrupt requests disabled 1b - STS[RF] CPU interrupt requests enabled
11 CA1IE	Capture A 1 Interrupt Enable This bit allows the STS[CFA1] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CA1DE]. 0b - Interrupt request disabled for STS[CFA1] 1b - Interrupt request enabled for STS[CFA1]
10 CA0IE	Capture A 0 Interrupt Enable This bit allows the STS[CFA0] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CA0DE]. 0b - Interrupt request disabled for STS[CFA0]. 1b - Interrupt request enabled for STS[CFA0].
9 CB1IE	Capture B 1 Interrupt Enable This bit allows the STS[CFB1] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CB1DE]. 0b - Interrupt request disabled for STS[CFB1]. 1b - Interrupt request enabled for STS[CFB1].
8 CB0IE	Capture B 0 Interrupt Enable This bit allows the STS[CFB0] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CB0DE]. 0b - Interrupt request disabled for STS[CFB0]. 1b - Interrupt request enabled for STS[CFB0].
7 CX1IE	Capture X 1 Interrupt Enable This bit allows the STS[CFX1] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CX1DE]. 0b - Interrupt request disabled for STS[CFX1]. 1b - Interrupt request enabled for STS[CFX1].
6 CX0IE	Capture X 0 Interrupt Enable This bit allows the STS[CFX0] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CX0DE]. 0b - Interrupt request disabled for STS[CFX0]. 1b - Interrupt request enabled for STS[CFX0].
5-0 CMPIE	Compare Interrupt Enables These bits enable the STS[CMPF] flags to cause a compare interrupt request to the CPU. 00_0000b - The corresponding STS[CMPF] bit will not cause an interrupt request. 00_0001b - The corresponding STS[CMPF] bit will cause an interrupt request.

73.5.21 DMA Enable Register (SM0DMAEN - SM3DMAEN)

73.5.21.1 Offset

Register	Offset
SM0DMAEN	28h
SM1DMAEN	88h
SM2DMAEN	E8h
SM3DMAEN	148h

73.5.21.2 Function

Contains controls for DMA.

73.5.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						VALDE	FAND	CAPTDE		CA1DE	CA0DE	CB1DE	CB0DE	CX1DE	CX0DE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.21.4 Fields

Field	Function
15-10 —	Reserved
9 VALDE	Value Registers DMA Enable This bit enables DMA write requests for the VALx and FRACVALx registers when STS[RF] is set. Reset clears this bit. 0b - DMA write requests disabled 1b - Enabled. DMA write requests for the VALx and FRACVALx registers enabled
8 FAND	FIFO Watermark AND Control This bit works in conjunction with the DMAEN[CAPTDE] field when it is set to watermark mode (DMAEN[CAPTDE] = 01). While DMAEN[CAxDE], DMAEN[CBxDE], and DMAEN[CXxDE] determine which FIFO watermarks the DMA read request is sensitive to. This bit determines if the selected watermarks are AND'ed together or OR'ed together to create the request.

Table continues on the next page...

Field	Function
	0b - Selected FIFO watermarks are OR'ed together. 1b - Selected FIFO watermarks are AND'ed together.
7-6 CAPTDE	Capture DMA Enable Source Select These bits select the source of enabling the DMA read requests for the capture FIFOs. Reset clears these bits. 00b - Read DMA requests disabled. 01b - Exceeding a FIFO watermark sets the DMA read request. This requires at least one of DMAEN[CA1DE], DMAEN[CA0DE], DMAEN[CB1DE], DMAEN[CB0DE], DMAEN[CX1DE], or DMAEN[CX0DE] to be set to determine which watermark(s) the DMA request is sensitive. 10b - A local synchronization (VAL1 matches counter) sets the read DMA request. 11b - A local reload (STS[RF] being set) sets the read DMA request.
5 CA1DE	Capture A1 FIFO DMA Enable This bit enables DMA read requests for the Capture A1 FIFO data when STS[CFA1] is set. Reset clears this bit. Do not set this bit and INTEN[CA1IE].
4 CA0DE	Capture A0 FIFO DMA Enable This bit enables DMA read requests for the Capture A0 FIFO data when STS[CFA0] is set. Reset clears this bit. Do not set both this bit and INTEN[CA0IE].
3 CB1DE	Capture B1 FIFO DMA Enable This bit enables DMA read requests for the Capture B1 FIFO data when STS[CFB1] is set. Reset clears this bit. Do not set both this bit and INTEN[CB1IE].
2 CB0DE	Capture B0 FIFO DMA Enable This bit enables DMA read requests for the Capture B0 FIFO data when STS[CFB0] is set. Reset clears this bit. Do not set both this bit and INTEN[CB0IE].
1 CX1DE	Capture X1 FIFO DMA Enable This bit enables DMA read requests for the Capture X1 FIFO data when STS[CFX1] is set. Reset clears this bit. Do not set both this bit and INTEN[CX1IE].
0 CX0DE	Capture X0 FIFO DMA Enable This bit enables DMA read requests for the Capture X0 FIFO data when STS[CFX0] is set. Reset clears this bit. Do not set both this bit and INTEN[CX0IE].

73.5.22 Output Trigger Control Register (SM0TCTRL - SM3TCTRL)

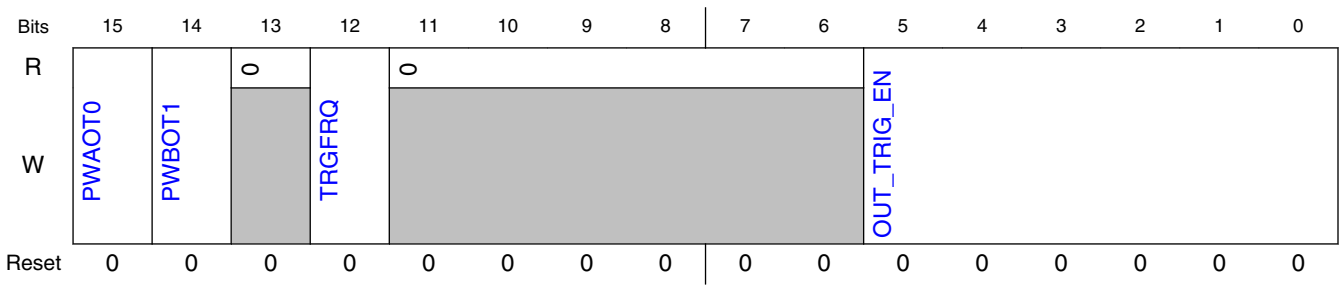
73.5.22.1 Offset

Register	Offset
SM0TCTRL	2Ah
SM1TCTRL	8Ah
SM2TCTRL	EAh
SM3TCTRL	14Ah

73.5.22.2 Function

Contains trigger controls.

73.5.22.3 Diagram



73.5.22.4 Fields

Field	Function
15 PWAOT0	Output Trigger 0 Source Select This bit selects which signal to bring out on the PWM's PWM_OUT_TRIG0 port. The output trigger port is often connected to routing logic on the chip. This control bit allows the PWM_A output signal to be driven onto the output trigger port so it can be sent to the chip routing logic. 0b - Route the PWM_OUT_TRIG0 signal to PWM_OUT_TRIG0 port. 1b - Route the PWM_A output to the PWM_OUT_TRIG0 port.
14 PWBOT1	Output Trigger 1 Source Select This bit selects which signal to bring out on the PWM's PWM_OUT_TRIG1 port. The output trigger port is often connected to routing logic on the chip. This control bit allows the PWM_B output signal to be driven onto the output trigger port so it can be sent to the chip routing logic. 0b - Route the PWM_OUT_TRIG1 signal to PWM_OUT_TRIG1 port. 1b - Route the PWM_B output to the PWM_OUT_TRIG1 port.
13 —	Reserved
12 TRGFRQ	Trigger Frequency This bit allows control over the frequency of the trigger outputs when using non-zero values of CTRL[LDFQ]. 0b - Trigger outputs are generated during every PWM period even if the PWM is not reloaded every period due to CTRL[LDFQ] being non-zero. 1b - Trigger outputs are generated only during the final PWM period prior to a reload opportunity when the PWM is not reloaded every period due to CTRL[LDFQ] being non-zero.
11-6 —	Reserved

Table continues on the next page...

Field	Function
5-0 OUT_TRIG_EN	Output Trigger Enables These bits enable the generation of PWM_OUT_TRIG0 and PWM_OUT_TRIG1 outputs based on the counter value matching the value in one or more of the VAL0-5 registers. **NOTE:** Due to delays in creating the PWM outputs, the output trigger signals lead the PWM output edges by 2-3 clock cycles depending on the fractional cycle value being used. 1x_xxxxb - PWM_OUT_TRIG1 will set when the counter value matches the VAL5 value. x1_xxxxb - PWM_OUT_TRIG0 will set when the counter value matches the VAL4 value. xx_1xxxb - PWM_OUT_TRIG1 will set when the counter value matches the VAL3 value. xx_x1xxb - PWM_OUT_TRIG0 will set when the counter value matches the VAL2 value. xx_xx1xb - PWM_OUT_TRIG1 will set when the counter value matches the VAL1 value. xx_xxx1b - PWM_OUT_TRIG0 will set when the counter value matches the VAL0 value.

73.5.23 Fault Disable Mapping Register 0 (SM0DISMAP0 - SM3DISMAP0)

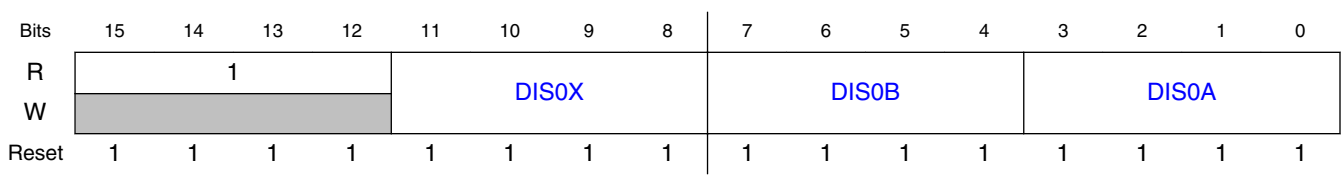
73.5.23.1 Offset

Register	Offset
SM0DISMAP0	2Ch
SM1DISMAP0	8Ch
SM2DISMAP0	ECh
SM3DISMAP0	14Ch

73.5.23.2 Function

This register determines which PWM pins are disabled by the fault protection inputs. Reset sets all of the bits in the fault disable mapping register.

73.5.23.3 Diagram



73.5.23.4 Fields

Field	Function
15-12 —	Reserved
11-8 DIS0X	PWM_X Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_X output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0X[0] is associated with FAULT0. DIS0X[1] is associated with FAULT1. DIS0X[2] is associated with FAULT2. DIS0X[3] is associated with FAULT3. A reset sets all bits in this field.
7-4 DIS0B	PWM_B Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_B output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0B[0] is associated with FAULT0. DIS0B[1] is associated with FAULT1. DIS0B[2] is associated with FAULT2. DIS0B[3] is associated with FAULT3. A reset sets all bits in this field.
3-0 DIS0A	PWM_A Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_A output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0A[0] is associated with FAULT0. DIS0A[1] is associated with FAULT1. DIS0A[2] is associated with FAULT2. DIS0A[3] is associated with FAULT3. A reset sets all bits in this field.

73.5.24 Deadtime Count Register 0 (SM0DTCNT0 - SM3DTCNT0)

73.5.24.1 Offset

Register	Offset
SM0DTCNT0	30h
SM1DTCNT0	90h
SM2DTCNT0	F0h
SM3DTCNT0	150h

73.5.24.2 Function

Deadtime operation applies only to complementary channel operation. The values written to the DTCNTx registers are in terms of IPBus clock cycles regardless of the setting of CTRL[PRSC] and/or CTRL2[CLK_SEL]. Reset sets the deadtime count registers to a default value of 0x07FF, selecting a deadtime of 2047 IPBus clock cycles. The DTCNTx registers are not byte accessible.

73.5.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DTCNT0															
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

73.5.24.4 Fields

Field	Function
15-0 DTCNT0	DTCNT0 The DTCNT0 field is interpreted differently depending on whether or not the fractional delays are being used (Set FRCTRL[FRAC23_EN] to 1 when MCTRL[IPOL]=0, or set FRCTRL[FRAC45_EN] to 1 when MCTRL[IPOL]=1). If the fractional delays are off, then the upper 5 bits of DTCNT0 are ignored and the remaining 11 bits are used to specify the number of cycles of deadtime. In this case, the maximum value is 0x07FF which indicates 2047 cycles of deadtime. If the fractional delays are being used, then the upper 11 bits of DTCNT0 represent the number of whole cycles of deadtime and the lower 5 bits of each register represent the fractional cycle added to the whole number. In this case, the maximum value is 0xFFFF which represents 2047 31/32 cycles of deadtime. The DTCNT0 field is used to control the deadtime during 0 to 1 transitions of the PWM_A output (assuming normal polarity).

73.5.25 Deadtime Count Register 1 (SM0DTCNT1 - SM3DTCNT1)

73.5.25.1 Offset

Register	Offset
SM0DTCNT1	32h

Table continues on the next page...

Register	Offset
SM1DTCNT1	92h
SM2DTCNT1	F2h
SM3DTCNT1	152h

73.5.25.2 Function

Deadtime operation applies only to complementary channel operation. The values written to the DTCNTx registers are in terms of IPBus clock cycles regardless of the setting of CTRL[PRSC] and/or CTRL2[CLK_SEL]. Reset sets the deadtime count registers to a default value of 0x07FF, selecting a deadtime of 2047 IPBus clock cycles. The DTCNTx registers are not byte accessible.

73.5.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DTCNT1															
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

73.5.25.4 Fields

Field	Function
15-0 DTCNT1	DTCNT1 The DTCNT1 field is interpreted differently depending on whether or not the fractional delays are being used (Set FRCTRL[FRAC23_EN] to 1 when MCTRL[IPOL]=0, or set FRCTRL[FRAC45_EN] to 1 when MCTRL[IPOL]=1). If the fractional delays are off, then the upper 5 bits of DTCNT1 are ignored and the remaining 11 bits are used to specify the number of cycles of deadtime. In this case, the maximum value is 0x07FF which indicates 2047 cycles of deadtime. If the fractional delays are being used, then the upper 11 bits of DTCNT1 represent the number of whole cycles of deadtime and the lower 5 bits of each register represent the fractional cycle added to the whole number. In this case, the maximum value is 0xFFFF which represents 2047 31/32 cycles of deadtime. The DTCNT1 field is used to control the deadtime during 0 to 1 transitions of the PWM_B output (assuming normal polarity).

73.5.26 Capture Control A Register (SM0CAPTCTRLA)

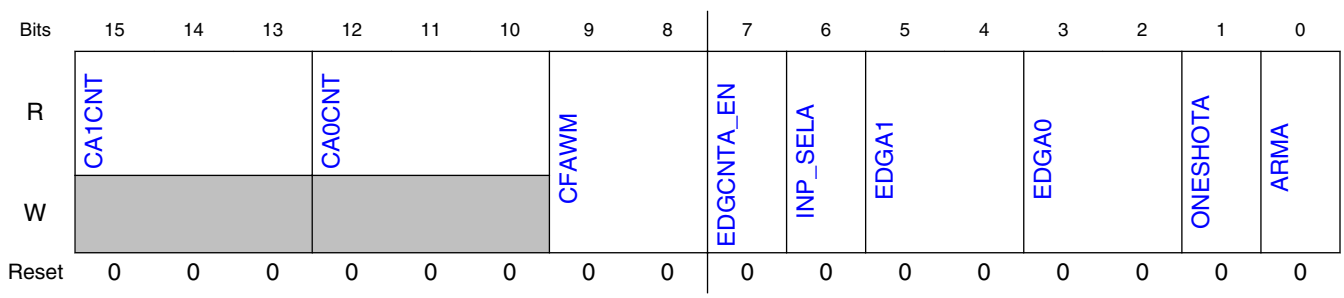
73.5.26.1 Offset

Register	Offset
SM0CAPTCTRLA	34h

73.5.26.2 Function

Contains capture controls for mode A.

73.5.26.3 Diagram



73.5.26.4 Fields

Field	Function
15-13 CA1CNT	Capture A1 FIFO Word Count This field reflects the number of words in the Capture A1 FIFO. (FIFO depth is 1.)
12-10 CA0CNT	Capture A0 FIFO Word Count This field reflects the number of words in the Capture A0 FIFO. (FIFO depth is 1.)
9-8 CFAWM	Capture A FIFOs Water Mark This field represents the water mark level for capture A FIFOs. The capture flags, STS[CFA1] and STS[CFA0], are not set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTA_EN	Edge Counter A Enable This field enables the edge counter which counts rising and falling edges on the PWM_A input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled

Table continues on the next page...

Field	Function
6 INP_SELA	Input Select A This field selects between the raw PWM_A input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_A input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLA[EDGA0] and CAPTCTRLA[EDGA1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRA[EDGA0] and/or CAPTCTRLA[EDGA1] fields to enable one or both of the capture registers.
5-4 EDGA1	Edge A 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGA0	Edge A 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTA	One Shot Mode A This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLA[ARMA] is cleared. No further captures are performed until CAPTCTRLA[ARMA] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLA[ARMA] is then cleared.
0 ARMA	Arm A Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLA[EDGAx] is enabled.

73.5.27 Capture Compare A Register (SM0CAPTCOMPA)

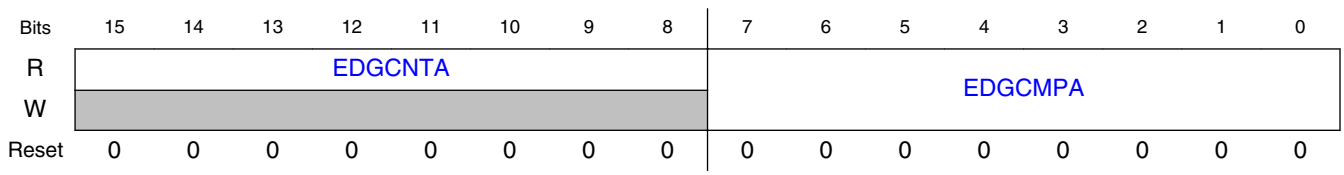
73.5.27.1 Offset

Register	Offset
SM0CAPTCOMPA	36h

73.5.27.2 Function

Contains capture and compare values for mode A.

73.5.27.3 Diagram



73.5.27.4 Fields

Field	Function
15-8 EDGCNTA	Edge Counter A This field contains the edge counter value for the PWM_A input capture circuitry.
7-0 EDGCMPA	Edge Compare A This field is the compare value associated with the edge counter for the PWM_A input capture circuitry.

73.5.28 Capture Control B Register (SM0CAPTCTRLB)

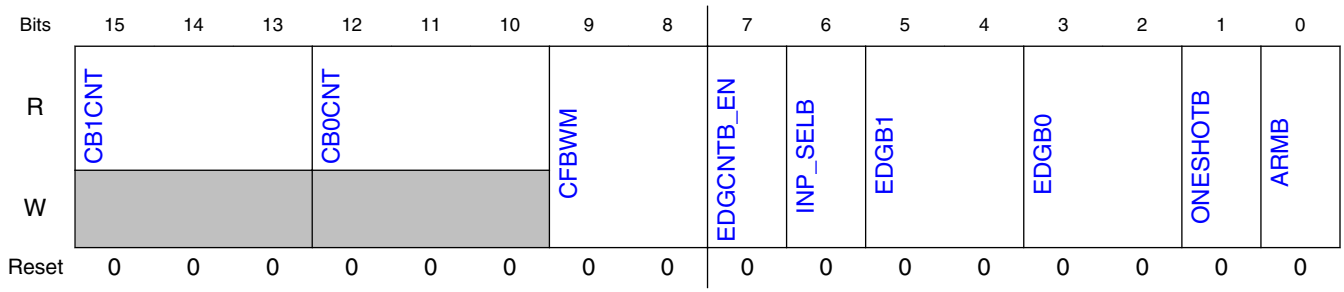
73.5.28.1 Offset

Register	Offset
SM0CAPTCTRLB	38h

73.5.28.2 Function

Contains capture controls for mode B.

73.5.28.3 Diagram



73.5.28.4 Fields

Field	Function
15-13 CB1CNT	Capture B1 FIFO Word Count This field reflects the number of words in the Capture B1 FIFO. (FIFO depth is 1.)
12-10 CB0CNT	Capture B0 FIFO Word Count This field reflects the number of words in the Capture B0 FIFO. (FIFO depth is 1.)
9-8 CFBWM	Capture B FIFOs Water Mark This field represents the water mark level for capture B FIFOs. The capture flags, STS[CFB1] and STS[CFB0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTB_EN	Edge Counter B Enable This field enables the edge counter which counts rising and falling edges on the PWM_B input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELB	Input Select B This field selects between the raw PWM_B input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_B input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLB[EDGB0] and CAPTCTRLB[EDGB1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRB[EDGB0] and/or CAPTCTRLB[EDGB1] fields to enable one or both of the capture registers.
5-4 EDGB1	Edge B 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGB0	Edge B 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTB	One Shot Mode B This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLB[ARMB] is cleared. No further captures are performed until CAPTCTRLB[ARMB] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLB[ARMB] is then cleared.
0 ARMB	Arm B Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLB[EDGBx] is enabled.

73.5.29 Capture Compare B Register (SM0CAPTCOMPB)

73.5.29.1 Offset

Register	Offset
SM0CAPTCOMPB	3Ah

73.5.29.2 Function

Contains capture and compare values for mode B.

73.5.29.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTB								EDGCMPB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.29.4 Fields

Field	Function
15-8 EDGCNTB	Edge Counter B This field contains the edge counter value for the PWM_B input capture circuitry.
7-0 EDGCMPB	Edge Compare B This field is the compare value associated with the edge counter for the PWM_B input capture circuitry.

73.5.30 Capture Control X Register (SM0CAPTCTRLX)

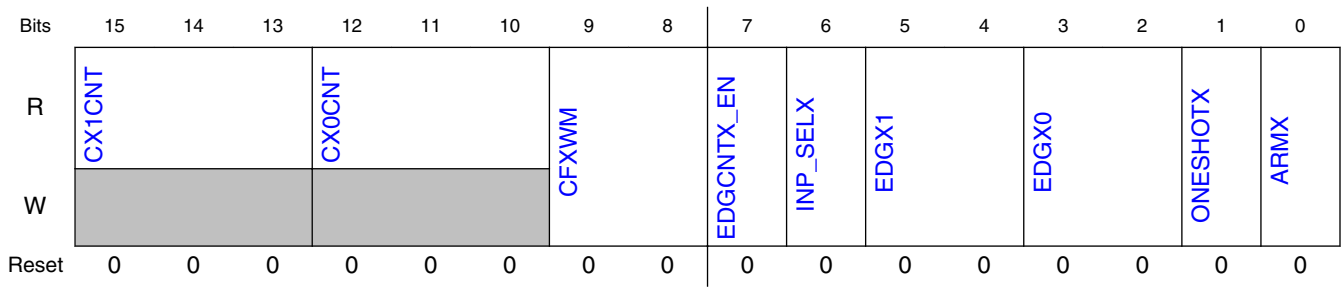
73.5.30.1 Offset

Register	Offset
SM0CAPTCTRLX	3Ch

73.5.30.2 Function

Contains capture controls for mode X.

73.5.30.3 Diagram



73.5.30.4 Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

73.5.31 Capture Compare X Register (SM0CAPTCOMPX)

73.5.31.1 Offset

Register	Offset
SM0CAPTCOMPX	3Eh

73.5.31.2 Function

Contains capture and control values for mode X.

73.5.31.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTX								EDGCMPX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.31.4 Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

73.5.32 Capture Value 0 Register (SM0CVAL0)

73.5.32.1 Offset

Register	Offset
SM0CVAL0	40h

73.5.32.2 Function

Writing this register generates bus transfer error.

73.5.32.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.32.4 Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.33 Capture Value 0 Cycle Register (SM0CVAL0CYC)

73.5.33.1 Offset

Register	Offset
SM0CVAL0CYC	42h

73.5.33.2 Function

Writing this register generates bus transfer error.

73.5.33.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL0CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.33.4 Fields

Field	Function
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.34 Capture Value 1 Register (SM0CVAL1)

73.5.34.1 Offset

Register	Offset
SM0CVAL1	44h

73.5.34.2 Function

Writing this register generates bus transfer error.

73.5.34.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.34.4 Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.35 Capture Value 1 Cycle Register (SM0CVAL1CYC)

73.5.35.1 Offset

Register	Offset
SM0CVAL1CYC	46h

73.5.35.2 Function

Writing this register generates bus transfer error.

73.5.35.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.35.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

73.5.36 Capture Value 2 Register (SM0CVAL2)

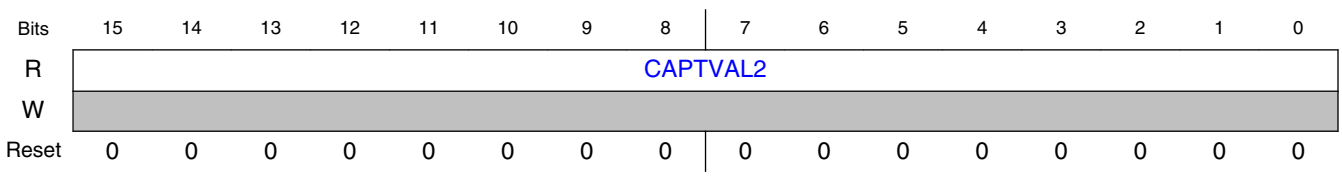
73.5.36.1 Offset

Register	Offset
SM0CVAL2	48h

73.5.36.2 Function

Writing this register generates bus transfer error.

73.5.36.3 Diagram



73.5.36.4 Fields

Field	Function
15-0 CAPTVAL2	Capture Value 2 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA0]. Each capture increases the value of CAPTCTRLA[CA0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.37 Capture Value 2 Cycle Register (SM0CVAL2CYC)

73.5.37.1 Offset

Register	Offset
SM0CVAL2CYC	4Ah

73.5.37.2 Function

Writing this register generates bus transfer error.

73.5.37.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL2CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.37.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL2CYC	Capture Value 2 Cycle This field stores the cycle number corresponding to the value captured in CVAL2. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.38 Capture Value 3 Register (SM0CVAL3)

73.5.38.1 Offset

Register	Offset
SM0CVAL3	4Ch

73.5.38.2 Function

Writing this register generates bus transfer error.

73.5.38.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.38.4 Fields

Field	Function
15-0 CAPTVAL3	Capture Value 3 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA1]. Each capture increases the value of CAPTCTRLA[CA1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.39 Capture Value 3 Cycle Register (SM0CVAL3CYC)

73.5.39.1 Offset

Register	Offset
SM0CVAL3CYC	4Eh

73.5.39.2 Function

Writing this register generates bus transfer error.

73.5.39.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL3CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.39.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL3CYC	Capture Value 3 Cycle This field stores the cycle number corresponding to the value captured in CVAL3. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.40 Capture Value 4 Register (SM0CVAL4)

73.5.40.1 Offset

Register	Offset
SM0CVAL4	50h

73.5.40.2 Function

Writing this register generates bus transfer error.

73.5.40.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.40.4 Fields

Field	Function
15-0 CAPTVAL4	Capture Value 4

Field	Function
	This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB0]. Each capture increases the value of CAPTCTRLB[CB0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.41 Capture Value 4 Cycle Register (SM0CVAL4CYC)

73.5.41.1 Offset

Register	Offset
SM0CVAL4CYC	52h

73.5.41.2 Function

Writing this register generates bus transfer error.

73.5.41.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL4CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.41.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL4CYC	Capture Value 4 Cycle This field stores the cycle number corresponding to the value captured in CVAL4. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.42 Capture Value 5 Register (SM0CVAL5)

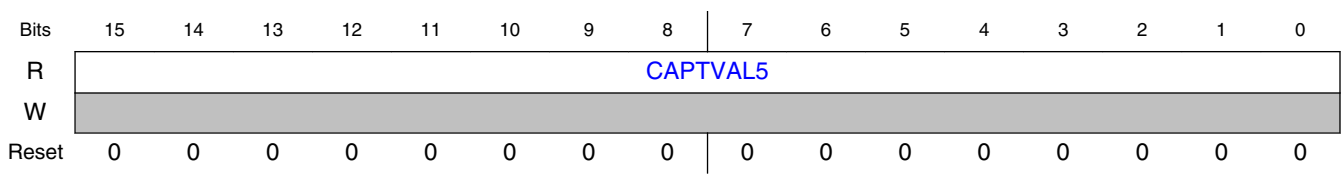
73.5.42.1 Offset

Register	Offset
SM0CVAL5	54h

73.5.42.2 Function

Writing this register generates bus transfer error.

73.5.42.3 Diagram



73.5.42.4 Fields

Field	Function
15-0 CAPTVAL5	Capture Value 5 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB1]. Each capture increases the value of CAPTCTRLB[CB1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.43 Capture Value 5 Cycle Register (SM0CVAL5CYC)

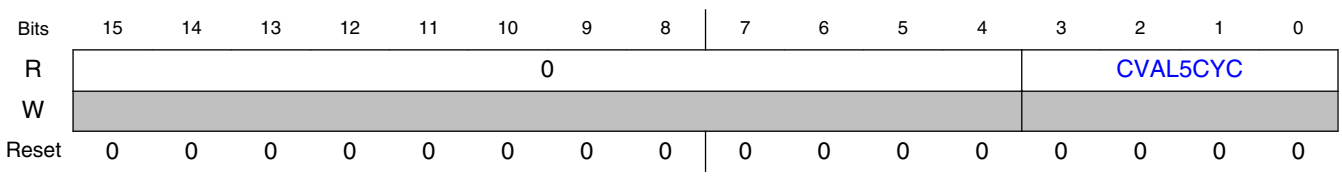
73.5.43.1 Offset

Register	Offset
SM0CVAL5CYC	56h

73.5.43.2 Function

Writing this register generates bus transfer error.

73.5.43.3 Diagram



73.5.43.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL5CYC	Capture Value 5 Cycle This field stores the cycle number corresponding to the value captured in CVAL5. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.44 Capture Control A Register (SM1CAPTCTRLA)

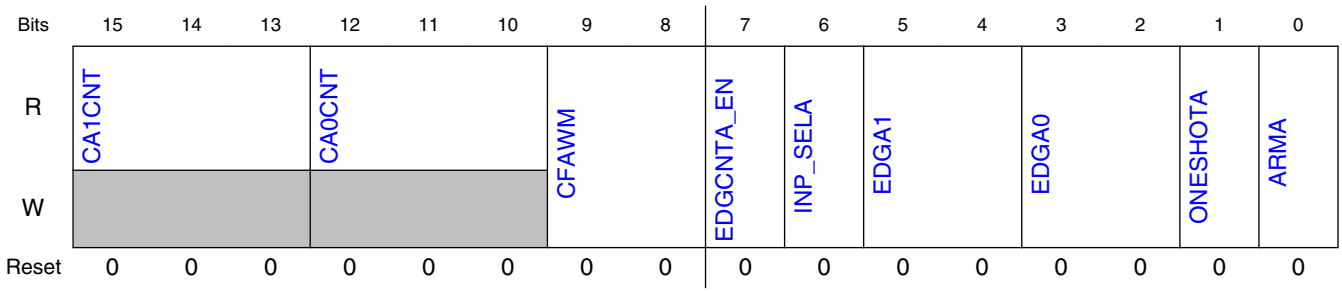
73.5.44.1 Offset

Register	Offset
SM1CAPTCTRLA	94h

73.5.44.2 Function

Contains capture controls for mode A.

73.5.44.3 Diagram



73.5.44.4 Fields

Field	Function
15-13 CA1CNT	Capture A1 FIFO Word Count This field reflects the number of words in the Capture A1 FIFO. (FIFO depth is 1.)
12-10 CA0CNT	Capture A0 FIFO Word Count This field reflects the number of words in the Capture A0 FIFO. (FIFO depth is 1.)
9-8 CFAWM	Capture A FIFOs Water Mark This field represents the water mark level for capture A FIFOs. The capture flags, STS[CFA1] and STS[CFA0], are not set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTA_EN	Edge Counter A Enable This field enables the edge counter which counts rising and falling edges on the PWM_A input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELA	Input Select A This field selects between the raw PWM_A input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_A input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLA[EDGA0] and CAPTCTRLA[EDGA1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRA[EDGA0] and/or CAPTCTRLA[EDGA1] fields to enable one or both of the capture registers.
5-4 EDGA1	Edge A 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGA0	Edge A 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTA	One Shot Mode A This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLA[ARMA] is cleared. No further captures are performed until CAPTCTRLA[ARMA] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLA[ARMA] is then cleared.
0 ARMA	Arm A Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLA[EDGAx] is enabled.

73.5.45 Capture Compare A Register (SM1CAPTCOMPA)

73.5.45.1 Offset

Register	Offset
SM1CAPTCOMPA	96h

73.5.45.2 Function

Contains capture and compare values for mode A.

73.5.45.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTA								EDGCMPA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.45.4 Fields

Field	Function
15-8 EDGCNTA	Edge Counter A This field contains the edge counter value for the PWM_A input capture circuitry.
7-0 EDGCMPA	Edge Compare A This field is the compare value associated with the edge counter for the PWM_A input capture circuitry.

73.5.46 Capture Control B Register (SM1CAPTCTRLB)

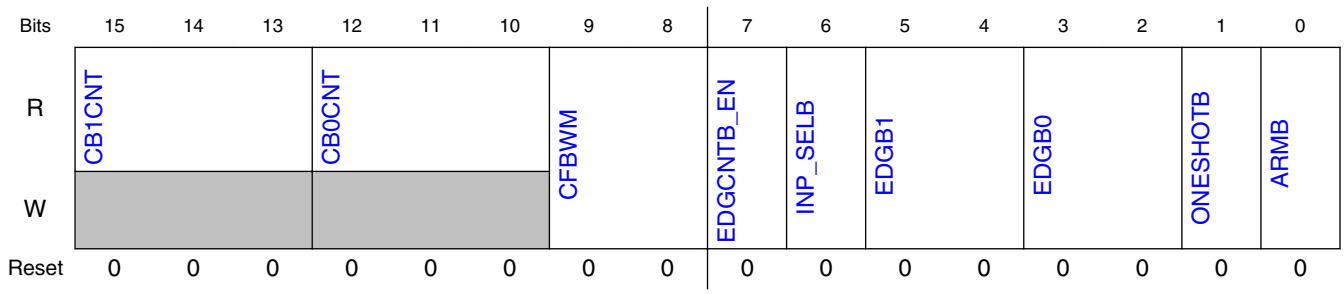
73.5.46.1 Offset

Register	Offset
SM1CAPTCTRLB	98h

73.5.46.2 Function

Contains capture controls for mode B.

73.5.46.3 Diagram



73.5.46.4 Fields

Field	Function
15-13 CB1CNT	Capture B1 FIFO Word Count This field reflects the number of words in the Capture B1 FIFO. (FIFO depth is 1.)
12-10 CB0CNT	Capture B0 FIFO Word Count This field reflects the number of words in the Capture B0 FIFO. (FIFO depth is 1.)
9-8 CFBWM	Capture B FIFOs Water Mark This field represents the water mark level for capture B FIFOs. The capture flags, STS[CFB1] and STS[CFB0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTB_EN	Edge Counter B Enable This field enables the edge counter which counts rising and falling edges on the PWM_B input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELB	Input Select B This field selects between the raw PWM_B input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_B input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLB[EDGB0] and CAPTCTRLB[EDGB1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRB[EDGB0] and/or CAPTCTRLB[EDGB1] fields to enable one or both of the capture registers.
5-4 EDGB1	Edge B 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGB0	Edge B 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTB	One Shot Mode B This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLB[ARMB] is cleared. No further captures are performed until CAPTCTRLB[ARMB] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLB[ARMB] is then cleared.
0 ARMB	Arm B Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLB[EDGBx] is enabled.

73.5.47 Capture Compare B Register (SM1CAPTCOMPB)

73.5.47.1 Offset

Register	Offset
SM1CAPTCOMPB	9Ah

73.5.47.2 Function

Contains capture and compare values for mode B.

73.5.47.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTB								EDGCMPB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.47.4 Fields

Field	Function
15-8 EDGCNTB	Edge Counter B This field contains the edge counter value for the PWM_B input capture circuitry.
7-0 EDGCMPB	Edge Compare B This field is the compare value associated with the edge counter for the PWM_B input capture circuitry.

73.5.48 Capture Control X Register (SM1CAPTCTRLX)

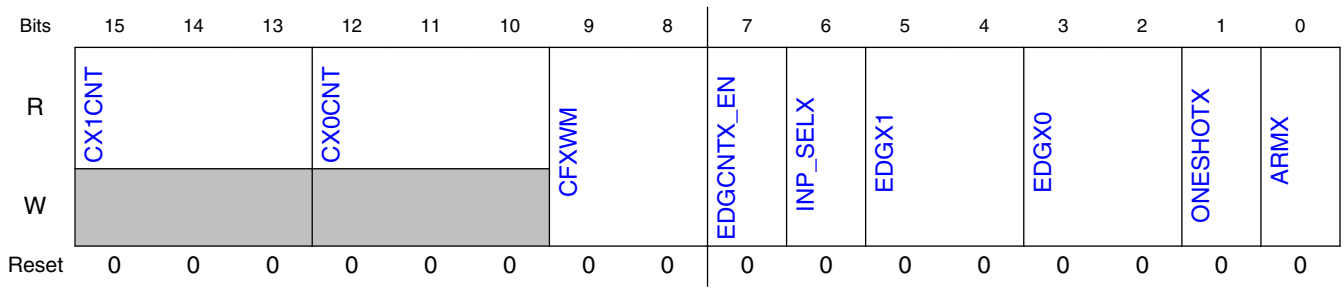
73.5.48.1 Offset

Register	Offset
SM1CAPTCTRLX	9Ch

73.5.48.2 Function

Contains capture controls for mode X.

73.5.48.3 Diagram



73.5.48.4 Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

73.5.49 Capture Compare X Register (SM1CAPTCOMPX)

73.5.49.1 Offset

Register	Offset
SM1CAPTCOMPX	9Eh

73.5.49.2 Function

Contains capture and control values for mode X.

73.5.49.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTX								EDGCMPX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.49.4 Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

73.5.50 Capture Value 0 Register (SM1CVAL0)

73.5.50.1 Offset

Register	Offset
SM1CVAL0	A0h

73.5.50.2 Function

Writing this register generates bus transfer error.

73.5.50.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.50.4 Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.51 Capture Value 0 Cycle Register (SM1CVAL0CYC)

73.5.51.1 Offset

Register	Offset
SM1CVAL0CYC	A2h

73.5.51.2 Function

Writing this register generates bus transfer error.

73.5.51.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL0CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.51.4 Fields

Field	Function
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.52 Capture Value 1 Register (SM1CVAL1)

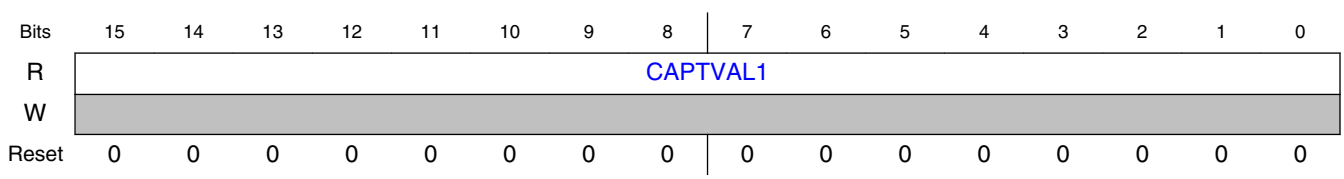
73.5.52.1 Offset

Register	Offset
SM1CVAL1	A4h

73.5.52.2 Function

Writing this register generates bus transfer error.

73.5.52.3 Diagram



73.5.52.4 Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.53 Capture Value 1 Cycle Register (SM1CVAL1CYC)

73.5.53.1 Offset

Register	Offset
SM1CVAL1CYC	A6h

73.5.53.2 Function

Writing this register generates bus transfer error.

73.5.53.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.53.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

73.5.54 Capture Value 2 Register (SM1CVAL2)

73.5.54.1 Offset

Register	Offset
SM1CVAL2	A8h

73.5.54.2 Function

Writing this register generates bus transfer error.

73.5.54.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.54.4 Fields

Field	Function
15-0 CAPTVAL2	Capture Value 2 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA0]. Each capture increases the value of CAPTCTRLA[CA0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.55 Capture Value 2 Cycle Register (SM1CVAL2CYC)

73.5.55.1 Offset

Register	Offset
SM1CVAL2CYC	AAh

73.5.55.2 Function

Writing this register generates bus transfer error.

73.5.55.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL2CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.55.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL2CYC	Capture Value 2 Cycle This field stores the cycle number corresponding to the value captured in CVAL2. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.56 Capture Value 3 Register (SM1CVAL3)

73.5.56.1 Offset

Register	Offset
SM1CVAL3	ACh

73.5.56.2 Function

Writing this register generates bus transfer error.

73.5.56.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.56.4 Fields

Field	Function
15-0 CAPTVAL3	Capture Value 3 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA1]. Each capture increases the value of CAPTCTRLA[CA1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.57 Capture Value 3 Cycle Register (SM1CVAL3CYC)

73.5.57.1 Offset

Register	Offset
SM1CVAL3CYC	AEh

73.5.57.2 Function

Writing this register generates bus transfer error.

73.5.57.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL3CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.57.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL3CYC	Capture Value 3 Cycle This field stores the cycle number corresponding to the value captured in CVAL3. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.58 Capture Value 4 Register (SM1CVAL4)

73.5.58.1 Offset

Register	Offset
SM1CVAL4	B0h

73.5.58.2 Function

Writing this register generates bus transfer error.

73.5.58.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.58.4 Fields

Field	Function
15-0 CAPTVAL4	Capture Value 4

Field	Function
	This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB0]. Each capture increases the value of CAPTCTRLB[CB0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.59 Capture Value 4 Cycle Register (SM1CVAL4CYC)

73.5.59.1 Offset

Register	Offset
SM1CVAL4CYC	B2h

73.5.59.2 Function

Writing this register generates bus transfer error.

73.5.59.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL4CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.59.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL4CYC	Capture Value 4 Cycle This field stores the cycle number corresponding to the value captured in CVAL4. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.60 Capture Value 5 Register (SM1CVAL5)

73.5.60.1 Offset

Register	Offset
SM1CVAL5	B4h

73.5.60.2 Function

Writing this register generates bus transfer error.

73.5.60.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.60.4 Fields

Field	Function
15-0 CAPTVAL5	Capture Value 5 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB1]. Each capture increases the value of CAPTCTRLB[CB1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.61 Capture Value 5 Cycle Register (SM1CVAL5CYC)

73.5.61.1 Offset

Register	Offset
SM1CVAL5CYC	B6h

73.5.61.2 Function

Writing this register generates bus transfer error.

73.5.61.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL5CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.61.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL5CYC	Capture Value 5 Cycle This field stores the cycle number corresponding to the value captured in CVAL5. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.62 Capture Control A Register (SM2CAPTCTRLA)

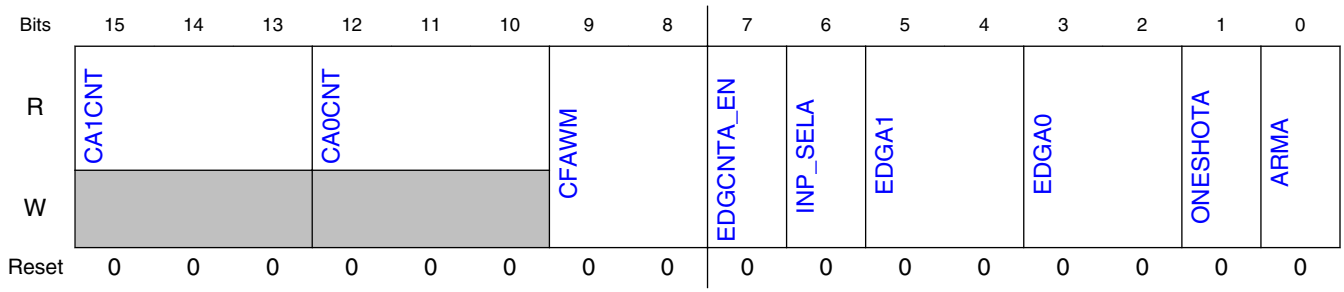
73.5.62.1 Offset

Register	Offset
SM2CAPTCTRLA	F4h

73.5.62.2 Function

Contains capture controls for mode A.

73.5.62.3 Diagram



73.5.62.4 Fields

Field	Function
15-13 CA1CNT	Capture A1 FIFO Word Count This field reflects the number of words in the Capture A1 FIFO. (FIFO depth is 1.)
12-10 CA0CNT	Capture A0 FIFO Word Count This field reflects the number of words in the Capture A0 FIFO. (FIFO depth is 1.)
9-8 CFAWM	Capture A FIFOs Water Mark This field represents the water mark level for capture A FIFOs. The capture flags, STS[CFA1] and STS[CFA0], are not set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTA_EN	Edge Counter A Enable This field enables the edge counter which counts rising and falling edges on the PWM_A input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELA	Input Select A This field selects between the raw PWM_A input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_A input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLA[EDGA0] and CAPTCTRLA[EDGA1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRA[EDGA0] and/or CAPTCTRLA[EDGA1] fields to enable one or both of the capture registers.
5-4 EDGA1	Edge A 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGA0	Edge A 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTA	One Shot Mode A This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLA[ARMA] is cleared. No further captures are performed until CAPTCTRLA[ARMA] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLA[ARMA] is then cleared.
0 ARMA	Arm A Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLA[EDGAx] is enabled.

73.5.63 Capture Compare A Register (SM2CAPTCOMPA)

73.5.63.1 Offset

Register	Offset
SM2CAPTCOMPA	F6h

73.5.63.2 Function

Contains capture and compare values for mode A.

73.5.63.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTA								EDGCMPA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.63.4 Fields

Field	Function
15-8 EDGCNTA	Edge Counter A This field contains the edge counter value for the PWM_A input capture circuitry.
7-0 EDGCMPA	Edge Compare A This field is the compare value associated with the edge counter for the PWM_A input capture circuitry.

73.5.64 Capture Control B Register (SM2CAPTCTRLB)

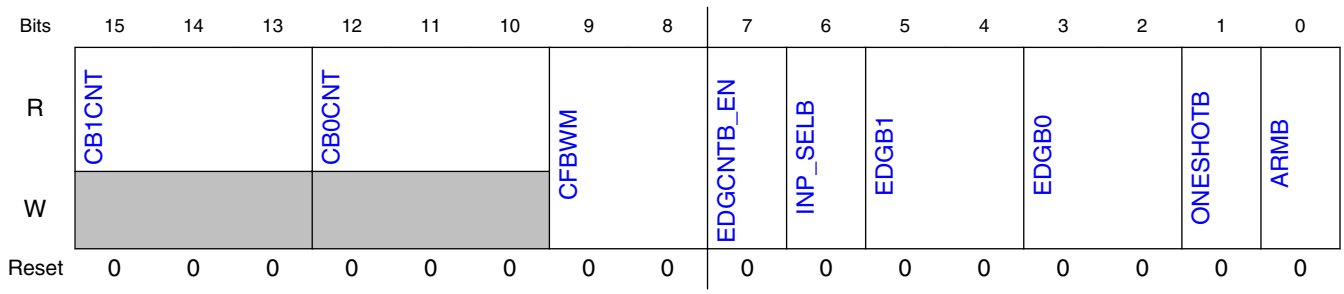
73.5.64.1 Offset

Register	Offset
SM2CAPTCTRLB	F8h

73.5.64.2 Function

Contains capture controls for mode B.

73.5.64.3 Diagram



73.5.64.4 Fields

Field	Function
15-13 CB1CNT	Capture B1 FIFO Word Count This field reflects the number of words in the Capture B1 FIFO. (FIFO depth is 1.)
12-10 CB0CNT	Capture B0 FIFO Word Count This field reflects the number of words in the Capture B0 FIFO. (FIFO depth is 1.)
9-8 CFBWM	Capture B FIFOs Water Mark This field represents the water mark level for capture B FIFOs. The capture flags, STS[CFB1] and STS[CFB0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTB_EN	Edge Counter B Enable This field enables the edge counter which counts rising and falling edges on the PWM_B input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELB	Input Select B This field selects between the raw PWM_B input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_B input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLB[EDGB0] and CAPTCTRLB[EDGB1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRB[EDGB0] and/or CAPTCTRLB[EDGB1] fields to enable one or both of the capture registers.
5-4 EDGB1	Edge B 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGB0	Edge B 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTB	One Shot Mode B This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLB[ARMB] is cleared. No further captures are performed until CAPTCTRLB[ARMB] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLB[ARMB] is then cleared.
0 ARMB	Arm B Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLB[EDGBx] is enabled.

73.5.65 Capture Compare B Register (SM2CAPTCOMPB)

73.5.65.1 Offset

Register	Offset
SM2CAPTCOMPB	FAh

73.5.65.2 Function

Contains capture and compare values for mode B.

73.5.65.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTB								EDGCMPB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.65.4 Fields

Field	Function
15-8 EDGCNTB	Edge Counter B This field contains the edge counter value for the PWM_B input capture circuitry.
7-0 EDGCMPB	Edge Compare B This field is the compare value associated with the edge counter for the PWM_B input capture circuitry.

73.5.66 Capture Control X Register (SM2CAPTCTRLX)

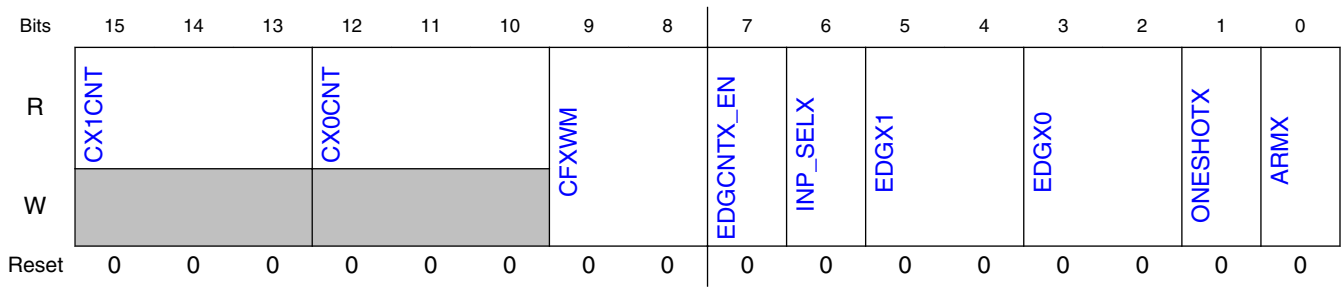
73.5.66.1 Offset

Register	Offset
SM2CAPTCTRLX	FCh

73.5.66.2 Function

Contains capture controls for mode X.

73.5.66.3 Diagram



73.5.66.4 Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

73.5.67 Capture Compare X Register (SM2CAPTCOMPX)

73.5.67.1 Offset

Register	Offset
SM2CAPTCOMPX	FEh

73.5.67.2 Function

Contains capture and control values for mode X.

73.5.67.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTX								EDGCMPX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.67.4 Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

73.5.68 Capture Value 0 Register (SM2CVAL0)

73.5.68.1 Offset

Register	Offset
SM2CVAL0	100h

73.5.68.2 Function

Writing this register generates bus transfer error.

73.5.68.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.68.4 Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.69 Capture Value 0 Cycle Register (SM2CVAL0CYC)

73.5.69.1 Offset

Register	Offset
SM2CVAL0CYC	102h

73.5.69.2 Function

Writing this register generates bus transfer error.

73.5.69.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL0CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.69.4 Fields

Field	Function
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.70 Capture Value 1 Register (SM2CVAL1)

73.5.70.1 Offset

Register	Offset
SM2CVAL1	104h

73.5.70.2 Function

Writing this register generates bus transfer error.

73.5.70.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.70.4 Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.71 Capture Value 1 Cycle Register (SM2CVAL1CYC)

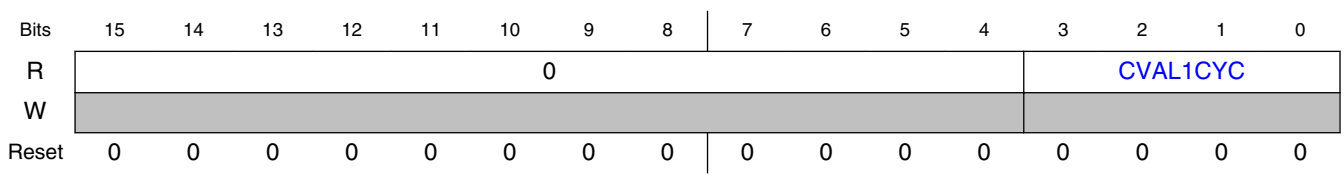
73.5.71.1 Offset

Register	Offset
SM2CVAL1CYC	106h

73.5.71.2 Function

Writing this register generates bus transfer error.

73.5.71.3 Diagram



73.5.71.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

73.5.72 Capture Value 2 Register (SM2CVAL2)

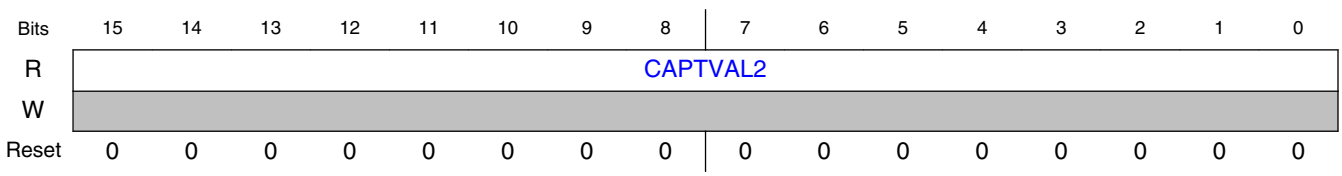
73.5.72.1 Offset

Register	Offset
SM2CVAL2	108h

73.5.72.2 Function

Writing this register generates bus transfer error.

73.5.72.3 Diagram



73.5.72.4 Fields

Field	Function
15-0 CAPTVAL2	Capture Value 2 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA0]. Each capture increases the value of CAPTCTRLA[CA0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.73 Capture Value 2 Cycle Register (SM2CVAL2CYC)

73.5.73.1 Offset

Register	Offset
SM2CVAL2CYC	10Ah

73.5.73.2 Function

Writing this register generates bus transfer error.

73.5.73.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL2CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.73.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL2CYC	Capture Value 2 Cycle This field stores the cycle number corresponding to the value captured in CVAL2. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.74 Capture Value 3 Register (SM2CVAL3)

73.5.74.1 Offset

Register	Offset
SM2CVAL3	10Ch

73.5.74.2 Function

Writing this register generates bus transfer error.

73.5.74.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.74.4 Fields

Field	Function
15-0 CAPTVAL3	Capture Value 3 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA1]. Each capture increases the value of CAPTCTRLA[CA1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.75 Capture Value 3 Cycle Register (SM2CVAL3CYC)

73.5.75.1 Offset

Register	Offset
SM2CVAL3CYC	10Eh

73.5.75.2 Function

Writing this register generates bus transfer error.

73.5.75.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL3CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.75.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL3CYC	Capture Value 3 Cycle This field stores the cycle number corresponding to the value captured in CVAL3. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.76 Capture Value 4 Register (SM2CVAL4)

73.5.76.1 Offset

Register	Offset
SM2CVAL4	110h

73.5.76.2 Function

Writing this register generates bus transfer error.

73.5.76.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.76.4 Fields

Field	Function
15-0 CAPTVAL4	Capture Value 4

Field	Function
	This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB0]. Each capture increases the value of CAPTCTRLB[CB0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.77 Capture Value 4 Cycle Register (SM2CVAL4CYC)

73.5.77.1 Offset

Register	Offset
SM2CVAL4CYC	112h

73.5.77.2 Function

Writing this register generates bus transfer error.

73.5.77.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL4CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.77.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL4CYC	Capture Value 4 Cycle This field stores the cycle number corresponding to the value captured in CVAL4. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.78 Capture Value 5 Register (SM2CVAL5)

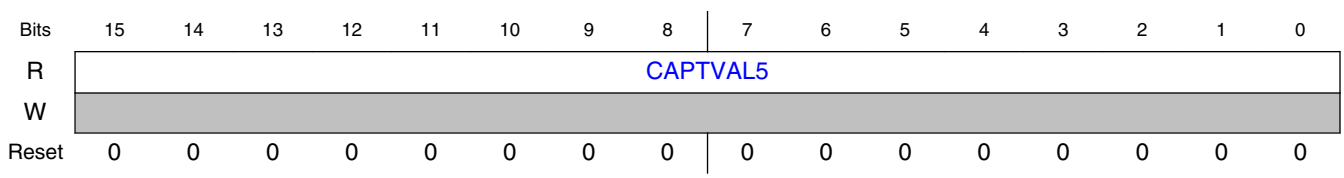
73.5.78.1 Offset

Register	Offset
SM2CVAL5	114h

73.5.78.2 Function

Writing this register generates bus transfer error.

73.5.78.3 Diagram



73.5.78.4 Fields

Field	Function
15-0 CAPTVAL5	Capture Value 5 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB1]. Each capture increases the value of CAPTCTRLB[CB1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.79 Capture Value 5 Cycle Register (SM2CVAL5CYC)

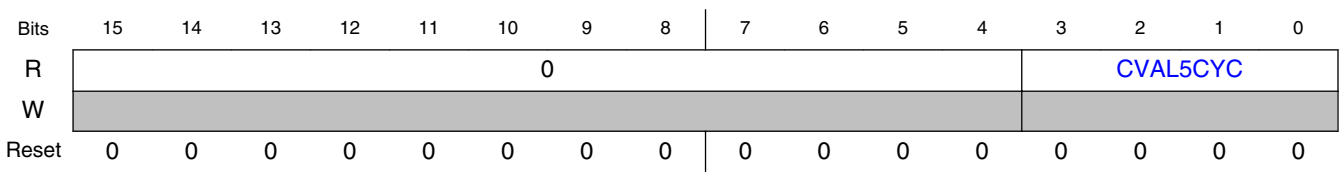
73.5.79.1 Offset

Register	Offset
SM2CVAL5CYC	116h

73.5.79.2 Function

Writing this register generates bus transfer error.

73.5.79.3 Diagram



73.5.79.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL5CYC	Capture Value 5 Cycle This field stores the cycle number corresponding to the value captured in CVAL5. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.80 Capture Control A Register (SM3CAPTCTRLA)

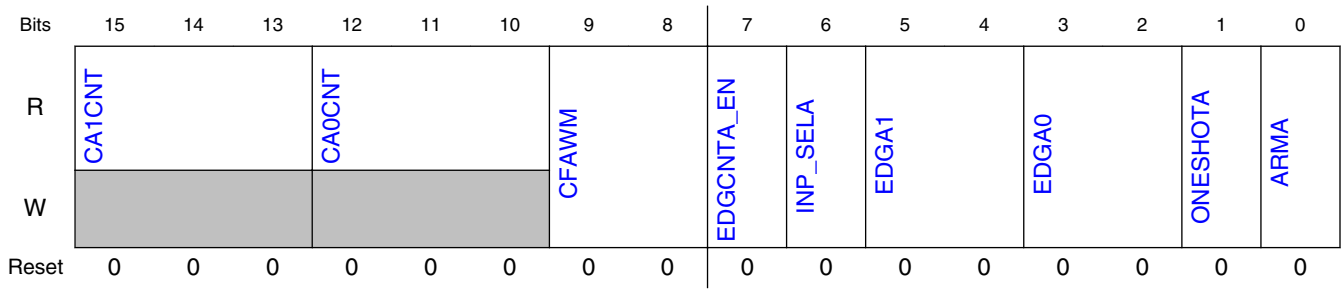
73.5.80.1 Offset

Register	Offset
SM3CAPTCTRLA	154h

73.5.80.2 Function

Contains capture controls for mode A.

73.5.80.3 Diagram



73.5.80.4 Fields

Field	Function
15-13 CA1CNT	Capture A1 FIFO Word Count This field reflects the number of words in the Capture A1 FIFO. (FIFO depth is 1.)
12-10 CA0CNT	Capture A0 FIFO Word Count This field reflects the number of words in the Capture A0 FIFO. (FIFO depth is 1.)
9-8 CFAWM	Capture A FIFOs Water Mark This field represents the water mark level for capture A FIFOs. The capture flags, STS[CFA1] and STS[CFA0], are not set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTA_EN	Edge Counter A Enable This field enables the edge counter which counts rising and falling edges on the PWM_A input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELA	Input Select A This field selects between the raw PWM_A input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_A input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLA[EDGA0] and CAPTCTRLA[EDGA1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRA[EDGA0] and/or CAPTCTRLA[EDGA1] fields to enable one or both of the capture registers.
5-4 EDGA1	Edge A 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGA0	Edge A 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTA	One Shot Mode A This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLA[ARMA] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLA[ARMA] is cleared. No further captures are performed until CAPTCTRLA[ARMA] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLA[ARMA] is then cleared.
0 ARMA	Arm A Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLA[EDGAx] is enabled.

73.5.81 Capture Compare A Register (SM3CAPTCOMPA)

73.5.81.1 Offset

Register	Offset
SM3CAPTCOMPA	156h

73.5.81.2 Function

Contains capture and compare values for mode A.

73.5.81.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTA								EDGCMPA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.81.4 Fields

Field	Function
15-8 EDGCNTA	Edge Counter A This field contains the edge counter value for the PWM_A input capture circuitry.
7-0 EDGCMPA	Edge Compare A This field is the compare value associated with the edge counter for the PWM_A input capture circuitry.

73.5.82 Capture Control B Register (SM3CAPTCTRLB)

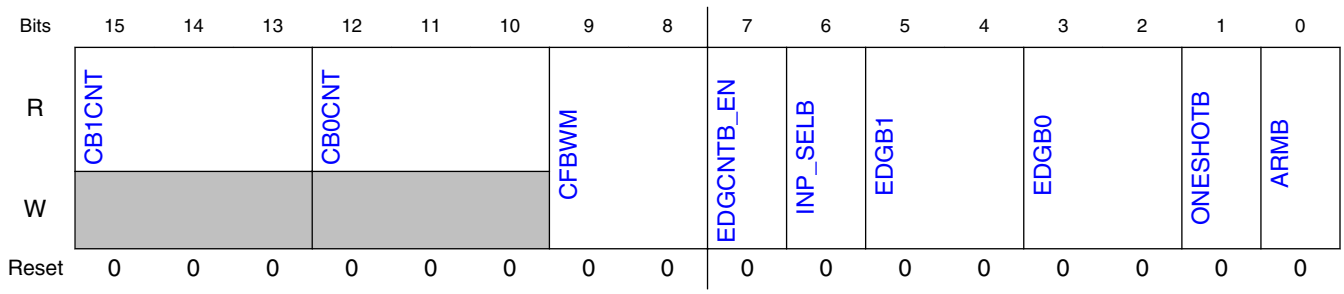
73.5.82.1 Offset

Register	Offset
SM3CAPTCTRLB	158h

73.5.82.2 Function

Contains capture controls for mode B.

73.5.82.3 Diagram



73.5.82.4 Fields

Field	Function
15-13 CB1CNT	Capture B1 FIFO Word Count This field reflects the number of words in the Capture B1 FIFO. (FIFO depth is 1.)
12-10 CB0CNT	Capture B0 FIFO Word Count This field reflects the number of words in the Capture B0 FIFO. (FIFO depth is 1.)
9-8 CFBWM	Capture B FIFOs Water Mark This field represents the water mark level for capture B FIFOs. The capture flags, STS[CFB1] and STS[CFB0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTB_EN	Edge Counter B Enable This field enables the edge counter which counts rising and falling edges on the PWM_B input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELB	Input Select B This field selects between the raw PWM_B input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_B input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLB[EDGB0] and CAPTCTRLB[EDGB1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRB[EDGB0] and/or CAPTCTRLB[EDGB1] fields to enable one or both of the capture registers.
5-4 EDGB1	Edge B 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGB0	Edge B 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTB	One Shot Mode B This bit selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after CAPTCTRLB[ARMB] is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and CAPTCTRLB[ARMB] is cleared. No further captures are performed until CAPTCTRLB[ARMB] is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and CAPTCTRLB[ARMB] is then cleared.
0 ARMB	Arm B Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLB[EDGBx] is enabled.

73.5.83 Capture Compare B Register (SM3CAPTCOMPB)

73.5.83.1 Offset

Register	Offset
SM3CAPTCOMPB	15Ah

73.5.83.2 Function

Contains capture and compare values for mode B.

73.5.83.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTB								EDGCMPB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.83.4 Fields

Field	Function
15-8 EDGCNTB	Edge Counter B This field contains the edge counter value for the PWM_B input capture circuitry.
7-0 EDGCMPB	Edge Compare B This field is the compare value associated with the edge counter for the PWM_B input capture circuitry.

73.5.84 Capture Control X Register (SM3CAPTCTRLX)

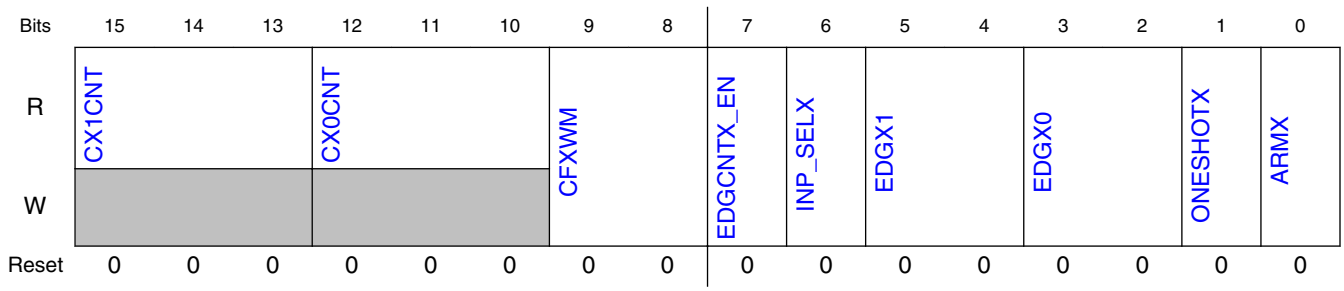
73.5.84.1 Offset

Register	Offset
SM3CAPTCTRLX	15Ch

73.5.84.2 Function

Contains capture controls for mode X.

73.5.84.3 Diagram



73.5.84.4 Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], will not be set until the word count of the corresponding FIFO is greater than this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event.

Table continues on the next page...

Field	Function
	00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

73.5.85 Capture Compare X Register (SM3CAPTCOMPX)

73.5.85.1 Offset

Register	Offset
SM3CAPTCOMPX	15Eh

73.5.85.2 Function

Contains capture and control values for mode X.

73.5.85.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EDGCNTX								EDGCMPX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.85.4 Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

73.5.86 Capture Value 0 Register (SM3CVAL0)

73.5.86.1 Offset

Register	Offset
SM3CVAL0	160h

73.5.86.2 Function

Writing this register generates bus transfer error.

73.5.86.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.86.4 Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.87 Capture Value 0 Cycle Register (SM3CVAL0CYC)

73.5.87.1 Offset

Register	Offset
SM3CVAL0CYC	162h

73.5.87.2 Function

Writing this register generates bus transfer error.

73.5.87.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL0CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.87.4 Fields

Field	Function
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.88 Capture Value 1 Register (SM3CVAL1)

73.5.88.1 Offset

Register	Offset
SM3CVAL1	164h

73.5.88.2 Function

Writing this register generates bus transfer error.

73.5.88.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.88.4 Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.89 Capture Value 1 Cycle Register (SM3CVAL1CYC)

73.5.89.1 Offset

Register	Offset
SM3CVAL1CYC	166h

73.5.89.2 Function

Writing this register generates bus transfer error.

73.5.89.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.89.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

73.5.90 Capture Value 2 Register (SM3CVAL2)

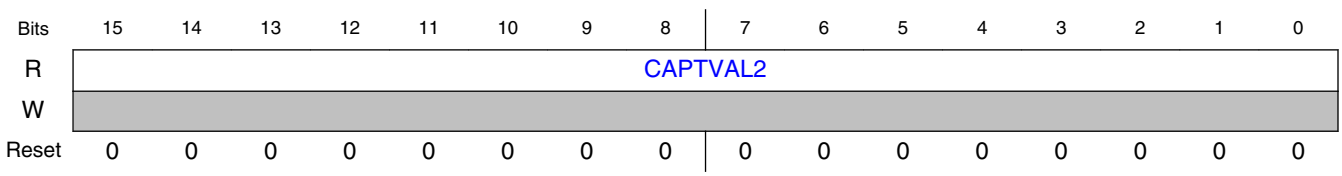
73.5.90.1 Offset

Register	Offset
SM3CVAL2	168h

73.5.90.2 Function

Writing this register generates bus transfer error.

73.5.90.3 Diagram



73.5.90.4 Fields

Field	Function
15-0 CAPTVAL2	Capture Value 2 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA0]. Each capture increases the value of CAPTCTRLA[CA0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.91 Capture Value 2 Cycle Register (SM3CVAL2CYC)

73.5.91.1 Offset

Register	Offset
SM3CVAL2CYC	16Ah

73.5.91.2 Function

Writing this register generates bus transfer error.

73.5.91.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL2CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.91.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL2CYC	Capture Value 2 Cycle This field stores the cycle number corresponding to the value captured in CVAL2. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.92 Capture Value 3 Register (SM3CVAL3)

73.5.92.1 Offset

Register	Offset
SM3CVAL3	16Ch

73.5.92.2 Function

Writing this register generates bus transfer error.

73.5.92.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.92.4 Fields

Field	Function
15-0 CAPTVAL3	Capture Value 3 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLA[EDGA1]. Each capture increases the value of CAPTCTRLA[CA1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLA[CA1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.93 Capture Value 3 Cycle Register (SM3CVAL3CYC)

73.5.93.1 Offset

Register	Offset
SM3CVAL3CYC	16Eh

73.5.93.2 Function

Writing this register generates bus transfer error.

73.5.93.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL3CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.93.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL3CYC	Capture Value 3 Cycle This field stores the cycle number corresponding to the value captured in CVAL3. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.94 Capture Value 4 Register (SM3CVAL4)

73.5.94.1 Offset

Register	Offset
SM3CVAL4	170h

73.5.94.2 Function

Writing this register generates bus transfer error.

73.5.94.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.94.4 Fields

Field	Function
15-0 CAPTVAL4	Capture Value 4

Field	Function
	This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB0]. Each capture increases the value of CAPTCTRLB[CB0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB0CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.95 Capture Value 4 Cycle Register (SM3CVAL4CYC)

73.5.95.1 Offset

Register	Offset
SM3CVAL4CYC	172h

73.5.95.2 Function

Writing this register generates bus transfer error.

73.5.95.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL4CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.95.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL4CYC	Capture Value 4 Cycle This field stores the cycle number corresponding to the value captured in CVAL4. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.96 Capture Value 5 Register (SM3CVAL5)

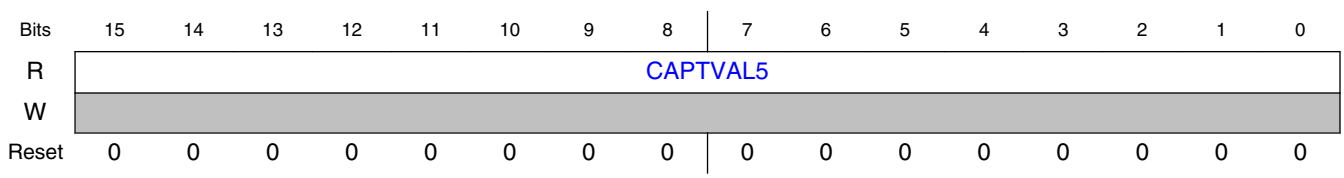
73.5.96.1 Offset

Register	Offset
SM3CVAL5	174h

73.5.96.2 Function

Writing this register generates bus transfer error.

73.5.96.3 Diagram



73.5.96.4 Fields

Field	Function
15-0 CAPTVAL5	Capture Value 5 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLB[EDGB1]. Each capture increases the value of CAPTCTRLB[CB1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLB[CB1CNT] by 1 until 0 is reached. This field is not byte accessible.

73.5.97 Capture Value 5 Cycle Register (SM3CVAL5CYC)

73.5.97.1 Offset

Register	Offset
SM3CVAL5CYC	176h

73.5.97.2 Function

Writing this register generates bus transfer error.

73.5.97.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL5CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.97.4 Fields

Field	Function
15-4 —	Reserved
3-0 CVAL5CYC	Capture Value 5 Cycle This field stores the cycle number corresponding to the value captured in CVAL5. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

73.5.98 Output Enable Register (OUTEN)

73.5.98.1 Offset

Register	Offset
OUTEN	180h

73.5.98.2 Function

Contains PWM output enables.

73.5.98.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PWMA_EN				PWMB_EN				PWMX_EN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.98.4 Fields

Field	Function
15-12 —	Reserved
11-8 PWMA_EN	PWM_A Output Enables This field enables the PWM_A outputs of submodules 3 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_A pin is being used for input capture. • 0b0 PWM_A output disabled. • 0b1 PWM_A output enabled.
7-4 PWMB_EN	PWM_B Output Enables This field enables the PWM_B outputs of submodules 3 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_B pin is being used for input capture. • 0b0 PWM_B output disabled. • 0b1 PWM_B output enabled.
3-0 PWMX_EN	PWM_X Output Enables This field enables the PWM_X outputs of submodules 3 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_X pin is being used for input capture or deadtime correction. • 0b0 PWM_X output disabled. • 0b1 PWM_X output enabled.

73.5.99 Mask Register (MASK)

73.5.99.1 Offset

Register	Offset
MASK	182h

73.5.99.2 Function

MASK is double buffered and does not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading MASK reads the buffered values and not necessarily the values currently in effect. This double buffering can be overridden by setting the UPDATE_MASK bits.

73.5.99.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					MASKA				MASKB				MASKX			
W	UPDATE_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.99.4 Fields

Field	Function
15-12 UPDATE_MASK	Update Mask Bits Immediately This field masks or unmasks the PWM_X (X = A, B, or X) outputs of submodules 3 - 0 respectively by forcing the MASK* (* = A, B or X) bits to be immediately updated within submodules 3 - 0, respectively without waiting for a FORCE_OUT event. These self-clearing bits always read as zero. Software may write to any or all of these bits and may set these bits in the same write operation that updates the MASKA, MASKB, and MASKX fields of this register. • 0b0 Normal operation. MASK* bits within the corresponding submodule are not updated until a FORCE_OUT event occurs within the submodule. • 0b1 Immediate operation. MASK* bits within the corresponding submodule are updated on the following clock edge after setting this bit. For example: • UPDATE_MASK[3]=1 updates the mask bits MASKA[3], MASKB[3], and MASKX[3]. • UPDATE_MASK[3:0]=0101 updates the mask bits MASKA[2], MASKA[0], MASKB[2], MASKB[0], MASKX[2], and MASKX[0].
11-8 MASKA	PWM_A Masks

Table continues on the next page...

Field	Function
	This field masks the PWM_A outputs of submodules 3 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_A output normal. • 0b1 PWM_A output masked. MASKA[3:0] masks PWM_A_3 through PWM_A_0
7-4 MASKB	PWM_B Masks This field masks the PWM_B outputs of submodules 3 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_B output normal. • 0b1 PWM_B output masked. MASKB[3:0] masks PWM_B_3 through PWM_B_0
3-0 MASKX	PWM_X Masks This field masks the PWM_X outputs of submodules 3 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_X output normal. • 0b1 PWM_X output masked. MASKX[3:0] masks PWM_X_3 through PWM_X_0

73.5.100 Software Controlled Output Register (SWCOUT)

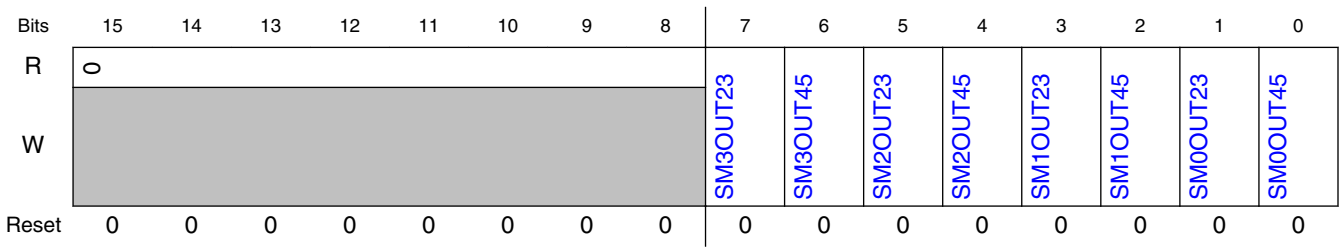
73.5.100.1 Offset

Register	Offset
SWCOUT	184h

73.5.100.2 Function

These bits are double buffered and do not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading these bits reads the buffered value and not necessarily the value currently in effect.

73.5.100.3 Diagram



73.5.100.4 Fields

Field	Function
15-8 —	Reserved
7 SM3OUT23	Submodule 3 Software Controlled Output 23 This field is used when DTSRCSEL[SM3SEL23] is set to 0b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 3 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 3 instead of PWM23.
6 SM3OUT45	Submodule 3 Software Controlled Output 45 This field is used when DTSRCSEL[SM3SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 3 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 3 instead of PWM45.
5 SM2OUT23	Submodule 2 Software Controlled Output 23 This field is used when DTSRCSEL[SM2SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 2 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 2 instead of PWM23.
4 SM2OUT45	Submodule 2 Software Controlled Output 45 This field is used when DTSRCSEL[SM2SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 2 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 2 instead of PWM45.
3 SM1OUT23	Submodule 1 Software Controlled Output 23 This field is used when DTSRCSEL[SM1SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 1 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 1 instead of PWM23.
2 SM1OUT45	Submodule 1 Software Controlled Output 45 This field is used when DTSRCSEL[SM1SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule.

Table continues on the next page...

Field	Function
	0b - A logic 0 is supplied to the deadtime generator of submodule 1 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 1 instead of PWM45.
1 SM0OUT23	Submodule 0 Software Controlled Output 23 This field is used when DTSRCSEL[SM0SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 0 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 0 instead of PWM23.
0 SM0OUT45	Submodule 0 Software Controlled Output 45 This field is used when DTSRCSEL[SM0SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 0 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 0 instead of PWM45.

73.5.101 PWM Source Select Register (DTSRCSEL)

73.5.101.1 Offset

Register	Offset
DTSRCSEL	186h

73.5.101.2 Function

The PWM source select bits are double buffered and do not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading these bits reads the buffered value and not necessarily the value currently in effect.

73.5.101.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SM3SEL23		SM3SEL45		SM2SEL23		SM2SEL45		SM1SEL23		SM1SEL45		SM0SEL23		SM0SEL45	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.101.4 Fields

Field	Function
15-14 SM3SEL23	Submodule 3 PWM23 Control Select This field selects possible overrides to the generated SM3PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM3PWM23 signal used by the deadtime logic. 01b - Inverted generated SM3PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM3OUT23] used by the deadtime logic. 11b - PWM3_EXTA signal used by the deadtime logic.
13-12 SM3SEL45	Submodule 3 PWM45 Control Select This field selects possible overrides to the generated SM3PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM3PWM45 signal used by the deadtime logic. 01b - Inverted generated SM3PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM3OUT45] used by the deadtime logic. 11b - PWM3_EXTB signal used by the deadtime logic.
11-10 SM2SEL23	Submodule 2 PWM23 Control Select This field selects possible overrides to the generated SM2PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM2PWM23 signal used by the deadtime logic. 01b - Inverted generated SM2PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM2OUT23] used by the deadtime logic. 11b - PWM2_EXTA signal used by the deadtime logic.
9-8 SM2SEL45	Submodule 2 PWM45 Control Select This field selects possible overrides to the generated SM2PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM2PWM45 signal used by the deadtime logic. 01b - Inverted generated SM2PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM2OUT45] used by the deadtime logic. 11b - PWM2_EXTB signal used by the deadtime logic.
7-6 SM1SEL23	Submodule 1 PWM23 Control Select This field selects possible overrides to the generated SM1PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM1PWM23 signal used by the deadtime logic. 01b - Inverted generated SM1PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM1OUT23] used by the deadtime logic. 11b - PWM1_EXTA signal used by the deadtime logic.
5-4 SM1SEL45	Submodule 1 PWM45 Control Select This field selects possible overrides to the generated SM1PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM1PWM45 signal used by the deadtime logic. 01b - Inverted generated SM1PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM1OUT45] used by the deadtime logic. 11b - PWM1_EXTB signal used by the deadtime logic.
3-2	Submodule 0 PWM23 Control Select

Table continues on the next page...

Field	Function
SM0SEL23	This field selects possible overrides to the generated SM0PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM0PWM23 signal used by the deadtime logic. 01b - Inverted generated SM0PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM0OUT23] used by the deadtime logic. 11b - PWM0_EXTA signal used by the deadtime logic.
1-0 SM0SEL45	Submodule 0 PWM45 Control Select This field selects possible overrides to the generated SM0PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM0PWM45 signal used by the deadtime logic. 01b - Inverted generated SM0PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM0OUT45] used by the deadtime logic. 11b - PWM0_EXTB signal used by the deadtime logic.

73.5.102 Master Control Register (MCTRL)

73.5.102.1 Offset

Register	Offset
MCTRL	188h

73.5.102.2 Function

In every 4-bit field in this register, each bit acts on a separate submodule. Accordingly, the description of every bit field refers to the effect of an individual bit.

73.5.102.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IPOL				RUN				0				LDOK			
W									CLDOK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.102.4 Fields

Field	Function
15-12 IPOL	Current Polarity This field corresponds to submodules 3 - 0, respectively. Each bit selects between PWM23 and PWM45 as the source for the generation of the complementary PWM pair output for the corresponding submodule. MCTRL[IPOL] is ignored in independent mode. MCTRL[IPOL] does not take effect until a FORCE_OUT event takes place in the appropriate submodule. Reading MCTRL[IPOL] reads the buffered value and not necessarily the value currently in effect. 0000b - PWM23 is used to generate complementary PWM pair in the corresponding submodule. 0001b - PWM45 is used to generate complementary PWM pair in the corresponding submodule.
11-8 RUN	Run This field enables the clocks to the PWM generator of submodules 3 - 0, respectively. The corresponding MCTRL[RUN] bit must be set for each submodule that is using its input capture functions or is using the local reload as its reload source. When this bit equals zero, the submodule counter is reset, and PWM outputs are held. A reset clears this field. 0000b - PWM counter is stopped, but PWM outputs hold the current state. 0001b - PWM counter is started in the corresponding submodule.
7-4 CLDOK	Clear Load Okay This field corresponds to submodules 3 - 0, respectively. Each write-only bit is used to clear the corresponding bit of MCTRL[LDOK]. Write a 1 to CLDOK to clear the corresponding MCTRL[LDOK] bit. If a reload occurs within a submodule with the corresponding MCTRL[LDOK] bit set at the same time that MCTRL[CLDOK] is written, then the reload in that submodule will not be performed and MCTRL[LDOK] will be cleared. CLDOK bit is self-clearing and always reads as a 0.
3-0 LDOK	Load Okay This field corresponds to submodules 3 - 0, respectively. Each read/set bit loads CTRL[PRSC] and the INIT, FRACVALx, and VALx registers of the corresponding submodule into a set of buffers. The buffered prescaler divisor, submodule counter modulus value, and PWM pulse width take effect at the next PWM reload if CTRL[LDMOD] is clear or if CTRL[LDMOD] is set. Set the corresponding MCTRL[LDOK] bit by reading it when it is logic zero and then writing a logic one to it. The VALx, FRACVALx, INIT, and CTRL[PRSC] registers of the corresponding submodule cannot be written while the corresponding MCTRL[LDOK] bit is set. In Master Reload Mode (CTRL2[RELOAD_SEL]=1), it is necessary to set the LDOK bit corresponding to submodule0; however, it is recommended to also set the LDOK bit of the slave submodules, to prevent unwanted writes to the registers in the slave submodules. The MCTRL[LDOK] bit is automatically cleared after the new values are loaded, or it can be manually cleared before a reload by writing a logic 1 to the appropriate MCTRL[CLDOK] bit. LDOK bits cannot be written with a zero. MCTRL[LDOK] can be set in DMA mode when the DMA indicates that it has completed the update of all CTRL[PRSC], INIT, FRACVALx, and VALx registers in the corresponding submodule. Reset clears LDOK field. 0000b - Do not load new values. 0001b - Load prescaler, modulus, and PWM values of the corresponding submodule.

73.5.103 Fault Control Register (FCTRL0)

73.5.103.1 Offset

Register	Offset
FCTRL0	18Ch

73.5.103.2 Function

For every 4-bit field in this register, the bits act on the fault inputs in order. For example, FLVL bits 15-12 act on faults 3-0, respectively.

73.5.103.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FLVL				FAUTO				FSAFE				FIE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.103.4 Fields

Field	Function
15-12 FLVL	Fault Level This field selects the active logic level of the individual fault inputs 3-0, respectively. A reset clears this field. 0000b - A logic 0 on the fault input indicates a fault condition. 0001b - A logic 1 on the fault input indicates a fault condition.
11-8 FAUTO	Automatic Fault Clearing This field selects automatic or manual clearing of faults 3-0, respectively. A reset clears this field. 0000b - Manual fault clearing. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL]. If neither FFULL nor FHALF is set, then the fault condition cannot be cleared. This is further controlled by FCTRL[FSAFE]. 0001b - Automatic fault clearing. PWM outputs disabled by this fault are enabled when FSTS[FFPINx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL] without regard to the state of FSTS[FFLAGx]. If neither FFULL nor FHALF is set, then the fault condition cannot be cleared.
7-4 FSAFE	Fault Safety Mode This field selects the safety mode during manual fault clearing. A reset clears this field.

Table continues on the next page...

Field	Function
	FSTS[FFPINx] may indicate that a fault condition still exists even though the actual fault signal at the FAULTx pin is clear due to the fault filter latency. 0000b - Normal mode. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL] without regard to the state of FSTS[FFPINx]. If neither FHALF nor FFULL is set, then the fault condition cannot be cleared. The PWM outputs disabled by this fault input will not be re-enabled until the actual FAULTx input signal de-asserts since the fault input will combinationally disable the PWM outputs (as programmed in DISMAPn). 0001b - Safe mode. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear and FSTS[FFPINx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL]. If neither FHLAF nor FFULL is set, then the fault condition cannot be cleared.
3-0 FIE	Fault Interrupt Enables This field enables CPU interrupt requests generated by the FAULTx pins. A reset clears this field. **NOTE:** The fault protection circuit is independent of the FIEx bit and is always active. If a fault is detected, the PWM outputs are disabled according to the disable mapping register. 0000b - FAULTx CPU interrupt requests disabled. 0001b - FAULTx CPU interrupt requests enabled.

73.5.104 Fault Status Register (FSTS0)

73.5.104.1 Offset

Register	Offset
FSTS0	18Eh

73.5.104.2 Function

Includes controls related to fault conditions.

73.5.104.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FHALF				FFPIN				FFULL				FFLAG			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.104.4 Fields

Field	Function
15-12 FHALF	Half Cycle Fault Recovery This field is used to control the timing for re-enabling the PWM outputs after a fault condition. These bits apply to both automatic and manual clearing of a fault condition. **NOTE:** Both FHALF and FFULL can be set so that the fault recovery occurs at the start of a full cycle and at the start of a half cycle (as defined by VAL0). If neither FHALF nor FFULL is set, then no fault recovery is possible. 0000b - PWM outputs are not re-enabled at the start of a half cycle. 0001b - PWM outputs are re-enabled at the start of a half cycle (as defined by VAL0).
11-8 FFPIN	Filtered Fault Pins These read-only bits reflect the current state of the filtered FAULTx pins converted to high polarity. A logic 1 indicates that a fault condition exists on the filtered FAULTx pin. A reset has no effect on this field. After the system reset de-asserts, these are the possible values of FFPIN: If FCTRL[FLVL] = 0 and FAULTx = 0, then FFPIN is set to 1. If FCTRL[FLVL] = 0 and FAULTx = 1, then FFPIN is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 0, then FFPIN is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 1, then FFPIN is set to 1.
7-4 FFULL	Full Cycle This field is used to control the timing for re-enabling the PWM outputs after a fault condition. These bits apply to both automatic and manual clearing of a fault condition. **NOTE:** Both FHALF and FFULL can be set so that the fault recovery occurs at the start of a full cycle and at the start of a half cycle (as defined by VAL0). If neither FHALF nor FFULL is set, then no fault recovery is possible. 0000b - PWM outputs are not re-enabled at the start of a full cycle 0001b - PWM outputs are re-enabled at the start of a full cycle
3-0 FFLAG	Fault Flags These read-only flags are set within two CPU cycles after a transition to active on the FAULTx pin. Clear this bit by writing a logic one to it. A reset clears this field. While the reset value is 0, these bits may be set to 1 by the time they can be read depending on the state of the fault input signals. After the system reset de-asserts, these are the possible values of FFLAG: If FCTRL[FLVL] = 0 and FAULTx = 0, then FFLAG is set to 1. If FCTRL[FLVL] = 0 and FAULTx = 1, then FFLAG is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 0, then FFLAG is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 1, then FFLAG is set to 1. 0000b - No fault on the FAULTx pin. 0001b - Fault on the FAULTx pin.

73.5.105 Fault Filter Register (FFILT0)

73.5.105.1 Offset

Register	Offset
FFILT0	190h

73.5.105.2 Function

The settings in this register are shared among each of the fault input filters within the fault channel.

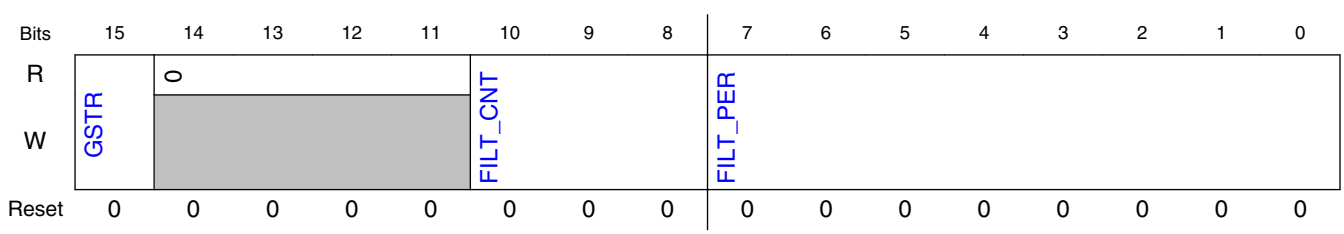
Input filter considerations include:

- Set the FILT_PER value such that the sampling period is larger than the period of the expected noise. This way a noise spike corrupts one sample. Select the FILT_CNT value to reduce and recognize the probability of noisy samples causing an incorrect transition. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the FILT_CNT+3 power.
- The values of FILT_PER and FILT_CNT must be traded off against the desire for minimal latency in recognizing input transitions. Turning on the input filter (setting FILT_PER to a non-zero value) introduces a latency of ((FILT_CNT+4) x FILT_PER x IPBus clock period).

Note

When the filter is enabled, there is a combinational path to disable the PWM outputs to ensure rapid response to fault conditions, and fault response if the PWM module loses its clock. The latency induced by the filter will be seen in the time to set FSTS[FFLAG] and FSTS[FFPIN].

73.5.105.3 Diagram



73.5.105.4 Fields

Field	Function
15 GSTR	Fault Glitch Stretch Enable This field is used to enable the fault glitch-stretching logic. This logic ensures that narrow fault glitches are stretched to be at least 2 IPBus clock cycles wide. In some cases, a narrow fault input can cause problems due to the short PWM output shutdown/re-activation time. The stretching logic ensures that a glitch on the fault input, when the fault filter is disabled, will be registered in the fault flags. 0b - Fault input glitch stretching is disabled. 1b - Input fault signals are stretched to at least 2 IPBus clock cycles.
14-11 —	Reserved
10-8 FILT_CNT	Fault Filter Count These bits represent the number of consecutive samples that must agree prior to the input filter accepting an input transition. The number of samples is the decimal value of this field plus three: the bit field value of 0-7 represents 3-10 samples, respectively. The value of FILT_CNT affects the input latency.
7-0 FILT_PER	Fault Filter Period This 8-bit field applies universally to all fault inputs. These bits represent the sampling period (in IPBus clock cycles) of the fault pin input filter. Each input is sampled multiple times at the rate specified by this field. If FILT_PER is 0x00 (default), then the input filter is bypassed. The value of FILT_PER affects the input latency. **NOTE:** When changing values for FILT_PER from one non-zero value to another non-zero value, first write a value of zero to clear the filter.

73.5.106 Fault Test Register (FTST0)

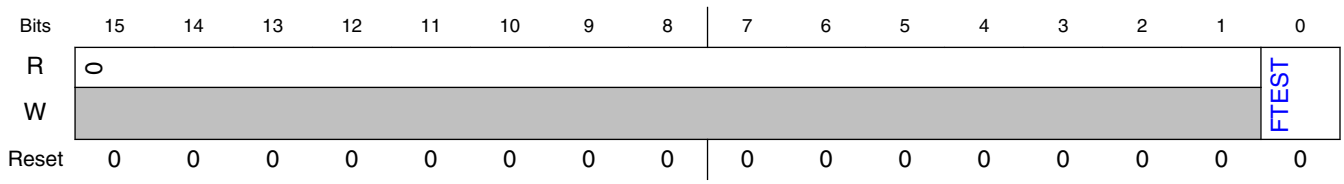
73.5.106.1 Offset

Register	Offset
FTST0	192h

73.5.106.2 Function

Contains FTEST field for fault simulation.

73.5.106.3 Diagram



73.5.106.4 Fields

Field	Function
15-1 —	Reserved
0 FTEST	Fault Test This field is used to simulate a fault condition. Setting this bit causes a simulated fault to be sent into all of the fault filters. The condition propagates to the fault flags and possibly the PWM outputs depending on the DISMAPn settings. Clearing this bit removes the simulated fault condition. 0b - No fault 1b - Cause a simulated fault

73.5.107 Fault Control 2 Register (FCTRL20)

73.5.107.1 Offset

Register	Offset
FCTRL20	194h

73.5.107.2 Function

Controls combinational link from fault inputs to PWM outputs.

73.5.107.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												NOCOMB			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

73.5.107.4 Fields

Field	Function
15-4 —	Reserved
3-0 NOCOMB	No Combinational Path From Fault Input To PWM Output This field is used to control the combinational path from the fault inputs to the PWM outputs. When these bits are low (default), the corresponding fault inputs have a combinational path to the PWM outputs that are sensitive to these fault inputs (as defined by DISMAP0). This combinational path is a safety feature that ensures the output is disabled even if the SOC has a failure of its clocking system. The combinational path also means that a pulse on the fault input can cause a brief disable of the PWM output even if the fault pulse is not wide enough to get through the input filter and be latched in the fault logic. Setting these bits removes the combinational path and uses the filtered and latched fault signals as the fault source to disable the PWM outputs. This eliminates fault glitches from creating PWM output glitches but also increases the latency to respond to a real fault. 0000b - There is a combinational link from the fault inputs to the PWM outputs. The fault inputs are combined with the filtered and latched fault signals to disable the PWM outputs. 0001b - The direct combinational path from the fault inputs to the PWM outputs is disabled and the filtered and latched fault signals are used to disable the PWM outputs.

Chapter 74
General Purpose Timer (GPT)

74.1 Chip-specific GPT information

Table 74-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

74.2 Overview

The General Purpose Timer (GPT) is a 32-bit up-counter with clock source and frequency scaling options to provide a wide range of timing rates for various system applications. External hardware can send the GPT a "capture event" signal on an input pin to load the current timer value into a register for software to read. This capture signal can be configured to trigger the GPT on a rising or/and falling edge. Also, when the timer matches a preset value, the GPT can generate an interrupt for software and a "compare event" signal on output pins for external hardware.

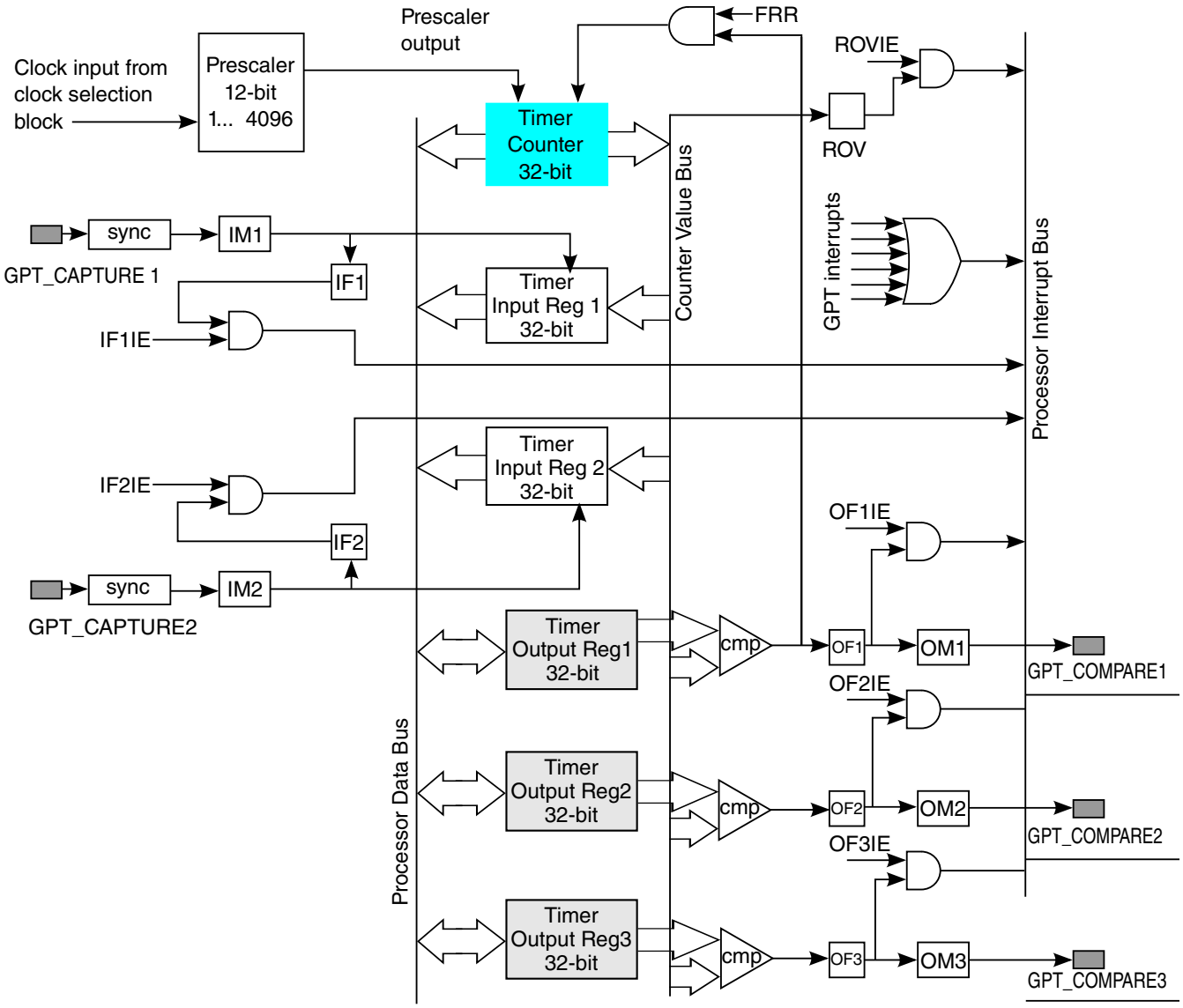


Figure 74-1. GPT Block Diagram

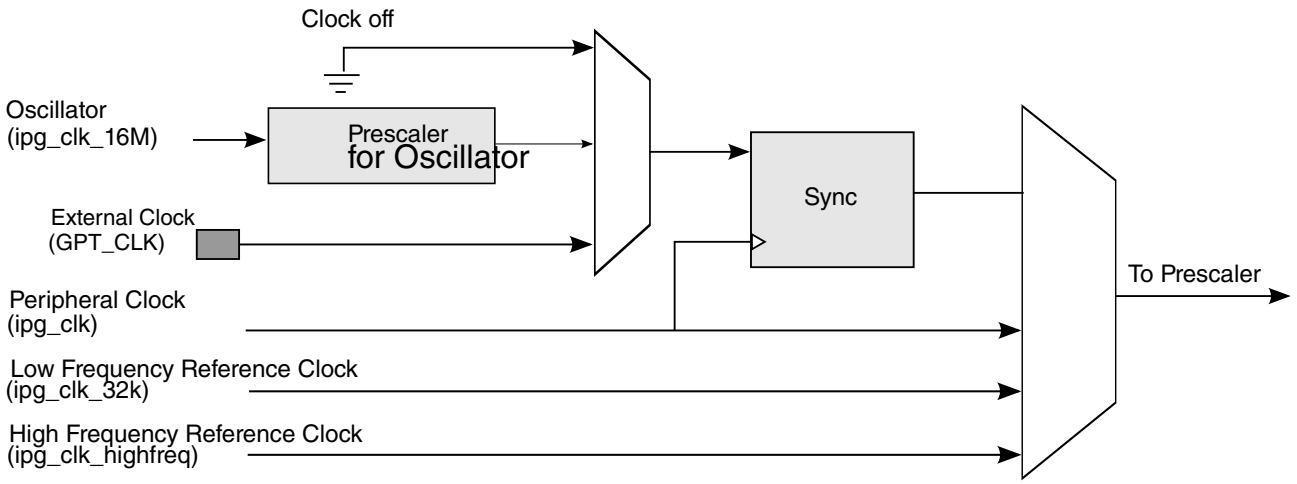


Figure 74-2. GPT Counter Clocks Diagram

74.2.1 Features

- One 32-bit up-counter with clock source selection, including external clock.
- Two input capture channels with a programmable trigger edge.
- Three output compare channels with a programmable output mode. A "forced compare" feature is also available.
- Can be programmed to be *active* in low power and debug modes.
- Interrupt generation at capture, compare, and rollover events.
- Restart or free-run modes for counter operations.

74.3 External Signals

The GPT follows the IP Bus protocol for interfacing with the processor core. The GPT does not have *any interface signals with any other module inside the chip,* except for the clock and reset inputs (from the clock and reset controller module) and for the interrupt signals *to* the processor interrupt handler. There are functional and clock inputs, and functional output signals going outside the chip boundary.

The following table describes all block signals that connect off-chip. Please see IOMUX Controller for GPT pin mux assignments.

Table 74-2. Off-Chip Module Signals

Name	Direction	Function	Reset State	Pull-Up
GPT_CLK	I	Input pin for an external clock that the counter can be operated at.	-	Passive Hysteresis
GPT_CAPTURE1	I	Input pin for a capture event for Input Capture Channel 1.	-	Passive
GPT_CAPTURE2	I	Input pin for a capture event for Input Capture Channel 2.	-	Passive
GPT_COMPARE1	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 1.	0	Passive
GPT_COMPARE2	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 2.	0	Passive
GPT_COMPARE3	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 3.	0	Passive

74.3.1 External Clock Input

The GPT counter can be operated using an external clock from outside the device, and this is the input pin used for that purpose. The external clock input (GPT_CLK) is treated as asynchronous to the peripheral clock (ipg_clk). To ensure proper operations of GPT, the external clock input frequency should be less than 1/4 of frequency of the peripheral clock (ipg_clk). Hysteresis characteristics on this pad will be required because this is a clock input.

74.3.2 Input Capture Trigger Signals

The GPT counter value can be stored in a register, triggered by an event from *outside the device*. A positive / negative edge on these GPT_CAPTURE*n* signals can trigger this capture event. These signals are treated as asynchronous to the peripheral clock (ipg_clk). Only those transitions which occur *at least a single clock cycle* (the clock selected to run the counter) *after the previous recorded transition* are guaranteed to trigger a capture event.

74.3.3 Output Compare Signals

The GPT_COMPARE*n* signals indicate that an output compare event has occurred on the corresponding Output Compare Channel *n*; see Output Compare.

74.4 Clocks

The clock that is input to the prescaler can be selected from multiple clock sources. The following table describes the clock sources for GPT. See Clock Controller Module (CCM) for clock source connection, configuration and gating information.

Table 74-3. GPT Clocks

Clock name	Description
Peripheral Clock (ipg_clk)	This is the internal clock used to access the module registers. If the Peripheral Clock or the External Clock (GPT_CLK) is selected as the Clock Source (CLKSRC=001 or 011), then the Peripheral Clock will be ON in normal GPT operations. In Low Power modes, if the GPT is programmed to be disabled (STOPEN or WAITEN or DOZEN=0), then the Peripheral Clock (ipg_clk) can be switched OFF.
Low Reference Clock (ipg_clk_32k)	This low-frequency 32-kHz reference clock (provided by the CCM) is intended to be ON in Low Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the Low Reference Clock in Low Power mode.
High-Frequency Clock (ipg_clk_highfreq)	Provided by the Clock Controller Module (CCM), this high-frequency reference clock is intended to be ON in Normal Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the High Frequency Clock *in Normal Power mode*.
Oscillator Clock (ipg_clk_16M)	This 16-MHz oscillator reference clock is intended to be used against frequency changes of the Peripheral Clock (ipg_clk) to provide a more accurate timer clock for system operation. The CCM provides the 16 MHz Oscillator Clock *without* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode. Synchronization is done in GPT module. Before synchronization, the 16 MHz Oscillator Clock is divided by a dedicated prescaler to make sure the clock frequency less than half of System Bus Clock (ahb_clk).
External Clock (GPT_CLK)	The External Clock comes from *outside* the device and can be selected to run the GPT counter. The External Clock frequency is limited to < 1/4 frequency of the Peripheral Clock (ipg_clk) for proper GPT operations. Note that in Low Power modes, *if* the Peripheral Clock (ipg_clk) is not available, then the External Clock *cannot be used* to run the counter.

The clock input source is configured using the clock source field (CLKSRC, in the GPT_CR control register). The clock input to the prescaler can be disabled by programming the CLKSRC bits (of the GPT_CR control register) to 000. **The CLKSRC field value should be changed** *only after disabling the GPT* (by setting the EN bit in the GPT_CR to 0).

The PRESCALER field selects the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value (from 1 to 4096) and can be changed *at any time*. A change in the value of the PRESCALER field *immediately affects* the output clock frequency.

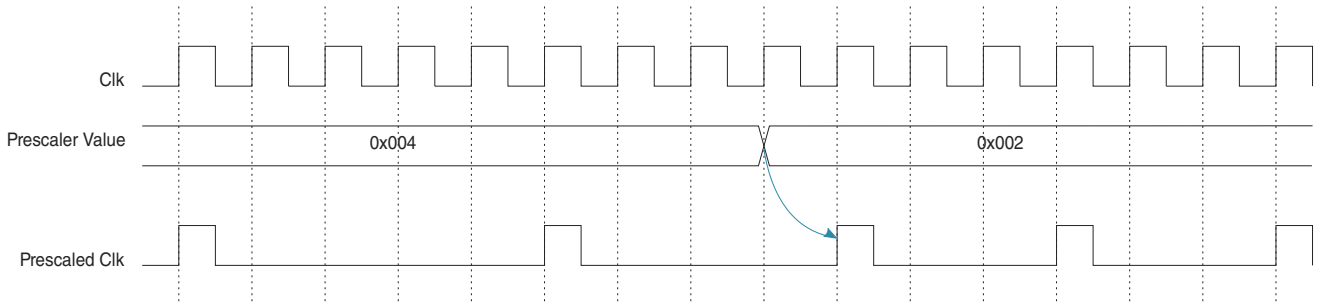


Figure 74-3. Prescaler Value Change Timing Diagram

74.5 Functional Description

This section provides a complete functional description of the GPT.

74.5.1 Operating Modes

The GPT counter can be programmed to work in either of two modes: Restart mode or Free-Run mode.

74.5.1.1 Restart Mode

In Restart mode (selectable through the GPT Control Register GPT_CR), when the counter reaches the compared value, the counter resets and starts again from 0x00000000. The Restart feature is associated only with Compare Channel 1.

Any write access to the Compare register of Channel 1 will reset the GPT counter. This is done to avoid possibly missing a compare event when compare value is changed from a higher value to lower value while counting is proceeding.

For the other two compare channels, when the compare event occurs the counter is *not reset*.

74.5.1.2 Free-Run Mode

In Free-Run mode, when compare events occur for all 3 channels, the counter is *not reset*; instead the counter continues to count until 0xffffffff, and then rolls over (to 0x00000000).

74.5.2 Operation

The General Purpose Timer (GPT) has a single counter (GPT_CNT) that is a 32-bit free-running *up-counter,* which starts counting *after it is enabled by software* (EN=1). The counter's clock source is the output of the prescaler labelled "Prescaler output" in Figure 74-1.

- If the GPT timer is disabled (EN=0), then the Main Counter *and* Prescaler Counter freeze their current count values. The ENMOD bit determines the value of the GPT counter when the EN bit is set and the Counter is enabled again.
  - If the ENMOD bit is set (=1), then the Main Counter and Prescaler Counter values are reset to 0, when GPT is enabled (EN=1).
  - If ENMOD bit is programmed to 0, then the Main Counter and Prescaler Counter restart counting from their frozen values, when GPT is enabled again (EN=1).
- If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter freeze at their current count values *when* GPT enters low power mode. When GPT exits a low power mode, the Main Counter and Prescaler Counter start counting from their frozen values *regardless* of the ENMOD bit value. Note that the GPT_CNT can be read *at any time* by the processor, and that *both* Input Capture Channels use the *same* counter (GPT_CNT).
- A hardware reset resets all the GPT registers to their respective reset values. All registers except the Output Compare Registers (OCR1, OCR2, OCR3) obtain a value of 0x0. The Compare registers are reset to 0xFFFF_FFFF.
- The software reset (SWR bit in the GPT_CR control register) resets *all* of the register bits *except* the EN, ENMOD, STOPEN, WAITEN, and DBGEN bits. The state of these bits is not affected by a software reset. Note that a software reset can be given *while the GPT is disabled.*

74.5.2.1 Input Capture

There are two Input Capture Channels, and each Input Capture Channel has a dedicated capture pin, capture register and input edge detection/selection logic. Each input capture function has an associated status flag, and can cause the processor to make an interrupt service request.

When a selected edge transition occurs on an Input Capture pin, the contents of the GPT_CNT is captured on the corresponding capture register and the appropriate interrupt status flag is set. An interrupt request can be generated when the transition is detected *if* its corresponding enable bit is set (in the Interrupt Register). The capture can be programmed to occur on the input pin's rising edge, falling edge, on both rising and

falling edges, or the capture can be disabled. The events are synchronized with the clock that was selected to run the counter. Only those transitions that occur at least one clock cycle (clock selected to run the counter) *after* the previous recorded transition will be guaranteed to trigger a capture event. There can be up to one clock cycle of uncertainty in the latching of the input transition. The Input Capture registers can be read *at any time* without affecting their values.

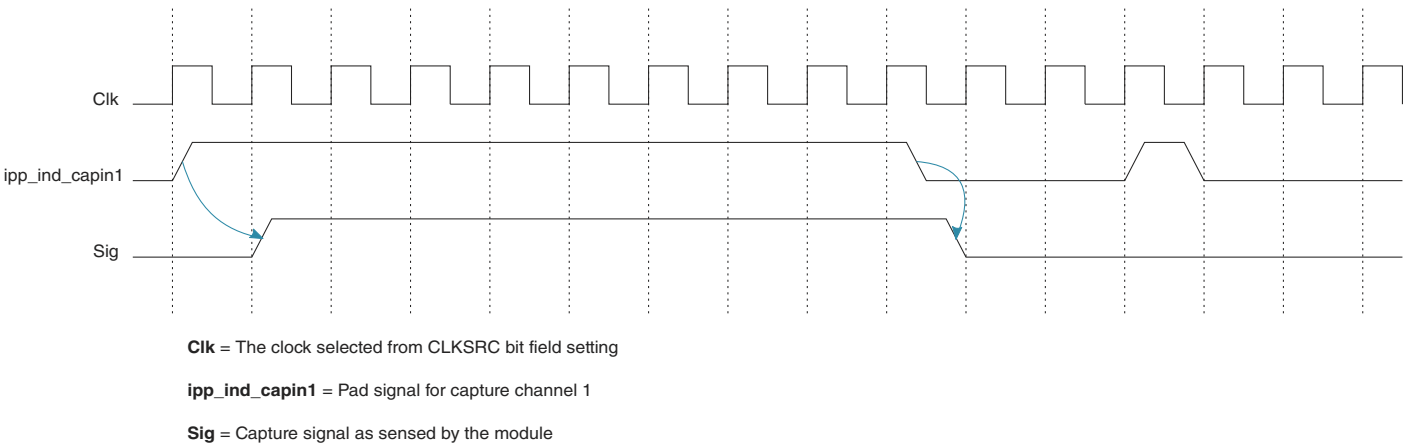


Figure 74-4. Input Capture Event Timing

74.5.2.2 Output Compare

The three Output Compare Channels *use the same counter* (GPT_CNT) as the Input Capture Channels. When the programmed content of an Output Compare register matches the value in GPT_CNT, an output compare status flag is set and an interrupt is generated (if the corresponding bit is set in the interrupt register). Consequently, the Output Compare timer pin is set, cleared, toggled, not affected at all or provide an active-low pulse for one input clock period according to the Output Compare Operating Mode bits in the Control Register (CR[OM*n*]).

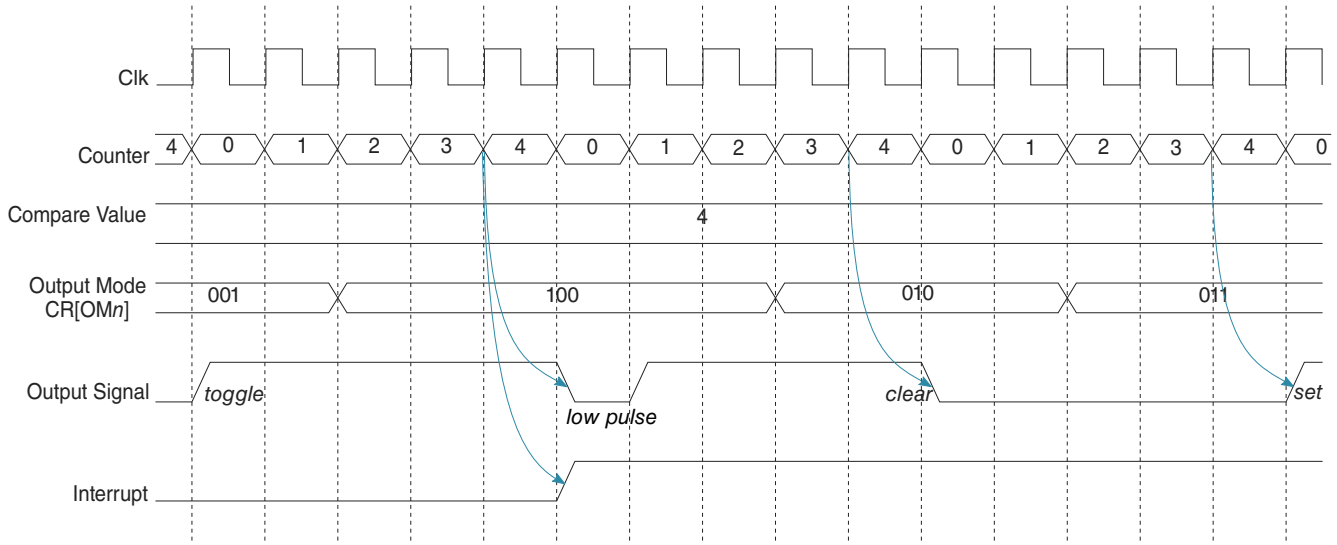


Figure 74-5. Output Compare and Interrupt Timing

There is also a "forced-compare" feature that allows the software to generate a compare event when required, *without the condition of the counter value that is equal to the compare value*. The action taken as a result of a forced compare is the same as when an output compare match occurs, *except that the status flags are not set and no interrupt can be generated*. Forced channels take programmed action immediately after the write to the force-compare bits. These bits are self-negating and always read as zeros.

74.5.2.3 Interrupts

There are 6 different interrupts that are generated by the GPT. If the selected clock for running the counter is available, then *all interrupts can be generated in Low Power and Debug modes*.

- Rollover Interrupt

  The Rollover Interrupt is generated when the GPT counter reaches 0xffffffff, then resets to 0x00000000 and continues counting. The Rollover Interrupt is enabled by the ROVIE bit in the GPT_IR register; the associated status bit is the ROV bit in the GPT_SR register.

- Input Capture Interrupt 1, 2

  After a capture event occurs, the associated Input Capture Channel generates an interrupt. The "capture event" interrupts are enabled by the IF2IE and IF1IE bits (in the GPT_IR register); the associated status bits are IF2 and IF1 (in the GPT_SR register). The capture of the counter value because of a capture event is *not affected*

by a pending capture interrupt. The Capture register is updated with a new counter value when a capture event occurs, regardless of whether that Capture Channels' interrupt has been serviced or not.

- Output Compare Interrupt 1, 2, 3

  After a compare event occurs, the associated Output Compare Channel generates an interrupt. The "compare event" interrupts are enabled by the OF3IE, OF2IE, and OF1IE bits (in the GPT_IR register); the associated status bits are OF3, OF2, and OF1 (in the GPT_SR register). A "forced compare" does not generate an interrupt.

A *cumulative* interrupt line is also present, which is asserted whenever any of the above interrupts are posted. The cumulative interrupt line has *no* associated enables or status bits.

74.5.2.4 Low Power Mode Behavior

In Low Power modes, if the clock from the selected clock source is available (except for the External Clock (GPT_CLK), which can be used *only if* the Peripheral Clock (ipg_clk) is available), the counter will continue to run depending on whether the control bit for that mode is set. If the clock is not present or if the corresponding low power bit in the GPT_CR control register is 0, the Main Counter and the Prescaler Counter freeze at their current values and resume counting (from their frozen values) when the Low Power mode is exited.

74.5.2.5 Debug Mode Behavior

In Debug mode, the modules in the device have the option of continuing to run or be halted.

- If the DBGEN bit is set, then the GPT timer will continue to run in Debug mode.
- If the DBGEN bit is not set (in the GPT_CR control register), then the GPT timer is halted.

74.6 Initialization/ Application Information

74.6.1 Selecting the Clock Source

The Clock Source field in the Control Register (CR[CLKSRC]) selects the clock source. CLKSRC should be changed only after disabling the GPT (CR[EN]=0). To change the clock source, follow this sequence:

1. Disable the GPT: CR[EN]=0
2. Disable GPT interrupts: Clear the Interrupt Register (IR)
3. Configure the Output Mode to unconnected/ disconnected for all channels: CR[OM*n*]=0
4. Disable the Input Capture Modes: CR[IM*n*]=0
5. Change the clock source CR[CLKSRC] to the desired value.
6. Assert a GPT software reset: CR[SWR]=1
7. Clear the flags in the GPT Status Register (SR) by writing 1s to them.
8. Reset the GPT counter to zero: CR[ENMOD]=1
9. Enable the GPT: CR[EN]=1
10. Enable GPT interrupts: For the desired interrupts, write 1s to their enable bits in IR.

74.7 Memory map/register definition

74.7.1 GPT register descriptions

The GPT has 10 user-accessible 32-bit registers used to configure, operate and monitor the GPT.

A write access to the GPT Status Registers (Read-only registers ICR1, ICR2, CNT) generates a bus exception.

74.7.1.1 GPT memory map

GPT1 base address: 400E_C000h

GPT2 base address: 400F_0000h

GPT3 base address: 400F_4000h

GPT4 base address: 400F_8000h

GPT5 base address: 400F_C000h

GPT6 base address: 4010_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	GPT Control Register (CR)	32	RW	0000_0000
4	GPT Prescaler Register (PR)	32	RW	0000_0000
8	GPT Status Register (SR)	32	W1C	0000_0000
C	GPT Interrupt Register (IR)	32	RW	0000_0000
10 - 18	GPT Output Compare Register (OCR1 - OCR3)	32	RW	FFFF_FFFF
1C - 20	GPT Input Capture Register (ICR1 - ICR2)	32	RO	0000_0000
24	GPT Counter Register (CNT)	32	RO	0000_0000

74.7.1.2 GPT Control Register (CR)

The GPT Control Register (CR) contains configuration options for GPT operations, such as selecting the clock source for the counter and selecting a channel's output pin response when a compare event occurs. The CR register also includes the control bit (SWR) to trigger a software reset.

NOTE

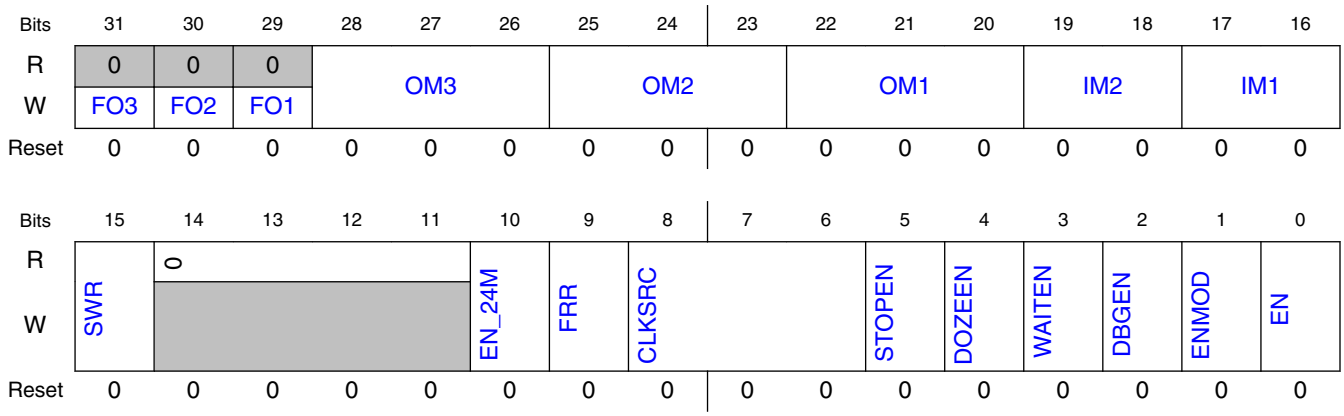
Setting the SWR bit resets **all of the GPT registers** to their default reset values except for the EN, ENMOD, STOPEN, DOZEEN, WAITEN, and DBGEN bits in this register.

For applications where precise timing is critical, be aware that writes to the CR register take effect after one cycle of the module's register bus clock (1 wait state of latency). Reads to the register occur immediately (0 wait states).

74.7.1.2.1 Offset

Register	Offset
CR	0h

74.7.1.2.2 Diagram



74.7.1.2.3 Fields

Field	Description
31-29 FOn	Force Output Compare for Channel n By setting the FO*n* bit, software can force the timer Output Compare *n* pin to respond according to the pin action programmed in the OM*n* bits (in this register). This bit is self-clearing and always reads as zero. **NOTE:** The OF*n* flag in the Status register (SR) is not set. 0 - No effect 1 - Trigger the programmed response on the pin.
28-26: OM3 25-23: OM2 22-20: OM1	Output Compare Operating Mode for Channel n Selects the response on the output pin when a compare event occurs. 000 - Output disabled. No response on pin. 001 - Toggle output pin 010 - Clear output pin 011 - Set output pin 1xx - Generate a low pulse that is one input clock cycle wide on the output pin. When OMn is first programmed as 1xx, the output pin is set to one immediately on the next input clock (if it was not one already). "Input clock" here refers to the clock selected by the CLKSRC field of this register.
19-18: IM2 17-16: IM1	Input Capture Operating Mode for Channel n Selects what transition on the input pin triggers a capture event. 00 - Capture disabled 01 - Capture on rising edge only 10 - Capture on falling edge only 11 - Capture on both edges
15 SWR	Software Reset Setting SWR resets **all of the registers** to their default reset values except for the DOZEEN, EN, ENMOD, STOPEN, WAITEN, and DBGEN bits in this register.

Table continues on the next page...

Field	Description
	The SWR bit remains set while the module is resetting. The SWR bit automatically clears when the reset procedure finishes. 0 - GPT is not in software reset state 1 - GPT is in software reset state
14-11 —	Reserved Software should write 0s to this reserved field. This field always reads as zero.
10 EN_24M	Enable Oscillator Clock Input Enables the 16-MHz clock input from an oscillator. 0 - Disable 1 - Enable
9 FRR	Free-Run or Restart Mode Selects the behavior of the GPT when a compare event in channel 1 occurs. 0 - Restart mode. After a compare event, the counter resets to 0x0000_0000 and resumes counting. 1 - Free-Run mode. After a compare event, the counter continues counting until 0xFFFF_FFFF and then rolls over to 0.
8-6 CLKSRC	Clock Source Select Selects which clock goes to the prescaler and subsequently runs the GPT counter. **NOTE:** The CLKSRC field value should be changed only after disabling the GPT (EN=0). For other programming requirements when changing the clock source, see section Selecting the Clock Source. 000 - No clock 001 - Peripheral Clock (ipg_clk) 010 - High Frequency Reference Clock (ipg_clk_highfreq) 011 - External Clock 100 - Low Frequency Reference Clock (ipg_clk_32k) 101 - Oscillator as Reference Clock (ipg_clk_16M)
5 STOPEN	GPT Stop Mode Enable Enables GPT operation *during Stop mode*. 0 - GPT is disabled in Stop mode. 1 - GPT is enabled in Stop mode.
4 DOZEEN	GPT Doze Mode Enable Enables GPT operation *during Doze mode*. 0 - GPT is disabled in Doze mode. 1 - GPT is enabled in Doze mode.
3 WAITEN	GPT Wait Mode Enable Enables GPT operation *during Wait mode*. 0 - GPT is disabled in Wait mode. 1 - GPT is enabled in Wait mode.
2 DBGEN	GPT Debug Mode Enable Enables GPT operation *during Debug mode*.

Table continues on the next page...

Field	Description
	0 - GPT is disabled in Debug mode. 1 - GPT is enabled in Debug mode.
1 ENMOD	GPT Enable Mode When the GPT is disabled (EN=0), both the Main Counter and Prescaler Counter *freeze their current count values*. The ENMOD bit selects the value of the GPT counter when GPT is enabled again (EN=1). **NOTE:** If GPT is programmed to be disabled in a low power mode (STOP/WAIT), the Main Counter and Prescaler Counter *freeze at their current count values* when the GPT enters low power mode. When GPT exits low power mode, the Main Counter and Prescaler Counter start counting from their frozen values, regardless of the ENMOD bit value. **NOTE:** A software reset (SWR=1) clears the Main Counter and Prescaler Counter values, regardless of the value of EN or ENMOD bits. 0 - The Main Counter and Prescaler Counter restart counting from their frozen values after GPT is enabled (EN=1). 1 - The Main Counter and Prescaler Counter values are reset to 0 after GPT is enabled (EN=1).
0 EN	GPT Enable Enables the GPT module. **NOTE:** Before setting the EN bit, *all registers should be properly programmed*. 0 - GPT is disabled. 1 - GPT is enabled.

74.7.1.3 GPT Prescaler Register (PR)

The GPT Prescaler Register (PR) contains the *divide value* of the clock that runs the counter.

74.7.1.3.1 Offset

Register	Offset
PR	4h

74.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER24M				PRESCALER											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

74.7.1.3.3 Fields

Field	Description
31-16 —	Reserved Software should write 0s to these reserved bits. These bits always read as zero.
15-12 PRESCALER24M	Prescaler divide value for the oscillator clock The 16-MHz oscillator clock is divided by [PRESCALER24M + 1] before selected by the CLKSRC field. If the 16-MHz oscillator clock is not selected, this field takes no effect. 0000 - Divide by 1 0001 - Divide by 2 1111 - Divide by 16
11-0 PRESCALER	Prescaler divide value The clock selected by the CLKSRC field is divided by [PRESCALER + 1], and then used to run the counter. **NOTE:** A change in the value of PRESCALER causes the Prescaler counter to reset and a new count period to start immediately. See Clocks for an example timing diagram. 000000000000 - Divide by 1 000000000001 - Divide by 2 111111111111 - Divide by 4096

74.7.1.4 GPT Status Register (SR)

The GPT Status Register (SR) contains flags to indicate that a counter has rolled over, and if any event has occurred on the Input Capture and Output Compare channels. The flags are cleared by writing a 1 to them.

74.7.1.4.1 Offset

Register	Offset
SR	8h

74.7.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										ROV	IF2	IF1	OF3	OF2	OF1
W											W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

74.7.1.4.3 Fields

Field	Description
31-6 —	Reserved Software should write 0s to these reserved bits. These bits always read as zero.
5 ROV	Rollover Flag ROV indicates that the counter has reached its *maximum possible value* and *rolled over* to 0 (from which the counter continues counting). ROV is set only when the counter has reached 0xFFFFFFFF in both Restart and Free-Run modes. 0 - Rollover has not occurred. 1 - Rollover has occurred.
4-3 IFn	Input Capture Flag for Channel n IF*n* indicates that a capture event has occurred on Input Capture channel *n*. 0 - Capture event has not occurred. 1 - Capture event has occurred.
2-0 OFn	Output Compare Flag for Channel n OF*n* indicates that a compare event has occurred on Output Compare channel *n*. 0 - Compare event has not occurred. 1 - Compare event has occurred.

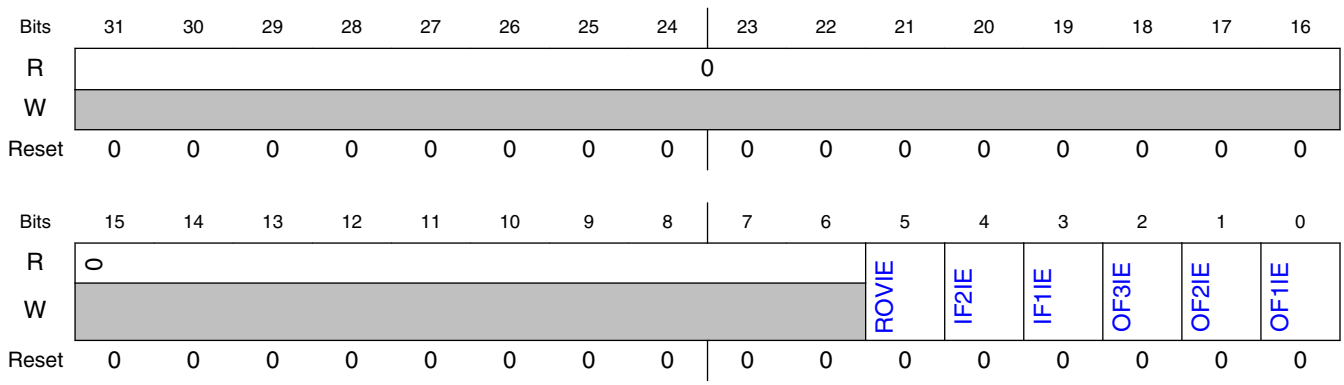
74.7.1.5 GPT Interrupt Register (IR)

Contains enable bits that control whether interrupts are generated after rollover, input capture and output compare events.

74.7.1.5.1 Offset

Register	Offset
IR	Ch

74.7.1.5.2 Diagram



74.7.1.5.3 Fields

Field	Description
31-6 —	Reserved • Writing a value to these reserved bits does not affect GPT operations. • These reserved bits are always read as zero.
5 ROVIE	Rollover Interrupt Enable Enables the Rollover interrupt. 0 - Disable 1 - Enable
4-3 IFnIE	Input Capture Flag for Channel n Interrupt Enable Enables the Input Capture Flag for Channel *n* interrupt. 0 - Disable 1 - Enable
2-0	Output Compare Flag for Channel n Interrupt Enable

Field	Description
OFnIE	Enables the Output Compare Flag for Channel *n* interrupt. 0 - Disable 1 - Enable

74.7.1.6 GPT Output Compare Register (OCR1 - OCR3)

The GPT Output Compare Register *n* (OCR*n*) holds the value that determines when a compare event is generated on the corresponding Output Compare Channel *n*.

NOTE

The behavior of OCR1 is different compared with OCR2 and OCR3:

- A write access to OCR1 while in Restart mode (CR[FRR]=0) resets the GPT counter.
- Writes to OCR1 take effect after one cycle of the module's register bus clock (1 wait state of latency). Reads to the register occur immediately (0 wait states).

74.7.1.6.1 Offset

Register	Offset
OCR1	10h
OCR2	14h
OCR3	18h

74.7.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

74.7.1.6.3 Fields

Field	Description
31-0 COMP	Compare Value When the counter value equals COMP, a compare event is generated.

74.7.1.7 GPT Input Capture Register (ICR1 - ICR2)

The read-only GPT Input Capture Register *n* (ICR*n*) holds the value that was in the counter during the last capture event on the corresponding Input Capture Channel *n*.

74.7.1.7.1 Offset

Register	Offset
ICR1	1Ch
ICR2	20h

74.7.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

74.7.1.7.3 Fields

Field	Description
31-0 CAPT	Capture Value After a capture event on Input Capture Channel *n* occurs, the current value of the counter is loaded into the corresponding ICR*n*[CAPT].

74.7.1.8 GPT Counter Register (CNT)

The read-only GPT Counter Register (CNT) is the main counter's register. CNT can be read at any time *without affecting the counting process* of the GPT.

74.7.1.8.1 Offset

Register	Offset
CNT	24h

74.7.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

74.7.1.8.3 Fields

Field	Description
31-0 COUNT	Counter Value COUNT shows the current value of the GPT counter.

Chapter 75
Periodic Interrupt Timer (PIT)

75.1 Chip-specific PIT information

Table 75-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments
PIT channel to DMA channel correspondence	-	PIT Channel Assignments For Periodic DMA Triggering

There are 4 PIT channels in this device.

75.2 Introduction

PIT includes an array of timers that can be used to raise interrupts and trigger DMA channels. Some of the registers listed in PIT register descriptions are used to configure these timers, and after the expiry of a timer, the respective registers indicate a timeout status, which can be used to raise interrupts and triggers.

75.2.1 Block diagram

The following figure shows the block diagram of PIT module.

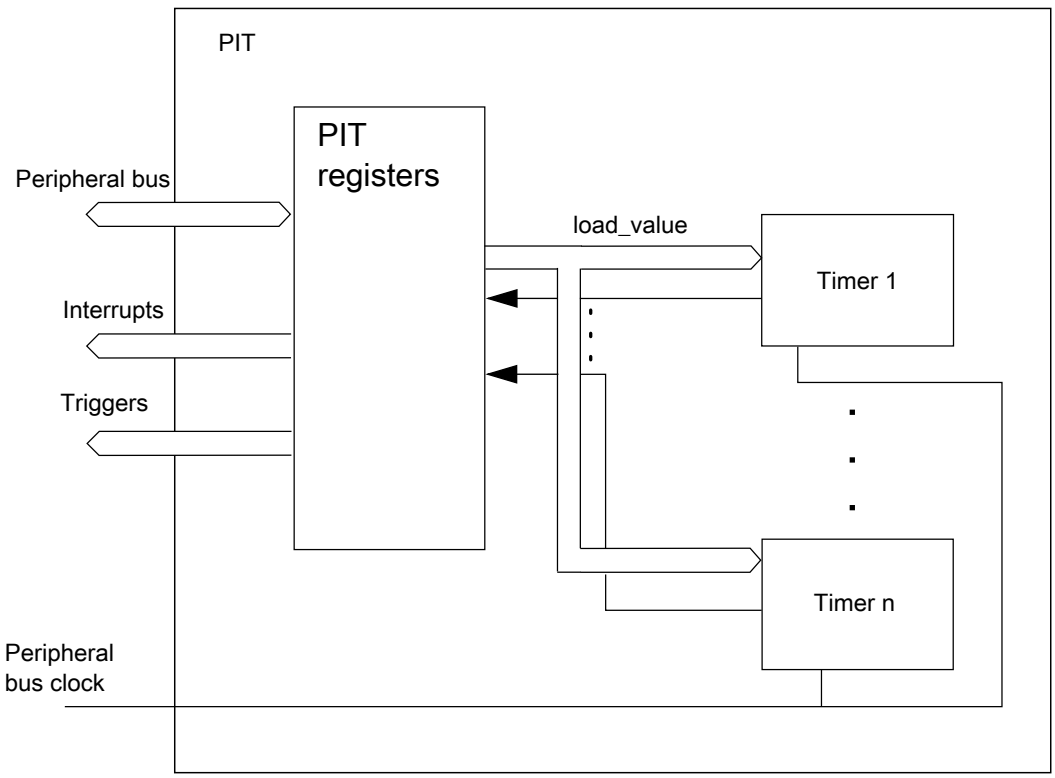


Figure 75-1. Block diagram of PIT

75.2.2 Features

The key features of the module are these:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer
- All the timers use a downcounter

75.3 Modes of operation

This subsection briefly describes all operating modes supported by PIT.

- Run mode

  All functional parts of the PIT are running during normal Run mode.

- Stop mode

75.4 Functional description

This section provides the functional description of the module.

75.4.1 General operation

This section provides detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts, and each interrupt is available on a separate interrupt line.

75.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start values again. Each time a timer reaches 0, it generates a trigger pulse and sets the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted by first disabling and then enabling the timer with TCTRLn[TEN]. See the following figure.

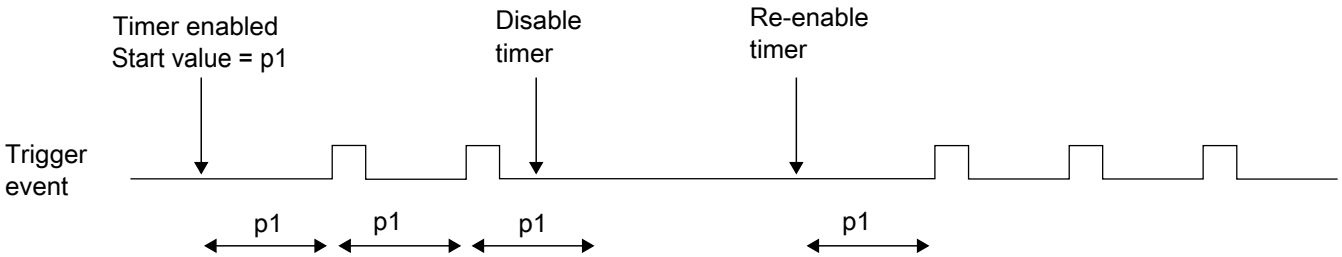


Figure 75-2. Stopping and starting a timer

The counter period of a running timer can be modified by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

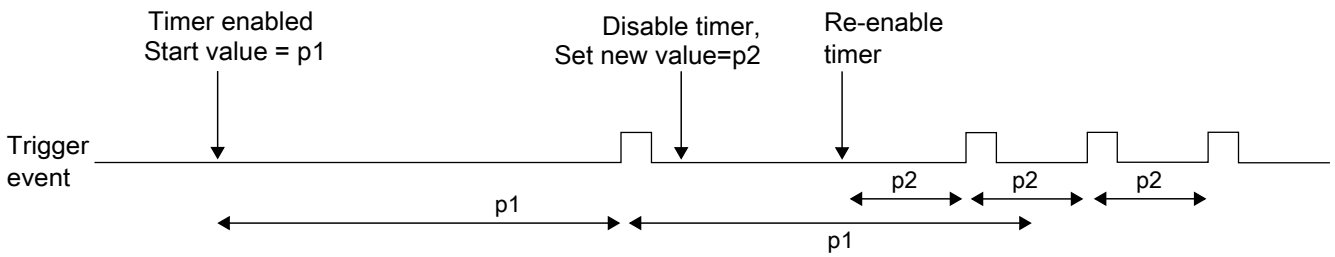


Figure 75-3. Modifying running timer period

It is also possible to change the counter period without restarting the timer, by writing LDVAL with the new load value. This value then loads after the next trigger event. See the following figure.

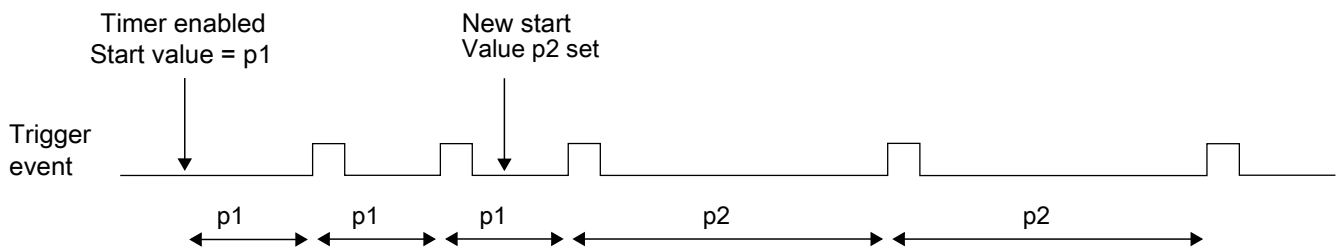


Figure 75-4. Dynamically setting a new load value

NOTE

- If MCR[FRZ] is written as 1 and when the timer is nearing 0 (CVALn =0x0), the debug mode command may not make it to the IP before the timer expires and generates a trigger.

- Debug Mode will be ignored if CVALn =0x0, but the trigger will remain asserted until the mode is removed. The user is recommended to exit Debug mode and then clear the interrupt TFLGn[TIF].
- If the timers are to be freezed for debug, sufficient time must be ensured for the IP to react.

75.4.1.2 Debug mode

In the Debug mode, the timers are frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

75.4.2 Interrupts

All the timers support interrupt generation.

Timer interrupts can be enabled by setting TCTRLn[TIE].

TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

75.4.3 Chained timers

When a timer has chain mode enabled, it counts after the previous timer has expired. So if timer n-1 counts down to 0, counter n decrements the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

See Example configuration for chained timers.

75.4.4 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.

- Timers 1 and 2 are available.
- An interrupt is raised every 1 minute.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN].

TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2


// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

75.4.5 Example configuration for the Lifetime Timer

To configure the Lifetime Timer, channels 0 and 1 need to be chained together.

First, the PIT module needs to be activated by writing a 0 to the MDIS bit in the CTRL register, and then the LDVAL registers need to be set to the maximum value.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
```

```
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1


// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

To access the Lifetime Timer, read first LTMR64H and then LTMR64L.

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

75.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F, respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. Also, TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1
```

```
// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

75.6 Memory Map and Registers

75.6.1 PIT register descriptions

This section provides a detailed description of all registers accessible in the PIT module.

- See the chip-specific PIT information for the number of PIT channels used in this MCU.

75.6.1.1 PIT memory map

PIT1 base address: 400D_8000h

PIT2 base address: 40CB_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	PIT Module Control Register (MCR)	32	RW	0000_0002
E0	PIT Upper Lifetime Timer Register (LTMR64H)	32	RO	0000_0000
E4	PIT Lower Lifetime Timer Register (LTMR64L)	32	RO	0000_0000
100	Timer Load Value Register (LDVAL0)	32	RW	0000_0000
104	Current Timer Value Register (CVAL0)	32	RO	0000_0000
108	Timer Control Register (TCTRL0)	32	RW	0000_0000
10C	Timer Flag Register (TFLG0)	32	W1C	0000_0000
110	Timer Load Value Register (LDVAL1)	32	RW	0000_0000
114	Current Timer Value Register (CVAL1)	32	RO	0000_0000
118	Timer Control Register (TCTRL1)	32	RW	0000_0000
11C	Timer Flag Register (TFLG1)	32	W1C	0000_0000
120	Timer Load Value Register (LDVAL2)	32	RW	0000_0000
124	Current Timer Value Register (CVAL2)	32	RO	0000_0000
128	Timer Control Register (TCTRL2)	32	RW	0000_0000
12C	Timer Flag Register (TFLG2)	32	W1C	0000_0000
130	Timer Load Value Register (LDVAL3)	32	RW	0000_0000
134	Current Timer Value Register (CVAL3)	32	RO	0000_0000
138	Timer Control Register (TCTRL3)	32	RW	0000_0000
13C	Timer Flag Register (TFLG3)	32	W1C	0000_0000

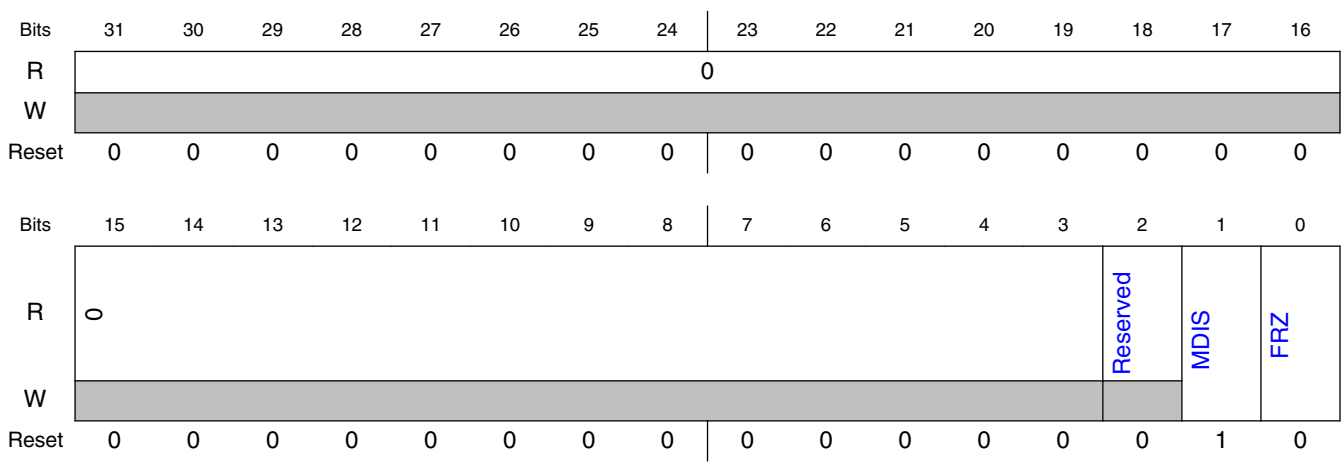
75.6.1.2 PIT Module Control Register (MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

75.6.1.2.1 Offset

Register	Offset
MCR	0h

75.6.1.2.2 Diagram



75.6.1.2.3 Fields

Field	Description
31-3 —	Reserved
2 —	Reserved
1 MDIS	Module Disable for PIT Disables the standard timers. You must write 1 to this field before setting up anything. 0 - Clock for standard PIT timers is enabled.

Table continues on the next page...

Field	Description
	1 - Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 - Timers continue to run in Debug mode. 1 - Timers are stopped in Debug mode.

75.6.1.3 PIT Upper Lifetime Timer Register (LTMR64H)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit Lifetime Timer.

75.6.1.3.1 Offset

Register	Offset
LTMR64H	E0h

75.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LTH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75.6.1.3.3 Fields

Field	Description
31-0 LTH	Life Timer value Shows the timer value of timer 1. If this register is read at a time t1, LTMR64L shows the value of timer 0 at time t1.

75.6.1.4 PIT Lower Lifetime Timer Register (LTMR64L)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit Lifetime Timer.

To use LTMR64H and LTMR64L, timer 0 and timer 1 need to be chained. To obtain the correct value, first read LTMR64H and then LTMR64L. The value for the LTMR64H register is set to CVAL1 register at the time of the first access and the value of the LTMR64L register is set to CVAL0 register at first access. Therefore, the application is not affected by the carry-over effects of the running counter.

75.6.1.4.1 Offset

Register	Offset
LTMR64L	E4h

75.6.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LTL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75.6.1.4.3 Fields

Field	Description
31-0 LTL	Life Timer value Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read.

75.6.1.5 Timer Load Value Register (LDVAL0 - LDVAL3)

These registers select the timeout period for the timer interrupts.

75.6.1.5.1 Offset

Register	Offset
LDVAL0	100h
LDVAL1	110h
LDVAL2	120h
LDVAL3	130h

75.6.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75.6.1.5.3 Fields

Field	Description
31-0 TSV	Timer Start Value Sets the timer start value. The timer counts down until it reaches 0, then generates an interrupt and loads this register value again. Writing a new value to this register does not restart the timer; instead the value is loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

75.6.1.6 Current Timer Value Register (CVAL0 - CVAL3)

These registers indicate the current timer position.

75.6.1.6.1 Offset

Register	Offset
CVAL0	104h
CVAL1	114h
CVAL2	124h
CVAL3	134h

75.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TVL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TVL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75.6.1.6.3 Fields

Field	Description
31-0	Current Timer Value
TVL	Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field because its value is unreliable. • The timer uses a downcounter. The timer values are frozen in the Debug mode if MCR[FRZ] is set.

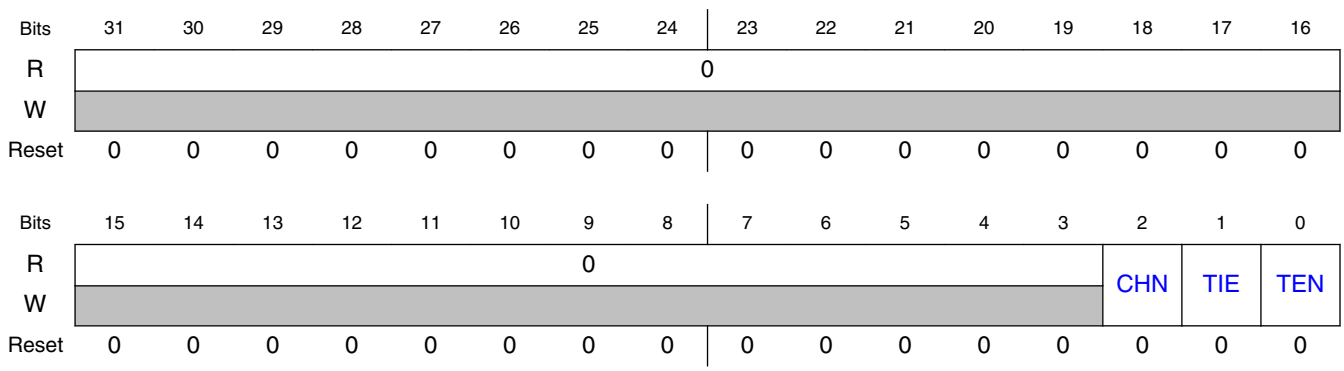
75.6.1.7 Timer Control Register (TCTRL0 - TCTRL3)

These registers contain the control bits for each timer.

75.6.1.7.1 Offset

Register	Offset
TCTRL0	108h
TCTRL1	118h
TCTRL2	128h
TCTRL3	138h

75.6.1.7.2 Diagram



75.6.1.7.3 Fields

Field	Description
31-3 —	Reserved
2 CHN	Chain Mode When activated, timer n-1 needs to expire before timer n (n is > 0) can decrement by 1. Timer 0 cannot be chained. 0 - Timer is not chained. 1 - Timer is chained to a previous timer. For example, for channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or if TFLGn[TIF] is set, enabling the interrupt causes an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 - Interrupt requests from Timer n are disabled. 1 - Interrupt is requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 - Timer n is disabled. 1 - Timer n is enabled.

75.6.1.8 Timer Flag Register (TFLG0 - TFLG3)

These registers hold the PIT interrupt flags.

75.6.1.8.1 Offset

Register	Offset
TFLG0	10Ch
TFLG1	11Ch
TFLG2	12Ch
TFLG3	13Ch

75.6.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

75.6.1.8.3 Fields

Field	Description
31-1 —	Reserved
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it and writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 - Timeout has not yet occurred. 1 - Timeout has occurred.

Chapter 76
Quad Timer (TMR)

76.1 Chip-specific TMR information

Table 76-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

76.2 Overview

Each timer module (TMR) contains four identical counter/timer groups. Each 16-bit counter/timer group contains a prescaler, a counter, a load register, a hold register, a capture register, two compare registers, two status and control registers, and one control register. All of the registers except the prescaler are read/writable.

NOTE

This document uses the terms "Timer" and "Counter" interchangeably because the counter/timers may perform either or both tasks.

The load register provides the initialization value to the counter when the counter's terminal value has been reached.

The hold register captures the counter's value when other counters are being read. This feature supports the reading of cascaded counters.

The capture register enables an external signal to take a "snap shot" of the counter's current value.

The COMP1 and COMP2 registers provide the values to which the counter is compared. If a match occurs, the OFLAG (TMR Output signal) can be set, cleared, or toggled. At match time, an interrupt is generated if enabled, and the new compare value is loaded into the COMP1 or COMP2 registers from CMPLD1 and CMPLD2 if enabled.

The prescaler provides different time bases useful for clocking the counter/timer.

The counter provides the ability to count internal or external events.

Within a timer module (set of four timer/counters), the four inputs are shared by the four counters, but the four outputs are dedicated to each counter.

76.2.1 Block Diagram

Each of the timer/counter groups within the quad-timer are shown in this figure.

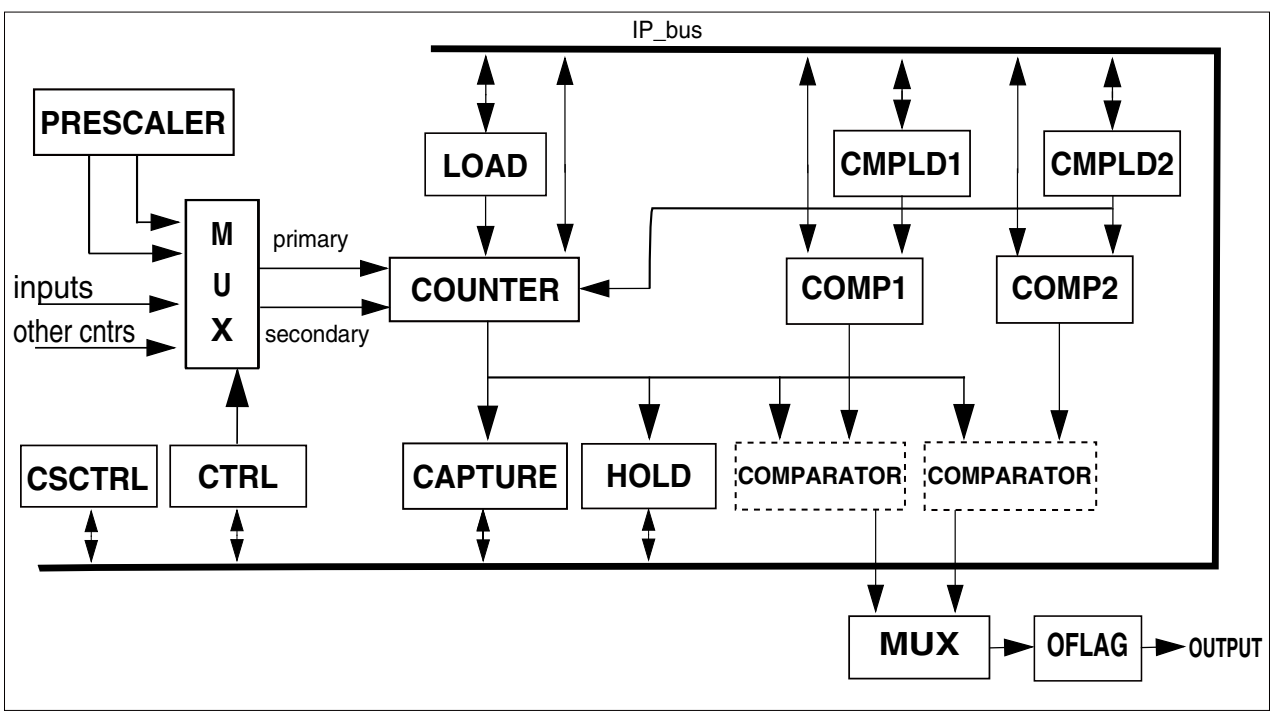


Figure 76-1. Quad Timer Block Diagram

76.2.2 Features

The TMR module design includes these distinctive features:

- Four 16-bit counters/timers
- Count up/down
- Counters are cascadable
- Programmable count modulo
- Max count rate equals BUS_CLOCK_ROOT/2 for external clocks
- Max count rate equals BUS_CLOCK_ROOT for internal clocks
- Count once or repeatedly
- Counters are preloadable
- Compare registers are preloadable (available with compare load feature)
- Counters can share available input pins
- Separate prescaler for each counter
- Each counter has capture and compare capability
- Programmable operation during debug mode
- Inputs may act as fault inputs
- Programmable input filter
- Counting start can be synchronized across counters

76.3 Modes of Operation

The TMR module design operates in only a single mode of operation: Functional Mode. The various counting modes are detailed in the Functional Description.

76.4 Functional Description

76.4.1 General

The counter/timer has two basic modes of operation: it can count internal or external events, or it can count an internal clock source while an external input signal is asserted, thus timing the width of the external input signal.

- The counter can count the rising, falling, or both edges of the selected input pin.
- The counter can decode and count quadrature encoded input signals.
- The counter can count up and down using dual inputs in a "count with direction" format.
- The counter's terminal count value (modulo) is programmable.
    - The value that is loaded into the counter after reaching its terminal count is programmable.
- The counter can count repeatedly, or it can stop after completing one count cycle.
- The counter can be programmed to count to a programmed value and then immediately reinitialize, or it can count through the compare value until the count "rolls over" to zero.

The external inputs to each counter/timer are shareable among each of the four counter/timers within the module. The external inputs can be used as:

- Count commands
- Timer commands
- They can trigger the current counter value to be "captured"
- They can be used to generate interrupt requests

The polarity of the external inputs are selectable.

The primary output of each timer/counter is the output signal OFLAG. The OFLAG output signal can be:

- Set, cleared, or toggled when the counter reaches the programmed value.
- The OFLAG output signal may be output to an external pin instead of having that pin serve as a timer input.
- The OFLAG output signal enables each counter to generate square waves, PWM, or pulse stream outputs.
- The polarity of the OFLAG output signal is selectable.

Any counter/timer can be assigned as a master. A master's compare signal can be broadcast to the other counter/timers within the module. The other counters can be configured to reinitialize their counters and/or force their OFLAG output signals to predetermined values when a master's counter/timer compare event occurs.

76.4.2 Usage of Compare Registers

The dual compare registers (COMP1 and COMP2) provide a bidirectional modulo count capability. The COMP1 register is used when the counter is *counting up*, and the COMP2 register is used when the counter is *counting down*. Alternating compare mode is the only exception.

The COMP1 register should be set to the desired maximum count value or FFFFh to indicate the maximum unsigned value prior to roll-over, and the COMP2 register should be set to the minimum count value or 0000h to indicate the minimum unsigned value prior to roll-under.

If CTRL[OUTMODE] is set to 100, the OFLAG will toggle while using alternating compare registers. In this variable frequency PWM mode, the COMP2 value defines the desired pulse width of the on time, and the COMP1 register defines the off time.

Use caution when changing COMP1 and COMP2 while the counter is active. If the counter has already passed the new value, it will count to FFFFh or 0000h, roll over, then begin counting toward the new value. The check is: Count=CMPx, *not* Count > COMP1 or Count < COMP2.

The use of the CMPLD1 and CMPLD2 registers to compare values will help to minimize this problem.

76.4.3 Usage of Compare Load Registers

The CMPLD1, CMPLD2, and CSCTRL registers offer a high degree of flexibility for loading compare registers with user-defined values on different compare events. To ensure correct functionality while using these registers we strongly suggest using the following method described in this section.

The purpose of the compare load feature is to allow quicker updating of the compare registers. In the past, a compare register could be updated using interrupts. However, because of the latency between an interrupt event occurring and the service of that interrupt, there was the possibility that the counter may have already counted past the

new compare value by the time the compare register was updated by the interrupt service routine. The counter would then continue counting until it rolled over and reached the new compare value.

To address this, the compare registers are now updated in hardware in the same way the counter register is re-initialized to the value stored in the load register. The compare load feature allows the user to calculate new compare values and store them in to the comparator load registers. When a compare event occurs, the new compare values in the comparator load registers are written to the compare registers eliminating the use of software to do this.

The compare load feature is intended to be used in variable frequency PWM mode. The COMP1 register determines the pulse width for the logic low part of OFLAG and COMP2 determines the pulse width for the logic high part of OFLAG. The period of the waveform is determined by the COMP1 and COMP2 values and the frequency of the primary clock source. See the following figure.

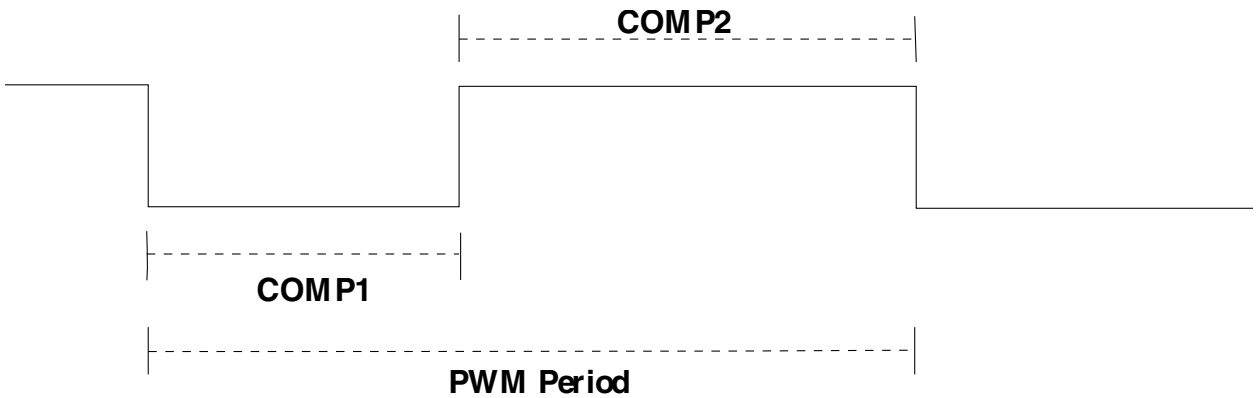


Figure 76-2. Variable PWM Waveform

Should we desire to update the duty cycle or period of the above waveform, we would need to update the COMP1 and COMP2 values using the compare load feature.

76.4.4 Usage of the Capture Register

The capture register stores a copy of the counter's value when an input edge (positive, negative, or both) is detected. After a capture event occurs, no further updating of the capture register will occur until the SCTRL[IEF] (input edge flag) is cleared by writing a zero to the SCTRL[IEF].

76.4.5 Functional Modes

The selected external count signals are sampled at the TMR's base clock rate and then run through a transition detector. The maximum count rate is one-half of the TMR's base clock rate. Internal clock sources can be used to clock the counters at the TMR's base clock rate.

If a counter is programmed to count to a specific value and then stop, the CTRL[CM] field is cleared when the count terminates.

76.4.5.1 Stop Mode

If CTRL[CM] is set to '000', the counter is inert. No counting will occur. Stop mode will also disable the interrupts caused by input transitions on a selected input pin.

76.4.5.2 Count Mode

If CTRL[CM] is set to '001', the counter will count the rising edges of the selected clock source. This mode is useful for generating periodic interrupts for timing purposes, or counting external events such as "widgets" on a conveyor belt passing a sensor. If the selected input is inverted by setting SCTRL[IPS] (input polarity select), then the negative edge of the selected external input signal is counted.

Example: 76.4.5.2.1 Count Pulses from External Source

```
//    This example uses TMR1 to count pulse (actually counts rising edges of the pulse)
//      from an external source (QT3).
//
void Pulse_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=3,SCS=0,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0x0600);           /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x00);
  setReg(TMR1_CNTR,0x00);             /* Reset counter register */
  setReg(TMR1_LOAD,0x00);             /* Reset load register */
  setRegBitGroup(TMR1_CTRL,CM,0x01);  /* Run counter */
}
```

Example: 76.4.5.2.2 Generate Periodic Interrupt By Counting Internal Clocks

```
//   This example generates an interrupt every 100ms,
//     assuming the chip is operating at 60 MHz.
//
```

```
// It does this by using the IP_bus_clk divided by 128 as the counter clock source.
// The counter then counts to 46874 where it matches the COMP1 value.
// At that time an interrupt is generated, the counter is reloaded and
// the next COMP1 value is loaded from CMPLD1.
//
void TimerInt_Init(void)
{
  /* TMR0_CTRL: CM=0,PCS=0,SCS=0,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR0_CTRL,0x20);              /* Stop all functions of the timer */
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR0_SCTRL,0x00);
  setReg(TMR0_LOAD,0x00);             /* Reset load register */
  setReg(TMR0_COMP1,46874);           /* Set up compare 1 register */
  setReg(TMR0_CMPLD1,46874);          /* Also set the compare preload register */
  /* TMR0_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,TCF2EN=0,TCF1EN=1,
                  TCF2=0,TCF1=0,CL2=0,CL1=1 */
  setReg(TMR0_CSCTRL,0x41);           /* Enable compare 1 interrupt and */
                                       /*   compare 1 preload */
  setRegBitGroup(TMR0_CTRL,PCS,0xF); /* Primary Count Source to IP_bus_clk / 128 */
  setReg(TMR0_CNTR,0x00);             /* Reset counter register */
  setRegBitGroup(TMR0_CTRL,CM,0x01); /* Run counter */
}
```

76.4.5.3 Edge-Count Mode

If CTRL[CM] is set to '010', the counter will count both edges of the selected external clock source. This mode is useful for counting the changes in the external environment, such as a simple encoder wheel.

Example: 76.4.5.3.1 Count Both Edges of External Source Signal

```
//  This example uses TMR1 to count pulse (actually counts both edges of the pulse)
//    from an external source (QT3).
//
void Pulse_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=3,SCS=0,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0x0600);           /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x00);
  setReg(TMR1_CNTR,0x00);             /* Reset counter register */
  setReg(TMR1_LOAD,0x00);             /* Reset load register */
  setRegBitGroup(TMR1_CTRL,CM,0x02);  /* Run counter */
}
```

76.4.5.4 Gated-Count Mode

If CTRL[CM] is set to '011', the counter will count while the selected secondary input signal is high. This mode is used to time the duration of external events. If the selected input is inverted by setting SCTRL[IPS] (input polarity select), then the counter will count while the selected secondary input is low.

Example: 76.4.5.4.1 Capture Duration of External Pulse

```
//   This example uses TMR1 to determine the duration of an external pulse.
//
//   The IP_bus clock is used as the primary counter. If the duration of the
//     external pulse is longer than 0.001 seconds one of the other IP_bus clock
//     dividers can be used. If the pulse duration is longer than 0.128 seconds
//     an external clock source will have to be used as the primary clock source.
//
void Pulse1_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=8,SCS=1,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0x1080);            /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x00);
  setReg(TMR1_CNTR,0x00);             /* Reset counter register */
  setReg(TMR1_LOAD,0x00);             /* Reset load register */
  setRegBitGroup(TMR1_CTRL,CM,0x03);  /* Run counter */
}
```

76.4.5.5 Quadrature-Count Mode

If CTRL[CM] is set to '100', the counter will decode the primary and secondary external inputs as quadrature encoded signals. Quadrature signals are usually generated by rotary or linear sensors used to monitor movement of motor shafts or mechanical equipment. The quadrature signals are square waves that are 90 degrees out of phase. The decoding of quadrature signal provides both count and direction information.

This figure shows a timing diagram illustrating the basic operation of a quadrature incremental position encoder.

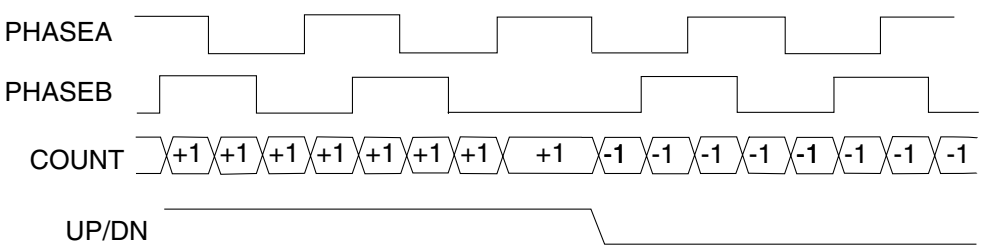


Figure 76-3. Quadrature Incremental Position Encoder

Example: 76.4.5.5.1 Quadrature Count Mode Example

```
//  This example uses TMR0 for counting states of a quadrature position encoder.
//
//  Timer input 0 is used as the primary count source (PHASEA).
//  Timer input 1 is used as the secondary count source (PHASEB).
//
void Pulse_Init(void)
{
  /* TMR0_CTRL: CM=0,PCS=0,SCS=1,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
```

```
  setReg(TMRC0_CTRL,0x80);              /* Set up mode */
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=1 */
  setReg(TMR0_SCTRL,0x00);
  setReg(TMR0_CNTR,0x00);             /* Reset counter register */
  setReg(TMR0_LOAD,0x00);             /* Reset load register */
  setReg(TMR0_COMP1,0xFFFF);           /* Set up compare 1 register */
  setReg(TMR0_COMP2,0x00);             /* Set up compare 2 register */
  /* TMR0_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR0_CSCTRL,0x00);
  setRegBitGroup(TMR0_CTRL,CM,0x04);  /* Run counter */
}
```

76.4.5.6 Quadrature-Count Mode with Index Input

As an extension to the quadrature count mode discussed in the previous paragraph, some rotary shafts have a HOME or INDEX indicator. This would be a third input to the timer that is used to reset the timer's counter.

In this example, channel 0 is used to decode the quadrature inputs, but it doesn't actually count. Because its upper and lower limits are both set to 0, its output is cascaded count up and count down signals each time the quadrature inputs indicate a change in count. Channel 1 works in cascaded count mode receiving its counting instructions from channel 0. When an input capture event occurs, channel 1 is programmed to reset its counter value. The channel 1 counter contains the position value for the shaft.

Example: 76.4.5.6.1 Quadrature Count Mode with Index Input Example

```
//  This example uses TMR0and TMR1 for counting states of a quadrature position encoder.
//
// Timer input 0 is used as the primary count source (PHASEA).
// Timer input 1 is used as the secondary count source (PHASEB).
// Timer input 2 is used as the index input source (INDEX).
//
void Pulse_Init(void)
{
  /* TMR0_CTRL: CM=0,PCS=0,SCS=1,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR0_CTRL,0xA0);              /* Set up mode */
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR0_SCTRL,0x00);
  setReg(TMR0_CNTR,0x00);             /* Reset counter register */
  setReg(TMR0_LOAD,0x00);             /* Reset load register */
  setReg(TMR0_COMP1,0x00);           /* Set up compare 1 register */
  setReg(TMR0_COMP2,0x00);             /* Set up compare 2 register */
  /* TMR0_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR0_CSCTRL,0x00);
/* TMR1_CTRL: CM=7,PCS=100,SCS=2,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0xEA00);            /* Set up capture edge */
/* TMR1_SCTRL: Capture_Mode=10 */
  setReg(TMR1_SCTRL,0x0080);
/* TMR1_CSTRL: ROC=1 */
  setReg(TMR1_CTRL,0x0800);            /* Set up reload on capture */
setRegBitGroup(TMR0_CTRL,CM,0x04);  /* Run counter */
}
```

76.4.5.7 Signed-Count Mode

If CTRL[CM] is set to '101', the counter counts the primary clock source while the selected secondary source provides the selected count direction (up/down).

Example: 76.4.5.7.1 Signed Count Mode Example

```
//  This example uses TMR0 for signed mode counting.
//
//  Timer input 2 is used as the primary count source.
//  Timer input 1 is used to determine the count direction.
//
void Pulse_Init(void)
{
  /* TMR0_CTRL: CM=0,PCS=2,SCS=1,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR0_CTRL,0x0480);              /* Set up mode */
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=1,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR0_SCTRL,0x1000);
  setReg(TMR0_CNTR,0x00);              /* Reset counter register */
  setReg(TMR0_LOAD,0x00);              /* Reset load register */
  setRegBitGroup(TMR0_CTRL,CM,0x05);  /* Run counter */
}
```

76.4.5.8 Triggered-Count Mode 1

If CSCTRL[TCI] is clear and CTRL[CM] is set to '110', the counter will begin counting the primary clock source after a positive transition (negative edge if SCTRL[IPS]=1) of the secondary input occurs. The counting will continue until a compare event occurs or another positive input transition is detected. If a second input transition occurs before a terminal count is reached, counting will stop and SCTRL[TCF] (timer compare flag) will be set. Subsequent secondary input transitions will continue to restart and stop the counting until a compare event occurs.

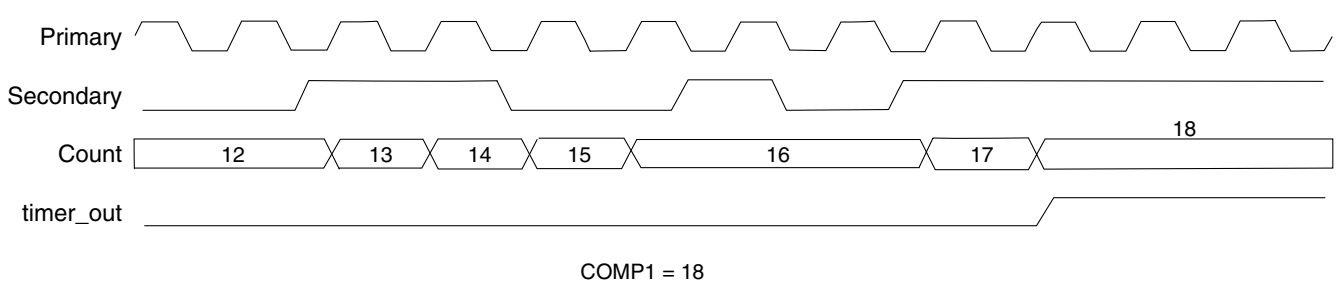


Figure 76-4. Triggered Count Mode 1 (CTRL[LENGTH]=0)

Example: 76.4.5.8.1 Triggered Count Mode 1 Example

```
//   This example uses TMR1 for triggered mode counting.
//
//   Timer input 3 is used as the primary count source.
//   Timer input 2 is used for the trigger input.
//
void Pulse_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=3,SCS=2,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0x0700);             /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=1,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x1000);
  setReg(TMR1_CNTR,0x00);               /* Reset counter register */
  setReg(TMR1_LOAD,0x00);               /* Reset load register */
  setReg(TMR1_COMP1,0x0012);             /* Set up compare 1 register */
  /* TMR1_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR1_CSCTRL,0x00);
  setRegBitGroup(TMR1_CTRL,CM,0x06);    /* Run counter */
}
```

76.4.5.9 Triggered-Count Mode 2

If CSCTRL[TCI] is set and CTRL[CM] is set to '110', the counter will begin counting the primary clock source after a positive transition (negative edge if SCTRL[IPS]=1) of the secondary input occurs. The counting will continue until a compare event occurs or another positive input transition is detected. If a second input transition occurs before a terminal count was reached, the counter will reload and continue counting. When CSCTRL[TCI] is set, the OFLAG output mode, CTRL[OUTMODE], should probably be set to '101' (cleared on init, set on compare) to ensure the output will be in a known state after the second input transition and subsequent reload takes place.

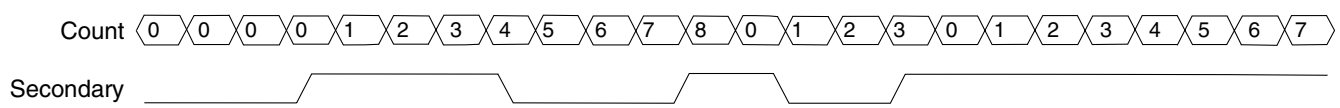


Figure 76-5. Triggered Count Mode 2 (CTRL[LENGTH]=0)

Example: 76.4.5.9.1 Triggered Count Mode 2Example

```
//   This example uses TMR1 for triggered mode counting.
//
//   Timer input 3 is used as the primary count source.
//   Timer input 2 is used for the trigger input.
//
void Pulse_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=3,SCS=2,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR1_CTRL,0x0700);             /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=1,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x1000);

  setReg(TMR1_CNTR,0x00);               /* Reset counter register */
```

```
  setReg(TMR1_LOAD,0x00);             /* Reset load register */
  setReg(TMR1_COMP1,0x0012);           /* Set up compare 1 register */
  /* TMR1_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR1_CSCTRL,0x00);
  setRegBitGroup(TMR1_CTRL,CM,0x06); /* Run counter */
}
```

76.4.5.10 One-Shot Mode

If CTRL[CM] is set to '110', and the counter is set to reinitialize at a compare event (CTRL[LENGTH]=1),and CTRL[OUTMODE] is set to '101' (cleared on init, set on compare), the counter works in a one-shot mode. An external event causes the counter to count, and when the terminal count is reached, the output is asserted. This delayed output can be used to provide timing delays.

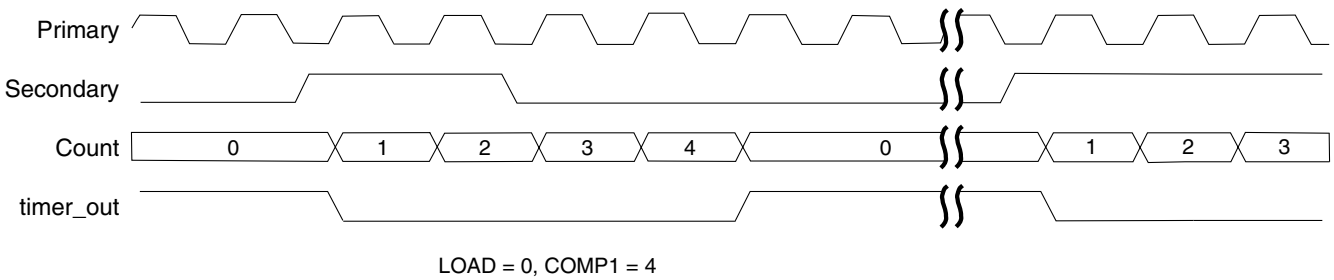


Figure 76-6. One-Shot Mode (CTRL[LENGTH]=1)

Example: 76.4.5.10.1 One-Shot Mode Example

```
//  This example uses TMR1 for one-shot mode counting.
//
//  Timer input 3 is used as the primary count source.
//  Timer input 2 is used for the trigger input.
//
void Pulse_Init(void)
{
  /* TMR1_CTRL: CM=0,PCS=3,SCS=2,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=5 */
  setReg(TMR1_CTRL,0x0725);           /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=1,IEF=0,IEFIE=0,IPS=0,INPUT=0,
              Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR1_SCTRL,0x1000);
  setReg(TMR1_CNTR,0x00);             /* Reset counter register */
  setReg(TMR1_LOAD,0x00);             /* Reset load register */
  setReg(TMR1_COMP1,0x0004);           /* Set up compare 1 register */
  /* TMR1_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR1_CSCTRL,0x00);
  setRegBitGroup(TMR1_CTRL,CM,0x06);  /* Run counter */
}
```

76.4.5.11 Cascade-Count Mode

If CTRL[CM] is set to '111', the counter's input is connected to the output of another selected counter. The counter will count up and down as compare events occur in the selected source counter. This cascade or daisy-chained mode enables multiple counters to be cascaded to yield longer counter lengths. When operating in cascade mode, a special high-speed signal path is used between modules rather than the OFLAG output signal. If the selected source counter is counting up and it experiences a compare event, the counter will be incremented. If the selected source counter is counting down and it experiences a compare event, the counter will be decremented.

Up to four counters may be cascaded to create a 64-bit wide synchronous counter. Check the data sheet to see if there are any frequency limits for cascaded counting mode.

Whenever any counter is read within a counter module, all of the counters' values within the module are captured in their respective hold registers. This action supports the reading of a cascaded counter chain. First read any counter of a cascaded counter chain, then read the hold registers of the other counters in the chain. The cascaded counter mode is synchronous.

Note

It is possible to connect counters together by using the other (non-cascade) counter modes and selecting the outputs of other counters as a clock source. In this case, the counters are operating in a ripple mode, where higher order counters will transition a clock later than a purely synchronous design.

Example: 76.4.5.11.1 Generate Periodic Interrupt Cascading Two Counters

```
// This example generates an interrupt every 30 seconds,
//    assuming the chip is operating at 60 MHz.
//
// To do this, counter 2 is used to count 60,000 IP_bus clocks, which means it
//    will compare and reload every 0.001 seconds.
// Counter 3 is cascaded and used to count the 0.001 second ticks and
//    generate the desired interrupt interval.
//
void TimerInt_Init(void)
{
  // Set counter 2 to count IP_bus clocks
  /* TMR2_CTRL: CM=0,PCS=8,SCS=0,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR2_CTRL,0x1020);           /* Stop all functions of the timer */
  // Set counter 3 as cascaded and to count counter 2 outputs
  /* TMR3_CTRL: CM=7,PCS=6,SCS=0,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=0 */
  setReg(TMR3_CTRL,0xEC20);            /* Set up cascade counter mode */
  /* TMR3_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR3_SCTRL,0x00);
  /* TMR2_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
  setReg(TMR2_SCTRL,0x00);
```

```
  setReg(TMR3_CNTR,0x00);             /* Reset counter register */
  setReg(TMR2_CNTR,0x00);
  setReg(TMR3_LOAD,0x00);             /* Reset load register */
  setReg(TMR2_LOAD,0x00);
  setReg(TMR3_COMP1, 30000-1);        /* milliseconds in 30 seconds */
  setReg(TMR3_CMPLD1,30000-1);
  setReg(TMR2_COMP1, 60000-1);        /* Set to cycle every milisecond */
  setReg(TMR2_CMPLD1,60000-1);
  /* TMR3_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=1,TCF2=0,TCF1=0,CL2=0,CL1=1 */
  setReg(TMR3_CSCTRL,0x41);           /* Enable compare 1 interrupt and */
                                      /* compare 1 preload              */
  /* TMR2_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=1 */
  setReg(TMR2_CSCTRL,0x01);           /* Enable Compare 1 preload */
  setRegBitGroup(TMR2_CTRL,CM,0x01);  /* Run counter */
}
```

76.4.5.12 Pulse-Output Mode

If CTRL[CM]=001, and CTRL[OUTMODE] is set to 111' (gated clock output), and CTRL[ONCE] is set, then the counter will output a pulse stream of pulses that has the same frequency of the selected clock source, and the number of output pulses is equal to the compare value minus the init value. This mode is useful for driving step motor systems.

Note

This does not work if CTRL[PCS] is set to 1000 (IP_bus/1).

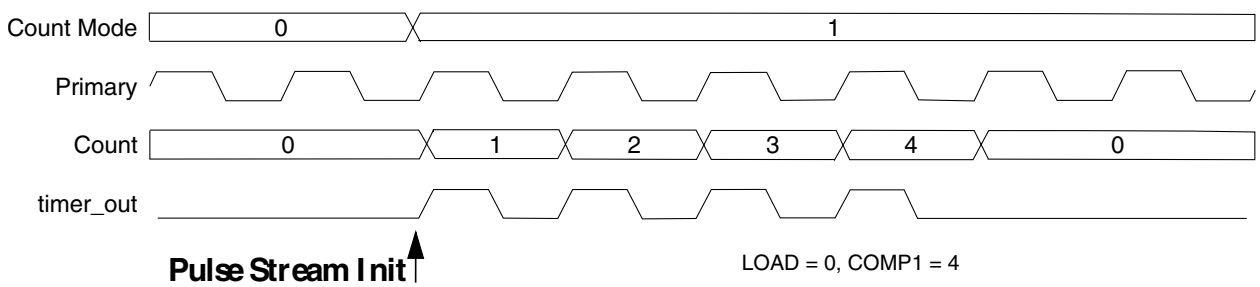


Figure 76-7. Pulse Output Mode

Example: 76.4.5.12.1 Pulse Outputs Using Two Counters

```
// This example generates six 10ms pulses, from QT1 output.
// Assuming the chip is operating at 60 MHz.
//
// To do this, timer 3 is used to generate a clock with a period of 10ms.
//
// Timer 1 is used to gate these clocks and count the number of pulses that have
//  been generated.
//
void PulseStream_Init(void)
{
// Select IP_bus_clk/16 as the clock source for Timer 3
```

```
  /* TMR3_CTRL: CM=0,PCS=0x0C,SCS=0,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=3 */
   setReg(TMR3_CTRL,0x1823);             /* Set up mode */
  /* TMR3_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=0 */
   setReg(TMR3_SCTRL,0x00);
   setReg(TMR3_LOAD,0x00);              /* Reset load register */
   setReg(TMR3_COMP1,37500-1);          /* (16 * 37500 ) / 60e6 = 0.01 sec */
 /* TMR3_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                   TCF2EN=0,TCF1EN=0,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR3_CSCTRL,0x00);             /* Set up comparator control register */



// Timer 3 output is the clock source for this timer.
  /* TMR1_CTRL: CM=0,PCS=7,SCS=0,ONCE=1,LENGTH=1,DIR=0,COINIT=0,OUTMODE=7 */
  setReg(TMR1_CTRL,0x0E67);             /* Set up mode */
  /* TMR1_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=0,OPS=0,OEN=1 */
   setReg(TMR1_SCTRL,0x01);
   setReg(TMR1_CNTR,0x00);              /* Reset counter register */
   setReg(TMR1_LOAD,0x00);              /* Reset load register */
   setReg(TMR1_COMP1,0x04);             /* Set up compare 1 register */

// set to interrupt after the last pulse
  /* TMR1_CSCTRL: DBG_EN=0,FAULT=0,ALT_LOAD=0,ROC=0,TCI=0,UP=0,OFLAG=0,
                  TCF2EN=0,TCF1EN=1,TCF2=0,TCF1=0,CL2=0,CL1=0 */
  setReg(TMR1_CSCTRL,0x40);             /* Set up comparator control register */
// Finally, start the counters running
  setReg(TMR3_CNTR,0);                  /* Reset counter */
  setRegBitGroup(TMR3_CTRL,CM,0x01);    /* Run source clock counter */
  setRegBitGroup(TMR1_CTRL,CM,0x01);    /* Run counter */
}
```

76.4.5.13 Fixed-Frequency PWM Mode

If CTRL[CM]=001, count through roll-over (CTRL[LENGTH]=0), continuous count (CTRL[ONCE]=0) and CTRL[OUTMODE] is '110' (set on compare, cleared on counter roll-over), then the counter output yields a pulse-width modulated (PWM) signal with a frequency equal to the count clock frequency divided by 65,536 and a pulse-width duty cycle equal to the compare value divided by 65,536. This mode of operation is often used to drive PWM amplifiers used to power motors and inverters.

Example: 76.4.5.13.1 Fixed-Frequency PWM Mode Example

```
// This example uses TMR0 for Fixed-Frequency PWM mode timing.
//
// The timer will count IP_bus clocks continuously until it rolls over.
// This results in a PWM period of 65536 / 60e6 = 1092.267 usec
//
// Initially, an output pulse width of 25 usec is generated ( 1500 / 60e6 )
//   giving a PWM ratio of 1500 / 65536 = 2.289%
// This pulse width can be changed by changing the COMP1 register value (using CMPLD1).
//
void PWM1_Init(void)
{
  setReg(TMR0_CNTR,0);           /* Reset counter */
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=1,OPS=0,OEN=1 */
```

```
  setReg(TMR0_SCTRL,0x05);       /* Enable output */
  setReg(TMR0_COMP1,1500-1);     /* Store initial value to the duty-compare register */
  /* TMR0_CTRL: CM=1,PCS=8,SCS=0,ONCE=0,LENGTH=0,DIR=0,COINIT=0,OUTMODE=6 */
  setReg(TMR0_CTRL,0x3006);      /* Run counter */
}
```

76.4.5.14 Variable-Frequency PWM Mode

If CTRL[CM]=001, count until compare (CTRL[LENGTH]=1), continuous count (CTRL[ONCE] = 0) and CTRL[OUTMODE] is '100' (toggle OFLAG and alternate compare registers), then the counter output yields a pulse-width modulated (PWM) signal whose frequency and pulse width is determined by the values programmed into the COMP1 and COMP2 registers, and the input clock frequency. This method of PWM generation has the advantage of allowing almost any desired PWM frequency and/or constant on or off periods. This mode of operation is often used to drive PWM amplifiers used to power motors and inverters. The CMPLD1 and CMPLD2 registers are especially useful for this mode, as they allow the programmer time to calculate values for the next PWM cycle while the PWM current cycle is underway.

To set up the TMR to run in variable frequency PWM mode with compare preload please use the following setup for the specific counter you would like to use. When performing the setup, update the TMR_CTRL register last because the counter will start counting if the count mode is changed to any value other than 000 (assuming the primary count source is already active).

Timer Control Register (CTRL)

- CM=001 (count rising edges of primary source)
- PCS=1000 (IP bus clock for best granularity for waveform timing)
- SCS=Any (ignored in this mode)
- ONCE=0 (want to count repeatedly)
- LENGTH=1 (want to count until compare value is reached and re-initialize counter register)
- DIR=Any (user's choice. The compare register values must be chosen carefully to account for things like roll-under, etc.)
- COINIT=0 (user can set this if they need this function)
- OUTMODE=100 (toggle OFLAG output using alternating compare registers)

Timer Status and Control Register (SCTRL)

- OEN = 1 (output enable to allow OFLAG output to be put on an external pin. Set this bit as needed.)
- OPS = Any (user's choice)
- Make sure the rest of the bits are cleared for this register. We will enable interrupts in the comparator status and control register instead of in this register.

Comparator Status and Control Register (CSCTRL)

- TCF2EN=1 (allow interrupt to be issued when CSCTRL[TCF2] is set)
- TCF1EN=0 (do not allow interrupt to be issued when CSCTRL[TCF1] is set)
- TCF1=0 (clear timer compare 1 interrupt source flag. This is set when counter register equals compare register 1 value and OFLAG is low)
- TCF2=0 (clear timer compare 2 interrupt source flag. This is set when counter register equals compare register 2 value and OFLAG is high)
- CL1=10 (load compare register when CSCTRL[TCF2] is asserted)
- CL2=01 (load compare register when CSCTRL[TCF1] is asserted)

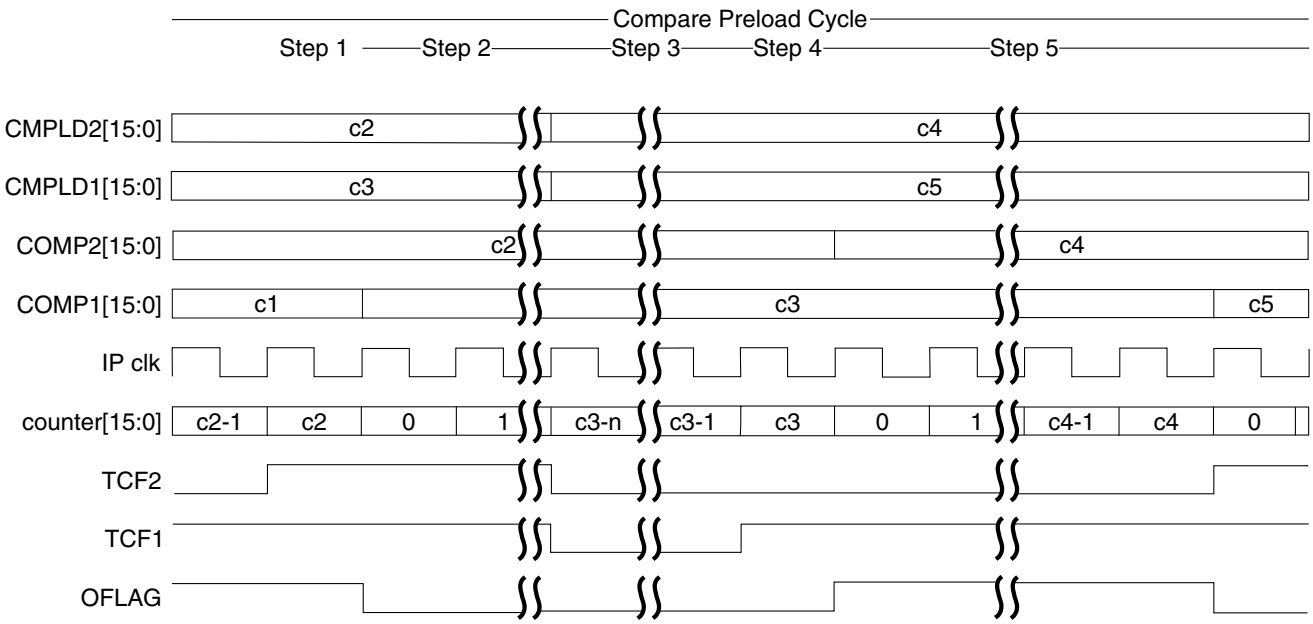
Interrupt Service Routines

To service the CSCTRL[TCF2] interrupts generated by the timer, the interrupt controller must be configured to enable the interrupts for the particular timer being used. Additionally the user will need to write an interrupt service routine to do at a minimum the following:

- Clear CSCTRL[TCF2] and CSCTRL[TCF1] flags.
- Calculate and write new values for both CMPLD1 and CMPLD2.

Timing

This figure contains the timing for using the compare preload feature. The compare preload cycle begins with a compare event on COMP2 causing CSCTRL[TCF2] to be asserted. COMP1 is loaded with the value in the CMPLD1 (c3) one IP bus clock later. In addition an interrupt is asserted by the timer and the interrupt service routine is executed during which both comparator load registers are updated with new values (c4 and c5). When CSCTRL[TCF1] is asserted, COMP2 is loaded with the value in CMPLD2 (c4). And on the subsequent CSCTRL[TCF2] event, COMP1 is loaded with the value in CMPLD1 (c5). The cycle starts over again as an interrupt is asserted and the interrupt service routine clears CSCTRL[TCF1] and CSCTRL[TCF2] and calculates new values for CMPLD1 and CMPLD2.



Step 1-- CNTR matches COMP2 value. CSCTRL[TCF2] is asserted and an interrupt request is generated.

Step 2-- One clock later, OFLAG toggles, CMPLD1 is copied to COMP1, LOAD is copied to CNTR, the counter starts counting.

Step 3-- The interrupt service routine clears CSCTRL[TCF1] and CSCTRL[TCF2] and the ISR loads CMPLD1 and CMPLD2 with the values for the next cycle. The counter continues counting until CNTR matches COMP1.

Step 4-- CSCTRL[TCF1] is asserted. One clock later, OFLAG toggles, CMPLD2 is copied to COMP2, LOAD is copied to CNTR and the counter starts counting.

Step 5--The counter continues counting until CNTR matches COMP2.

Figure 76-8. Compare Load Timing

Example: 76.4.5.14.1 Variable Frequency PWM Mode

```
// This example starts with an 11 msec with a 31 msec cycle.
// Assuming the chip is operating at 60 MHz, the timer use IP_bus_clk/32 as its
//   clock source.
//
// Initial pulse period: 60e6/32 clocks/sec * 31 ms = 58125 total clocks in period
// Initial pulse width: 60e6/32 clocks/sec * 11 ms = 20625 clocks in pulse
//
//
// Once the initial values of COMP1/CMPLD1 and COMP2/CMPLD2 are set the pulse width
//   can be varied by load new values of CMPLD1 and CMPLD2 on each compare interrupt.
//    (See Usage of Compare Load Registers.)
//
void PPG1_Init(void)
{
  setReg(TMR0_LOAD,0);                /* Clear load register */
  setReg(TMR0_CNTR,0);                /* Clear counter */
```

```
  /* TMR0_SCTRL: TCF=0,TCFIE=0,TOF=0,TOFIE=0,IEF=0,IEFIE=0,IPS=0,INPUT=0,
                 Capture_Mode=0,MSTR=0,EEOF=0,VAL=0,FORCE=1,OPS=0,OEN=1 */
  setReg(TMR0_SCTRL,5);                /* Set Status and Control Register */
  // Set compare preload operation and enable an interrupt on compare2 events.
  /* TMR0_CSCTRL: TCF2EN=1,TCF1EN=0,TCF2=0,TCF1=0,CL21=0,CL20=1,CL11=1,CL10=0 */
  setReg(TMR0_CSCTRL,0x86);            /* Set Comparator Status and Control Register */

  setReg(TMR0_COMP1, 20625-1);         /* Set the pulse width of the off time */
  setReg(TMR0_CMPLD1,20625-1);         /* Set the pulse width of the off time */
  setReg(TMR0_COMP2, 58125-20625-1);   /* Set the pulse width of the on time */
  setReg(TMR0_CMPLD2,58125-20625-1);   /* Set the pulse width of the on time */
  /* TMR0_CTRL: CM=1,PCS=0xD,SCS=0,ONCE=0,LENGTH=1,DIR=0,COINIT=0,OUTMODE=4 */
  setRegBits(TMR0_CTRL,0x3A24);        /* Set variable PWM mode and run counter */
}
```

76.4.6 Low Speed Operation

Some applications require that the TMR operate at a very low speed using a clock that is of lower frequency that the system clock frequency. This difference in clock speeds means that the CPU can't access the TMR registers. To accommodate this application the TMR has an input signal called tmr_cnts_freeze that can be used to reload the counter with the contents of the LOAD register and to reset the OFLAG. The software would first set the TMR control registers while the TMR is operating at the system clock speed and then would assert this signal prior to changing the clocks to the TMR. Once the TMR is operating off of a slower clock the tmr_cnts_freeze signal would be de-asserted and the TMR would start counting from a known state using a low speed clock.

76.5 Resets

76.5.1 General

The TMR module can be reset only by the RST_B signal. This forces all registers to their reset state and clears the OFLAG signal if it is asserted. The counter will be turned off until the settings in the control register are changed.

Table 76-2. Reset Summary

Reset	Priority	Source	Characteristics
RST_B	n/a	Hardware Reset	Full System Reset

76.6 Clocks

76.6.1 General

The timer only receives BUS_CLK_ROOT .

76.7 Interrupts

76.7.1 General

The TMR module can generate 20 interrupts, Five for each of the four counters/channels.

Table 76-3. Interrupt Summary

Core Interrupt	Interrupt	Description
TMR Channel 0	TMR0_COMP_IRQ_B	Compare Interrupt Request for Timer Channel 0
	TMR0_COMP1_IRQ_B	Compare 1 Interrupt Request for Timer Channel 0
	TMR0_COMP2_IRQ_B	Compare 2 Interrupt Request for Timer Channel 0
	TMR0_OVF_IRQ_B	Overflow Interrupt Request for Timer Channel 0
	TMR0_EDGE_IRQ_B	Input Edge Interrupt Request for Timer Channel 0
TMR Channel 1	TMR1_COMP_IRQ_B	Compare Interrupt Request for Timer Channel 1
	TMR1_COMP1_IRQ_B	Compare 1 Interrupt Request for Timer Channel 1
	TMR1_COMP2_IRQ_B	Compare 2 Interrupt Request for Timer Channel 1
	TMR1_OVF_IRQ_B	Overflow Interrupt Request for Timer Channel 1
	TMR1_EDGE_IRQ_B	Input Edge Interrupt Request for Timer Channel 1
TMR Channel 2	TMR2_COMP_IRQ_B	Compare Interrupt Request for Timer Channel 2
	TMR2_COMP1_IRQ_B	Compare 1 Interrupt Request for Timer Channel 2
	TMR2_COMP2_IRQ_B	Compare 2 Interrupt Request for Timer Channel 2
	TMR2_OVF_IRQ_B	Overflow Interrupt Request for Timer Channel 2
	TMR2_EDGE_IRQ_B	Input Edge Interrupt Request for Timer Channel 2
TMR Channel 3	TMR3_COMP_IRQ_B	Compare Interrupt Request for Timer Channel 3
	TMR3_COMP1_IRQ_B	Compare 1 Interrupt Request for Timer Channel 3
	TMR3_COMP2_IRQ_B	Compare 2 Interrupt Request for Timer Channel 3
	TMR3_OVF_IRQ_B	Overflow Interrupt Request for Timer Channel 3
	TMR3_EDGE_IRQ_B	Input Edge Interrupt Request for Timer Channel 3

76.7.2 Description of Interrupt Operation

76.7.2.1 Timer Compare Interrupts

These interrupts are generated when a successful compare occurs between a counter and it's compare registers while SCTRL[TCFIE] is set. These interrupts are cleared by writing a zero to the appropriate SCTRL[TCF].

When a timer compare interrupt is set in TMR_SCTRL and the Compare Load registers are available, one of the following two interrupts will also be asserted.

76.7.2.1.1 Timer Compare 1 Interrupts (Available with Compare Load Feature)

These interrupts are generated when a successful compare occurs between a counter and its COMP1 register while CSCTRL[TCF1EN] is set. These interrupts are cleared by writing a zero to the appropriate CSCTRL[TCF1].

76.7.2.1.2 Timer Compare 2 Interrupts (Available with Compare Load Feature)

These interrupts are generated when a successful compare occurs between a counter and its COMP2 register while CSCTRL[TCF2EN] is set. These interrupts are cleared by writing a zero to the appropriate CSCTRL[TCF2].

76.7.2.2 Timer Overflow Interrupts

These interrupts are generated when a counter rolls over its maximum value while SCTRL[TOFIE] is set. These interrupts are cleared by writing zero to the appropriate SCTRL[TOF].

76.7.2.3 Timer Input Edge Interrupts

These interrupts are generated by a transition of the input signal (either positive or negative depending on IPS setting) while SCTRL[IEFIE] is set. These interrupts are cleared by writing a zero to the appropriate SCTRL[IEF].

76.8 DMA

The TMR module can generate twelve DMA requests: three for each of the four counters/channels. Refer to the following table.

Table 76-4. DMA Summary

DMA Request	DMA Enable	Description
Channels 0-3	DMA[IEFDE]	CAPT contains a value
	DMA[CMPLD1DE]	CMPLD1 needs an update
	DMA[CMPLD2DE]	CMPLD2 needs an update

76.9 Memory map/register definition

76.9.1 TMR register descriptions

The address of a register is the sum of a base address and an address offset. The base address is defined at the chip level and the address offset is defined at the module level. Make certain to check which quad timer is available on the chip being used, and which timer channels have external I/O.

76.9.1.1 TMR memory map

TMR1 base address: 4015_C000h

TMR2 base address: 4016_0000h

TMR3 base address: 4016_4000h

TMR4 base address: 4016_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Timer Channel Compare Register 1 (COMP10)	16	RW	0000
1	Timer Channel Compare Register 2 (COMP20)	16	RW	0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
2	Timer Channel Capture Register (CAPT0)	16	RW	0000
3	Timer Channel Load Register (LOAD0)	16	RW	0000
4	Timer Channel Hold Register (HOLD0)	16	RW	0000
5	Timer Channel Counter Register (CNTR0)	16	RW	0000
6	Timer Channel Control Register (CTRL0)	16	RW	0000
7	Timer Channel Status and Control Register (SCTRL0)	16	RW	0000
8	Timer Channel Comparator Load Register 1 (CMPLD10)	16	RW	0000
9	Timer Channel Comparator Load Register 2 (CMPLD20)	16	RW	0000
A	Timer Channel Comparator Status and Control Register (CSCTRL0)	16	RW	0000
B	Timer Channel Input Filter Register (FILT0)	16	RW	0000
C	Timer Channel DMA Enable Register (DMA0)	16	RW	0000
F	Timer Channel Enable Register (ENBL)	16	RW	000F
10	Timer Channel Compare Register 1 (COMP11)	16	RW	0000
11	Timer Channel Compare Register 2 (COMP21)	16	RW	0000
12	Timer Channel Capture Register (CAPT1)	16	RW	0000
13	Timer Channel Load Register (LOAD1)	16	RW	0000
14	Timer Channel Hold Register (HOLD1)	16	RW	0000
15	Timer Channel Counter Register (CNTR1)	16	RW	0000
16	Timer Channel Control Register (CTRL1)	16	RW	0000
17	Timer Channel Status and Control Register (SCTRL1)	16	RW	0000
18	Timer Channel Comparator Load Register 1 (CMPLD11)	16	RW	0000
19	Timer Channel Comparator Load Register 2 (CMPLD21)	16	RW	0000
1A	Timer Channel Comparator Status and Control Register (CSCTRL1)	16	RW	0000
1B	Timer Channel Input Filter Register (FILT1)	16	RW	0000
1C	Timer Channel DMA Enable Register (DMA1)	16	RW	0000
20	Timer Channel Compare Register 1 (COMP12)	16	RW	0000
21	Timer Channel Compare Register 2 (COMP22)	16	RW	0000
22	Timer Channel Capture Register (CAPT2)	16	RW	0000
23	Timer Channel Load Register (LOAD2)	16	RW	0000
24	Timer Channel Hold Register (HOLD2)	16	RW	0000
25	Timer Channel Counter Register (CNTR2)	16	RW	0000
26	Timer Channel Control Register (CTRL2)	16	RW	0000
27	Timer Channel Status and Control Register (SCTRL2)	16	RW	0000
28	Timer Channel Comparator Load Register 1 (CMPLD12)	16	RW	0000
29	Timer Channel Comparator Load Register 2 (CMPLD22)	16	RW	0000
2A	Timer Channel Comparator Status and Control Register (CSCTRL2)	16	RW	0000
2B	Timer Channel Input Filter Register (FILT2)	16	RW	0000
2C	Timer Channel DMA Enable Register (DMA2)	16	RW	0000
30	Timer Channel Compare Register 1 (COMP13)	16	RW	0000
31	Timer Channel Compare Register 2 (COMP23)	16	RW	0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
32	Timer Channel Capture Register (CAPT3)	16	RW	0000
33	Timer Channel Load Register (LOAD3)	16	RW	0000
34	Timer Channel Hold Register (HOLD3)	16	RW	0000
35	Timer Channel Counter Register (CNTR3)	16	RW	0000
36	Timer Channel Control Register (CTRL3)	16	RW	0000
37	Timer Channel Status and Control Register (SCTRL3)	16	RW	0000
38	Timer Channel Comparator Load Register 1 (CMPLD13)	16	RW	0000
39	Timer Channel Comparator Load Register 2 (CMPLD23)	16	RW	0000
3A	Timer Channel Comparator Status and Control Register (CSCTRL3)	16	RW	0000
3B	Timer Channel Input Filter Register (FILT3)	16	RW	0000
3C	Timer Channel DMA Enable Register (DMA3)	16	RW	0000

76.9.1.2 Timer Channel Compare Register 1 (COMP10 - COMP13)

76.9.1.2.1 Offset

Register	Offset
COMP10	0h
COMP11	10h
COMP12	20h
COMP13	30h

76.9.1.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARISON_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.2.3 Fields

Field	Description
15-0	Comparison Value 1

Field	Description
COMPARISON_1	This read/write register stores the value used for comparison with the counter value in count up mode.

76.9.1.3 Timer Channel Compare Register 2 (COMP20 - COMP23)

76.9.1.3.1 Offset

Register	Offset
COMP20	1h
COMP21	11h
COMP22	21h
COMP23	31h

76.9.1.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARISON_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.3.3 Fields

Field	Description
15-0 COMPARISON_2	Comparison Value 2 This read/write register stores the value used for comparison with the counter value in count down mode or alternating compare mode.

76.9.1.4 Timer Channel Capture Register (CAPT0 - CAPT3)

76.9.1.4.1 Offset

Register	Offset
CAPT0	2h
CAPT1	12h
CAPT2	22h
CAPT3	32h

76.9.1.4.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAPTURE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.4.3 Fields

Field	Description
15-0 CAPTURE	Capture Value This read/write register stores the value captured from the counter.

76.9.1.5 Timer Channel Load Register (LOAD0 - LOAD3)

76.9.1.5.1 Offset

Register	Offset
LOAD0	3h
LOAD1	13h
LOAD2	23h
LOAD3	33h

76.9.1.5.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOAD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.5.3 Fields

Field	Description
15-0 LOAD	Timer Load Register This read/write register stores the value used to initialize the counter after counter compare or re-initialization.

76.9.1.6 Timer Channel Hold Register (HOLD0 - HOLD3)

76.9.1.6.1 Offset

Register	Offset
HOLD0	4h
HOLD1	14h
HOLD2	24h
HOLD3	34h

76.9.1.6.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HOLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.6.3 Fields

Field	Description
15-0 HOLD	HOLD This read/write register stores the counter's values of specific channels whenever any of the four counters within a module is read.

76.9.1.7 Timer Channel Counter Register (CNTR0 - CNTR3)

76.9.1.7.1 Offset

Register	Offset
CNTR0	5h
CNTR1	15h
CNTR2	25h
CNTR3	35h

76.9.1.7.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.7.3 Fields

Field	Description
15-0 COUNTER	COUNTER This read/write register is the counter for the corresponding channel in a timer module.

76.9.1.8 Timer Channel Control Register (CTRL0 - CTRL3)

76.9.1.8.1 Offset

Register	Offset
CTRL0	6h
CTRL1	16h
CTRL2	26h
CTRL3	36h

76.9.1.8.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CM			PCS				SCS		ONCE	LENGTH	DIR	COINIT	OUTMODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.8.3 Fields

Field	Description
15-13 CM	Count Mode These bits control the basic counting and behavior of the counter. 000 - No operation 001 - Count rising edges of primary source[1] 010 - Count rising and falling edges of primary source[2] 011 - Count rising edges of primary source while secondary input high active 100 - Quadrature count mode, uses primary and secondary sources 101 - Count rising edges of primary source; secondary source specifies direction[3] 110 - Edge of secondary source triggers primary count until compare 111 - Cascaded counter mode (up/down)[4]
12-9 PCS	Primary Count Source These bits select the primary count source. **NOTE:** A timer selecting its own output for input is not a legal choice. The result is no counting. 0000 - Counter 0 input pin 0001 - Counter 1 input pin 0010 - Counter 2 input pin 0011 - Counter 3 input pin

Table continues on the next page...

Field	Description
	0100 - Counter 0 output 0101 - Counter 1 output 0110 - Counter 2 output 0111 - Counter 3 output 1000 - IP bus clock divide by 1 prescaler 1001 - IP bus clock divide by 2 prescaler 1010 - IP bus clock divide by 4 prescaler 1011 - IP bus clock divide by 8 prescaler 1100 - IP bus clock divide by 16 prescaler 1101 - IP bus clock divide by 32 prescaler 1110 - IP bus clock divide by 64 prescaler 1111 - IP bus clock divide by 128 prescaler
8-7 SCS	Secondary Count Source These bits identify the external input pin to be used as a count command or timer command. The selected input can trigger the timer to capture the current value of CNTR . The selected input can also be used to specify the count direction. The selected signal can also be used as a fault input when CSCTRL[FAULT] is set. The polarity of the signal can be inverted by SCTRL[IPS]. 00 - Counter 0 input pin 01 - Counter 1 input pin 10 - Counter 2 input pin 11 - Counter 3 input pin
6 ONCE	Count Once This bit selects continuous or one shot counting mode. 0 - Count repeatedly. 1 - Count until compare and then stop. If counting up, a successful compare occurs when the counter reaches a COMP1 value. If counting down, a successful compare occurs when the counter reaches a COMP2 value. When output mode $4 is used, the counter re-initializes after reaching the COMP1 value, continues to count to the COMP2 value, and then stops.
5 LENGTH	Count Length This bit determines whether the counter: • counts to the compare value and then re-initializes itself to the value specified in the LOAD (or CMPLD2) register, or • continues counting past the compare value to the binary roll over. 0 - Count until roll over at $FFFF and then continue by re-initializing the counter from the LOAD register. 1 - Count until compare, then re-initialize using the LOAD register. If counting up, a successful compare occurs when the counter reaches a COMP1 value. If counting down, a successful compare occurs when the counter reaches a COMP2 value. When output mode $4 is used, alternating values of COMP1 and COMP2 are used to generate successful comparisons. For example, the counter counts until a COMP1 value is reached, re-initializes, counts until COMP2 value is reached, re-initializes, counts until COMP1 value is reached, and so on.
4 DIR	Count Direction This bit selects either the normal count direction up, or the reverse direction, down. 0 - Count up.

Table continues on the next page...

Field	Description
	1 - Count down.
3 COINIT	Co-Channel Initialization This bit enables another counter/timer within the module to force the re-initialization of this counter/timer when it has an active compare event. 0 - Co-channel counter/timers cannot force a re-initialization of this counter/timer 1 - Co-channel counter/timers may force a re-initialization of this counter/timer
2-0 OUTMODE	Output Mode These bits determine the mode of operation for the OFLAG output signal. 000 - Asserted while counter is active 001 - Clear OFLAG output on successful compare 010 - Set OFLAG output on successful compare 011 - Toggle OFLAG output on successful compare 100 - Toggle OFLAG output using alternating compare registers 101 - Set on compare, cleared on secondary source input edge 110 - Set on compare, cleared on counter rollover 111 - Enable gated clock output while counter is active

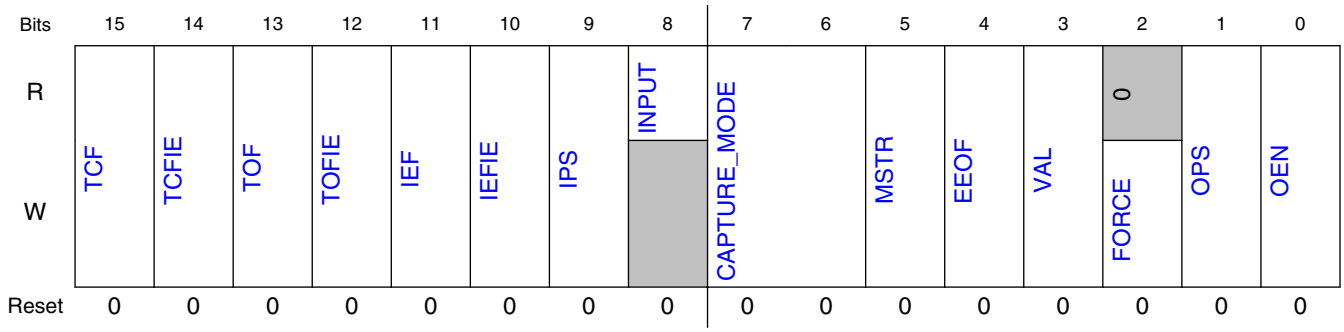
1. Rising edges are counted only when SCTRL[IPS] = 0. Falling edges are counted when SCTRL[IPS] = 1. If the primary count source is IP bus clock divide by 1, only rising edges are counted regardless of the value of SCTRL[IPS].
2. IP bus clock divide by 1 cannot be used as a primary count source in edge count mode.
3. Rising edges are counted only when SCTRL[IPS] = 0. Falling edges are counted when SCTRL[IPS] = 1.
4. The primary count source must be set to one of the counter outputs.

76.9.1.9 Timer Channel Status and Control Register (SCTRL0 - SCTRL3)

76.9.1.9.1 Offset

Register	Offset
SCTRL0	7h
SCTRL1	17h
SCTRL2	27h
SCTRL3	37h

76.9.1.9.2 Diagram



76.9.1.9.3 Fields

Field	Description
15 TCF	Timer Compare Flag This bit is set when a successful compare occurs. This bit is cleared by writing a zero to this bit location.
14 TCFIE	Timer Compare Flag Interrupt Enable This bit (when set) enables interrupts when TCF is set.
13 TOF	Timer Overflow Flag This bit is set when the counter rolls over its maximum value $FFFF or $0000 (depending on count direction). This bit is cleared by writing a zero to this bit location.
12 TOFIE	Timer Overflow Flag Interrupt Enable This bit (when set) enables interrupts when TOF is set.
11 IEF	Input Edge Flag This bit is set when CAPTMODE is enabled and a proper input transition occurs (on an input selected as a secondary count source) while the count mode does not equal 000. This bit is cleared by writing a zero to this bit position. This bit can also be cleared automatically by a read of CAPT when DMA[IEFDE] is set. **NOTE:** Setting the input polarity select bit (IPS) changes the edge to be detected. Also, the control register's secondary count source determines which external input pin is monitored by the detection circuitry.
10 IEFIE	Input Edge Flag Interrupt Enable This bit (when set) enables interrupts when IEF is set. **Restriction:** Do not set both this bit and DMA[IEFDE].
9 IPS	Input Polarity Select This bit (when set) inverts the input signal polarity.
8 INPUT	External Input Signal This read-only bit reflects the current state of the external input pin selected via the secondary count source after application of IPS and filtering.
7-6 CAPTURE_MODE	Input Capture Mode These bits specify the operation of the capture register as well as the operation of the input edge flag. The input source is the secondary count source. 00 - Capture function is disabled

Table continues on the next page...

Field	Description
	01 - Load capture register on rising edge (when IPS=0) or falling edge (when IPS=1) of input 10 - Load capture register on falling edge (when IPS=0) or rising edge (when IPS=1) of input 11 - Load capture register on both edges of input
5 MSTR	Master Mode This bit (when set) enables the compare function's output to be broadcasted to the other counters/timers in the module. This signal then can be used to re-initialize the other counters and/or force their OFLAG signal outputs.
4 EEOF	Enable External OFLAG Force This bit (when set) enables the compare from another counter/timer within the same module to force the state of this counter's OFLAG output signal.
3 VAL	Forced OFLAG Value This bit determines the value of the OFLAG output signal when software triggers a FORCE command.
2 FORCE	Force OFLAG Output This write only bit forces the current value of VAL to be written to the OFLAG output. This bit always reads as a zero. VAL and FORCE can be written simultaneously in a single write operation. Write to FORCE only if the counter is disabled. Setting this bit while the counter is enabled may yield unpredictable results.
1 OPS	Output Polarity Select This bit determines the polarity of the OFLAG output signal. 0 - True polarity. 1 - Inverted polarity.
0 OEN	Output Enable This bit determines the direction of the external pin. 0 - The external pin is configured as an input. 1 - The OFLAG output signal is driven on the external pin. Other timer groups using this external pin as their input see the driven value. The polarity of the signal is determined by OPS.

76.9.1.10 Timer Channel Comparator Load Register 1 (CMPLD10 - CMPLD13)

76.9.1.10.1 Offset

Register	Offset
CMPLD10	8h
CMPLD11	18h
CMPLD12	28h
CMPLD13	38h

76.9.1.10.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARATOR_LOAD_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.10.3 Fields

Field	Description
15-0 COMPARATOR_LOAD_1	COMPARATOR_LOAD_1 This read/write register is the comparator 1 preload value for the COMP1 register for the corresponding channel in a timer module.

76.9.1.11 Timer Channel Comparator Load Register 2 (CMPLD20 - CMPLD23)

76.9.1.11.1 Offset

Register	Offset
CMPLD20	9h
CMPLD21	19h
CMPLD22	29h
CMPLD23	39h

76.9.1.11.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARATOR_LOAD_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.11.3 Fields

Field	Description
15-0 COMPARATOR_LOAD_2	COMPARATOR_LOAD_2 This read/write register is the comparator 2 preload value for the COMP2 register for the corresponding channel in a timer module.

76.9.1.12 Timer Channel Comparator Status and Control Register (CSCTRL0 - CSCTRL3)

76.9.1.12.1 Offset

Register	Offset
CSCTRL0	Ah
CSCTRL1	1Ah
CSCTRL2	2Ah
CSCTRL3	3Ah

76.9.1.12.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DBG_EN		FAULT	ALT_LOAD	ROC	TCI	UP	OFLAG	TCF2EN	TCF1EN	TCF2	TCF1	CL2		CL1	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

76.9.1.12.3 Fields

Field	Description
15-14 DBG_EN	Debug Actions Enable These bits allow the TMR module to perform certain actions in response to the chip entering debug mode. 00 - Continue with normal operation during debug mode. (default) 01 - Halt TMR counter during debug mode. 10 - Force TMR output to logic 0 (prior to consideration of SCTRL[OPS]).

Table continues on the next page...

Field	Description
	11 - Both halt counter and force output to 0 during debug mode.
13 FAULT	Fault Enable The selected secondary input acts as a fault signal so that the timer OFLAG is cleared when the secondary input is set. When the secondary input is used in this mode, there is no resynchronization of the input so that there is a combinational path to clear the OFLAG. Fault inputs less than two clock periods wide will not be latched. Latched faults will be cleared the next time that the counter logic sets the OFLAG. 0 - Fault function disabled. 1 - Fault function enabled.
12 ALT_LOAD	Alternative Load Enable This bit allows for an alternative method for loading the counter during modulo counting. Normally, the counter can be loaded only with the value from the LOAD register. When this bit is set, the counter is loaded from the LOAD register when counting up and a match with COMP1 occurs, or the counter is loaded from the CMPLD2 register when counting down and a match with COMP2 occurs. 0 - Counter can be re-initialized only with the LOAD register. 1 - Counter can be re-initialized with the LOAD or CMPLD2 registers depending on count direction.
11 ROC	Reload on Capture This bit enables the capture function to cause the counter to be reloaded from the LOAD register. 0 - Do not reload the counter on a capture event. 1 - Reload the counter on a capture event.
10 TCI	Triggered Count Initialization Control This bit is used during triggered count mode, CTRL[CM] = 110, to enable the counter to be re-initialized when a second trigger occurs while the counter is still counting. Normally, the second trigger causes the counting to stop/pause until a third trigger occurs. With this bit set, a second trigger event causes the counter to re-initialize and continue counting. 0 - Stop counter upon receiving a second trigger event while still counting from the first trigger event. 1 - Reload the counter upon receiving a second trigger event while still counting from the first trigger event.
9 UP	Counting Direction Indicator This read-only bit is used during quadrature count mode, CTRL[CM] = 100, to read the direction of the last count. CTRL[DIR] reverses the sense of this bit. 0 - The last count was in the DOWN direction. 1 - The last count was in the UP direction.
8 OFLAG	Output flag This read only bit shows the state of the Timer's internal OFLAG signal prior to consideration of polarity or debug mode.
7 TCF2EN	Timer Compare 2 Interrupt Enable An interrupt is issued when both this bit and TCF2 are set.
6 TCF1EN	Timer Compare 1 Interrupt Enable An interrupt is issued when both this bit and TCF1 are set.
5 TCF2	Timer Compare 2 Interrupt Flag When set, this bit indicates a successful comparison of the timer and the the COMP2 register has occurred. This bit is sticky, and will remain set until explicitly cleared by writing a zero to this bit location.
4	Timer Compare 1 Interrupt Flag

Table continues on the next page...

Field	Description
TCF1	When set, this bit indicates a successful comparison of the timer and the the COMP1 register has occurred. This bit is sticky, and will remain set until explicitly cleared by writing a zero to this bit location.
3-2 CL2	Compare Load Control 2 These bits control when COMP2 is preloaded with the value from CMPLD2. 00 - Never preload 01 - Load upon successful compare with the value in COMP1 10 - Load upon successful compare with the value in COMP2 11 - Reserved
1-0 CL1	Compare Load Control 1 These bits control when COMP1 is preloaded with the value from CMPLD1. 00 - Never preload 01 - Load upon successful compare with the value in COMP1 10 - Load upon successful compare with the value in COMP2 11 - Reserved

76.9.1.13 Timer Channel Input Filter Register (FILT0 - FILT3)

The FILT register programs the values for the filtering of the corresponding input without regard for the fact that any timer channel can use the input as a count source.

Input filter considerations:

- Set the FILT_PER value such that the sampling period is larger than the period of the expected noise. In this way, a noise spike will corrupt only one sample. Choose the FILT_CNT value to reduce the probability that noisy samples cause an incorrect transition to be recognized. The probability of an incorrect transistion is defined as the probability of an incorrect sample raised to the power of (FILT_CNT + 3).
- The values of FILT_PER and FILT_CNT must also be balanced against the desire for minimal latency in recognizing input transitions. Turning on the input filter (setting FILT_PER to a non-zero value) introduces a latency of (((FILT_CNT + 3) x FILT_PER) + 2) IP bus clock periods.

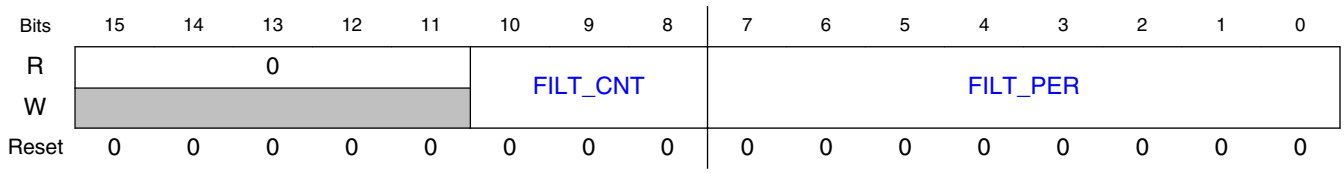
76.9.1.13.1 Offset

Register	Offset
FILT0	Bh
FILT1	1Bh

Table continues on the next page...

Register	Offset
FILT2	2Bh
FILT3	3Bh

76.9.1.13.2 Diagram



76.9.1.13.3 Fields

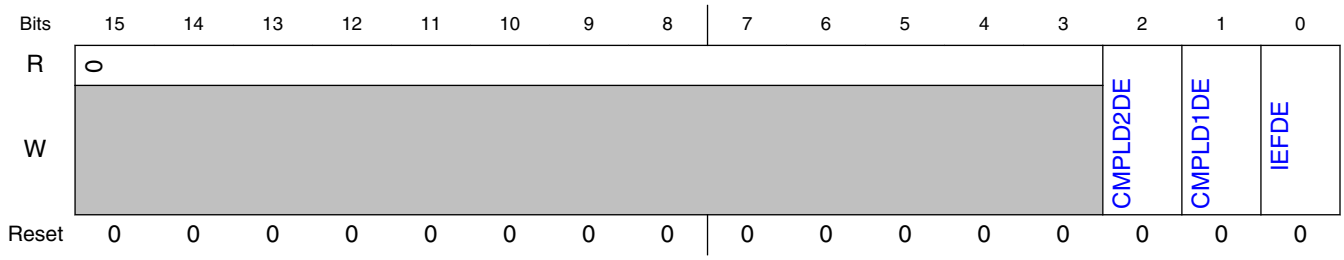
Field	Description
15-11 —	RESERVED
10-8 FILT_CNT	Input Filter Sample Count These bits represent the number of consecutive samples that must agree prior to the input filter accepting an input transition. A value of 0x0 represents 3 samples. A value of 0x7 represents 10 samples. The value of FILT_CNT affects the input latency.
7-0 FILT_PER	Input Filter Sample Period These bits represent the sampling period (in IP bus clock cycles) of the TMR input signals. Each input is sampled multiple times at the rate specified by this field. If FILT_PER is 0x00 (default), then the input filter is bypassed. The value of FILT_PER affects the input latency. When changing values for FILT_PER from one non-zero value to another non-zero value, write a value of zero first to clear the filter.

76.9.1.14 Timer Channel DMA Enable Register (DMA0 - DMA3)

76.9.1.14.1 Offset

Register	Offset
DMA0	Ch
DMA1	1Ch
DMA2	2Ch
DMA3	3Ch

76.9.1.14.2 Diagram



76.9.1.14.3 Fields

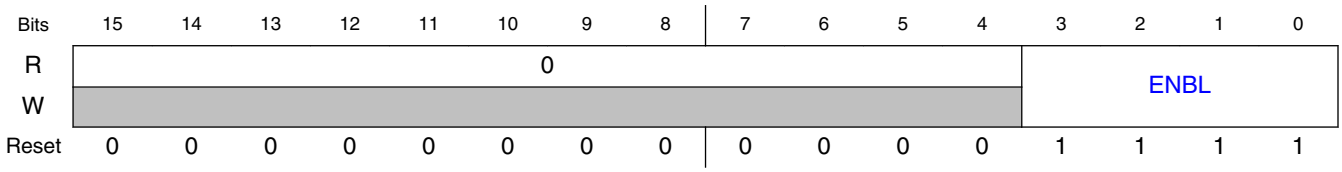
Field	Description
15-3 —	RESERVED
2 CMPLD2DE	Comparator Preload Register 2 DMA Enable Setting this bit enables DMA write requests for CMPLD2 whenever data is transferred out of the CMPLD2 register into the CNTR or COMP2 registers.
1 CMPLD1DE	Comparator Preload Register 1 DMA Enable Setting this bit enables DMA write requests for CMPLD1 whenever data is transferred out of the CMPLD1 register into the COMP1 register.
0 IEFDE	Input Edge Flag DMA Enable Setting this bit enables DMA read requests for CAPT when SCTRL[IEF] is set. **Restriction:** Do not set both this bit and SCTRL[IEFIE].

76.9.1.15 Timer Channel Enable Register (ENBL)

76.9.1.15.1 Offset

Register	Offset
ENBL	Fh

76.9.1.15.2 Diagram



76.9.1.15.3 Fields

Field	Description
15-4 —	RESERVED
3-0 ENBL	Timer Channel Enable These bits enable the prescaler (if it is being used) and counter in each channel. Multiple ENBL bits can be set at the same time to synchronize the start of separate counters. If an ENBL bit is set, then the corresponding channel starts its counter as soon as the CTRL[CM] field has a value other than 0. When an ENBL bit is clear, the corresponding counter maintains its current value. 0000 - Timer channel is disabled. 0001 - Timer channel is enabled. (default)

Chapter 77
Quadrature Decoder (QDC)

77.1 Chip-specific QDC information

Table 77-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

77.2 Overview

QDC interfaces to position and speed sensors used in industrial motor control applications. QDC uses the following five input signals from those position and speed sensors to decode shaft position, revolution count, and speed:

- PHASEA
- PHASEB
- INDEX
- TRIGGER
- HOME

77.2.1 Block diagram

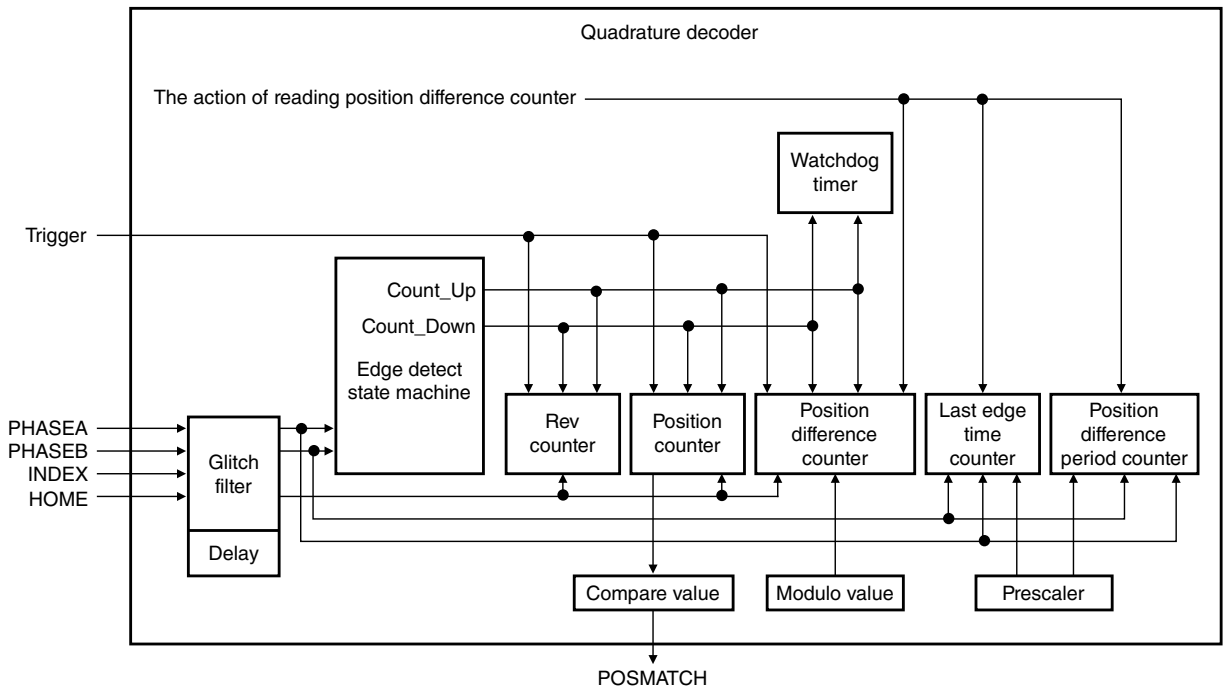


Figure 77-1. QDC block diagram

77.2.2 Features

- Logic to decode quadrature signals
- Inputs that you could connect to a general purpose timer to perform low-speed velocity measurements
- Digital filter for inputs that you could configure according to your requirements
- Quadrature decoder filter that can be bypassed
- 32-bit position counter capable of modulo counting
- Position counter that you, or external events, could initialize
- 16-bit position difference register
- Compare function to indicate when the shaft has reached a defined position
- Watchdog timer to detect a non-rotating shaft condition
- 16-bit revolution counter that could be preloaded
- Maximum count frequency that equals the peripheral clock rate
- Availability of an optional interrupt when both PHASEA and PHASEB inputs change in the same cycle

77.2.3 Decoder Block Diagram

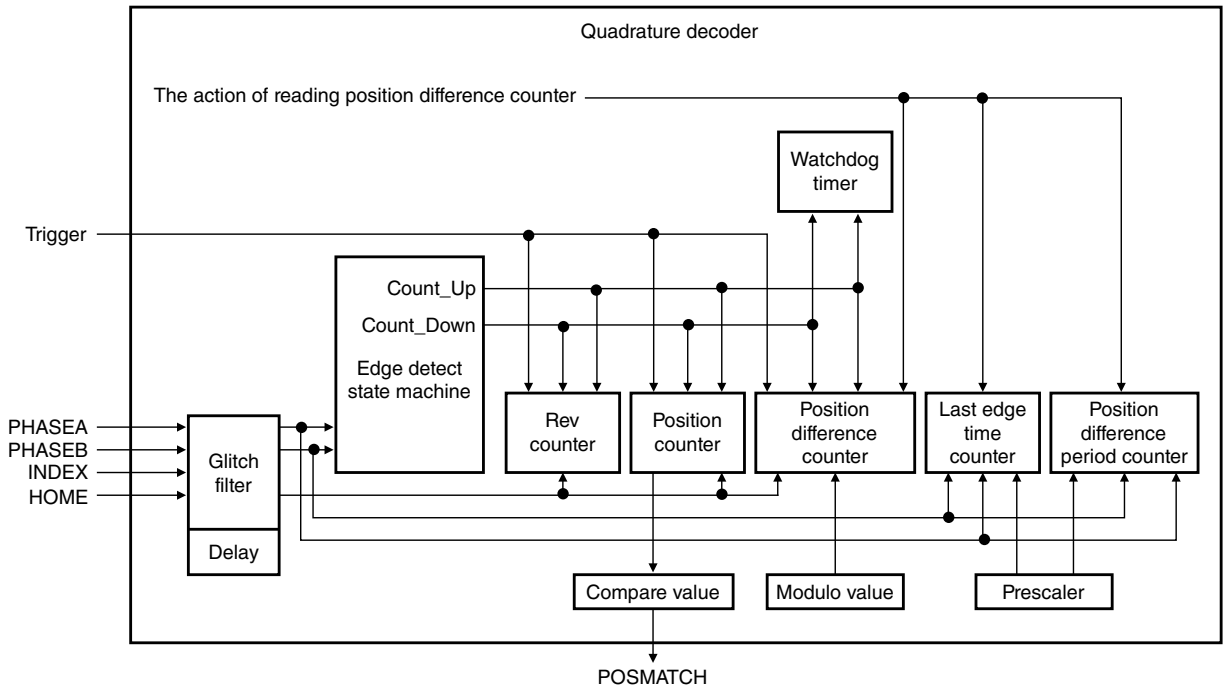


Figure 77-2. Quadrature Decoder Block Diagram

77.2.3.1 Glitch filter

The quadrature decoder logic must sense signal transitions, and the signal inputs first run through a glitch filter. This glitch filter has a digital delay line, which samples multiple time points on the signal and verifies a stable new signal state before outputting this new signal state to the internal quadrature decoder logic. To adapt to a variety of signal bandwidths, the sample rate of this delay line is programmable.

77.2.3.2 Edge detect state machine

The edge detect state machine looks for changes in the four possible states of the filtered PHASEA and PHASEB inputs, which calculates the direction of motion. Format this information as Count_Up and Count_Down signals. Each counter includes its hold register. These signals route into up to three up/down counters:

- Position counter
- Revolution counter
- Position difference counter

77.2.3.3 Position counter

The 32-bit position counter calculates up or down on every count pulse, which the difference between PHASEA and PHASEB generates. The position counter acts as integration information, whose count value is proportional to position. Count_Up and Count_Down signals determine the direction of the count. You may initialize the position counters to a predetermined value by one of 4 different methods:

- Software-triggered event
- INDEX signal transition
- HOME signal transition
- Rising edge of a trigger event

You can program the INDEX and HOME signals to interrupt the processor. Whenever you read the position counters (Upper Position Counter (UPOS) or Lower Position Counter (LPOS)), a snapshot of the following counters are placed into their respective hold registers:

- Position counter
- Position difference counter (only if CTRL3[PMEN] is 0)
- Revolution counter

77.2.3.4 Position difference counter

The 16-bit position difference counter contains the position difference value occurring between each read of the position register. The position register counts up or down on every count pulse. The position difference counter acts as a differentiator whose count value is proportional to the change in position since the last time the position counter was read.

When you read the position registers, the position difference counter, or the revolution counter, the contents of Position Difference Counter (POSD) are copied into Position Difference Hold (POSDH), and the position difference counter becomes 0.

If CTRL3[PMEN] is 1, a read of the position difference counter, and not a read of the revolution counter or position counters, updates the position difference counter and its hold register.

77.2.3.5 Position difference counter hold

Position Difference Hold (POSDH) stores a copy of the position difference counter when you read the position register. When you read the position register, the position difference counter, or the revolution counter, the position difference counter's contents are copied into Position Difference Hold (POSDH), and the position difference counter becomes 0.

If CTRL3[PMEN] is 1, a read of the position difference counter, and not a read of the revolution counter or position counters, updates Position Difference Hold (POSDH).

77.2.3.6 Revolution counter

The 16-bit up/down revolution counter counts or integrate revolutions by counting index pulses. Determine the direction of the count by the Count_Up and Count_Down signals, determined by the Phase A and B inputs. A different count direction on the rising and falling edges of the index pulse indicates that the quad encoder changed direction on the index pulse.

77.2.3.7 Watchdog timer

The watchdog timer ensures that the algorithm indicates motion in the shaft. Two successive counts indicate proper operation and reset the timer.

- The timeout value is programmable.
- When a timeout occurs, you can generate an interrupt to the processor.

77.2.3.8 Pulse accumulator functionality

You can program the logic to integrate only selected transitions of the PHASEA signal. This way, you can use the position counter as a pulse accumulator.

- The count direction is up.
- The INDEX input can optionally initialize the pulse accumulator.

77.2.3.9 Last edge time counter

The 16-bit last edge time counter contains the time since the last edge on PHASEA or PHASEB. The last edge time counter counts up on every prescaled clock pulse.

When you read the position difference counter, copy the content of LASTEDGE[LASTEDGE] into Last Edge Time Hold (LASTEDGEH).

77.2.3.10 Position difference period counter

The position difference period counter:

- Contains the accumulated time from the last time you read the position difference counter.

- Counts up on every prescaled clock pulse.
- Loads from the last edge time counter whenever you read the position difference register.

77.3 Functional description

The following timing diagram shows the basic operation of an incremental position QDC.

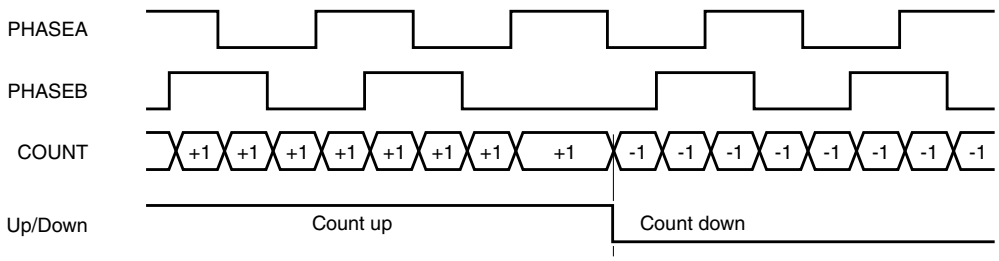


Figure 77-3. QDC signals

77.3.1 Positive versus negative direction

A typical quad encoder has three outputs: PHASEA signal, PHASEB signal, and INDEX pulse (not shown).

- If PHASEA leads PHASEB, then motion is in the positive direction.
- If PHASEA trails PHASEB, then motion is in the negative direction.

You can integrate the transitions on these phases to yield position or differentiate to yield velocity. Design the QDC to perform these functions in hardware.

77.3.2 Holding Registers and Initializing Registers

Hold registers are associated with 3 types of counters:

- Position
- Position difference
- Revolution

When any of the counter registers are read, the contents of each counter register is written to the corresponding hold register. Taking a snapshot of the counters' values allows a consistent view of a system's position and the velocity to be attained. If CTRL3[PMEN] is set, then the POSDH register will only be updated when the POSD counter is read and not when other counters are read. To capture a time stamp of when these registers are read, use the POSMATCH output with a timer channel.

The position counter is 32 bits wide. To ensure that the position counter can be reliably initialized with two 16-bit accesses, two registers (an upper and a lower initialization register) are provided. The upper initialization register (UINIT) and lower initialization register (LINIT) should be loaded with the desired value. Next, the position counter can be loaded by writing 1 to the Software-Triggered Initialization of Position Counters UPOS and LPOS bit (CTRL[SWIP]). Alternatively, either the INDEX-Triggered Initialization of Position Counters UPOS and LPOS bit (CTRL[XIP]) or the Enable HOME to Initialize Position Counters UPOS and LPOS bit (CTRL[HIP]) can enable the position counter to be initialized in response to a HOME or INDEX signal transition.

77.3.3 Speed measurement method

This section explains the speed measurement method. It includes a speed measurement algorithm.

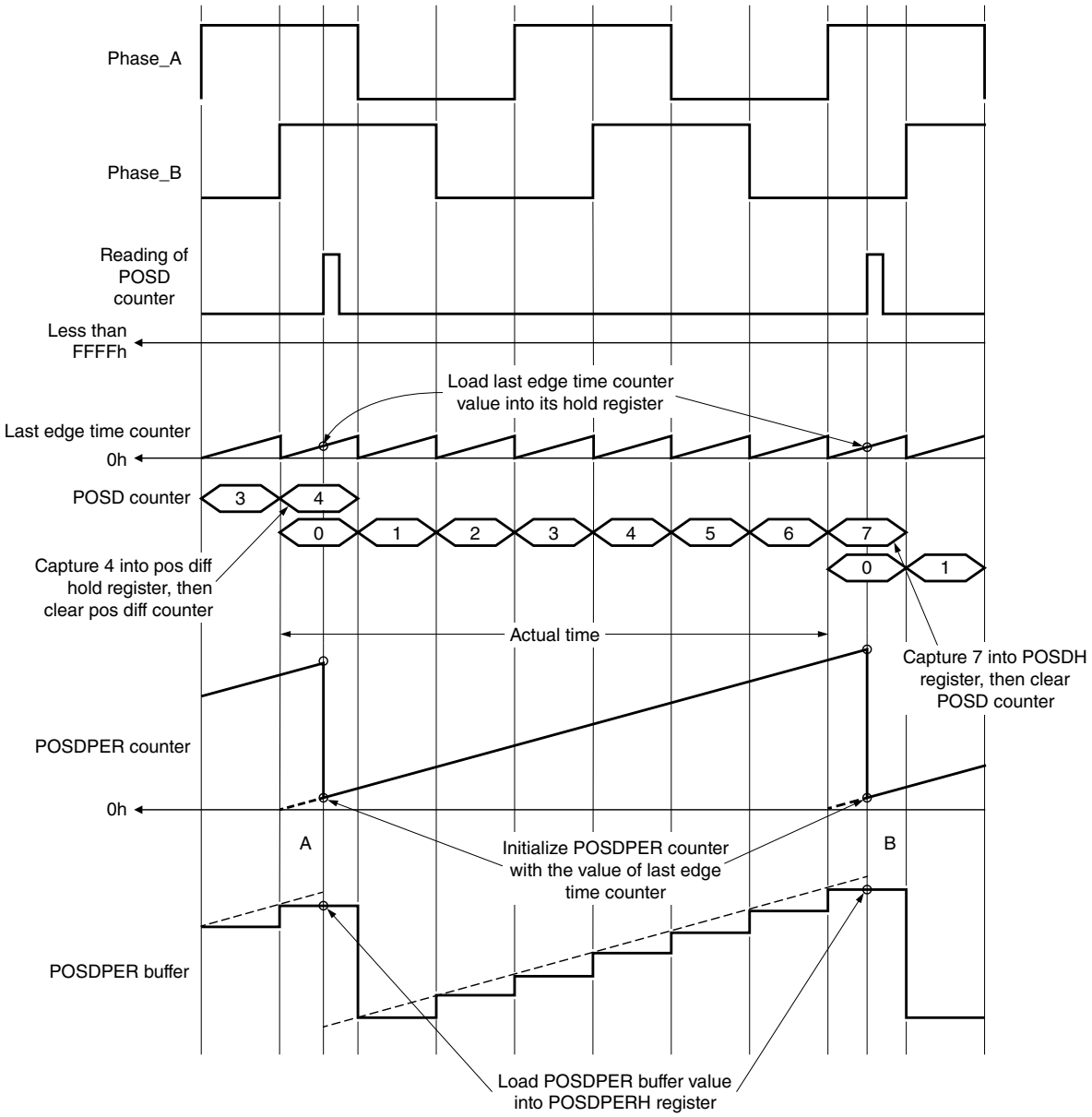


Figure 77-4. Counters behavior and updating mechanism

Reading of Position Difference Counter (POSD) occurs at time point A and time point B. “Actual time” represents the time duration between the first phase_a/b edges right before time points A and B. At time point B, after reading Position Difference Counter (POSD), Position Difference Period Hold (POSDPERH) contains the time length of “Actual time” and Position Difference Hold (POSDH) includes the value of phase_a/b pulses within that “Actual time”. You can calculate the speed based on Position Difference Period Hold (POSDPERH) and Position Difference Hold (POSDH). The following figure shows a visualized explanation of speed measurement with this method:

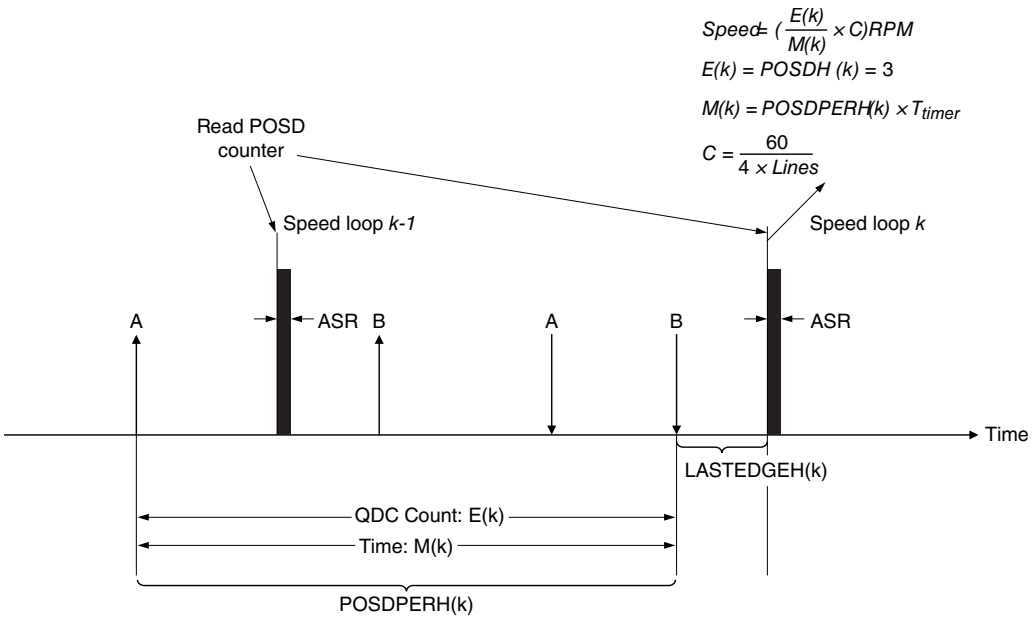
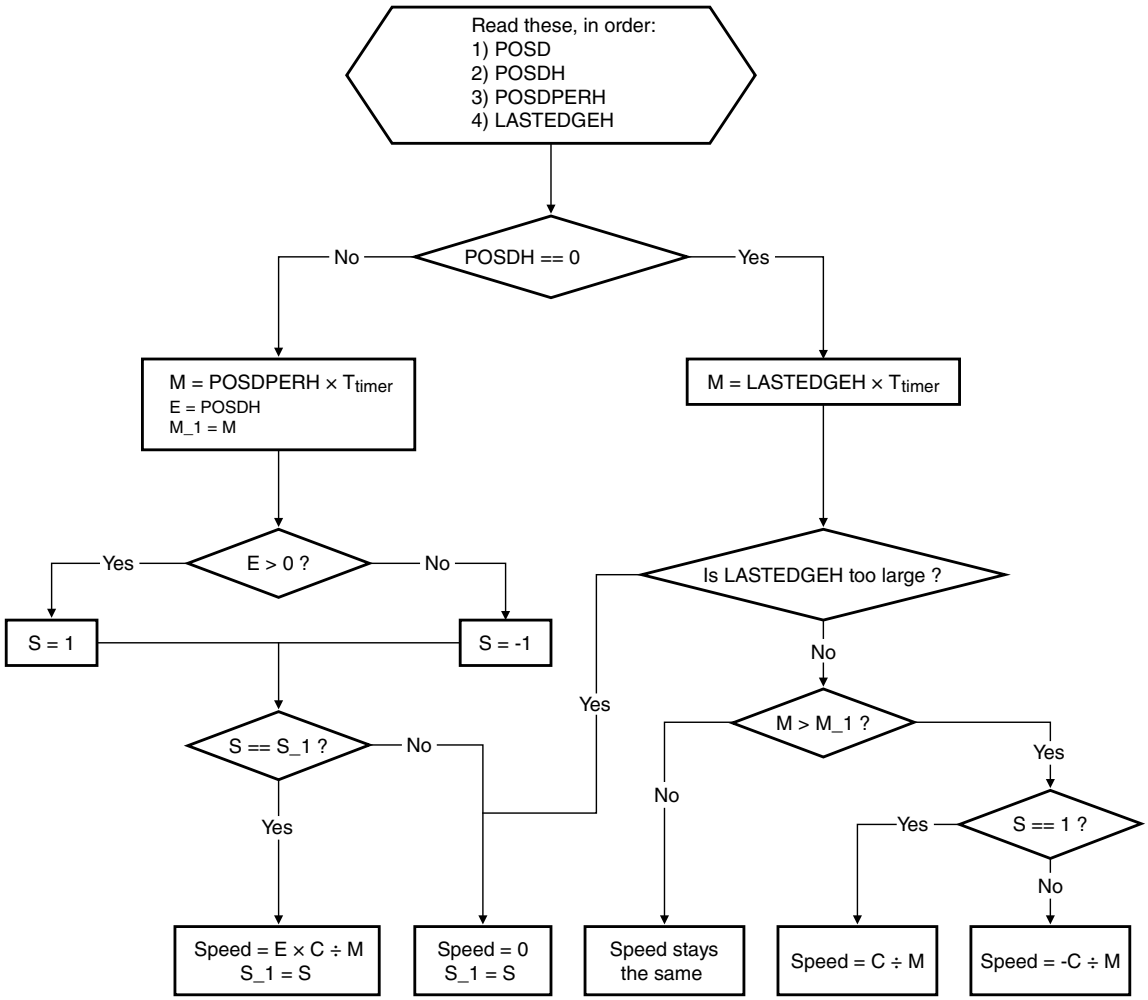


Figure 77-5. High speed measurement with method

77.3.3.1 Speed calculation algorithm

However, to cover all the cases in real applications, a simple speed measurement algorithm is necessary. The following figure shows the speed measurement algorithm.



S: Indicates whether E = 0 in this POSD reading operation. If E = 0, then S = -1. If E > 0, then S = 1.
S_1: Indicates whether E = 0 in the previous POSD reading operation. If E=0, then S = -1. If E > 0, then S=1.
M: Indicates the M value in this POSD reading operation.
M_1: Indicates the M value in the previous POSD reading operation.

Figure 77-6. Flowchart of speed calculation with method

77.3.3.2 Speed calculation of motor's starting from standstill (Chip is out of reset)

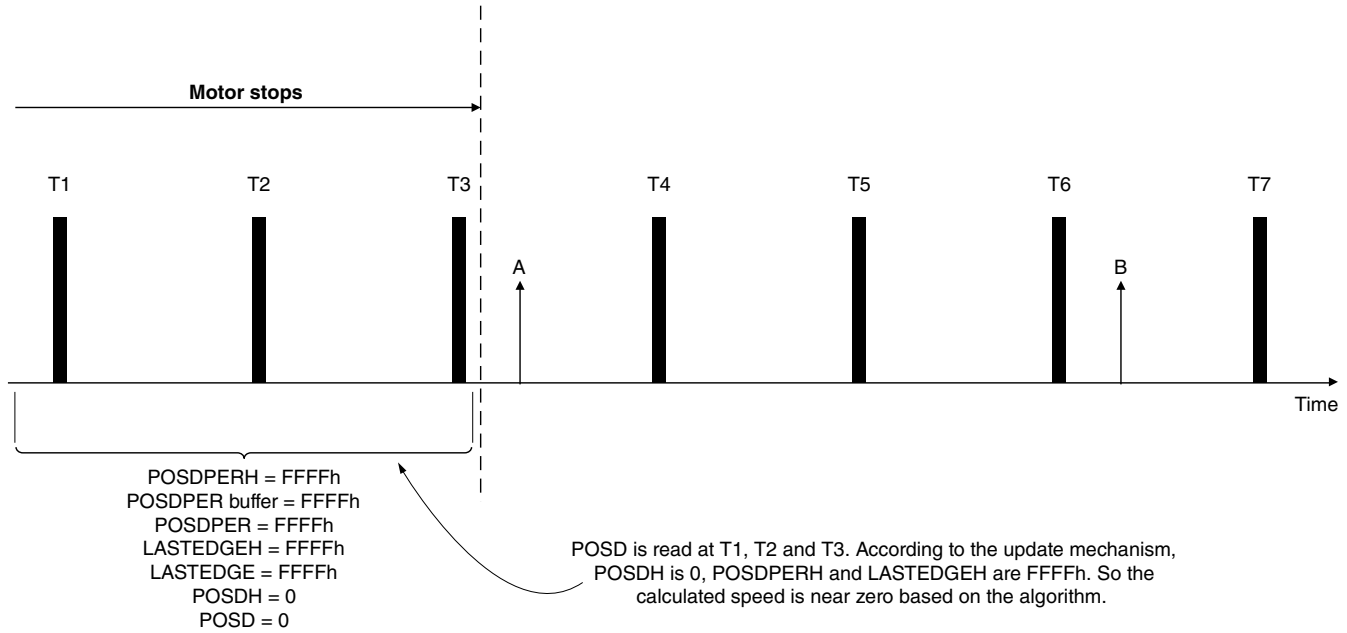


Figure 77-7. Speed measurement at time point T1~T3

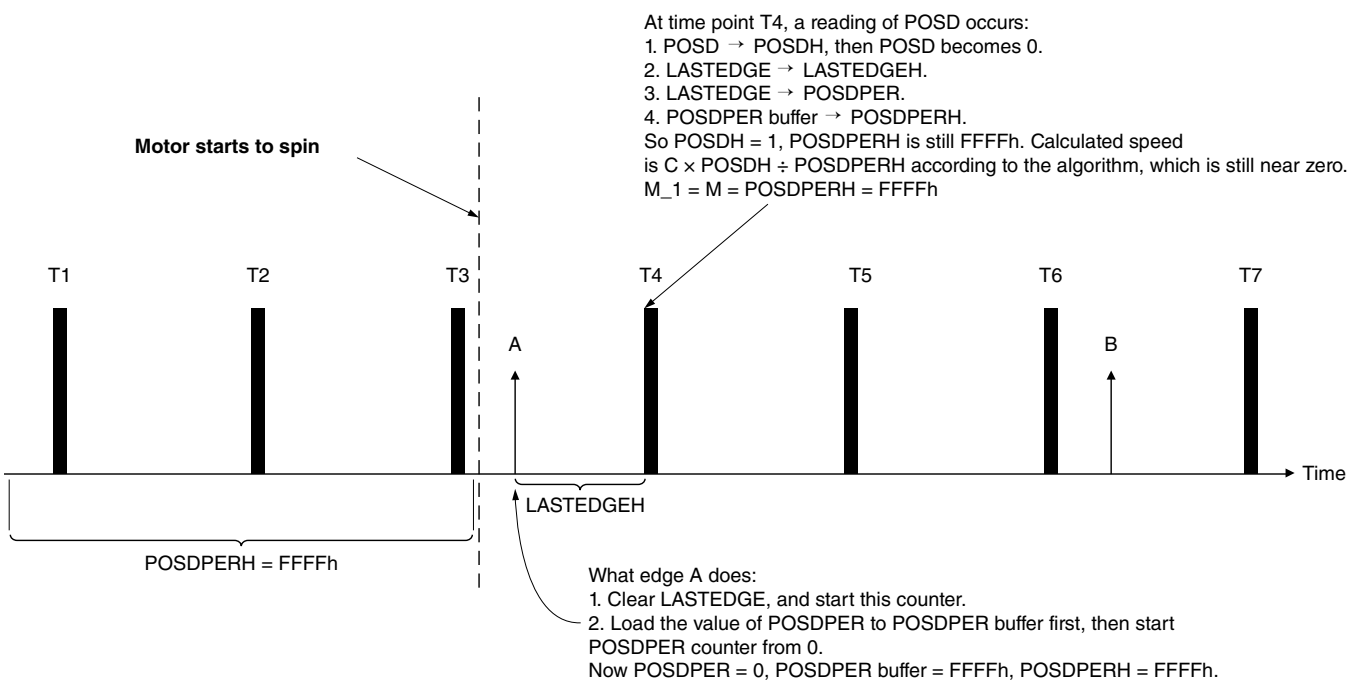


Figure 77-8. Speed measurement at time point T4

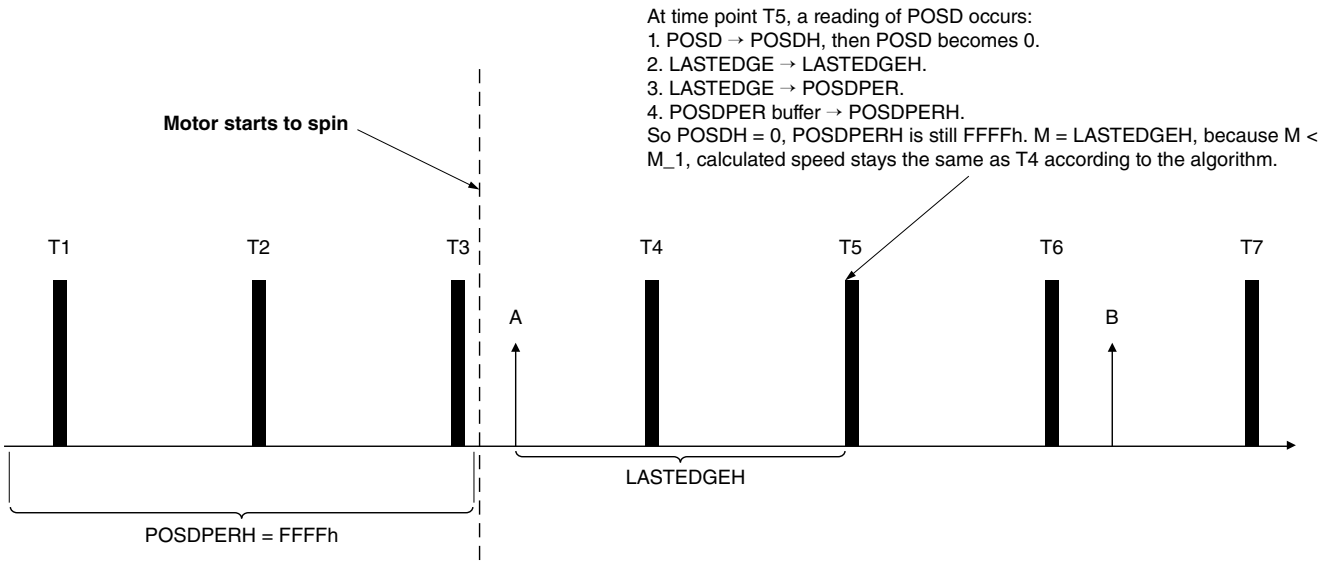


Figure 77-9. Speed measurement at time point T5

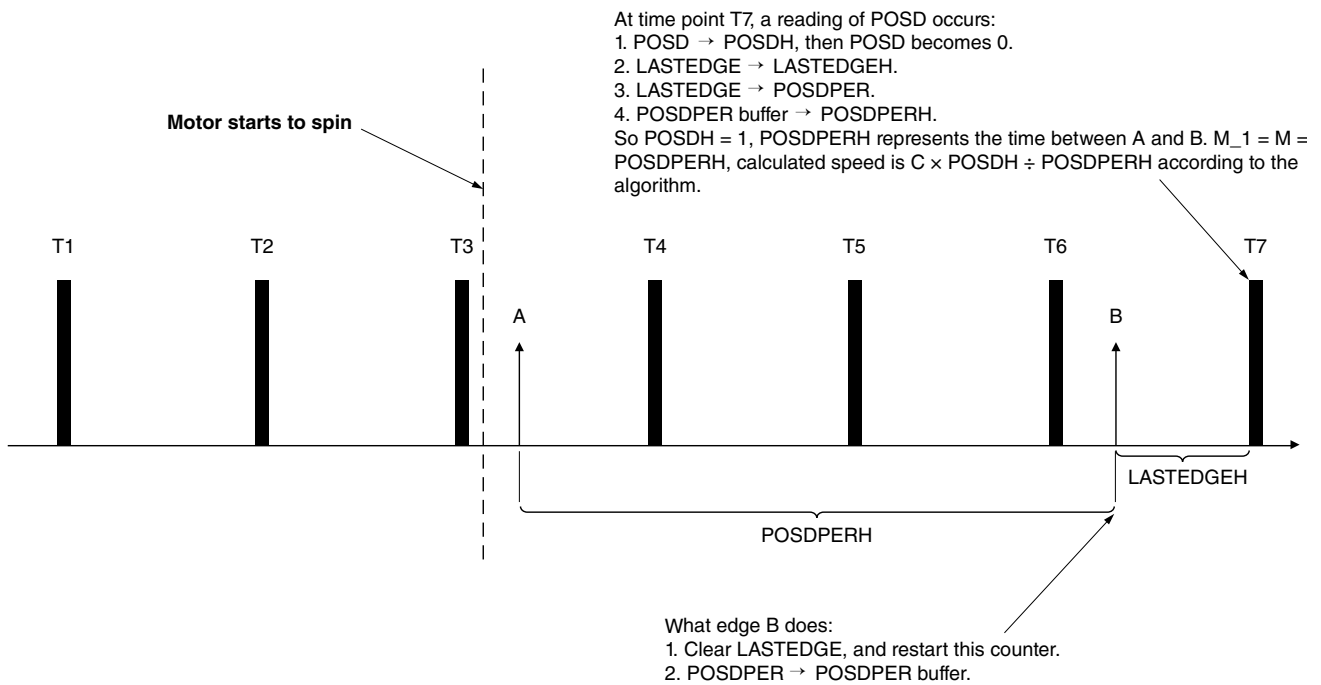


Figure 77-10. Speed measurement at time point T7

77.3.3.3 Speed calculation when motor stops

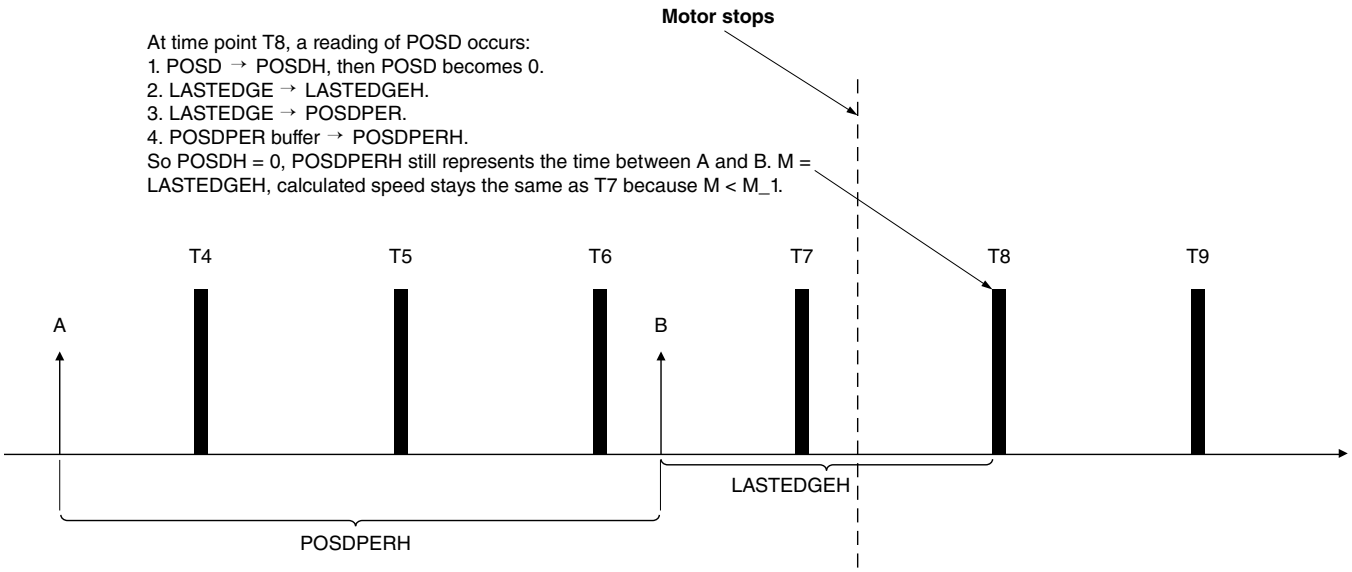


Figure 77-11. Speed measurement at time point T8

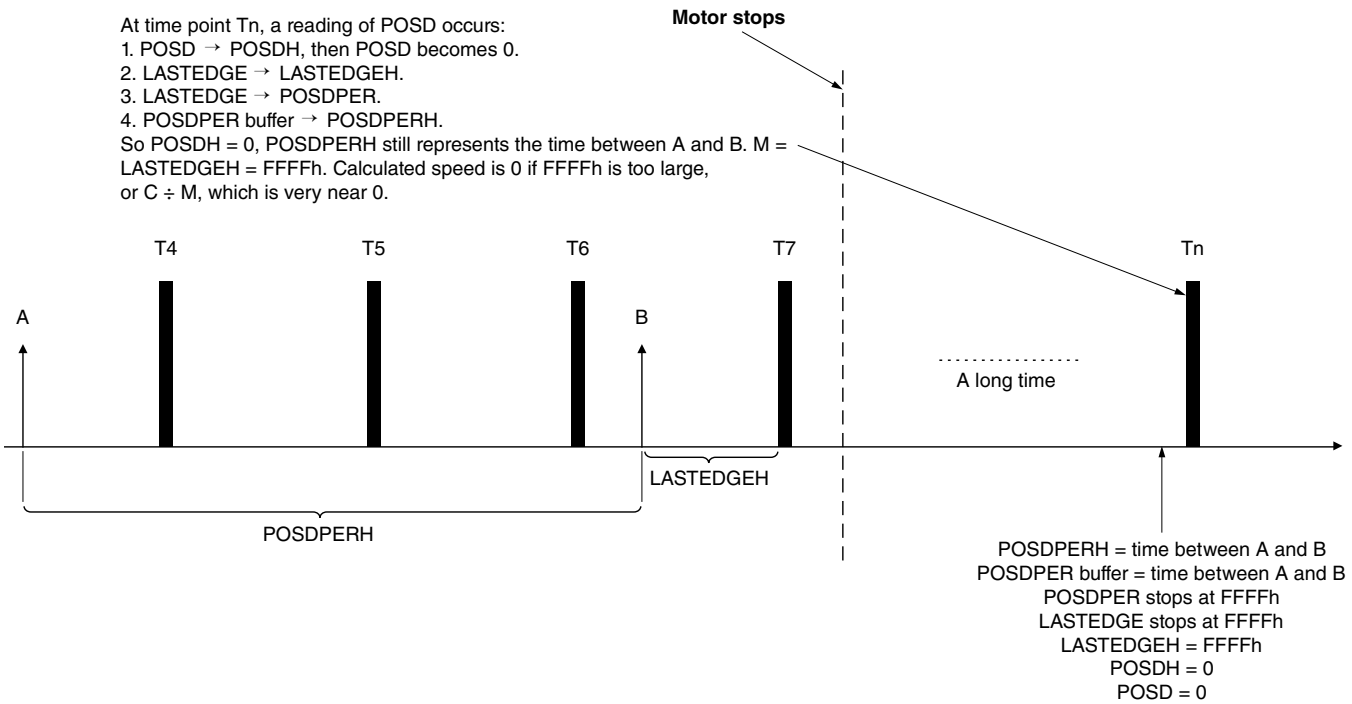


Figure 77-12. Speed measurement at time point Tn

77.3.3.4 Speed calculation when motor starts again

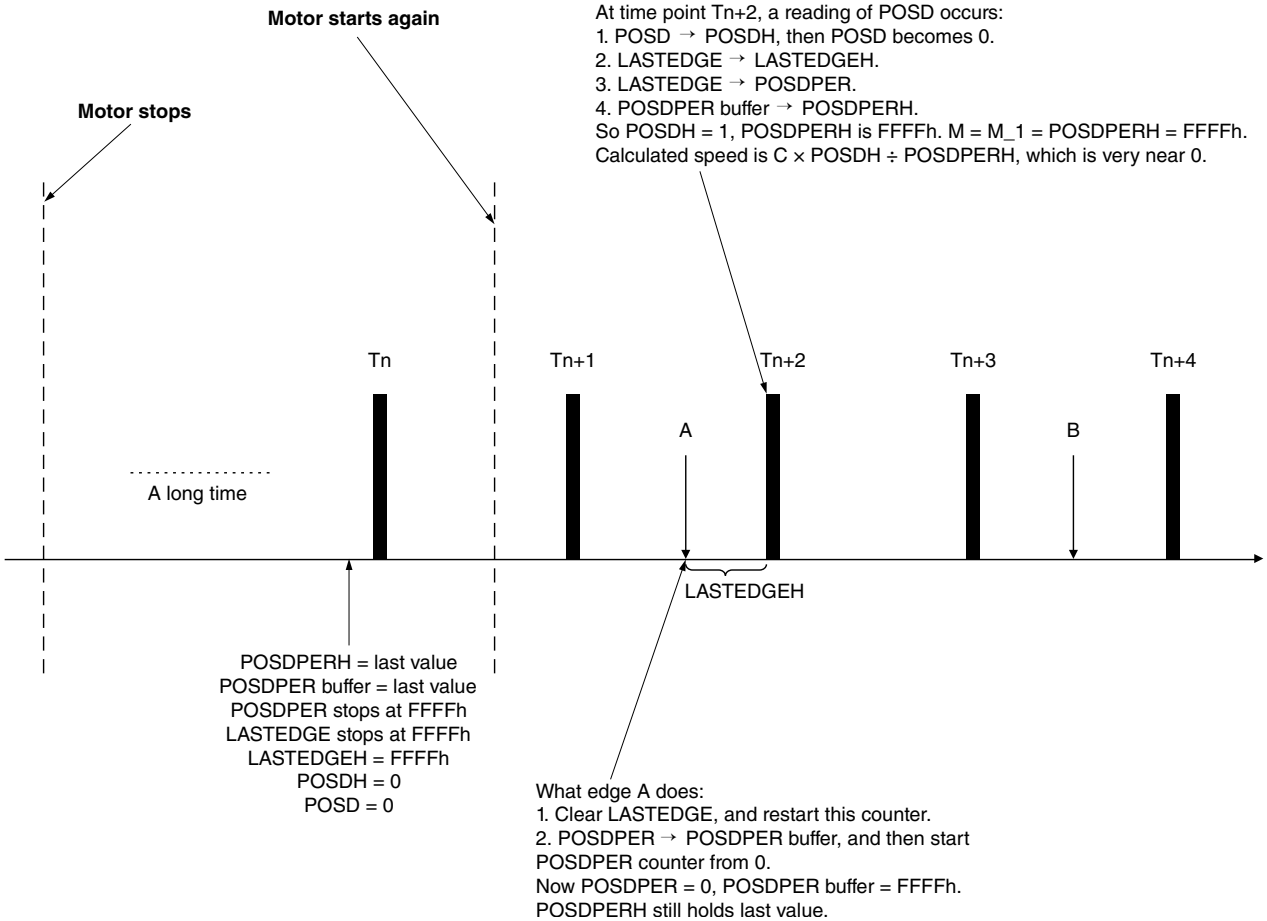


Figure 77-13. Speed measurement at time point Tn+2

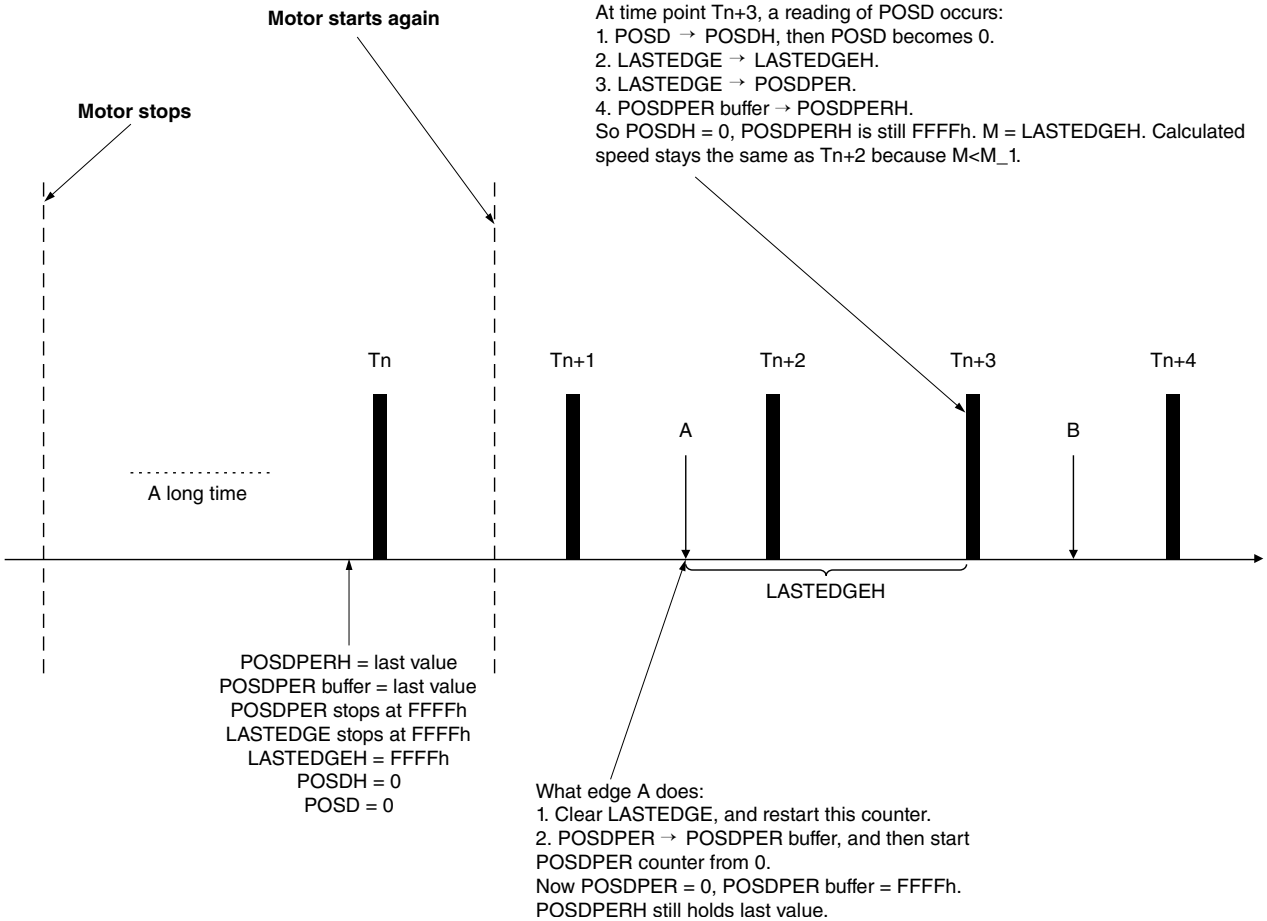


Figure 77-14. Speed measurement at time point Tn+3

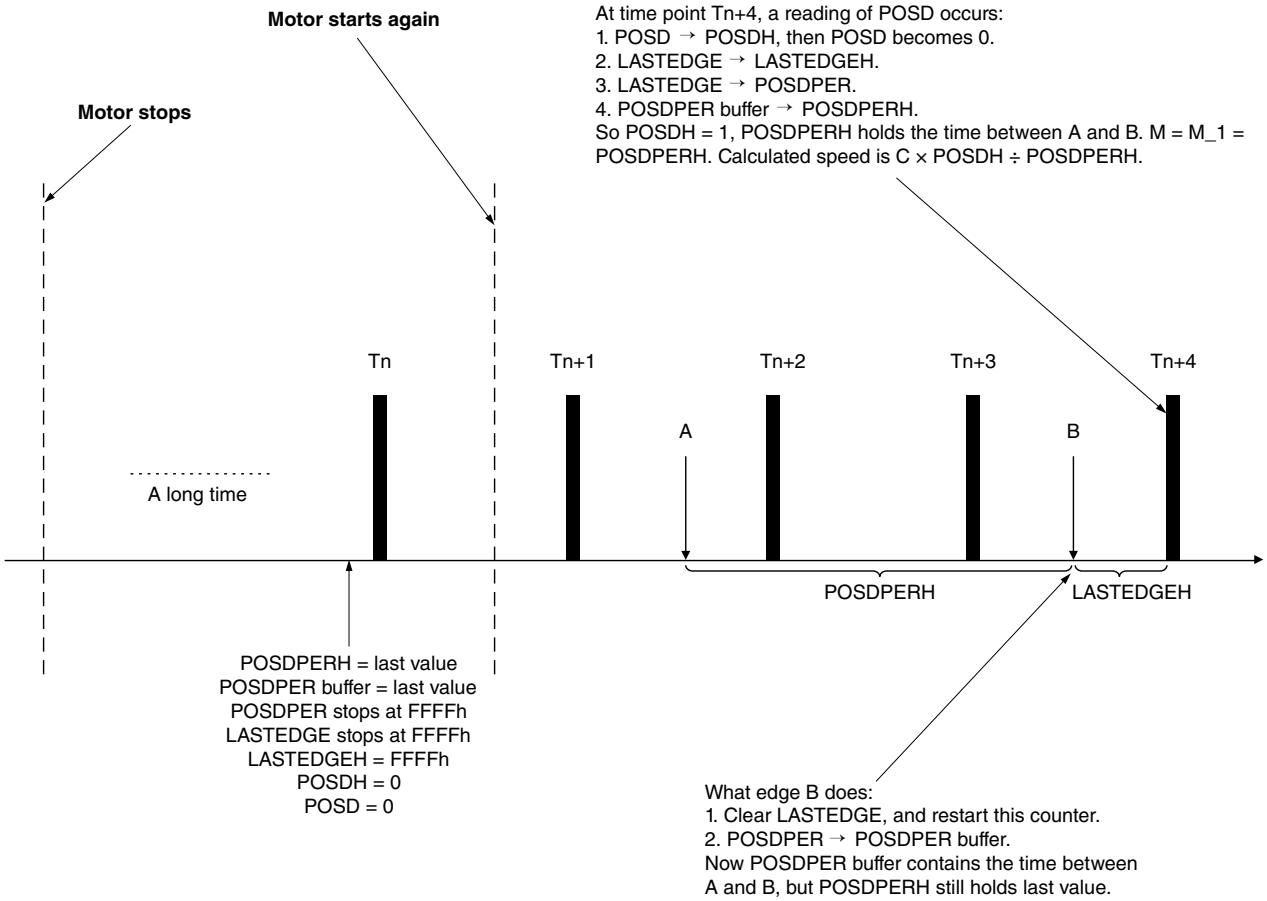


Figure 77-15. Speed measurement at time point Tn+4

77.3.4 Clocks

This module requires only the IPBus clock in normal operation.

77.3.5 Resets

There are no special requirements. Any system reset resets this module.

77.3.6 Interrupts

Table 77-2. Interrupt summary

Core interrupt	Interrupt flag	Interrupt enable	Description
ipi_int_home	CTRL[HIRQ]	CTRL[HIE]	HOME signal transition interrupt

Table continues on the next page...

Table 77-2. Interrupt summary (continued)

Core interrupt	Interrupt flag	Interrupt enable	Description
ipi_int_index	CTRL[XIRQ]	CTRL[XIE]	INDEX signal transition interrupt or roll-over/ under interrupt
	CTRL2[ROIRQ]	CTRL2[ROIE]	
	CTRL2[RUIRQ]	CTRL2[RUIE]	
ipi_int_wdog	CTRL[DIRQ]	CTRL[DIE]	Watchdog timeout interrupt
ipi_int_cmp	CTRL[CMPIRQ]	CTRL[CMPIE]	Compare match interrupt
ipi_int_sab	CTRL2[SABIRQ]	CTRL2[SABIE]	Simultaneous PHASEA and PHASEB change interrupt

77.4 Memory map and registers

This section describes the module memory map and registers.

77.4.1 QDC register descriptions

The address of a register is the sum of a base address and an address offset.

- Base address is defined at the MCU level
- Address offset is defined at the module level

For the base address, see the chip-specific documentation. All memory locations base and offsets are given in hex.

77.4.1.1 QDC memory map

ENC1 base address: 4017_4000h

ENC2 base address: 4017_8000h

ENC3 base address: 4017_C000h

ENC4 base address: 4018_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (CTRL)	16	RW	0000h
1h	Input Filter (FILT)	16	RW	0000h
2h	Watchdog Timeout (WTR)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3h	Position Difference Counter (POSD)	16	RW	0000h
4h	Position Difference Hold (POSDH)	16	R	0000h
5h	Revolution Counter (REV)	16	RW	0000h
6h	Revolution Hold (REVH)	16	R	0000h
7h	Upper Position Counter (UPOS)	16	RW	0000h
8h	Lower Position Counter (LPOS)	16	RW	0000h
9h	Upper Position Hold (UPOSH)	16	R	0000h
Ah	Lower Position Hold (LPOSH)	16	R	0000h
Bh	Upper Initialization (UINIT)	16	RW	0000h
Ch	Lower Initialization (LINIT)	16	RW	0000h
Dh	Input Monitor (IMR)	16	R	0000h
Eh	Test (TST)	16	RW	0000h
Fh	Control 2 (CTRL2)	16	RW	0000h
10h	Upper Modulus (UMOD)	16	RW	0000h
11h	Lower Modulus (LMOD)	16	RW	0000h
12h	Upper Position Compare (UCOMP)	16	RW	FFFFh
13h	Lower Position Compare (LCOMP)	16	RW	FFFFh
14h	Last Edge Time (LASTEDGE)	16	R	FFFFh
15h	Last Edge Time Hold (LASTEDGEH)	16	R	FFFFh
16h	Position Difference Period Counter (POSDPER)	16	R	FFFFh
17h	Position Difference Period Buffer (POSDPERBFR)	16	R	FFFFh
18h	Position Difference Period Hold (POSDPERH)	16	R	FFFFh
19h	Control 3 (CTRL3)	16	RW	0000h

77.4.1.2 Control (CTRL)

77.4.1.2.1 Offset

Register	Offset
CTRL	0h

77.4.1.2.2 Function

Controls basic functions on HOME and INDEX signals and part of basic functions.

77.4.1.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HIRQ	HIE	HIP	HNE	0	REV	PH1	XIRQ	XIE	XIP	XNE	DIRQ	DIE	WDE	CMPIRQ	CMPIE
W	W1C				SWIP			W1C				W1C			W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.2.4 Fields

Field	Function
15 HIRQ	HOME Signal Transition Interrupt Request Specifies whether a transition has occurred on the HOME signal. This field becomes 1 when a HOME signal transitions, according to CTRL[HNE]. If this field and CTRL[HIE] are 1, then a HOME interrupt occurs. 0b - Not occurred 1b - Occurred
14 HIE	HOME Interrupt Enable Enables or disables HOME signal interrupts. 0b - Disable 1b - Enable
13 HIP	Enable HOME to Initialize Position Counters UPOS and LPOS Enables the position counter, which the HOME signal initializes. 0b - No action 1b - HOME signal initializes the position counter
12 HNE	Use Negative Edge of HOME Input Selects using the positive and negative edge of the HOME input. 0b - Use positive-going edge-to-trigger initialization of position counters UPOS and LPOS 1b - Use negative-going edge-to-trigger initialization of position counters UPOS and LPOS
11 SWIP	Software-Triggered Initialization of Position Counters UPOS and LPOS Specifies that writing 1 to this field transfers the Upper Initialization (UINIT) and Lower Initialization (LINIT) contents to Upper Position Counter (UPOS) and Lower Position Counter (LPOS) (thereby initializing those counters). 0b - No action 1b - Initialize position counter
10 REV	Enable Reverse Direction Counting Selects the direction of the count (how you interpret the quadrature signal). 0b - Counts normally 1b - Counts in the reverse direction
9 PH1	Enable Signal Phase Count Mode Bypasses or uses the quadrature decoder logic. 0b - Uses the standard quadrature decoder, where PHASEA and PHASEB represent a two-phase quadrature signal.

Table continues on the next page...

Field	Function
	1b - Bypasses the quadrature decoder. A positive transition of the PHASEA input generates a count signal. PHASEB input and CTRL[REV] controls the counter direction. If the value of CTRL[REV] and PHASEB are identical; then count is up. If the value of CTRL[REV] and PHASEB is different, then count is down.
8 XIRQ	INDEX Pulse Interrupt Request Specifies whether the INDEX pulse has occurred. This field becomes 1 when a transition on the INDEX signal occurs, according to the use negative edge of CTRL[XNE]. If this field and CTRL[XIE] become 1, then an INDEX interrupt occurs. 0b - Not occurred 1b - Occurred
7 XIE	INDEX Pulse Interrupt Enable Enables or disables INDEX pulse interrupts. 0b - Disable 1b - Enable
6 XIP	INDEX Triggered Initialization of Position Counters UPOS and LPOS Enables or disables the position counter that the INDEX pulse initializes. 0b - Does not initialize 1b - Initializes
5 XNE	Select Positive and Negative Edge of INDEX Pulse Selects the positive and negative edge of the INDEX pulse. 0b - Use positive edge 1b - Use negative edge
4 DIRQ	Watchdog Timeout Interrupt Request Specifies whether the watchdog timeout interrupt has occurred. This field also becomes 0 when CTRL[WDE] = 0. 0b - Not occurred 1b - Occurred
3 DIE	Watchdog Timeout Interrupt Enable Enables or disables the watchdog timeout interrupts. 0b - Disable 1b - Enable
2 WDE	Watchdog Enable Enables or disables the watchdog timer that monitors the PHASEA and PHASEB inputs for motor movement. 0b - Disable 1b - Enable
1 CMPIRQ	Compare Interrupt Request Specifies whether the COMP match has occurred. This field becomes 1 when the counter matches the COMP value. 0b - No match has occurred 1b - COMP match has occurred
0 CMPIE	Compare Interrupt Enable Enables the compare interrupt. 0b - Disable 1b - Enable

77.4.1.3 Input Filter (FILT)

77.4.1.3.1 Offset

Register	Offset
FILT	1h

77.4.1.3.2 Function

Specifies:

- The input filter sample period.
- The filter sample count.

Choose FILT[FILT_PER] such that the sampling period is larger than the period of the expected noise. Doing this means that a noise spike will only corrupt one sample.

Choose FILT[FILT_CNT] to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of FILT_CNT+3.

You must also trade off the values of FILT[FILT_PER] and FILT[FILT_CNT] against the need for minimal latency in recognizing valid input transitions. Turning on the input filter (write a non-zero value to FILT[FILT_PER]) introduces a latency of ((FILT[FILT_CNT] + 3)×FILT[FILT_PER]) FILT clock cycles + 2 IPBus clock periods.

You can measure the filter latency:

- Drive the quadrature decoder inputs (PHASEA, PHASEB, INDEX, HOME).
- Monitor the filtered output in Input Monitor (IMR).
- Determine how many IPBus clock cycles it takes before the output shows up, using the following equations, where f is FILT[FILT_PER], and s is FILT[FILT_CNT].
  - DELAY = f×(s+3)(FILT clock cycles) + 1 (IPBus clock cycles)(to read the filtered output)
  - DELAY = f×(s+3)(FILT clock cycles) + 2 (IPBus clock cycles) (to monitor the output in the IMR)

One additional IPBus clock cycle is required to read the filtered output, and two more IPBus clock cycles are required to monitor the filtered output in Input Monitor (IMR). The sample rate sets when it reaches the number f. The following examples use the preceding equations:

- Example: when f = 0, the filter is bypassed: DELAY = 1 or 2 clock cycles.
- Example: when f = 5 and s = 2: DELAY = 5×(2+3) FILT clock cycles + (1 or 2) IPBus clock cycles

77.4.1.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	0			0		FILT_CNT			FILT_PER							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.3.4 Fields

Field	Function
15-13	Reserved
12-11 —	Reserved
10-8 FILT_CNT	Input Filter Sample Count Represents the number of consecutive samples that must agree, before the input filter accepts an input transition. • A value of 0h represents three samples • A value of 7h represents ten samples The value of this field affects the input latency.
7-0 FILT_PER	Input Filter Sample Period Represents the decoder input signals' sampling period (in IPBus clock cycles). Sample each input multiple times at the rate this field specifies. • If FILT_PER is 00h, then the input filter is bypassed. Bypassing the digital filter enables the position/position difference counters to operate with count rates up to the IPBus frequency. • The value of this field affects the input latency. • When you change the value of this field from a non-zero value to another non-zero value, write 0 first to clear the filter.

77.4.1.4 Watchdog Timeout (WTR)

77.4.1.4.1 Offset

Register	Offset
WTR	2h

77.4.1.4.2 Function

Stores the timeout count for the quadrature decoder module watchdog timer. This quadrature decoder module watchdog timer is different from any other watchdog timer(s) that may also be present in the chip.

77.4.1.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WDOG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.4.4 Fields

Field	Function
15-0 WDOG	WDOG Represents the number of clock cycles, plus one clock cycle that the watchdog timer counts before timing out (and optionally generating an interrupt).

77.4.1.5 Position Difference Counter (POSD)

77.4.1.5.1 Offset

Register	Offset
POSD	3h

77.4.1.5.2 Function

Contains the position change in value occurring between each read of the position register. You can use the value of this register to calculate velocity.

- This register computes up or down on every count pulse.
- This register acts as a differentiator, whose count value is proportional to the change in position since the last time you read the position counter.
- When you read this register, its contents are copied into Position Difference Hold (POSDH) and this register becomes 0.

77.4.1.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POSD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.5.4 Fields

Field	Function
15-0 POSD	POSD Specifies the position change in value between each read of the position register.

77.4.1.6 Position Difference Hold (POSDH)

77.4.1.6.1 Offset

Register	Offset
POSDH	4h

77.4.1.6.2 Function

Contains a snapshot of the value of Position Difference Counter (POSD). You can use the value of this register to calculate velocity.

77.4.1.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POSDH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.6.4 Fields

Field	Function
15-0	POSDH

Field	Function
POSDH	Holds the Position Difference Counter (POSD) value.

77.4.1.7 Revolution Counter (REV)

77.4.1.7.1 Offset

Register	Offset
REV	5h

77.4.1.7.2 Function

Contains the current value of the revolution counter.

77.4.1.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.7.4 Fields

Field	Function
15-0 REV	REV Contains the current value of the revolution counter.

77.4.1.8 Revolution Hold (REVH)

77.4.1.8.1 Offset

Register	Offset
REVH	6h

77.4.1.8.2 Function

Contains a snapshot of the value of Revolution Counter (REV).

77.4.1.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	REVH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.8.4 Fields

Field	Function
15-0 REVH	REVH Holds the Revolution Counter (REV) value.

77.4.1.9 Upper Position Counter (UPOS)

77.4.1.9.1 Offset

Register	Offset
UPOS	7h

77.4.1.9.2 Function

Contains the upper (most significant) half of the position counter. This is the binary count from the position counter.

77.4.1.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.9.4 Fields

Field	Function
15-0 POS	POS Contains the upper (most significant) half of the position counter.

77.4.1.10 Lower Position Counter (LPOS)

77.4.1.10.1 Offset

Register	Offset
LPOS	8h

77.4.1.10.2 Function

Contains the lower (least significant) half of the position counter. This is the binary count from the position counter.

77.4.1.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.10.4 Fields

Field	Function
15-0 POS	POS Contains the lower (least significant) half of the position counter.

77.4.1.11 Upper Position Hold (UPOSH)

77.4.1.11.1 Offset

Register	Offset
UPOSH	9h

77.4.1.11.2 Function

Contains a snapshot of Upper Position Counter (UPOS).

77.4.1.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POSH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.11.4 Fields

Field	Function
15-0 POSH	POSH Holds the Upper Position Counter (UPOS) value.

77.4.1.12 Lower Position Hold (LPOSH)

77.4.1.12.1 Offset

Register	Offset
LPOSH	Ah

77.4.1.12.2 Function

Contains a snapshot of Lower Position Counter (LPOS).

77.4.1.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POSH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.12.4 Fields

Field	Function
15-0 POSH	POSH Holds the Lower Position Counter (LPOS) value.

77.4.1.13 Upper Initialization (UINIT)

77.4.1.13.1 Offset

Register	Offset
UINIT	Bh

77.4.1.13.2 Function

Contains the value to initialize Upper Position Counter (UPOS).

77.4.1.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INIT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.13.4 Fields

Field	Function
15-0 INIT	INIT Contains the value to initialize Upper Position Counter (UPOS).

77.4.1.14 Lower Initialization (LINIT)

77.4.1.14.1 Offset

Register	Offset
LINIT	Ch

77.4.1.14.2 Function

Contains the value to initialize Lower Position Counter (LPOS).

77.4.1.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.14.4 Fields

Field	Function
15-0 INIT	INIT Contains the value to initialize Lower Position Counter (LPOS).

77.4.1.15 Input Monitor (IMR)

77.4.1.15.1 Offset

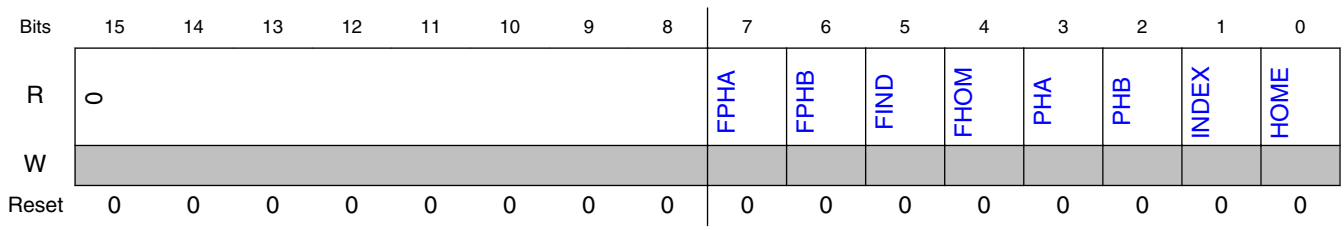
Register	Offset
IMR	Dh

77.4.1.15.2 Function

Contains the values of the raw and filtered PHASEA, PHASEB, INDEX, and HOME input signals. The reset value depends on the values of the raw and filtered values of PHASEA, PHASEB, INDEX, and HOME input signals:

- If you connect these input pins to a pullup, then bits 0–7 of this register are all ones.
- If you connect these input pins to a pulldown device, then bits 0–7 of this register are all zeros.

77.4.1.15.3 Diagram



77.4.1.15.4 Fields

Field	Function
15-8 —	Reserved
7 FPHA	FPHA Specifies the filtered version of PHASEA input.
6 FPHB	FPHB Specifies the filtered version of PHASEB input.
5 FIND	FIND Specifies the filtered version of INDEX input.
4 FHOM	FHOM Specifies the filtered version of HOME input.
3 PHA	PHA Specifies the raw PHASEA input.
2 PHB	PHB Specifies the raw PHASEB input.
1 INDEX	INDEX Specifies the raw INDEX input.
0 HOME	HOME Specifies the raw HOME input.

77.4.1.16 Test (TST)

77.4.1.16.1 Offset

Register	Offset
TST	Eh

77.4.1.16.2 Function

Controls and sets the frequency of a quadrature signal generator; it provides a quadrature test signal to the inputs of the quadrature decoder module.

- The TEST_COUNT value is counted down to zero when TST[TEN] and TST[TCE] are 1.
- Each count value of one represents a single quadrature cycle, which the position counter (Upper Position Counter (UPOS) and Lower Position Counter (LPOS)) interprets as a count of one if enabled (TST[TEN] and TST[TCE] are 1).
- Repeated writing of new values to TEST_COUNT can cause an extra phase transition and, therefore an extra count by the Position Counter.
- The period field determines each quadrature cycle phase's length (in IPBus clock cycles).

This register is a factory test feature; however, you may find it helpful during your software development and testing.

77.4.1.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEN	TCE	QDN	TEST_PERIOD					TEST_COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.16.4 Fields

Field	Function
15 TEN	Test Mode Enable Connects the test module to inputs of the quadrature decoder module. 0b - Disable 1b - Enable

Table continues on the next page...

Field	Function
14 TCE	Test Counter Enable Connects the test counter to inputs of the quadrature decoder module. 0b - Disable 1b - Enable
13 QDN	Quadrature Decoder Negative Signal Selects whether a negative or positive quadrature decoder signal is generated. 0b - Positive quadrature decoder signal 1b - Negative quadrature decoder signal
12-8 TEST_PERIOD	TEST_PERIOD Specifies the period of the quadrature phase in IPBus clock cycles.
7-0 TEST_COUNT	TEST_COUNT Specifies the number of quadrature advances to generate.

77.4.1.17 Control 2 (CTRL2)

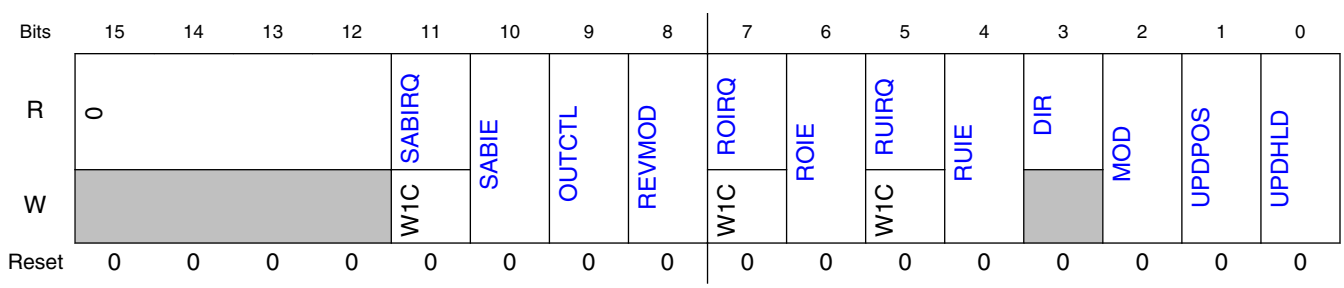
77.4.1.17.1 Offset

Register	Offset
CTRL2	Fh

77.4.1.17.2 Function

Specifies the status of interrupts, interrupt enable, and part of basic functions.

77.4.1.17.3 Diagram



77.4.1.17.4 Fields

Field	Function
15-12 —	Reserved
11 SABIRQ	Simultaneous PHASEA and PHASEB Change Interrupt Request Indicates that the PHASEA and PHASEB inputs changed simultaneously (within a single clock period). This event typically indicates an error condition, because quadrature coding requires only one of these inputs to change at a time. 0b - No simultaneous change has occurred 1b - A simultaneous change has occurred
10 SABIE	Simultaneous PHASEA and PHASEB Change Interrupt Enable Enables simultaneous PHASEA and PHASEB change interrupts when CTRL2[SABIRQ] becomes 1. 0b - Disable 1b - Enable
9 OUTCTL	Output Control Controls the pulsing of the POSMATCH output signal. This can control when a timer channel captures a timestamp. 0b - POSMATCH pulses when a match occurs between the position counters (POS) and the corresponding compare value (COMP) 1b - POSMATCH pulses when the UPOS, LPOS, REV, or POSD registers are read
8 REVMOD	Revolution Counter Modulus Enable Selects how Revolution Counter (REV) increments or decrements. By default, Revolution Counter (REV) is controlled based on the count direction and the INDEX pulse. As an option, you can control the revolution counter using the roll-over/under detection during modulo counting. 0b - Use INDEX pulse 1b - Use modulus counting roll-over or roll-under
7 ROIRQ	Roll-over Interrupt Request Indicates when the position counter (POS) rolls over from the MOD value to the INIT value or from FFFF_FFFFh to 0000_0000h. 0b - Did not occur 1b - Occurred
6 ROIE	Roll-over Interrupt Enable Enables roll-over interrupts when CTRL2[ROIRQ] is 1. This roll-over interrupt combines with the index interrupt signal. 0b - Disable 1b - Enable
5 RUIRQ	Roll-under Interrupt Request Indicates when the position counter (POS) rolls under from the INIT value to the MOD value or from 0000_0000h to FFFF_FFFFh. 0b - No roll-under has occurred 1b - Roll-under has occurred
4 RUIE	Roll-under Interrupt Enable Enables roll-under interrupts when CTRL2[RUIRQ] is 1. This roll-under interrupt combines with the index interrupt signal. 0b - Disable 1b - Enable
3 DIR	Count Direction Flag Indicates the direction of the last count.

Table continues on the next page...

Field	Function
	0b - Down direction 1b - Up direction

2
MOD
	Enable Modulo Counting
	Has multiple functions:
	• When this field is 1, it allows Upper Position Counter (UPOS) and Lower Position Counter (LPOS) to count in a modulo fashion, using MOD and INIT as the upper and lower bounds of the counting range. During modulo counting:
	  • When you indicate a "count up" and the position counter is equal to MOD, then the position counter reloads with the value of INIT.
	  • When you indicate a "count down" and the position counter is equal to INIT, then the position counter reloads with the value of MOD.
	• When this field is 0, then the values of MOD and INIT are ignored, and the position counter wraps to zero when counting up from FFFF_FFFFh, or wraps to FFFF_FFFFh when counting down from 0.
	
	0b - Disable
	1b - Enable

1 UPDPOS	Update Position Registers Has multiple functions: • When this field is 1, it allows the TRIGGER input to write 0 to Position Difference Counter (POSD), Revolution Counter (REV), Upper Position Counter (UPOS), and Lower Position Counter (LPOS). • When this field is 0, Position Difference Counter (POSD), Revolution Counter (REV), Upper Position Counter (UPOS), and Lower Position Counter (LPOS) ignore the TRIGGER input. Do not write 1 to this field when you use Last Edge Time (LASTEDGE) and Position Difference Period Counter (POSDPER) to measure speed. 0b - No action 1b - Clear
0 UPDHLD	Update Hold Registers Has multiple functions: • When this field is 1, it allows the TRIGGER input to update Position Difference Hold (POSDH), Revolution Hold (REVH), Upper Position Hold (UPOSH), and Lower Position Hold (LPOSH). • When this field is 0, the TRIGGER input does not update the hold registers (Position Difference Hold (POSDH), Revolution Hold (REVH), Upper Position Hold (UPOSH), and Lower Position Hold (LPOSH)). Do not write 1 to this field when you use Last Edge Time (LASTEDGE) and Position Difference Period Counter (POSDPER) to measure speed. Updating Position Difference Hold (POSDH) causes Position Difference Counter (POSD) to become 0. 0b - Disable 1b - Enable

77.4.1.18 Upper Modulus (UMOD)

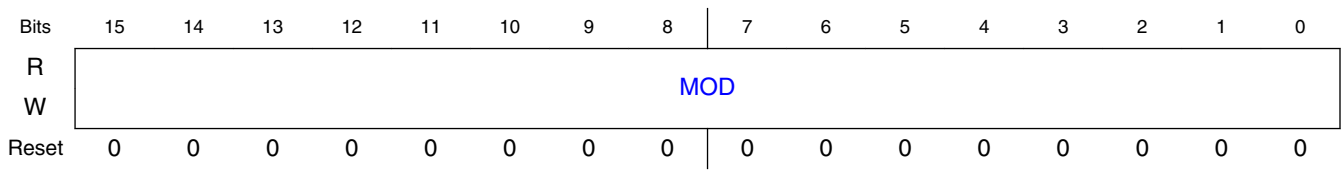
77.4.1.18.1 Offset

Register	Offset
UMOD	10h

77.4.1.18.2 Function

Contains the upper (most significant) half of the modulus register.

77.4.1.18.3 Diagram



77.4.1.18.4 Fields

Field	Function
15-0 MOD	MOD Specifies the upper (most significant) half of the modulus register. This field acts as the upper bound during modulo counting and as the upper reload value when rolling over from the lower bound.

77.4.1.19 Lower Modulus (LMOD)

77.4.1.19.1 Offset

Register	Offset
LMOD	11h

77.4.1.19.2 Function

Contains the lower (least significant) half of the modulus register.

77.4.1.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

77.4.1.19.4 Fields

Field	Function
15-0 MOD	MOD Specifies the lower (least significant) half of the modulus register. This field acts as the upper bound during modulo counting and as the upper reload value when rolling over from the lower bound.

77.4.1.20 Upper Position Compare (UCOMP)

77.4.1.20.1 Offset

Register	Offset
UCOMP	12h

77.4.1.20.2 Function

Specifies the upper (most significant) half of the position compare register. When the value of the position counter (POS) matches the value of the position compare register (COMP), CTRL[CMPIRQ] becomes 1, and the POSMATCH output is asserted.

77.4.1.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

77.4.1.20.4 Fields

Field	Function
15-0 COMP	COMP Specifies the upper (most significant) half of the position compare register.

77.4.1.21 Lower Position Compare (LCOMP)

77.4.1.21.1 Offset

Register	Offset
LCOMP	13h

77.4.1.21.2 Function

Contains the lower (least significant) half of the position compare register. When the value of the position counter (POS) matches the value of the position compare register (COMP), CTRL[CMPIRQ] becomes 1 and the POSMATCH output is asserted.

77.4.1.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

77.4.1.21.4 Fields

Field	Function
15-0 COMP	COMP Specifies the lower (least significant) half of the position compare register.

77.4.1.22 Last Edge Time (LASTEDGE)

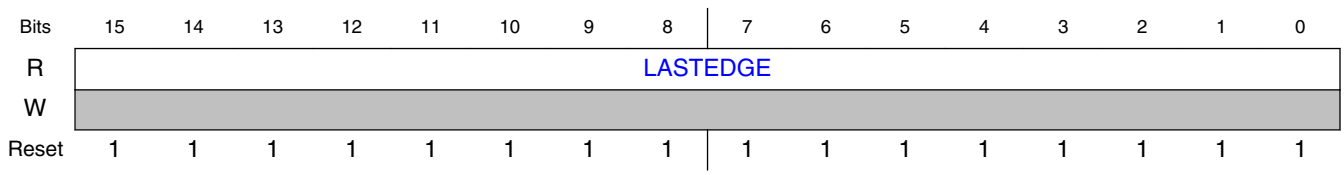
77.4.1.22.1 Offset

Register	Offset
LASTEDGE	14h

77.4.1.22.2 Function

Represents the time after the last edge occurred on PHASEA or PHASEB.

77.4.1.22.3 Diagram



77.4.1.22.4 Fields

Field	Function
15-0 LASTEDGE	Last Edge Time Counter Represents the time after the last edge occurred on PHASEA or PHASEB. This field counts up using the peripheral clock that CTRL3[PRSC] prescales. Any edge on PHASEA or PHASEB resets this field to 0 and starts counting. If the LASTEDGE count reaches FFFFh, the counting stops to prevent an overflow. Counting continues when an edge occurs on PHASEA or PHASEB.

77.4.1.23 Last Edge Time Hold (LASTEDGEH)

77.4.1.23.1 Offset

Register	Offset
LASTEDGEH	15h

77.4.1.23.2 Function

Holds register for the Last Edge Time (LASTEDGE) value.

77.4.1.23.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LASTEDGEH															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

77.4.1.23.4 Fields

Field	Function
15-0 LASTEDGEH	Last Edge Time Hold Holds the Last Edge Time (LASTEDGE) value when you read Position Difference Counter (POSD).

77.4.1.24 Position Difference Period Counter (POSDPER)

77.4.1.24.1 Offset

Register	Offset
POSDPER	16h

77.4.1.24.2 Function

Indicates the position-difference period.

77.4.1.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POSDPER															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

77.4.1.24.4 Fields

Field	Function
15-0 POSDPER	Position difference period

Field	Function
	Counts up using the peripheral clock that CTRL3[PRSC] prescales. Reading Position Difference Counter (POSD) loads the value of Last Edge Time (LASTEDGE) into Position Difference Period Counter (POSDPER). If Position Difference Period Counter (POSDPER) count reaches FFFFh, the counting stops to prevent an overflow. Counting continues when an edge occurs on PHASEA or PHASEB.

77.4.1.25 Position Difference Period Buffer (POSDPERBFR)

77.4.1.25.1 Offset

Register	Offset
POSDPERBFR	17h

77.4.1.25.2 Function

Buffers the value of Position Difference Period Counter (POSDPER).

77.4.1.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POSDPERBFR															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

77.4.1.25.4 Fields

Field	Function
15-0 POSDPERBFR	Position difference period buffer Updates with Position Difference Period Counter (POSDPER) value when any edge occurs on PHASEA or PHASEB.

77.4.1.26 Position Difference Period Hold (POSDPERH)

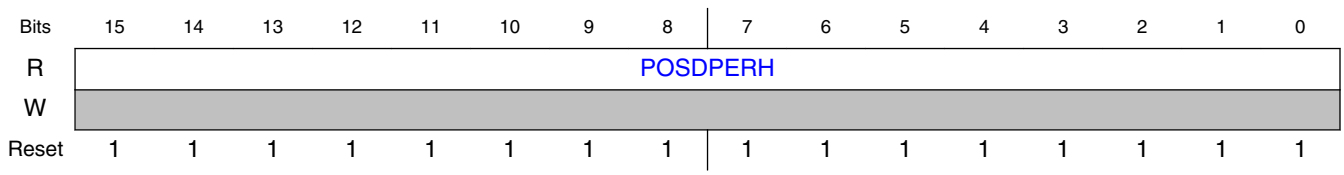
77.4.1.26.1 Offset

Register	Offset
POSDPERH	18h

77.4.1.26.2 Function

Holds register for Position Difference Period Counter (POSDPER) value.

77.4.1.26.3 Diagram



77.4.1.26.4 Fields

Field	Function
15-0 POSDPERH	Position difference period hold Updates with Position Difference Period Buffer (POSDPERBFR) value when you read Position Difference Counter (POSD).

77.4.1.27 Control 3 (CTRL3)

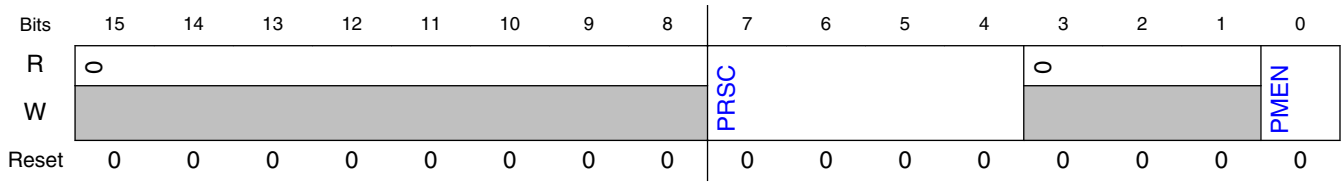
77.4.1.27.1 Offset

Register	Offset
CTRL3	19h

77.4.1.27.2 Function

Specifies prescaler values for the period clock divider and period measurement function.

77.4.1.27.3 Diagram



77.4.1.27.4 Fields

Field	Function
15-8 —	Reserved
7-4 PRSC	Prescaler Prescales the peripheral clock that LASTEDGE[LASTEDGE] and POSDPER[POSDPER] use. The resulting prescale clock value is 2^{PRSC}. The prescaler logic can divide the clock by a minimum of 1 and a maximum of 32,768.
3-1 —	Reserved
0 PMEN	Period Measurement Function Enable Specifies that the period-measurement functions of the following registers are used: • Last Edge Time (LASTEDGE) • Last Edge Time Hold (LASTEDGEH) • Position Difference Period Counter (POSDPER) • Position Difference Period Buffer (POSDPERBFR) • Position Difference Period Hold (POSDPERH) If this field is 0, Position Difference Counter (POSD) is loaded to Position Difference Hold (POSDH) and then becomes 0 whenever you read Position Difference Counter (POSD), Upper Position Counter (UPOS), Lower Position Counter (LPOS), or Revolution Counter (REV). If this field is 1, Position Difference Counter (POSD) is loaded to Position Difference Hold (POSDH) and then becomes 0 only when you read Position Difference Counter (POSD). 0b - Not used 1b - Used

Chapter 78
System Watchdog Timer (WDOG 1,2)

78.1 Chip-specific System Watchdog information

Table 78-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

System WDOG1 services the system reset signal (WDOG_RESET_B_DEB) to System Reset Controller (SRC).

System WDOG2 asserts the interrupt to SNVS to report a Security Violation (SV).

78.2 Overview

The Watchdog Timer (WDOG) protects against system failures by providing a method by which to escape from unexpected events or programming errors.

Once the WDOG is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG asserts the internal system reset signal, WDOG_RESET_B_DEB to the System Reset Controller (SRC).

There is also a provision for WDOG(ipp_wdog) signal assertion by timeout counter expiration. There is an option of programmable interrupt generation before the counter actually times out. The time at which the interrupt needs to be generated prior to counter timeout is programmable. There is a power down counter which is enabled out of any reset (POR, Warm/Cold). This counter has a fixed timeout period of 16 seconds, upon which it asserts the WDOG(ipp_wdog) signal.

Flow diagrams for the timeout counter, power down counter and interrupt operations are are shown in Flow Diagrams.

78.2.1 Block Diagram

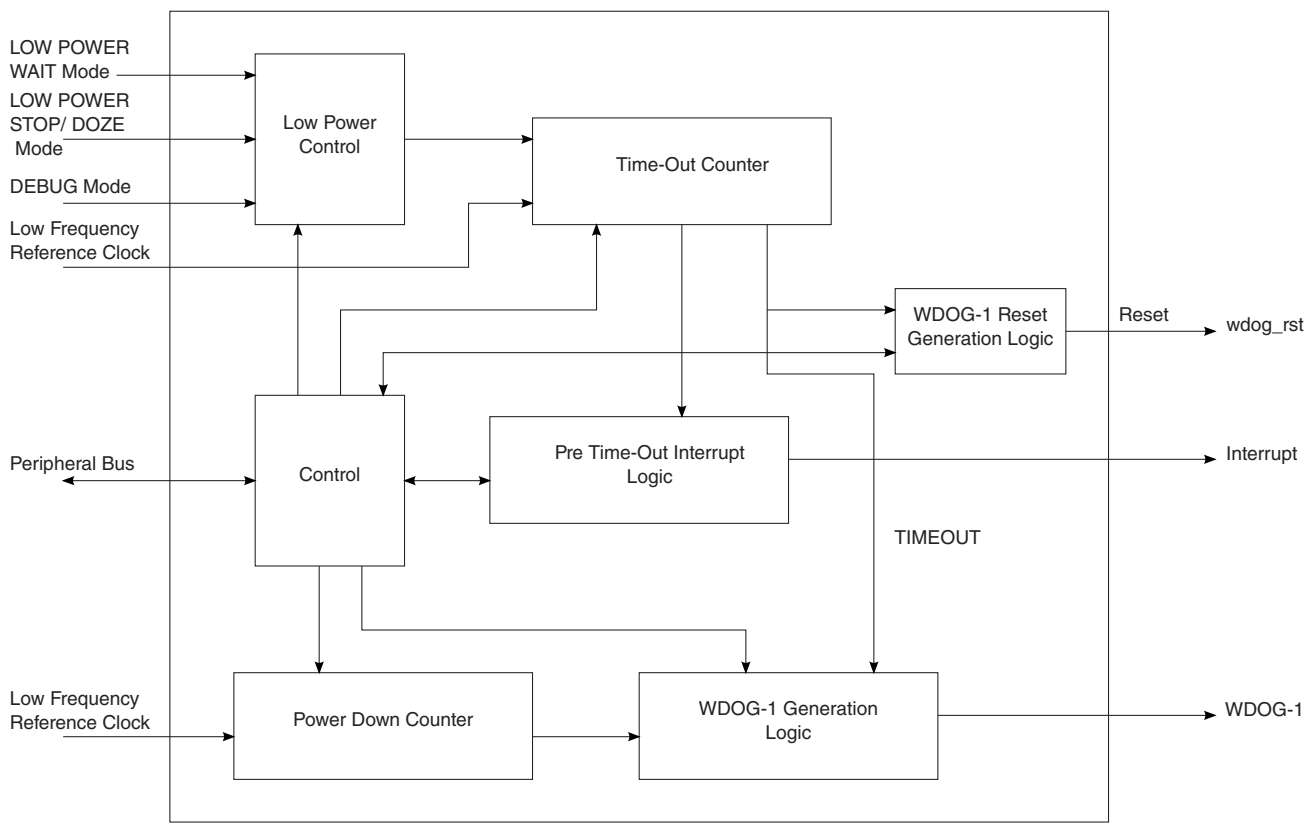


Figure 78-1. WDOG Diagram

78.2.2 Features

The WDOG features are listed below:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal.
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low-power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode
- Programmable interrupt generation prior to timeout
- The duration between interrupt and timeout events can be programmed from 0 to 127.5 seconds in steps of 0.5 seconds.
- Power down counter with fixed timeout period of 16 seconds, which if not disabled after reset will assert WDOG_B (ipp_wdog) signal low
- Power down counter will be enabled out of any reset (POR, Warm / Cold reset) by default.

78.3 Functional description

This section provides a complete functional description of the block.

78.3.1 Watchdog mechanism and system integration

The modules are disabled by default (after reset). WDOG1 will be configured during boot while WDOG2 is dedicated for secure world purposes and will be activated by TZ software if required. The TZ watchdog (TZ watchdog) module protects against TZ starvation by providing a method of escaping normal mode and forcing a switch to the TZ mode.TZ starvation is a situation where the normal OS prevents switching to the TZ mode. Such a situation is undesirable as it can compromise the system's security.

Once the TZ WDOG module is activated, it must be serviced by TZ on a periodic basis. If servicing does not take place, the timer times out. Upon a timeout, the TZ WDOG asserts a TZ-mapped interrupt that forces switching to the TZ mode. If it is still not serviced, the TZ WDOG asserts a security violation signal to the CSU. The TZ WDOG module cannot be programmed or de-activated by normal mode software.

The WDOG modules operate as follows:

- If servicing does not take place, the timer times out and the wdog_rst_b signal is activated (low)
- Interrupt can be generated before the counter actually times out

- The wdog_rst_b signal can be activated by software
- There is a power-down counter which gets enabled out of any reset. This counter has a fixed timeout period of 16 seconds upon which it will assert the ipp_wdog_b signal.

78.3.2 Timeout event

The WDOG provides timeout periods from 0.5 to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the timeout period by writing to the WDOG timeout field (WT[7:0]) in the Watchdog Control Register (WCR). The WDOG must be enabled by setting the WDE bit of Watchdog Control Register (WCR) for the timeout counter to start running. After the WDOG is enabled, the counter is activated, loads the timeout value and begins to count down from this programmed value. The timer will time out when the counter reaches zero and the WDOG outputs a system reset signal, WDOG_RESET_B_DEB and asserts WDOG_B (ipp_wdog_b) (WDT bit should be set in Watchdog Control Register (WCR)).

However, the timeout condition can be prevented by reloading the counter with the new timeout value (WT[7:0] of WDOG_WCR) if a service routine (see Servicing WDOG to reload the counter) is performed before the counter reaches zero. If any system errors occur which prevent the software from servicing the Watchdog Service Register (WSR), the timeout condition occurs. By performing the service routine, the WDOG reloads its counter to the timeout value indicated by bits WT[7:0] of the Watchdog Control Register (WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in Flow Diagrams.

NOTE

The timeout value is reloaded to the counter either at the time WDOG is enabled or after the service routine has been performed.

78.3.2.1 Servicing WDOG to reload the counter

To reload a timeout value to the counter, the proper service sequence begins by writing 0x_5555 followed by 0x_AAAA to the Watchdog Service Register (WSR). Any number of instructions can be executed between the two writes. If the WDOG_WSR is not loaded with 0x_5555 prior to writing 0x_AAAA to the WDOG_WSR, the counter is not

reloaded. If any value other than 0x_AAAA is written to the WDOG_WSR after 0x_5555, the counter is not reloaded. This service sequence will reload the counter with the timeout value WT[7:0] of Watchdog Control Register (WCR). The timeout value can be changed at any point; it is reloaded when WDOG is serviced by the core.

78.3.3 Interrupt event

Prior to timeout, the WDOG can generate an interrupt which can be considered a warning that timeout will occur shortly.

The duration between interrupt event and timeout event can be controlled by writing to the WICT field of Watchdog Interrupt Control Register (WICR). It can vary between 0 and 127.5 seconds. If the WDOG is serviced (Servicing WDOG to reload the counter) before the interrupt generation, the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WCR) and the interrupt will not be triggered.

78.3.4 Power-down counter event

The power-down counter inside WDOG will be enabled out of reset. This counter has a fixed timeout value of 16 seconds, after which it will drive the WDOG_B (ipp_wdog) signal low.

To prevent this, the software must disable this counter by clearing the PDE bit of Watchdog Miscellaneous Control Register (WMCR) within 16 seconds of reset deassertion. Once disabled, this counter can't be enabled again until the next system reset occurs. This feature is intended to prevent the hanging up of cores after reset, as WDOG is not enabled out of reset.

78.3.5 Low power modes

78.3.5.1 STOP and DOZE mode

If the WDOG timer disable bit for low power STOP and DOZE mode (WDZST) bit in the Watchdog Control Register (WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock. If the low power enable (WDZST) bit is set, the WDOG timer operation will be suspended in low power STOP or DOZE mode. Upon exiting low power STOP or DOZE mode, the WDOG operation returns to what it was prior to entering the STOP or DOZE mode.

78.3.5.2 WAIT mode

If the WDOG timer disable bit for low power WAIT mode (WDW) bit in the Watchdog Control Register (WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock i.e *ipg_clk_32k* clock (32.768 kHz frequency clock). If the low power WAIT enable (WDW) bit is set, the WDOG timer operation will be suspended. Upon exiting low power WAIT mode, the WDOG operation returns to what it was prior to entering the WAIT mode.

NOTE

The WDOG timer won't be able to detect events that happen for periods shorter than one low frequency reference clock cycle. For example, in repeated WAIT mode entry or exit, if the RUN mode time is less than one low frequency reference clock cycle and if the WDW bit is set, the WDOG timer may never time out, even though the system is in RUN mode for a finite duration; WDOG may not see a low frequency reference clock edge during its wake time.

78.3.6 Debug mode

The WDOG timer can be configured for continual operation, or for suspension during debug mode. If the WDOG debug enable (WDBG) bit is set in the Watchdog Control Register (WCR), the WDOG timer operation is suspended in debug mode. If the WDBG bit is set and the debug mode (*ipg_debug* signal assertion) is entered, WDOG timer operation is suspended after two low frequency reference (*ipg_clk_32k*) clocks. Similarly, WDOG timer operation continues after two low frequency reference clocks of debug mode exit. Register read and write accesses in debug mode continue to function normally. Also, while in debug mode, the WDE bit of Watchdog Control Register (WCR) can be enabled/disabled directly. If the WDOG debug enable (WDBG) bit is cleared then WDOG timer operation is not suspended. The power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WCR) is set/ cleared while in debug mode, it remains set/cleared even after exiting debug mode.

78.3.7 Operations

78.3.7.1 Watchdog reset generation

The WDOG generated reset signal WDOG_RESET_B_DEB is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WCR).
- WDOG timeout. See Timeout event.

The wdog_rst_b will be asserted for one clock cycle of low frequency reference clock for both a timeout condition and a software write occurrence. It remains asserted for 1 clock cycle of low frequency reference clock even if a system reset is asserted in between. Figure 78-3 shows the timing diagram of this signal due to a timeout condition.

78.3.7.2 WDOG_B generation

The WDOG asserts WDOG_B in the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WCR). WDOG_B signal remains asserted as long as the WDA bit is "0".
- WDOG timeout condition, WDT bit of Watchdog Control Register (WCR) must be set for this scenario. A description of the timeout condition can be found in the Timeout event. WDOG_B (ipp_wdog) signal remains asserted until a power-on reset (POR) occurs. It gets cleared after the POR occurs (not due to any other system reset). Figure 78-4 shows the timing diagram of WDOG_B (ipp_wdog) due to timeout condition.
- WDOG power-down counter timeout, PDE bit of Watchdog Miscellaneous Control Register (WMCR) should not be cleared for this scenario. A description of this counter can be found in the Power-down counter event. WDOG_B (ipp_wdog) signal remains asserted for one clock cycle of low frequency reference clock (*ipg_clk_32k*).

Figure 78-2 shows the scenarios under which WDOG_B(ipp_wdog_b) gets asserted.

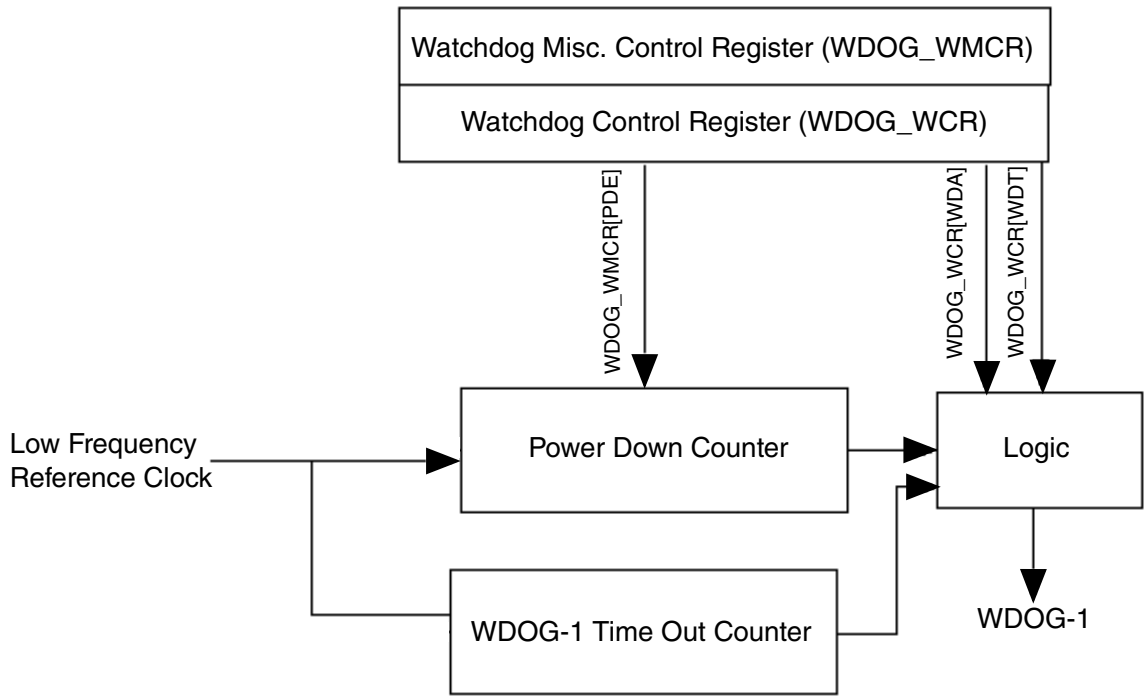


Figure 78-2. WDOG_B (ipp_wdog) generation

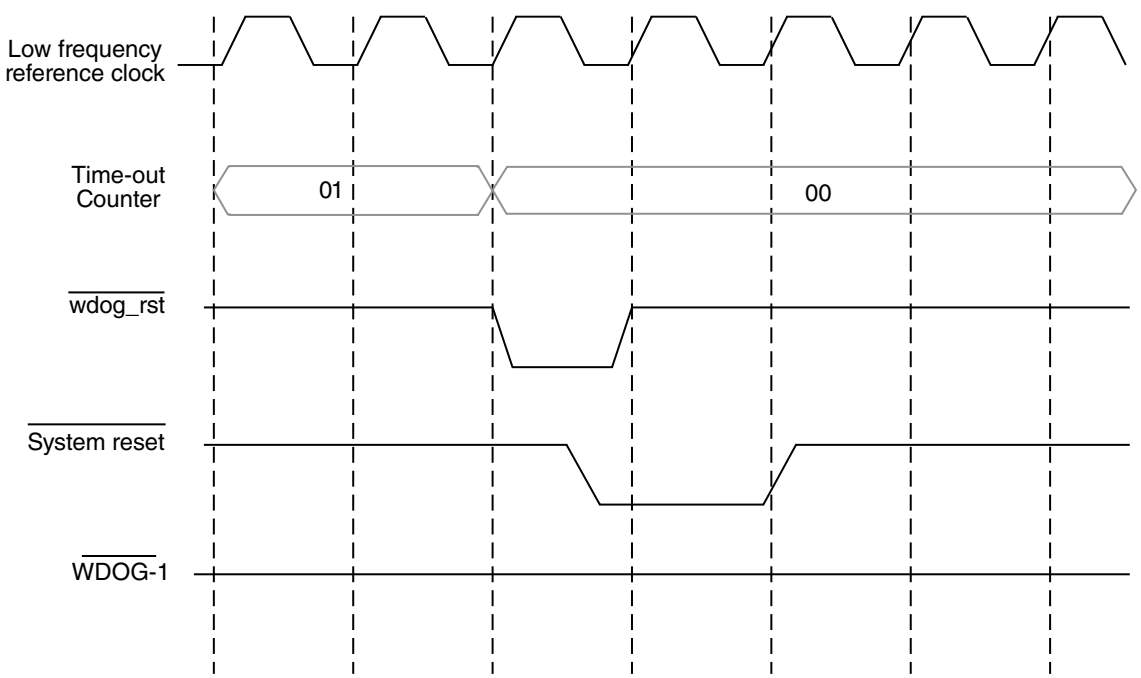


Figure 78-3. WDOG timeout condition/WDT bit is not set

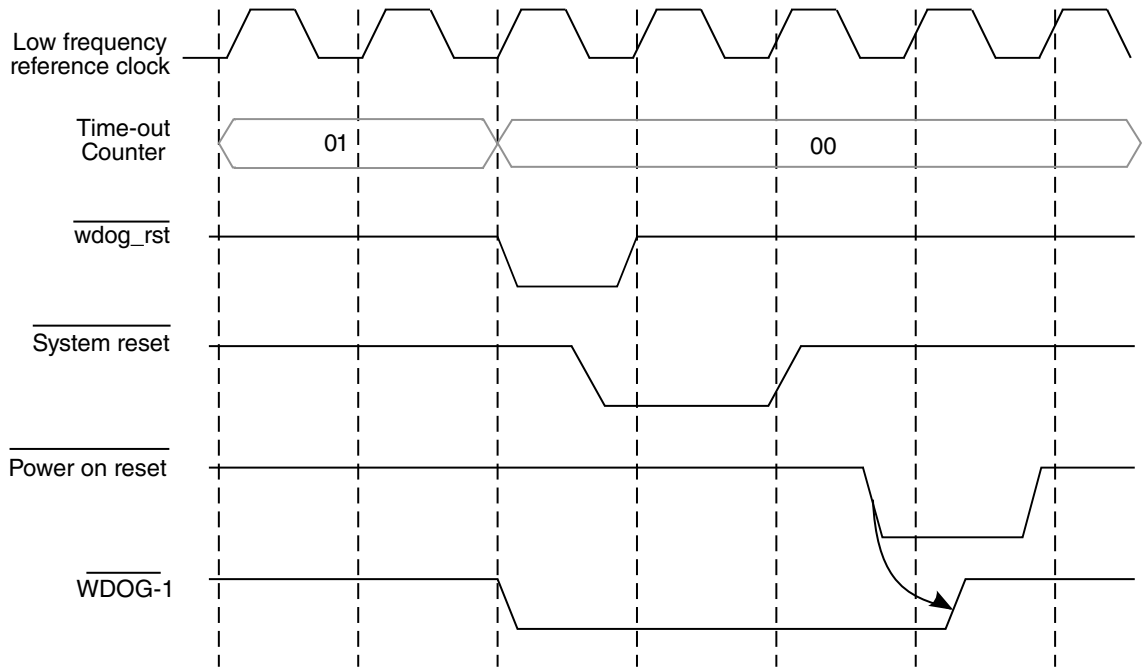


Figure 78-4. WDOG timeout condition/WDT bit is set

78.3.8 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG uses the low frequency reference clock for its counter and control operations. The peripheral bus clock is used for register read/write operations.

The following table describes the clock sources for WDOG. Please see for clock setting, configuration and gating information.

Table 78-2. WDOG Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	IP Global functional clock. All functionality inside the WDOG module is synchronized to this clock.
ipg_clk_s	ipg_clk_root	IP slave bus clock. This clock is synchronized to ipg_clk and is only used for register read/write operations.
ipg_clk_32k	ckil_sync_clk_root	Low frequency (32.768 kHz) clock that continues to run in low-power mode. It is assumed that the Clock Controller will provide this clock signal synchronized to ipg_clk in the normal mode, and switch to a non-synchronized signal in low-power mode when the ipg_clk is off.

78.3.9 Reset

The block is reset by a system reset and the WDOG counter will be disabled. The power-down counter is enabled and starts counting.

78.3.10 Interrupt

The WDOG has the feature of Interrupt generation before timeout.

The interrupt will be generated only if the WIE bit in Watchdog Interrupt Control Register (WICR) is set. The exact time at which the interrupt should occur (prior to timeout) depends on the value of WICT field of Watchdog Interrupt Control Register (WICR). For example, if the WICT field has a value 0x04, then the interrupt will be generated two seconds prior to timeout. Once the interrupt is triggered the WTIS bit in Watchdog Interrupt Control Register (WICR) will be set. The software needs to clear this bit to deassert the interrupt. If the WDOG is serviced before the interrupt generation then the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WCR) and interrupt would not be triggered.

78.3.11 Flow Diagrams

The flow diagrams for WDOG operation are shown below.

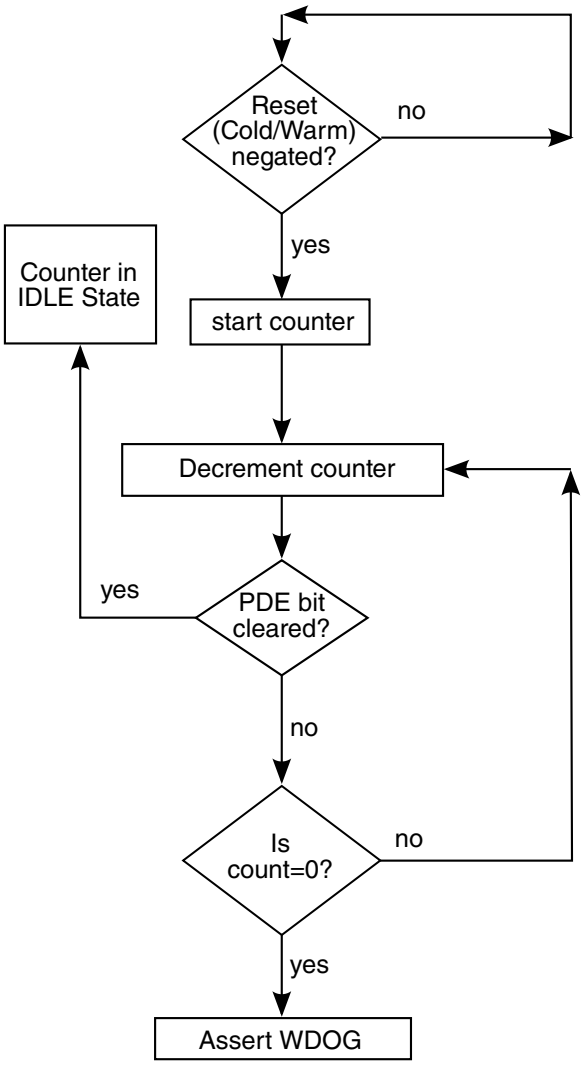


Figure 78-5. Power-Down Counter Flow Diagram

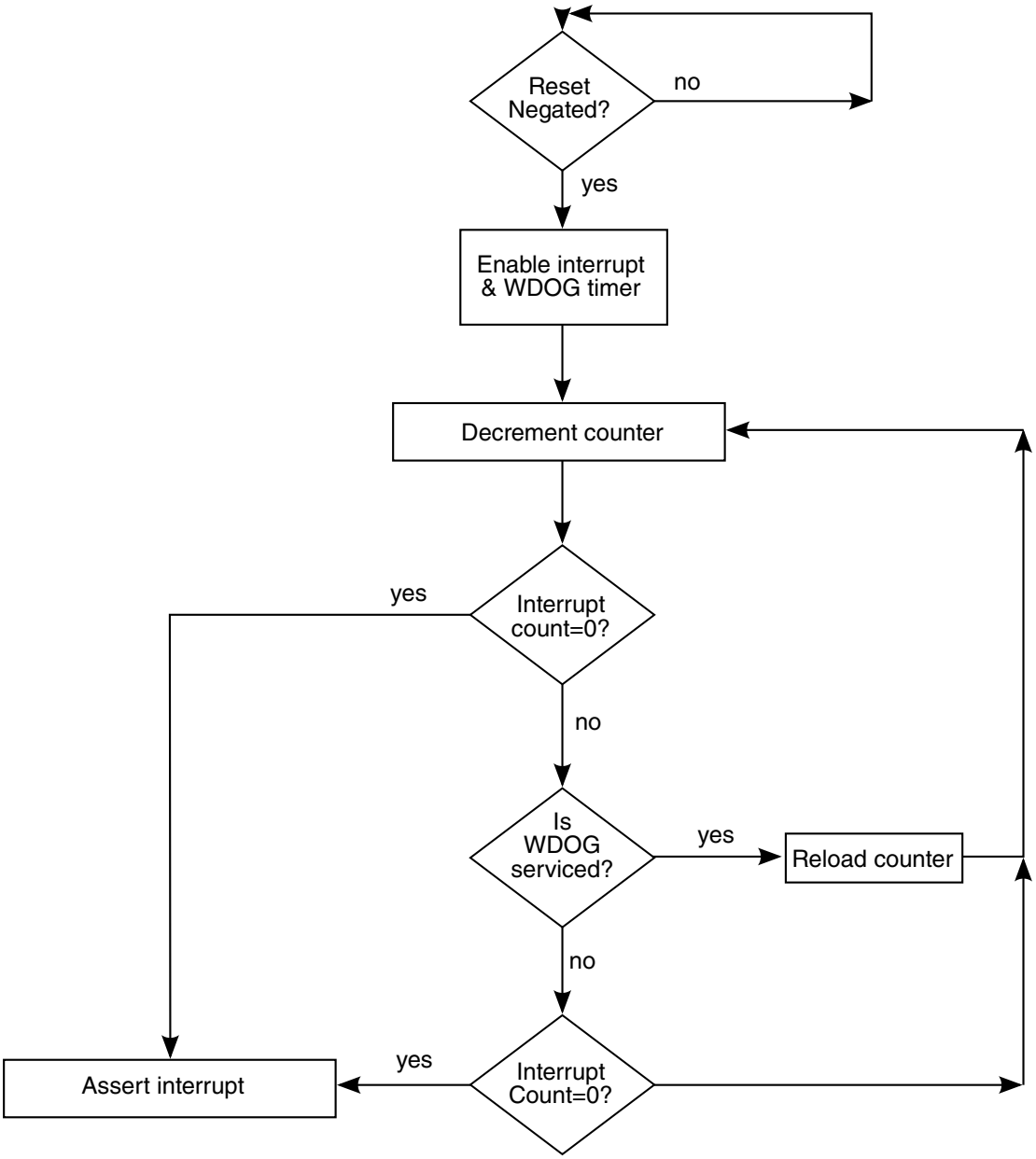


Figure 78-6. Interrupt Generation Flow Diagram

78.4 External signals

Table 78-3. WDOG External Signals

Signal	Description	Direction
WDOG_ANY	Global WDOG signal	IO
WDOG_B	This signal will power down the chip.	IO
WDOG_RESET_B_DEB	This signal is a reset source for the chip.	O

78.5 Initialization

The following sequence should be performed for WDOG initialization.

- PDE bit of Watchdog Miscellaneous Control Register (WMCR) should be cleared to disable the power down counter.
- WT field of Watchdog Control Register (WCR) should be programmed for sufficient timeout value.
- WDOG should be enabled by setting WDE bit of Watchdog Control Register (WCR) so that the timeout counter loads the WT field value of Watchdog Control Register (WCR) and starts counting.

78.6 WDOG Memory Map/Register Definition

78.6.1 WDOG register descriptions

The WDOG has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed,the WDOG will not generate a peripheral bus error but will behave normally, like a 16-Bit access, making read/write possible. A 32-Bit access should be avoided, as the system may go to an unknown state.

78.6.1.1 WDOG memory map

WDOG1 base address: 4003_0000h

WDOG2 base address: 4003_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Watchdog Control Register (WCR)	16	RW	0030
2	Watchdog Service Register (WSR)	16	RW	0000
4	Watchdog Reset Status Register (WRSR)	16	RO	0000
6	Watchdog Interrupt Control Register (WICR)	16	RW	0004
8	Watchdog Miscellaneous Control Register (WMCR)	16	RW	0001

78.6.1.2 Watchdog Control Register (WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG operation.

- WDZST, WDBG and WDW are write-once only bits. Once the software does a write access to these bits, they will be locked and cannot be reprogrammed until the next system reset assertion.
- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next system reset.
- WDT is also a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next POR. This bit does not get reset/cleared due to any system reset.

78.6.1.2.1 Offset

Register	Offset
WCR	0h

78.6.1.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WT								WDW	SRE	WDA	SRS	WDT	WDE	WDBG	WDZST
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

78.6.1.2.3 Fields

Field	Description
15-8 WT	WT Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG is enabled or after the service routine has been performed. For more information see Timeout event . 00000000 - - 0.5 Seconds (Default).

Table continues on the next page...

Field	Description
	00000001 - - 1.0 Seconds. 00000010 - - 1.5 Seconds. 00000011 - - 2.0 Seconds. 11111111 - - 128 Seconds.
7 WDW	WDW Watchdog Disable for Wait. This bit determines the operation of WDOG during Low Power WAIT mode. This is a write once only bit. 0 - Continue WDOG timer operation (Default). 1 - Suspend WDOG timer operation.
6 SRE	Software Reset Extension, an optional way to generate software reset adopt a new way to generate a more robust software reset. This bit can be set/cleared with IP bus and will be reset with power-on reset . 0 - using original way to generate software reset (default) 1 - using new way to generate software reset.
5 WDA	WDA WDOG_B assertion. Controls the software assertion of the WDOG_B signal. 0 - Assert WDOG_B output. 1 - No effect on system (Default).
4 SRS	SRS Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal WDOG_RESET_B_DEB . This bit automatically resets to "1" after it has been asserted to "0". **NOTE:** This bit does not generate the software reset to the block. 0 - Assert system reset signal. 1 - No effect on the system (Default).
3 WDT	WDT WDOG_B Time-out assertion. Determines if the WDOG_B gets asserted upon a Watchdog Time-out Event. This is a write-one once only bit. **NOTE:** There is no effect on WDOG_RESET_B_DEB (WDOG Reset) upon writing on this bit. WDOG_B gets asserted along with WDOG_RESET_B_DEB if this bit is set. 0 - No effect on WDOG_B (Default). 1 - Assert WDOG_B upon a Watchdog Time-out event.
2 WDE	WDE Watchdog Enable. Enables or disables the WDOG block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 - Disable the Watchdog (Default). 1 - Enable the Watchdog.
1 WDBG	WDBG Watchdog DEBUG Enable. Determines the operation of the WDOG during DEBUG mode. This bit is write once only. 0 - Continue WDOG timer operation (Default).

Table continues on the next page...

Field	Description
	1 - Suspend the watchdog timer.
0 WDZST	WDZST Watchdog Low Power. Determines the operation of the WDOG during low-power modes. This bit is write once-only. **NOTE:** The WDOG can continue/suspend the timer operation in the low-power modes (STOP and DOZE mode). 0 - Continue timer operation (Default). 1 - Suspend the watchdog timer.

78.6.1.3 Watchdog Service Register (WSR)

When enabled, the WDOG requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the timeout condition.

NOTE

Executing the service sequence will reload the WDOG timeout counter.

78.6.1.3.1 Offset

Register	Offset
WSR	2h

78.6.1.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

78.6.1.3.3 Fields

Field	Description
15-0 WSR	WSR

Field	Description
	Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0101010101010101 - Write to the Watchdog Service Register (WDOG_WSR). 1010101010101010 - Write to the Watchdog Service Register (WDOG_WSR).

78.6.1.4 Watchdog Reset Status Register (WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG. Read access to this register is with one wait state. Any write performed on this register will generate a Peripheral Bus Error .

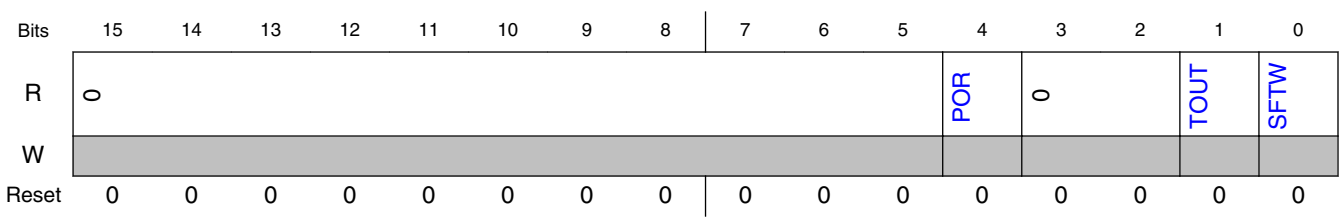
A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

78.6.1.4.1 Offset

Register	Offset
WRSR	4h

78.6.1.4.2 Diagram



78.6.1.4.3 Fields

Field	Description
15-5 —	Reserved.
4 POR	POR Power On Reset. Indicates whether the reset is the result of a power on reset. 0 - Reset is not the result of a power on reset. 1 - Reset is the result of a power on reset.
3-2 —	Reserved.
1 TOUT	TOUT Timeout. Indicates whether the reset is the result of a WDOG timeout. 0 - Reset is not the result of a WDOG timeout. 1 - Reset is the result of a WDOG timeout.
0 SFTW	SFTW Software Reset. Indicates whether the reset is the result of a WDOG software reset by asserting SRS bit 0 - Reset is not the result of a software reset. 1 - Reset is the result of a software reset.

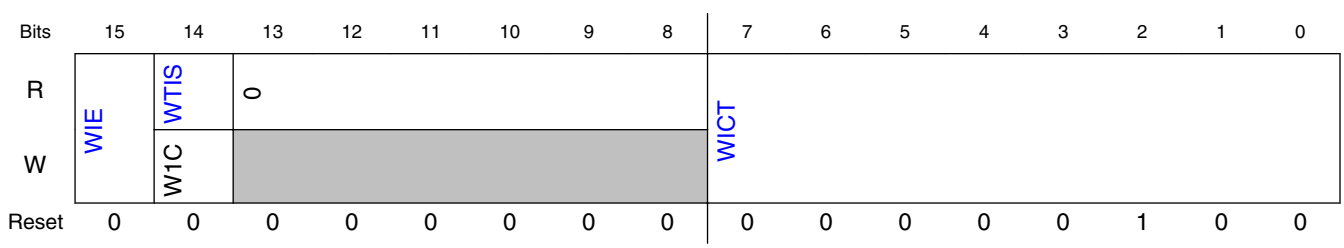
78.6.1.5 Watchdog Interrupt Control Register (WICR)

The WDOG_WICR controls the WDOG interrupt generation.

78.6.1.5.1 Offset

Register	Offset
WICR	6h

78.6.1.5.2 Diagram



78.6.1.5.3 Fields

Field	Description
15 WIE	WIE Watchdog Timer Interrupt enable bit. Reset value is 0. **NOTE:** This bit is a write once only bit. Once the software does a write access to this bit, it will get locked and cannot be reprogrammed until the next system reset assertion 0 - Disable Interrupt (Default). 1 - Enable Interrupt.
14 WTIS	WTIS Watchdog TImer Interrupt Status bit will reflect the timer interrupt status, whether interrupt has occurred or not. Once the interrupt has been triggered software must clear this bit by writing 1 to it. 0 - No interrupt has occurred (Default). 1 - Interrupt has occurred
13-8 —	Reserved.
7-0 WICT	WICT Watchdog Interrupt Count Time-out (WICT) field determines, how long before the counter time-out must the interrupt occur. The reset value is 0x04 implies interrupt will occur 2 seconds before time-out. The maximum value that can be programmed to WICT field is 127.5 seconds with a resolution of 0.5 seconds. **NOTE:** This field is write once only. Once the software does a write access to this field, it will get locked and cannot be reprogrammed until the next system reset assertion. 00000000 - WICT[7:0] = Time duration between interrupt and time-out is 0 seconds. 00000001 - WICT[7:0] = Time duration between interrupt and time-out is 0.5 seconds. 00000100 - WICT[7:0] = Time duration between interrupt and time-out is 2 seconds (Default). 11111111 - WICT[7:0] = Time duration between interrupt and time-out is 127.5 seconds.

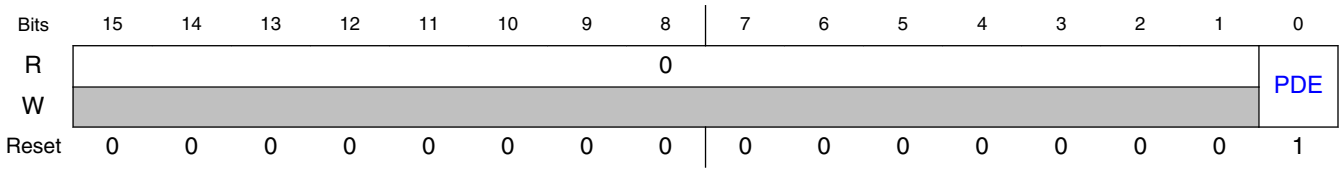
78.6.1.6 Watchdog Miscellaneous Control Register (WMCR)

WDOG_WMCR Controls the Power Down counter operation.

78.6.1.6.1 Offset

Register	Offset
WMCR	8h

78.6.1.6.2 Diagram



78.6.1.6.3 Fields

Field	Description
15-1 —	Reserved.
0 PDE	PDE Power Down Enable bit. Reset value of this bit is 1, which means the power down counter inside the WDOG is enabled after reset. The software must write 0 to this bit to disable the counter within 16 seconds of reset de-assertion. Once disabled this counter cannot be enabled again. See Power-down counter event for operation of this counter. **NOTE:** This bit is write-one once only bit. Once software sets this bit it cannot be reset until the next system reset. 0 - Power Down Counter of WDOG is disabled. 1 - Power Down Counter of WDOG is enabled (Default).

Chapter 79
CPU Watchdog Timer (WDOG 3,4)

79.1 Chip-specific CPU Watchdog information

Table 79-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

CPU WDOG3 services the CPU reset for the CM7.

CPU WDOG4 services the CPU reset for the CM4.

NOTE

- WDOG 1 and 2 are controlled by the System WDOG. Please see that chapter for more details.
- On this device, both LPO clock and INTCLK are 32 KHz clock generated by 32 KHz XTAL, which could be automatically switched to 32 KHz RCOSC upon XTAL clock loss. ERCLK is 1 MHz RCOSC clock generated by ANATOP.

79.2 Overview

The Watchdog Timer (WDOG) module is an independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the MCU.

79.2.1 Block diagram

The following figure shows a block diagram of the WDOG module.

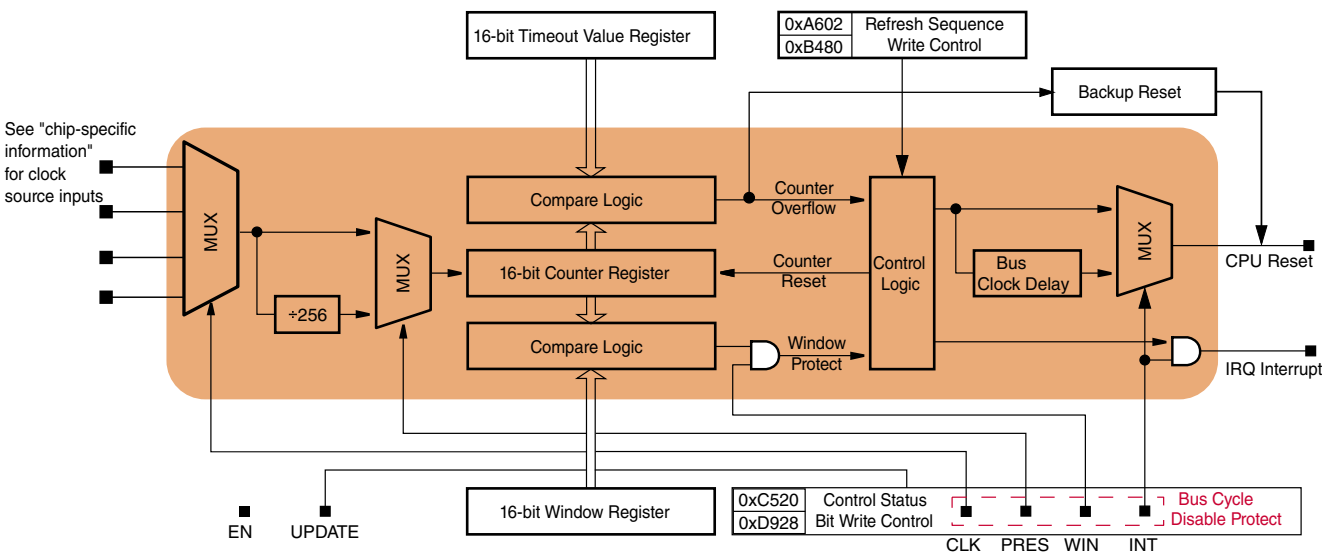


Figure 79-1. WDOG block diagram

79.2.2 Features

Features of the WDOG module include:

- Configurable clock source inputs independent from the bus clock
- Programmable timeout period
  - Programmable 16-bit timeout value
  - Optional fixed 256 clock prescaler when longer timeout periods are needed
- Robust write sequence for counter refresh

    - Refresh sequence of writing 0xA602 and then 0xB480
- Window mode option for the refresh mechanism
    - Programmable 16-bit window value
    - Provides robust check that program flow is faster than expected
    - Early refresh attempts trigger a reset.
- Optional timeout interrupt to allow post-processing diagnostics
    - Interrupt request to CPU with interrupt vector for an interrupt service routine (ISR)
    - Forced reset occurs 255 bus clocks after the interrupt vector fetch.
- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered.
- Robust write sequence for unlocking write-once configuration bits
    - Unlock sequence of writing 0xC520 and then 0xD928 for allowing updates to write-once configuration bits
    - Software must make updates within 255 bus clocks after unlocking and before WDOG closing unlock window.

79.3 Functional description

The WDOG module provides a fail safe mechanism to ensure the system can be reset to a known state of operation in case of system failure, such as the CPU clock stopping or there being a run away condition in the software code. The watchdog counter runs continuously off a selectable clock source and expects to be serviced (refreshed) periodically. If it is not, it generates a reset triggering event.

79.3.1 Watchdog refresh mechanism

The watchdog resets the MCU if the watchdog counter is not refreshed. A robust refresh mechanism makes it very unlikely that the watchdog can be refreshed by runaway code.

To refresh the watchdog counter, software must execute a refresh write sequence before the timeout period expires. In addition, if window mode is used, software must not start the refresh sequence until after the time value set in the WIN register. See the following figure.

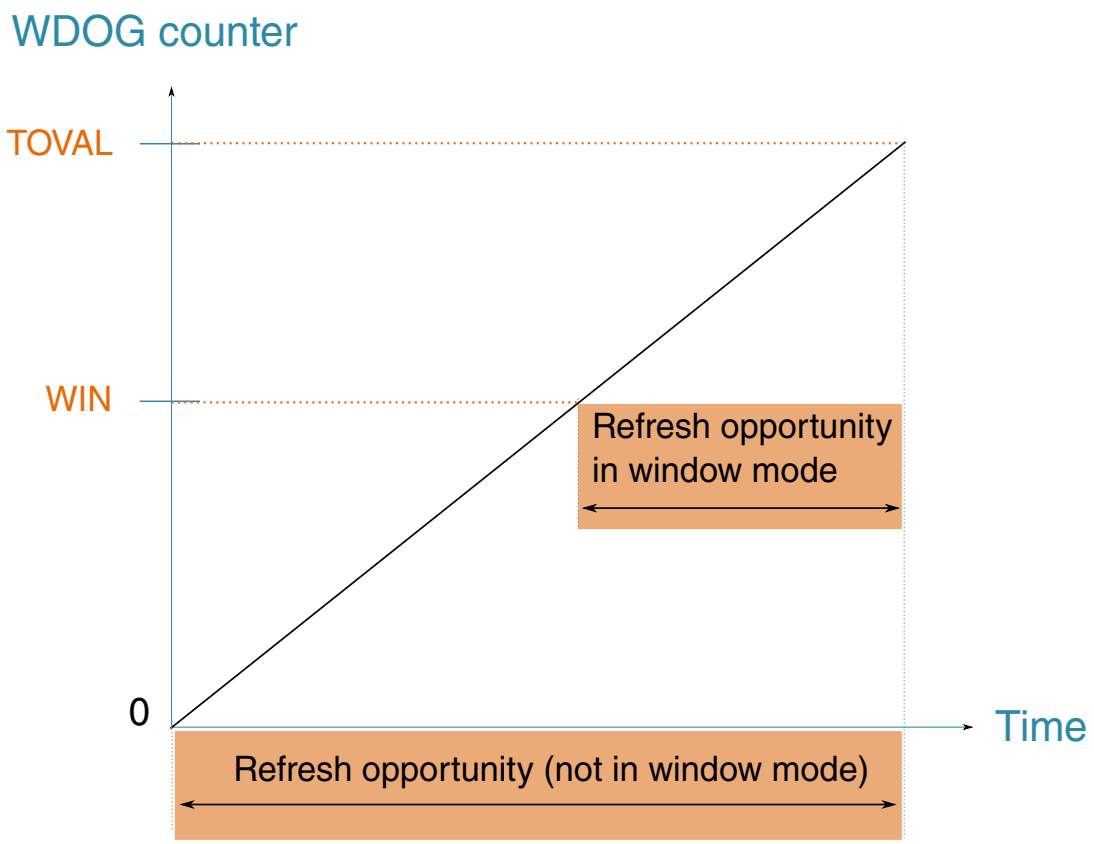


Figure 79-2. Refresh opportunity for the Watchdog counter

79.3.1.1 Window mode

Software finishing its main control loop faster than expected could be an indication of a problem. Depending on the requirements of the application, the WDOG can be programmed to force a reset when refresh attempts are early.

When Window mode is enabled, the watchdog must be refreshed after the counter has reached a minimum expected time value; otherwise, the watchdog resets the MCU. The minimum expected time value is specified in the WIN register. Setting CS[WIN] enables Window mode.

79.3.1.2 Refreshing the Watchdog

The refresh write sequence can be

- either two 16-bit writes (0xA602, 0xB480) or four 8-bit writes (0xA6, 0x02, 0xB4, 0x80) if WDOG_CS[CMD32EN] is 0;
- one 32-bit write (0xB480_A602) if WDOG_CS[CMD32EN] is 1.

to the CNT register. Both methods must occur before the WDG timeout; otherwise, the watchdog resets the MCU.

Note

Before starting the refresh sequence, disable the global interrupts. Otherwise, an interrupt could effectively invalidate the refresh sequence, if the interrupt occurs before the refresh writes finish. After the sequence finishes, restore the global interrupt control state.

The example codes can be found at the end of this chapter.

79.3.2 Configuring the Watchdog

79.3.2.1 Configuring the Watchdog Once

All watchdog control bits, timeout value, and window value are write-once after reset . This means that after a write has occurred they cannot be changed unless a reset occurs. This is guaranteed by the user configuring the window and timeout value first, followed by the other control bits, and ensuring that CS[UPDATE] is also set to 0.

This provides a robust mechanism to configure the watchdog and ensure that a runaway condition cannot mistakenly disable or modify the watchdog configuration after configured.

The new configuration takes effect only after all registers except CNT are written after reset. Otherwise, the WDOG uses the reset values by default. If window mode is not used (CS[WIN] is 0), writing to WIN is not required to make the new configuration take effect.

79.3.2.2 Reconfiguring the Watchdog

In some cases (like when supporting a bootloader function), you may want to reconfigure or disable the watchdog, *without forcing a reset first*.

- By setting CS[UPDATE] to 1 on the initial configuration of the watchdog after a reset, you can reconfigure the watchdog at any time by executing an unlock sequence.
- Conversely, if CS[UPDATE] remains 0, the only way to reconfigure the watchdog is by initiating a reset.

The unlock sequence is similar to the refresh sequence but uses different values.

79.3.2.2.1 Unlocking the Watchdog

The unlock sequence is a write to the CNT register of 0xC520 followed by 0xD928 within 31 bus clocks at any time after the watchdog has been configured. On completing the unlock sequence, the user must reconfigure the watchdog within 255 bus clocks; otherwise, the watchdog closes the unlock window.

NOTE

Due to the 255 bus clocks requirement for reconfiguring the watchdog, some delays must be inserted before executing STOP or WAIT instructions after reconfiguring the watchdog. This ensures that the watchdog's new configuration takes effect before the MCU enters low power mode. Otherwise, the MCU may not be waken up from low power mode.

The example codes can be found at end of this chapter.

79.3.3 Functionality in debug and low-power modes

By default, the watchdog is not functional in Debug mode, Wait mode, or Stop mode. However, the watchdog can remain functional in these modes as follows:

- **For Debug mode,** set CS[DBG]. (This way the watchdog is functional in Debug mode even when the CPU is held by the Debug module.)
- **For Wait mode,** set CS[WAIT].
- **For Stop mode,** set CS[STOP], CS[WAIT], and ensure the clock source is active in Stop mode.

NOTE

The watchdog can generate interrupt in Stop mode.

For Debug mode and Stop mode, in addition to the above configurations, a clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the watchdog cannot function.

79.3.4 Fast testing of the watchdog

Before executing application code in safety critical applications, users are required to test that the watchdog works as expected and resets the MCU. Testing every bit of a 16-bit counter by letting it run to the overflow value takes a relatively long time (64 k clocks).

To help minimize the startup delay for application code after reset, the watchdog has a feature to test the watchdog more quickly by splitting the counter into its constituent byte-wide stages. The low and high bytes are run independently and tested for timeout against the corresponding byte of the timeout value register. (For complete coverage when testing the high byte of the counter, the test feature feeds the input clock via the 8th bit of the low byte, thus ensuring that the overflow connection from the low byte to the high byte is tested.)

Using this test feature reduces the test time to 512 clocks (not including overhead, such as user configuration and reset vector fetches). To further speed testing, use a faster clock (such as the bus clock) for the counter reference.

On a power-on reset, the POR bit in the system reset register is set, indicating the user should perform the WDOG fast test.

79.3.4.1 Testing each byte of the counter

The test procedure follows these steps:

1. Program the preferred watchdog timeout value in the TOVAL register during the watchdog configuration time period.
2. Select a byte of the counter to test using the CS[TST] = 10b for the low byte; CS[TST] = 11b for the high byte.
3. Wait for the watchdog to timeout. Optionally, in the idle loop, increment RAM locations as a parallel software counter for later comparison. Because the RAM is not affected by a watchdog reset, the timeout period of the watchdog counter can be compared with the software counter to verify the timeout period has occurred as expected.
4. The watchdog counter times out and forces a reset.
5. Confirm the WDOG flag in the system reset register is set, indicating that the watchdog caused the reset. (The POR flag remains clear.)
6. Confirm that CS[TST] shows a test (10b or 11b) was performed.

If confirmed, the count and compare functions work for the selected byte. Repeat the procedure, selecting the other byte in step 2.

NOTE

CS[TST] is cleared by a POR only and not affected by other resets.

79.3.4.2 Entering user mode

After successfully testing the low and high bytes of the watchdog counter, the user can configure CS[TST] to 01b to indicate the watchdog is ready for use in application user mode. Thus if a reset occurs again, software can recognize the reset trigger as a real watchdog reset caused by runaway or faulty application code.

As an ongoing test when using the default clock source, software can periodically read the CNT register to ensure the counter is being incremented.

79.3.5 Clock source

The watchdog counter has the clock source options selected by programming CS[CLK]. See the "chip-specific information" section for available clock inputs and the default option for this device.

The options allow software to select a clock source independent of the bus clock for applications that need to meet more robust safety requirements. Using a clock source other than the bus clock ensures that the watchdog counter continues to run if the bus clock is somehow halted; see Backup reset.

NOTE

The *default* clock source of WDOG should be enabled after its functional reset is deasserted, for the WDOG module to function properly.

An optional fixed prescaler for all clock sources allows for longer timeout periods. When CS[PRES] is set, the clock source is prescaled by 256 before clocking the watchdog counter.

The following table summarizes the different watchdog timeout periods which could be available, as an example.

Table 79-2. Watchdog timeout availability

Reference clock	Prescaler	Watchdog time-out availability
REF_CLK	Pass through	1×RCP to 65535×RCP [1]
	Enable	256×RCP to 16776960×RCP

1. RCP means Reference Clock Period.

NOTE

When the programmer switches clock sources during reconfiguration, the watchdog hardware holds the counter at zero for 2.5 periods of the previous clock source and 2.5 periods of the new clock source after the configuration time period (255 bus clocks) ends. This delay ensures a smooth transition before restarting the counter with the new configuration.

79.3.6 Backup reset

NOTE

A clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the backup reset function is not available.

The backup reset function is a safeguard feature that independently generates a reset in case the main WDOG logic loses its clock (the bus clock) and can no longer monitor the counter. If the watchdog counter overflows twice in succession (without an intervening reset), the backup reset function takes effect and generates a reset.

Backup reset becomes valid when interrupt is enabled and the watchdog clock is from non bus clock. If interrupt is enabled, once the bus clock is cut off before exiting interrupt routine, the normal watchdog reset will be blocked. Under this case, the second overflow will cause backup reset directly.

79.3.7 Using interrupts to delay resets

- **When interrupts are enabled (CS[INT] = 1):** After a reset-triggering event (like a counter timeout or invalid refresh attempt), the watchdog first generates an interrupt request. Next, the watchdog delays 255 bus clocks (from the interrupt vector fetch, not the reset-triggering event) before forcing a reset, to allow the interrupt service routine (ISR) to perform tasks (like analyzing the stack to debug code).
- **When interrupts are disabled (CS[INT] = 0):** the watchdog does not delay before forcing a reset.

79.4 External signals

There are no external signals pinned out for this block.

79.5 Application Information

The watchdog is enabled by default after reset. To disable or reconfigure the watchdog, it is better to be done before the first watchdog timeout. It is suggested to disable or reconfigure the watchdog at the very beginning of the software code, e.g. beginning of the startup or main function.

NOTE

When the watchdog is configured by user, it needs at least 2.5 periods of watchdog clock to take effect. This means interval between two configures by user must be larger than 2.5 clocks.

NOTE

When Chip startup from BOOT ROM then jump to flash, the watchdog would be disabled in the beginning of bootloader, and enabled when bootloader exits. If there is any code in the flash program want to reconfigure the watchdog, it must be run 2.5 watchdog clocks later after the bootloader exits.

To disable or reconfigure the watchdog without forcing a reset, WDOG_CS[UPDATE] bit must be set during the initial configuration of the WDOG module. Then, the unlock sequence can be used at any time within the timeout limit to reconfigure the watchdog.

79.5.1 Disable Watchdog

To disable the watchdog, first do unlock sequence, then unset the WDOG_CS[EN] bit. The code snippet below shows an example for 32-bit write.

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
WDOG_CS &= ~WDOG_CS_EN_MASK; //disable watchdog
EnableInterrupts; //enable global interrupt
```

79.5.2 Disable Watchdog after Reset

All of watchdog registers are unlocked by reset. Therefore, unlock sequence is unnecessary, but it needs to write all of watchdog registers to make the new configuration take effect. The code snippet below shows an example of disabling watchdog after reset.

```
DisableInterrupts; // disable global interrupt
WDOG_CS &= ~WDOG_CS_EN_MASK; // disable watchdog
WDOG_TOVAL= 0xFFFF;
while(WDOG_CS[ULK]);  // waiting for lock
while(~WDOG_CS[RCS]); // waiting for new configuration to take effect
EnableInterrupts; // enable global interrupt
```

79.5.3 Configure Watchdog

The watchdog can be configured once by setting the WDOG_CS[UPDATE]=0. After that, the watchdog cannot be reconfigured until a reset. If set WDOG_CS[UPDATE]=1 when configuring the watchdog, the watchdog can be reconfigured without forcing a reset. The following example code shows how to configure the watchdog without window mode, clock source as LPO, interrupt enabled and timeout value to 256 clocks. The code snippet below shows an example for 32-bit write.

Configure once

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
while(WDOG_CS[ULK]==0);  //wait until registers are unlocked
WDOG_TOVAL = 256; //set timeout value
WDOG_CS = WDOG_CS_EN(1) | WDOG_CS_CLK(1) | WDOG_CS_INT(1) |
          WDOG_CS_WIN(0) | WDOG_CS_UPDATE(0);
while(WDOG_CS[RCS]==0); //wait until new configuration takes effect
EnableInterrupts; //enable global interrupt
```

Configure for reconfigurable

```
DisableInterrupts; //disable global interrupt
WDOG_CNT = 0xD928C520; //unlock watchdog
while(WDOG_CS[ULK]==0);  //wait until registers are unlocked
WDOG_TOVAL = 256; //set timeout value
WDOG_CS = WDOG_CS_EN(1) | WDOG_CS_CLK(1) | WDOG_CS_INT(1) |
          WDOG_CS_WIN(0) | WDOG_CS_UPDATE(1);
while(WDOG_CS[RCS]==0); //wait until new configuration takes effect
EnableInterrupts; //enable global interrupt
```

79.5.4 Refreshing the Watchdog

To refresh the watchdog and reset the watchdog counter to zero, a refresh sequence is required. The code snippet below shows an example for 32-bit write.

```
DisableInterrupts; // disable global interrupt
WDOG_CNT = 0xB480A602; // refresh watchdog
EnableInterrupts; // enable global interrupt
```

79.6 Memory map and register definition

79.6.1 WDOG register descriptions

79.6.1.1 WDOG memory map

WDOG3 base address: 4003_8000h

WDOG4 base address: 40C1_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Watchdog Control and Status Register (CS)	32	RW	0000_2180
4	Watchdog Counter Register (CNT)	32	RW	0000_0000
8	Watchdog Timeout Value Register (TOVAL)	32	RW	0000_7D00
C	Watchdog Window Register (WIN)	32	RW	0000_0000

79.6.1.2 Watchdog Control and Status Register (CS)

This section describes the function of Watchdog Control and Status Register.

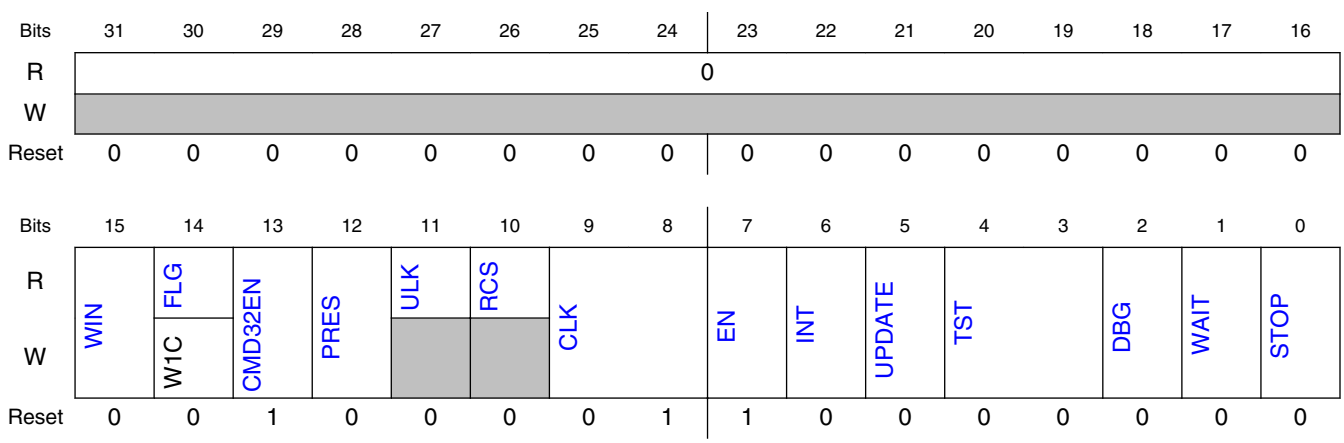
NOTE

TST is cleared (0:0) on POR only. Any other reset does not affect the value of this field.

79.6.1.2.1 Offset

Register	Offset
CS	0h

79.6.1.2.2 Diagram



79.6.1.2.3 Fields

Field	Description
31-16 —	Reserved
15 WIN	Watchdog Window This write-once bit enables window mode. See the Window mode section. 0 - Window mode disabled. 1 - Window mode enabled.
14 FLG	Watchdog Interrupt Flag This bit is an interrupt indicator when INT is set in control and status register 1. Write 1 to clear it. 0 - No interrupt occurred. 1 - An interrupt occurred.
13 CMD32EN	Enables or disables WDOG support for 32-bit (otherwise 16-bit or 8-bit) refresh/unlock command write words This is write-once field, and the user needs to unlock WDOG after writing this field for reconfiguration. 0 - Disables support for 32-bit refresh/unlock command write words. Only 16-bit or 8-bit is supported. 1 - Enables support for 32-bit refresh/unlock command write words. 16-bit or 8-bit is NOT supported.
12 PRES	Watchdog prescaler This write-once bit enables a fixed 256 pre-scaling of watchdog counter reference clock. (The block diagram shows this clock divider option.) 0 - 256 prescaler disabled. 1 - 256 prescaler enabled.
11 ULK	Unlock status This read-only bit indicates whether WDOG is unlocked or not. 0 - WDOG is locked. 1 - WDOG is unlocked.
10 RCS	Reconfiguration Success This read-only bit indicates whether the reconfiguration is successful or not. Default reset value is 0.This bit is set when new configuration takes effect, and is cleared by successful unlock command. 0 - Reconfiguring WDOG. 1 - Reconfiguration is successful.
9-8 CLK	Watchdog Clock This write-once field indicates the clock source that feeds the watchdog counter. See the "chip-specific information" section.
7 EN	Watchdog Enable This write-once bit enables the watchdog counter to start counting. 0 - Watchdog disabled. 1 - Watchdog enabled.
6 INT	Watchdog Interrupt

Table continues on the next page...

Field	Description
	This write-once bit configures the watchdog to immediately generate an interrupt request upon a reset-triggering event (timeout or illegal write to the watchdog), before forcing a reset. After the interrupt vector fetch (which comes after the reset-triggering event), the reset occurs after a delay of 255 bus clocks. 0 - Watchdog interrupts are disabled. Watchdog resets are not delayed. 1 - Watchdog interrupts are enabled. Watchdog resets are delayed by 255 bus clocks from the interrupt vector fetch.
5 UPDATE	Allow updates This write-once bit allows software to reconfigure the watchdog without a reset. 0 - Updates not allowed. After the initial configuration, the watchdog cannot be later modified without forcing a reset. 1 - Updates allowed. Software can modify the watchdog configuration registers within 255 bus clocks after performing the unlock write sequence.
4-3 TST	Watchdog Test Enables the fast test mode. The test mode allows software to exercise all bits of the counter to demonstrate that the watchdog is functioning properly. See the Fast testing of the watchdog section. This write-once field is cleared (0:0) on POR only. Any other reset does not affect the value of this field. 00 - Watchdog test mode disabled. 01 - Watchdog user mode enabled. (Watchdog test mode disabled.) After testing the watchdog, software should use this setting to indicate that the watchdog is functioning normally in user mode. 10 - Watchdog test mode enabled, only the low byte is used. CNT[CNTLOW] is compared with TOVAL[TOVALLOW]. 11 - Watchdog test mode enabled, only the high byte is used. CNT[CNTHIGH] is compared with TOVAL[TOVALHIGH].
2 DBG	Debug Enable This write-once bit enables the watchdog to operate when the chip is in debug mode. 0 - Watchdog disabled in chip debug mode. 1 - Watchdog enabled in chip debug mode.
1 WAIT	Wait Enable This write-once bit enables the watchdog to operate when the chip is in wait mode. 0 - Watchdog disabled in chip wait mode. 1 - Watchdog enabled in chip wait mode.
0 STOP	Stop Enable This write-once bit enables the watchdog to operate when the chip is in stop mode. 0 - Watchdog disabled in chip stop mode. 1 - Watchdog enabled in chip stop mode.

79.6.1.3 Watchdog Counter Register (CNT)

This section describes the watchdog counter register.

The watchdog counter register provides access to the value of the free-running watchdog counter. Software can read the counter register at any time.

Software cannot write directly to the watchdog counter; however, two write sequences to these registers have special functions:

1. The *refresh sequence* resets the watchdog counter to 0x0000. See the "Refreshing the Watchdog" section.
2. The *unlock sequence* allows the watchdog to be reconfigured without forcing a reset (when CS[UPDATE] = 1). See the "Configure for reconfigurable" section.

NOTE

All other writes to this register are illegal and force a reset.

79.6.1.3.1 Offset

Register	Offset
CNT	4h

79.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNTHIGH								CNTLOW							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

79.6.1.3.3 Fields

Field	Description
31-16 —	Reserved
15-8 CNTHIGH	High byte of the Watchdog Counter
7-0 CNTLOW	Low byte of the Watchdog Counter

79.6.1.4 Watchdog Timeout Value Register (TOVAL)

This section describes the watchdog timeout value register. TOVAL contains the 16-bit value used to set the timeout period of the watchdog.

The watchdog counter (CNT) is continuously compared with the timeout value (TOVAL). If the counter reaches the timeout value, the watchdog forces a reset triggering event.

NOTE

Do not write 0 to the TOVAL register (if CS[TST]=11b, then TOVALHIGH cannot be written as 0; if CS[TST]=10b, then TOVALLOW cannot be 0); otherwise, the watchdog always generates a reset.

79.6.1.4.1 Offset

Register	Offset
TOVAL	8h

79.6.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TOVALHIGH								TOVALLOW							
W																
Reset	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0

79.6.1.4.3 Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Field	Description
—	
15-8 TOVALHIGH	High byte of the timeout value
7-0 TOVALLOW	Low byte of the timeout value

79.6.1.5 Watchdog Window Register (WIN)

This section describes the watchdog window register. When window mode is enabled (CS[WIN] is set), The WIN register determines the earliest time that a refresh sequence is considered valid. See the Watchdog refresh mechanism section.

The WIN register value must be less than the TOVAL register value.

79.6.1.5.1 Offset

Register	Offset
WIN	Ch

79.6.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WINHIGH								WINLOW							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

79.6.1.5.3 Fields

Field	Description
31-16	Reserved

Table continues on the next page...

Field	Description
—	
15-8 WINHIGH	High byte of Watchdog Window
7-0 WINLOW	Low byte of Watchdog Window

Chapter 80
External Watchdog Module (EWM)

80.1 Chip-specific EWM information

Table 80-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM Low Power Clock Source	Clock Management Clock Control Module (CCM) Low Power Clock Source (lpo_clk[3:0])
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

NOTE

- lpo_clk_0: 16MHz RC Oscillator
- lpo_clk_1: 1MHz RC Oscillator
- lpo_clk_2: 32KHz clock generated by 32 KHZ XTALOSC which could be automatically switched to 32KHz RC OSC upon XTAL clock loss
- lpo_clk_3: IP Bus Clock

80.2 Overview

For safety, a redundant watchdog system, External Watchdog Monitor (EWM), is designed to monitor external circuits, as well as the MCU software flow. This provides a back-up mechanism to the internal watchdog that resets the MCU's CPU and peripherals.

The watchdog is generally used to monitor the flow and execution of embedded software within an MCU. The watchdog consists of a counter that if allowed to overflow, forces an internal reset (asynchronous) to all on-chip peripherals and optionally asserts the RESET_B pin to reset external devices/circuits. The overflow of the watchdog counter must not occur if the software code works well and services the watchdog to re-start the actual counter.

The EWM differs from the internal watchdog in that it does not reset the MCU's CPU and peripherals. The EWM provides an independent EWM_OUT_b signal that when asserted resets or places an external circuit into a safe mode. The EWM_OUT_b signal is asserted upon the EWM counter time-out. An optional external input EWM_in is provided to allow additional control of the assertion of EWM_OUT_b signal.

80.2.1 Block Diagram

This figure shows the EWM block diagram.

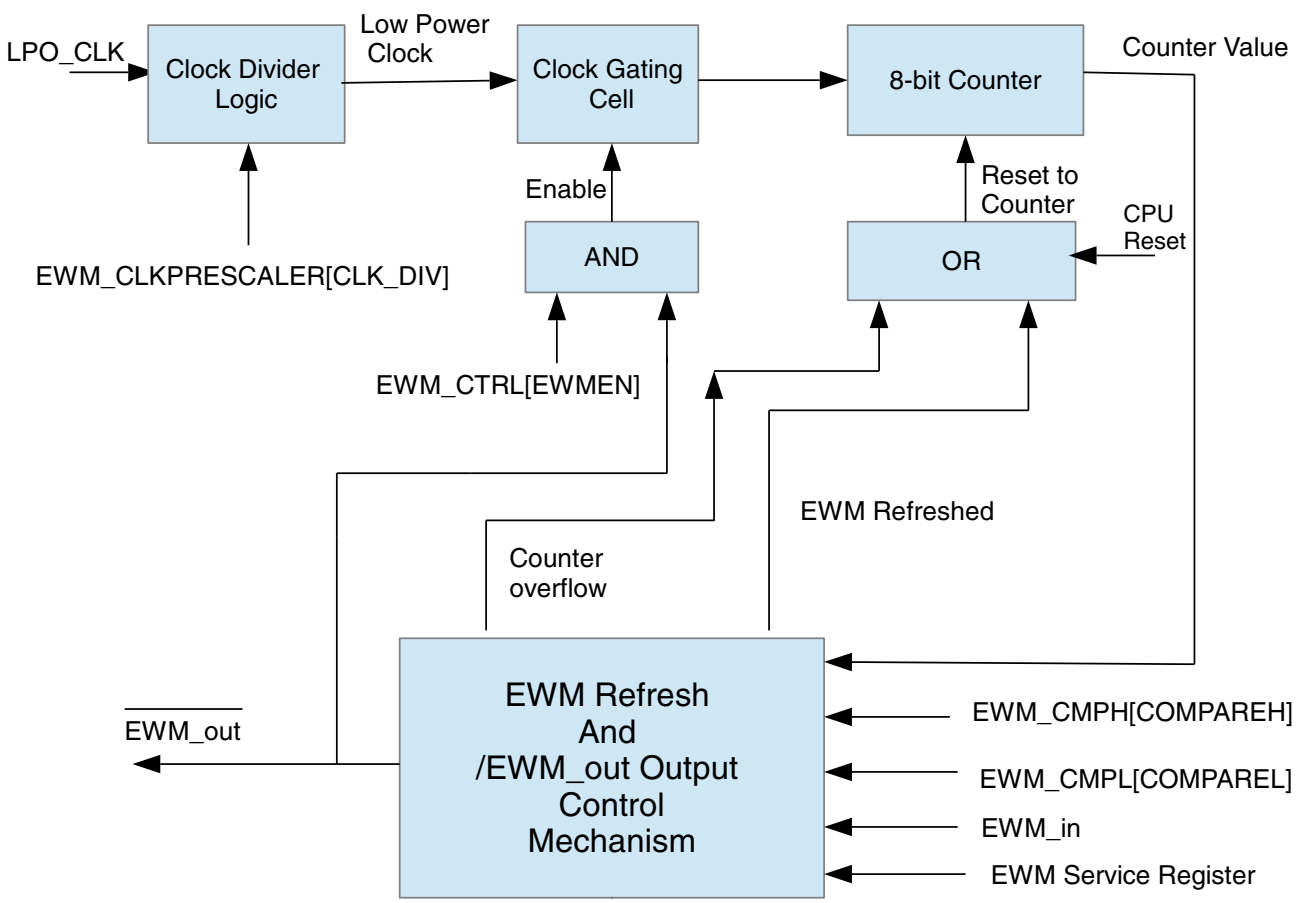


Figure 80-1. EWM Block Diagram

80.2.2 Features

Features of EWM module include:

- Independent LPO_CLK clock source
- Programmable time-out period specified in terms of number of EWM LPO_CLK clock cycles.
- Windowed refresh option
  - Provides robust check that program flow is faster than expected.
  - Programmable window.
  - Refresh outside window leads to assertion of EWM_OUT_b.
- Robust refresh mechanism
  - Write values of 0xB4 and 0x2C to EWM Refresh Register within 63 peripheral bus clock cycles.
- One output port, EWM_OUT_b, when asserted is used to reset or place the external circuit into safe mode.
- One Input port, EWM_in, allows an external circuit to control the assertion of the EWM_OUT_b signal.

80.3 Functional Description

The following sections describe functional details of the EWM module.

NOTE

When the BUS_CLK is lost, then EWM module doesn't generate the EWM_OUT_b signal and no refresh operation is possible

80.3.1 Modes of Operation

This section describes the module's operating modes.

80.3.1.1 Stop Mode

When the EWM is in stop mode, the CPU refreshes to the EWM cannot occur. On entry to stop mode, the EWM's counter freezes.

There are two possible ways to exit from Stop mode:

- On exit from stop mode through a reset, the EWM remains disabled.
- On exit from stop mode by an interrupt, the EWM is re-enabled, and the counter continues to be clocked from the same value prior to entry to stop mode.

Note the following if the EWM enters the stop mode during CPU refresh mechanism: At the exit from stop mode by an interrupt, refresh mechanism state machine starts from the previous state which means, if first refresh command is written correctly and EWM enters the stop mode immediately, the next command has to be written within the next *63* peripheral bus clocks after exiting from stop mode. User must mask all interrupts prior to executing EWM refresh instructions.

80.3.1.2 Wait Mode

The EWM module treats the stop and wait modes as the same. EWM functionality remains the same in both of these modes.

80.3.1.3 Debug Mode

Entry to debug mode has no effect on the EWM.

- If the EWM is enabled prior to entry of debug mode, it remains enabled.
- If the EWM is disabled prior to entry of debug mode, it remains disabled.

80.3.2 The EWM_OUT_b Signal

The EWM_OUT_b is a digital output signal used to gate an external circuit (application specific) that controls critical safety functions. For example, the EWM_OUT_b could be connected to the high voltage transistors circuits that control an AC motor in a large appliance.

The EWM_OUT_b signal remains deasserted when the EWM is being regularly refreshed by the CPU within the programmable refresh window, indicating that the application code is executed as expected.

The EWM_OUT_b signal is asserted in any of the following conditions:

- The EWM refresh occurs when the counter value is less than CMPL value.
- The EWM counter value reaches the CMPH value, and no EWM refresh has occurred.
- If functionality of EWM_in pin is enabled and EWM_in pin is asserted while refreshing the EWM.
- After any reset (by the virtue of the external pull-down mechanism on the EWM_OUT_b pin)

The EWM_OUT_b is asserted after any reset by the virtue of the external pull-down mechanism on the EWM_OUT_b signal. Then, to deassert the EWM_OUT_b signal, set EWMEN bit in the CTRL register to enable the EWM.

If the EWM_OUT_b signal shares its pad with a digital I/O pin, on reset this actual pad defers to being an input signal. The pad state is controlled by the EWM_OUT_b signal only after the EWM is enabled by the EWMEN bit in the CTRL register.

Note

EWM_OUT_b pad must be in pull down state when EWM functionality is used and when EWM is under Reset.

80.3.3 EWM_OUT_b pin state in low power modes

During Wait, Stop and Power Down modes the EWM_OUT_b pin enters a high-impedance state. A user has the option to control the logic state of the pin using an external pull device or by configuring the internal pull device. When the CPU enters a Run mode from Wait or Stop recovery, the pin resumes its previous state before entering Wait or Stop mode. When the CPU enters Run mode from Power Down, the pin returns to its reset state.

80.3.4 The EWM_in Signal

The EWM_in is a digital input signal for safety status of external safety circuits, that allows an external circuit to control the assertion of the EWM_OUT_b signal. For example, in the application, an external circuit monitors a critical safety function, and if there is fault with safety function, the external circuit can then actively initiate the EWM_OUT_b signal that controls the gating circuit.

The EWM_in signal is ignored if the EWM is disabled, or if INEN bit of CTRL register is cleared, as after any reset.

On enabling the EWM (setting the CTRL[EWMEN] bit) and enabling EWM_in functionality (setting the CTRL[INEN] bit), the EWM_in signal must be in the deasserted state prior to the CPU start refreshing the EWM. This ensures that the EWM_OUT_b stays in the deasserted state; otherwise, the EWM_OUT_b output signal is asserted.

Note

The user must update the CMPH and CMPL registers prior to enabling the EWM. After enabling the EWM, the counter resets to zero, therefore the user shall provide a reasonable time after a power-on reset for the external monitoring circuit to stabilize. The user shall also ensure that the EWM_in pin is deasserted.

80.3.5 EWM Counter

It is an 8-bit ripple counter fed from a clock source that is independent of the peripheral bus clock source. As the preferred time-out is between 1 ms and 100 ms the actual clock source should be in the kHz range.

The counter is reset to zero after the CPU reset, or when EWM refresh action completes, or at counter overflow. The counter value is not accessible to the CPU.

80.3.6 EWM Compare Registers

The compare registers CMPL and CMPH are write-once after a CPU reset and cannot be modified until another CPU reset occurs.

The EWM compare registers are used to create a refresh window to refresh the EWM module.

It is illegal to program CMPL and CMPH with same value. In this case, as soon as counter reaches (CMPL + 1), EWM_OUT_b is asserted.

80.3.7 EWM Refresh Mechanism

Other than the initial configuration of the EWM, the CPU can only access the EWM by the EWM Service Register. The CPU must access the EWM service register with correct write of unique data within the windowed time frame as determined by the CMPL and CMPH registers for correct EWM refresh operation. Therefore, three possible conditions can occur:

Table 80-2. EWM Refresh Mechanisms

Condition	Mechanism
An EWM refresh action completes when: CMPL < Counter < CMPH.	The software behaves as expected and the EWM counter is reset to zero. The EWM_OUT_b output signal remains in the deasserted state if, during the EWM refresh action, the EWM_in input has been in deasserted state.
An EWM refresh action completes when Counter < CMPL	The software refreshes the EWM before the windowed time frame, the counter is reset to zero and the EWM_OUT_b output signal is asserted irrespective of the input EWM_in signal.
Counter value reaches CMPH prior to completion of EWM refresh action.	Software has not refreshed the EWM. The EWM counter is reset to zero and the EWM_OUT_b output signal is asserted irrespective of the input EWM_in signal.

80.3.8 EWM Interrupt

When EWM_OUT_b is asserted, an interrupt request is generated to indicate the assertion of the EWM reset out signal. This interrupt is enabled when CTRL[INTEN] is set. Clearing this bit clears the interrupt request but does not affect EWM_OUT_b. The EWM_OUT_b signal can be deasserted only by forcing a system reset.

80.3.9 Selecting the EWM counter clock

There are four possible low power clock sources for the EWM counter. Select one of the available clock sources by programming CLKCTRL[CLKSEL].

80.3.10 Counter clock prescaler

The EWM counter clock source can be prescaled by a clock divider, by programming CLKPRESCALER[CLK_DIV]. This divided clock is used to run the EWM counter.

NOTE

The divided clock used to run the EWM counter must be no more than half the frequency of the bus clock.

80.4 EWM Signal Descriptions

The EWM has two external signals and internal options for the counter clock sources, as shown in the following table.

NOTE

All active-low signals are now represented with the suffix "_b" throughout the chapter.

Table 80-3. EWM Signal Descriptions

Signal	Description	I/O
EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active low.	I
EWM_OUT_b	EWM reset out signal	O
lpo_clk[3:0]	Low power clock sources for running counter	I

80.5 Memory Map/Register Definition

This section contains the module memory map and registers.

NOTE

EWM only supports 8-bit register access. 16-bit and 32-bit access are not possible.

80.5.1 EWM register descriptions

80.5.1.1 EWM memory map

EWM base address: 4002_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Control Register (CTRL)	8	RW	00
1	Service Register (SERV)	8	WORZ	00
2	Compare Low Register (CMPL)	8	RWONCE	00
3	Compare High Register (CMPH)	8	RWONCE	FF
4	Clock Control Register (CLKCTRL)	8	RWONCE	00
5	Clock Prescaler Register (CLKPRESCALER)	8	RWONCE	00

80.5.1.2 Control Register (CTRL)

The CTRL register is cleared by any reset.

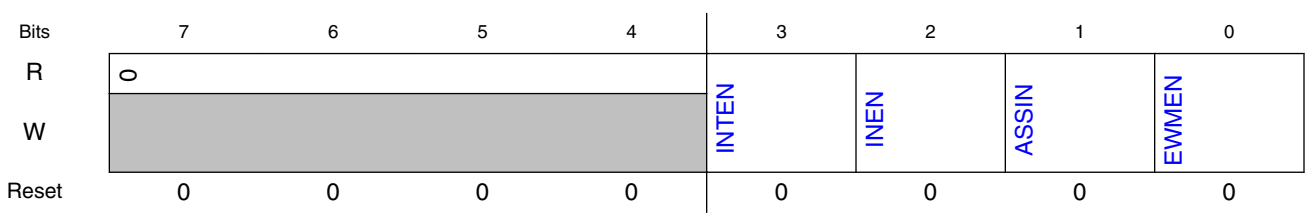
NOTE

INEN, ASSIN and EWMEN bits can be written once after a CPU reset. Modifying these bits more than once, generates a bus transfer error.

80.5.1.2.1 Offset

Register	Offset
CTRL	0h

80.5.1.2.2 Diagram



80.5.1.2.3 Fields

Field	Description
7-4	Reserved

Table continues on the next page...

Field	Description
—	
3 INTEN	Interrupt Enable. This bit when set and EWM_OUT_b is asserted, an interrupt request is generated. To de-assert interrupt request, user should clear this bit by writing 0. 0 - Deasserts the interrupt request. 1 - Generates an interrupt request, when EWM_OUT_b is asserted.
2 INEN	Input Enable. This bit when set, enables the EWM_in port. 0 - EWM_in port is disabled. 1 - EWM_in port is enabled.
1 ASSIN	EWM_in's Assertion State Select. Default assert state of the EWM_in signal is logic zero. Setting the ASSIN bit inverts the assert state of EWM_in signal to a logic one. 0 - Default assert state of the EWM_in signal. 1 - Inverts the assert state of EWM_in signal.
0 EWMEN	EWM enable. This bit when set, enables the EWM module. This resets the EWM counter to zero and deasserts the EWM_OUT_b signal. This bit when unset, keeps the EWM module disabled. It cannot be re-enabled until a next reset, due to the write-once nature of this bit. 0 - EWM module is disabled. 1 - EWM module is enabled.

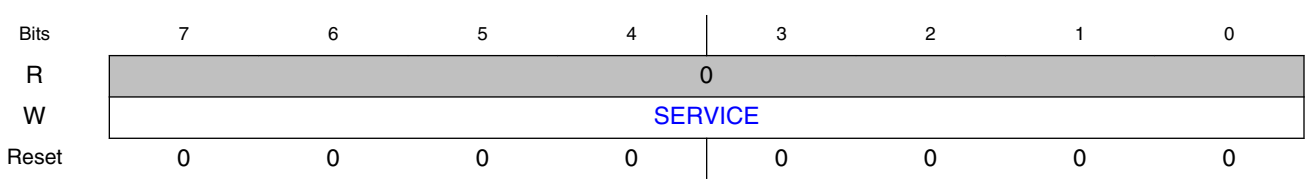
80.5.1.3 Service Register (SERV)

The SERV register provides the interface from the CPU to the EWM module. It is write-only and reads of this register return zero.

80.5.1.3.1 Offset

Register	Offset
SERV	1h

80.5.1.3.2 Diagram



80.5.1.3.3 Fields

Field	Description
7-0 SERVICE	SERVICE The EWM refresh mechanism requires the CPU to write two values to the SERV register: a first data byte of 0xB4, followed by a second data byte of 0x2C. The EWM refresh is invalid if either of the following conditions is true. • The first or second data byte is not written correctly. • The second data byte is not written within a fixed number of peripheral bus cycles of the first data byte. This fixed number of cycles is called *EWM_refresh_time* . 63 peripheral bus clock cycles are required for *EWM_refresh_time*

80.5.1.4 Compare Low Register (CMPL)

The CMPL register is reset to zero after a CPU reset. This provides no minimum time for the CPU to refresh the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

80.5.1.4.1 Offset

Register	Offset
CMPL	2h

80.5.1.4.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	COMPAREL							
Reset	0	0	0	0	0	0	0	0

80.5.1.4.3 Fields

Field	Description
7-0 COMPAREL	COMPAREL To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) minimum refresh time is required.

80.5.1.5 Compare High Register (CMPH)

The CMPH register is reset to 0xFF after a CPU reset. This provides a maximum of 256 clocks time, for the CPU to refresh the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

The valid values for CMPH are up to 0xFE because the EWM counter never expires when CMPH = 0xFF. The expiration happens only if EWM counter is greater than CMPH.

80.5.1.5.1 Offset

Register	Offset
CMPH	3h

80.5.1.5.2 Diagram

Bits	7	6	5	4	3	2	1	0
R W	COMPAREH							
Reset	1	1	1	1	1	1	1	1

80.5.1.5.3 Fields

Field	Description
7-0 COMPAREH	COMPAREH To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) maximum refresh time is required.

80.5.1.6 Clock Control Register (CLKCTRL)

This CLKCTRL register is reset to 0x00 after a CPU reset.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

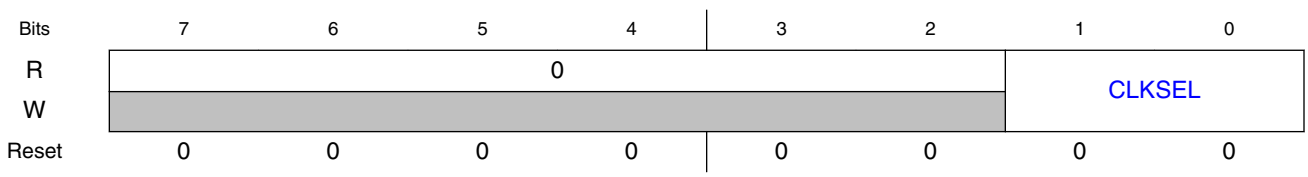
NOTE

User should select the required low power clock before enabling the EWM.

80.5.1.6.1 Offset

Register	Offset
CLKCTRL	4h

80.5.1.6.2 Diagram



80.5.1.6.3 Fields

Field	Description
7-2 —	reserved

Table continues on the next page...

Field	Description
1-0 CLKSEL	CLKSEL EWM has 4 possible low power clock sources for running EWM counter. One of the clock source can be selected by writing into this field. • 00 - lpo_clk[0] will be selected for running EWM counter. • 01 - lpo_clk[1] will be selected for running EWM counter. • 10 - lpo_clk[2] will be selected for running EWM counter. • 11 - lpo_clk[3] will be selected for running EWM counter.

80.5.1.7 Clock Prescaler Register (CLKPRESCALER)

This CLKPRESCALER register is reset to 0x00 after a CPU reset.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

Write the required prescaler value before enabling the EWM.

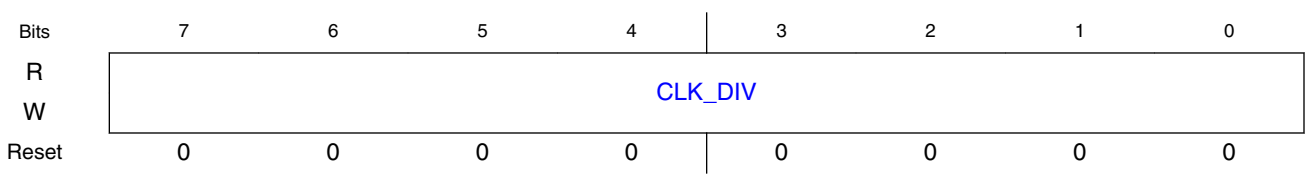
NOTE

The implementation of this register is chip-specific. See the Chip Configuration details.

80.5.1.7.1 Offset

Register	Offset
CLKPRESCALER	5h

80.5.1.7.2 Diagram



80.5.1.7.3 Fields

Field	Description
7-0 CLK_DIV	CLK_DIV Selected low power clock source for running the EWM counter can be prescaled as below. • Prescaled clock frequency = low power clock source frequency / (1 + CLK_DIV)

Chapter 81
On Chip Cross Triggers Overview

81.1 Overview

This chip integrates an on-chip cross trigger network. The following diagram shows the cross trigger network of this device.

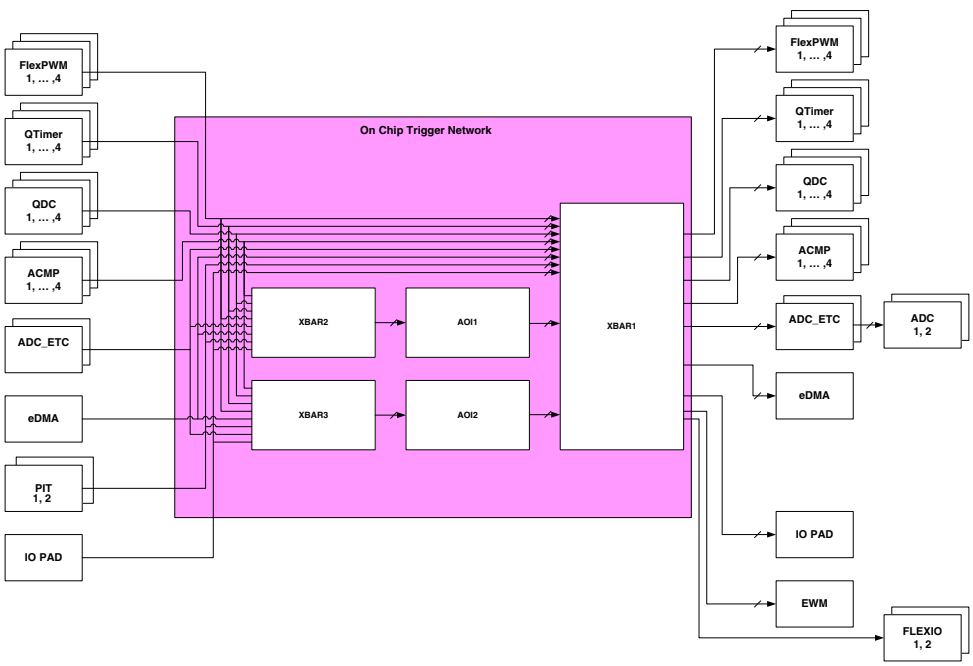


Figure 81-1. On-Chip Cross Trigger Network

81.1.1 Cross BAR (XBAR)

Each crossbar switch is an array of MUXes with shared inputs. Each mux output provides one output of the crossbar. The number of inputs and the number of MUXes/outputs is user configurable and registers are provided to select which of the shared inputs is routed

to each output. The crossbar switches are used to reconfigure data paths between peripherals (peripheral output to peripheral input) as well as between peripherals and GPIO.

81.1.2 And-Or-Inverter (AOI)

The AOI module provides an universal Boolean function generator using a four term sum of products expression, with each product term containing true or complement values of the four selected inputs (A, B, C, D).

Chapter 82
And-Or-Inverter (AOI)

82.1 Chip-specific AOI information

Table 82-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
XBAR resource assignments	XBAR	XBAR Resource Assignments

82.2 Overview

The AND/OR/INVERT (AOI) module supports the generation of a configurable number of EVENT signals. Each output EVENTn is a configurable and/or/invert function of four associated AOI inputs: An, Bn, Cn, and Dn.

This module is designed to be integrated in conjunction with one or more inter-peripheral crossbar switch (XBAR) modules. A crossbar switch is typically used to select the 4*n AOI inputs from among available peripheral outputs and GPIO signals. The n EVENTn outputs from the AOI module are typically used as additional inputs to a second crossbar switch, adding to it the ability to connect to its outputs an arbitrary 4-input boolean function of its other inputs.

The AOI controller is a slave peripheral module connecting event input indicators from a variety of device modules and generating event output signals that can be routed to an inter-peripheral crossbar switch or other peripherals. Its programming model is accessed through the standard IPS (Sky Blue) slave interface. The module is designed to be very configurable in terms of the functionality of its integrated AOI functions.

82.2.1 Block Diagram

The AOI module supports a configurable number of event outputs, where each event output represents a user-programmed combinational boolean function based on four event inputs. The key features of this module include:

- Four dedicated inputs for each event output
- User-programmable combinational boolean function evaluation for each event output
- Memory-mapped device connected to a slave peripheral (IPS) bus
- Configurable number of event outputs

NOTE

The connections from the AOI module outputs to other functions is SoC-specific.

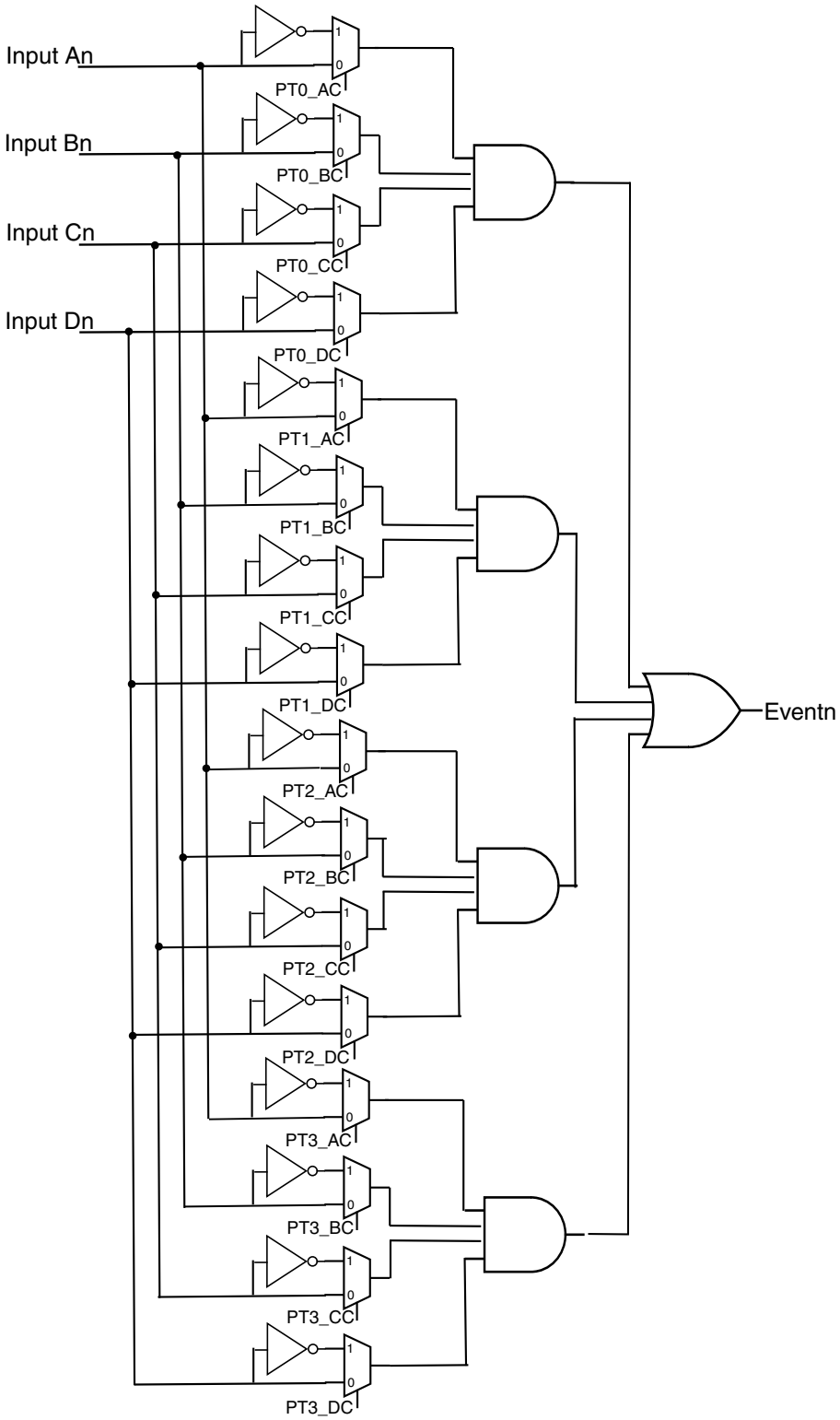


Figure 82-1. Simplified AOI Block Diagram

82.2.2 Features

The major features of the AOI module are summarized below:

- Highly programmable module for creating combinational boolean events for use as hardware triggers
  - Each channel has four event inputs and one output
  - Evaluates a combinational boolean expression as the sum of four products where each product term includes all four selected input sources available as true or complement values
  - Event output is formed as purely combinational logic and operates as a hardware trigger
- Memory-mapped device connected to the slave peripheral (IPS) bus
  - Programming model organized per channel for simplified software

82.2.3 Modes of Operation

The AOI module does not support any special modes of operation. As shown in Figure 82-1, its operation is primarily controlled by the selected event inputs and outputs. Additionally, as a memory-mapped device located on the slave peripheral bus, it responds based strictly on memory address for accesses to its programming model.

The AOI module resides in the slave peripheral *bus clock domain*.

82.3 Functional Description

The AOI is a highly programmable module for creating combinational boolean outputs for use as hardware triggers. Each AOI output channel, as shown in Figure 82-1, has one logic function:

- Evaluation of a combinational boolean expression as a sum of four products where each product term includes all four selected input sources available as true or complement values

A typical application of the AOI module is to be integrated with one or more inter-peripheral crossbar switch modules as illustrated in the following figure. The 20 external inputs are shared by two crossbar switch modules. The crossbar switch on the top is used to select the inputs to four 4-input AOI functions in the AOI module. The outputs of these four AOI functions are output from the AOI module and are added to the original 20 external inputs to provide a total of 24 inputs to the bottom crossbar switch. As a result, the bottom crossbar can not only direct any of the original 20 external inputs to any of its outputs, it can also now direct any one of four 4-input AOI functions of those external inputs to any of its outputs.

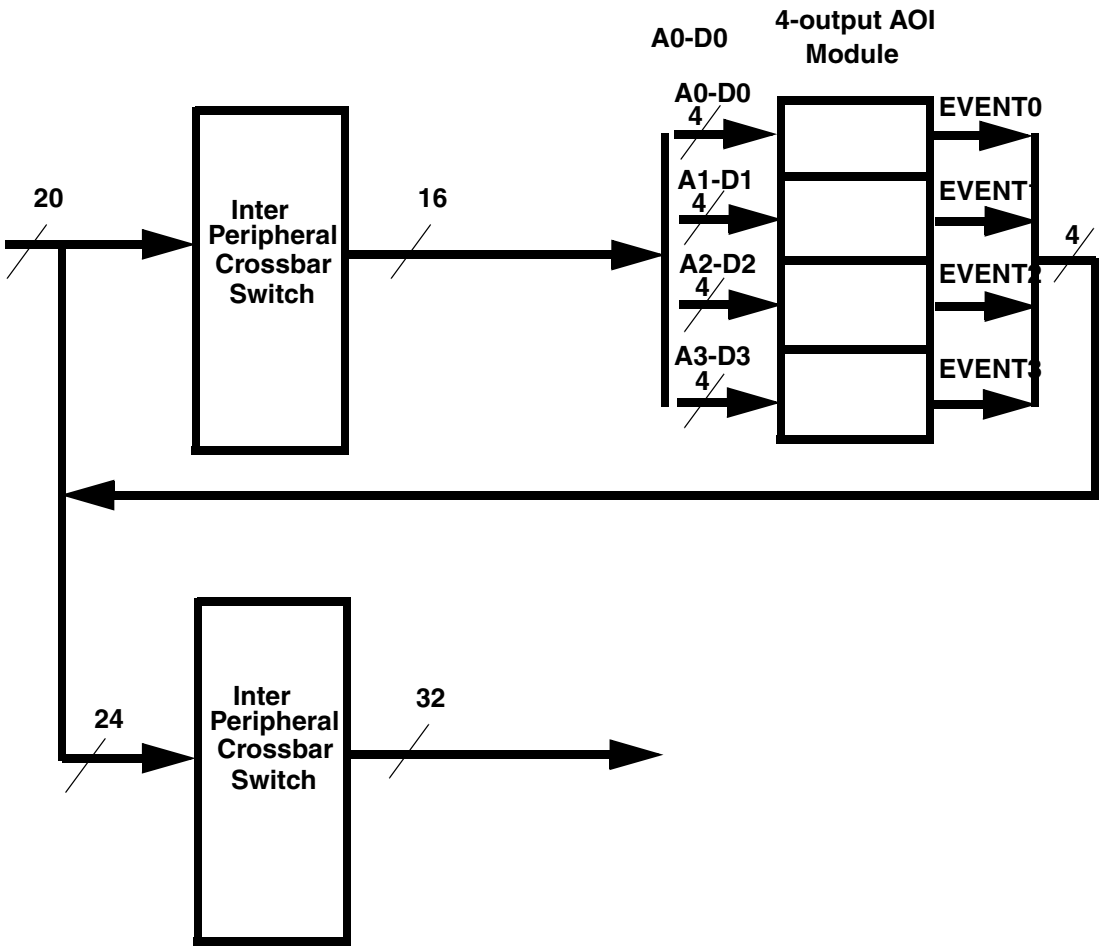


Figure 82-2. Integration Example of AOI with two Inter-Peripheral Crossbar Switches

82.3.1 Configuration Examples for the Boolean Function Evaluation

This section presents examples of the programming model configuration for simple boolean expressions.

The AOI module provides a universal boolean function generator using a four-term sum of products expression with each product term containing true or complement values of the four selected event inputs (A, B, C, D). Specifically, the event output is defined by the following “4 x 4” boolean expression:

```
EVENTn
    = (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 0
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 1
```

```
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 2
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 3
```

where each selected input term in each product term can be configured to produce a logical 0 or 1 or pass the true or complement of the selected event input. Each product term uses eight bits of configuration information, two bits for each of the four selected event inputs. The actual boolean expression implemented in each channel is:

```
EVENTn
    = (PT0_AC[0] & An |  PT0_AC[1] & ~An)// product term 0
                      & (PT0_BC[0] &  Bn | PTO_BC[1] & ~Bn)
                      & (PT0_CC[0] &  Cn | PTO_CC[1] & ~Cn)
                      & (PT0_DC[0] &  Dn | PTO_DC[1] & ~Dn)

    | (PT1_AC[0] & An |  PT1_AC[1] & ~An)// product term 1
                      & (PT1_BC[0] &  Bn | PT1_BC[1] & ~Bn)
                      & (PT1_CC[0] &  Cn | PT1_CC[1] & ~Cn)
                      & (PT1_DC[0] &  Dn | PT1_DC[1] & ~Dn)

    | (PT2_AC[0] & An |  PT2_AC[1] & ~An)// product term 2
                      & (PT2_BC[0] &  Bn | PT2_BC[1] & ~Bn)
                      & (PT2_CC[0] &  Cn | PT2_CC[1] & ~Cn)
                      & (PT2_DC[0] &  Dn | PT2_DC[1] & ~Dn)

    | (PT3_AC[0] & An |  PT3_AC[1] & ~An)// product term 3
                      & (PT3_BC[0] &  Bn | PT3_BC[1] & ~Bn)
                      & (PT3_CC[0] &  Cn | PT3_CC[1] & ~Cn)
                      & (PT3_DC[0] &  Dn | PT3_DC[1] & ~Dn)
```

where the bits of the combined {BFCRT01n,BFCRT23n} registers correspond to the `PT{0-3}_{A,B,C,D}C[1:0]` terms in the equation.

Consider the settings of the combined 32-bit {BFCRT01n,BFCRT23n} registers for several simple boolean expressions as shown in Table 82-2.

Table 82-2. BFCRTn Values for Simple Boolean Expressions

Event Output Expression	PT0	PT1	PT2	PT3	{BFCRT01, BFCRT23}
A & B	A & B	0	0	0	01011111_00000000_00000000_00000000
A & B & C	A & B & C	0	0	0	01010111_00000000_00000000_00000000
(A & B & C) + D	A & B & C	D	0	0	01010111_11111101_00000000_00000000
A + B + C + D	A	B	C	D	01111111_11011111_11110111_11111101
(A & ~B) + (~A & B)	A & ~B	~A & B	0	0	01101111_10011111_00000000_00000000

As can be seen in these examples, the resulting logic provides a simple yet powerful boolean function evaluation for defining an event output.

82.3.2 AOI Timing Between Inputs and Outputs

Each EVENTn output of the AOI module is a combination function of its four dedicated inputs An, Bn, Cn, and Dn. Propagation through the AOI and any associated inter-peripheral crossbar switch modules is intended to be single bus clock cycle.

82.3.3 Clocks

The bus clock is used for register accesses by the system, but AOI functions themselves are asynchronous and do not use a clock.

82.4 External Signal Description

The AOI module does not directly support any external interfaces. There may be package input signals (indirectly) connected to the module as event inputs, but since the *AOI does not include any input synchronization hardware*, this function must be handled before the event input signals are routed into the module.

82.5 Memory Map and Register Descriptions

The AOI module supports access to its programming model via a 16-bit peripheral bus connection. The module is designed to support 16-bit accesses only. Functionality for accesses of other widths is undefined.

82.5.1 AOI register descriptions

The AOI module supports a specific number of event outputs. Each output EVENTn outputs a four-term AOI function of four binary inputs: An, Bn, Cn, and Dn. A pair of 16-bit registers configures this four-term AOI function: The two registers BFCRT01n and BFCRT23n define the configuration for the evaluation of the Boolean function defining EVENTn, where *n* is the event output channel number. The BFCRT01n register defines the configuration of product terms 0 and 1, and the BFCRT23n register defines the configuration of product terms 2 and 3.

The AOI module provides a universal Boolean function generator using a four-term sum of products expression with each product term containing true or complement values of the four selected event inputs (An, Bn, Cn, Dn). Specifically, the EVENTn output is defined by the following "4 x 4" Boolean expression:

```
EVENTn
    = (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 0
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 1
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 2
    | (0,An,~An,1) & (0,Bn,~Bn,1) & (0,Cn,~Cn,1) & (0,Dn,~Dn,1)// product term 3
```

where each selected input of each product term can be configured to produce a logical 0 or 1 or pass the true or complement of the selected event input. Each product term uses 8 bits of configuration information, 2 bits for each of the four selected event inputs. The resulting logic provides a simple yet powerful Boolean function evaluation for defining an event output.

These AOI functions are combinational in nature and are intended to be sampled and used synchronously.

82.5.1.1 AOI memory map

AOI1 base address: 400B_8000h

AOI2 base address: 400B_C000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Boolean Function Term 0 and 1 Configuration Register for EVENTn (BFCRT010)	16	RW	0000
2	Boolean Function Term 2 and 3 Configuration Register for EVENTn (BFCRT230)	16	RW	0000
4	Boolean Function Term 0 and 1 Configuration Register for EVENTn (BFCRT011)	16	RW	0000
6	Boolean Function Term 2 and 3 Configuration Register for EVENTn (BFCRT231)	16	RW	0000
8	Boolean Function Term 0 and 1 Configuration Register for EVENTn (BFCRT012)	16	RW	0000
A	Boolean Function Term 2 and 3 Configuration Register for EVENTn (BFCRT232)	16	RW	0000
C	Boolean Function Term 0 and 1 Configuration Register for EVENTn (BFCRT013)	16	RW	0000
E	Boolean Function Term 2 and 3 Configuration Register for EVENTn (BFCRT233)	16	RW	0000

82.5.1.2 Boolean Function Term 0 and 1 Configuration Register for EVENTn (BFCRT010 - BFCRT013)

82.5.1.2.1 Offset

Register	Offset
BFCRT010	0h
BFCRT011	4h
BFCRT012	8h
BFCRT013	Ch

82.5.1.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PT0_AC		PT0_BC		PT0_CC		PT0_DC		PT1_AC		PT1_BC		PT1_CC		PT1_DC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

82.5.1.2.3 Fields

Field	Description
15-14 PT0_AC	Product term 0, A input configuration This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 0. 00 - Force the A input in this product term to a logical zero 01 - Pass the A input in this product term 10 - Complement the A input in this product term 11 - Force the A input in this product term to a logical one
13-12 PT0_BC	Product term 0, B input configuration This 2-bit field defines the Boolean evaluation associated with the selected input B in product term 0. 00 - Force the B input in this product term to a logical zero 01 - Pass the B input in this product term 10 - Complement the B input in this product term 11 - Force the B input in this product term to a logical one
11-10 PT0_CC	Product term 0, C input configuration This 2-bit field defines the Boolean evaluation associated with the selected input C in product term 0. 00 - Force the C input in this product term to a logical zero

Table continues on the next page...

Field	Description
	01 - Pass the C input in this product term
	10 - Complement the C input in this product term
	11 - Force the C input in this product term to a logical one
9-8	Product term 0, D input configuration
PT0_DC	This 2-bit field defines the Boolean evaluation associated with the selected input D in product term 0.
	00 - Force the D input in this product term to a logical zero
	01 - Pass the D input in this product term
	10 - Complement the D input in this product term
	11 - Force the D input in this product term to a logical one
7-6	Product term 1, A input configuration
PT1_AC	This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 1.
	00 - Force the A input in this product term to a logical zero
	01 - Pass the A input in this product term
	10 - Complement the A input in this product term
	11 - Force the A input in this product term to a logical one
5-4	Product term 1, B input configuration
PT1_BC	This 2-bit field defines the Boolean evaluation associated with the selected input B in product term 1.
	00 - Force the B input in this product term to a logical zero
	01 - Pass the B input in this product term
	10 - Complement the B input in this product term
	11 - Force the B input in this product term to a logical one
3-2	Product term 1, C input configuration
PT1_CC	This 2-bit field defines the Boolean evaluation associated with the selected input C in product term 1.
	00 - Force the C input in this product term to a logical zero
	01 - Pass the C input in this product term
	10 - Complement the C input in this product term
	11 - Force the C input in this product term to a logical one
1-0	Product term 1, D input configuration
PT1_DC	This 2-bit field defines the Boolean evaluation associated with the selected input D in product term 1.
	00 - Force the D input in this product term to a logical zero
	01 - Pass the D input in this product term
	10 - Complement the D input in this product term
	11 - Force the D input in this product term to a logical one

82.5.1.3 Boolean Function Term 2 and 3 Configuration Register for EVENTn (BFCRT230 - BFCRT233)

82.5.1.3.1 Offset

Register	Offset
BFCRT230	2h
BFCRT231	6h
BFCRT232	Ah
BFCRT233	Eh

82.5.1.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PT2_AC		PT2_BC		PT2_CC		PT2_DC		PT3_AC		PT3_BC		PT3_CC		PT3_DC	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

82.5.1.3.3 Fields

Field	Description
15-14 PT2_AC	Product term 2, A input configuration This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 2. 00 - Force the A input in this product term to a logical zero 01 - Pass the A input in this product term 10 - Complement the A input in this product term 11 - Force the A input in this product term to a logical one
13-12 PT2_BC	Product term 2, B input configuration This 2-bit field defines the Boolean evaluation associated with the selected input B in product term 2. 00 - Force the B input in this product term to a logical zero 01 - Pass the B input in this product term 10 - Complement the B input in this product term 11 - Force the B input in this product term to a logical one
11-10 PT2_CC	Product term 2, C input configuration This 2-bit field defines the Boolean evaluation associated with the selected input C in product term 2. 00 - Force the C input in this product term to a logical zero 01 - Pass the C input in this product term 10 - Complement the C input in this product term 11 - Force the C input in this product term to a logical one
9-8	Product term 2, D input configuration

Table continues on the next page...

Field	Description
PT2_DC	This 2-bit field defines the Boolean evaluation associated with the selected input D in product term 2. 00 - Force the D input in this product term to a logical zero 01 - Pass the D input in this product term 10 - Complement the D input in this product term 11 - Force the D input in this product term to a logical one
7-6 PT3_AC	Product term 3, A input configuration This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 3. 00 - Force the A input in this product term to a logical zero 01 - Pass the A input in this product term 10 - Complement the A input in this product term 11 - Force the A input in this product term to a logical one
5-4 PT3_BC	Product term 3, B input configuration This 2-bit field defines the Boolean evaluation associated with the selected input B in product term 3. 00 - Force the B input in this product term to a logical zero 01 - Pass the B input in this product term 10 - Complement the B input in this product term 11 - Force the B input in this product term to a logical one
3-2 PT3_CC	Product term 3, C input configuration This 2-bit field defines the Boolean evaluation associated with the selected input C in product term 3. 00 - Force the C input in this product term to a logical zero 01 - Pass the C input in this product term 10 - Complement the C input in this product term 11 - Force the C input in this product term to a logical one
1-0 PT3_DC	Product term 3, D input configuration This 2-bit field defines the Boolean evaluation associated with the selected input D in product term 3. 00 - Force the D input in this product term to a logical zero 01 - Pass the D input in this product term 10 - Complement the D input in this product term 11 - Force the D input in this product term to a logical one

Chapter 83
Inter-Peripheral Crossbar Switch A (XBARA)

83.1 Chip-specific XBAR information

Table 83-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

On this device, for XBAR1 (see the XBARA chapter), the number of inputs is 144, and the number of outputs is 175. For XBAR2 and XBAR3 (see the XBARB chapter), the number of inputs is 98 , and the number of outputs is 16.

83.2 Overview

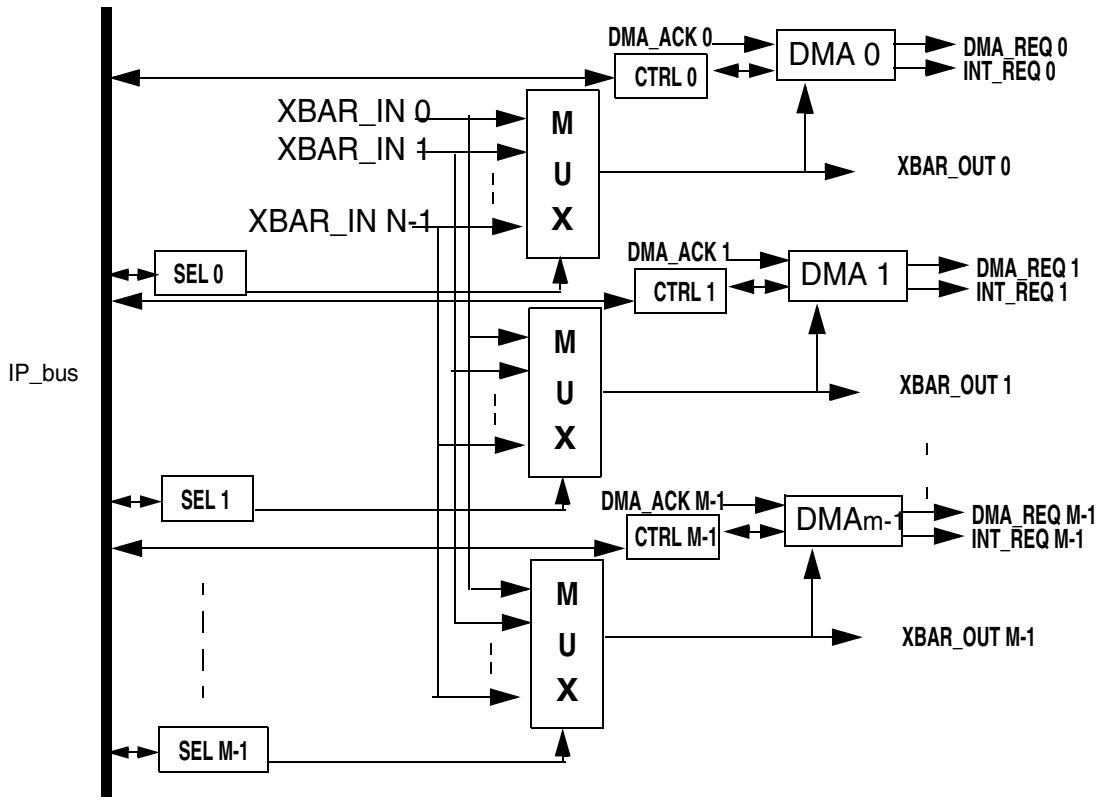
This module implements an array of M N-input combinational muxes. All muxes share the same N inputs in the same order, but each mux has its own independent select field.

The intended application of this module is to provide a flexible crossbar switch function that allows any input (typically from external GPIO or internal module outputs) to be connected to any output (typically to external GPIO or internal module inputs) under user control. This is used to allow user configuration of data paths between internal modules and between internal modules and GPIO.

A subset of the muxes can be configured to support edge detection and either interrupt or DMA request generation based on detected signal edges on the mux output. This allows signal transitions on the signals feeding the crossbar to trigger interrupts or initiate data transfers via DMA into or out of other system modules.

83.2.1 Block Diagram

The block diagram for XBAR is shown in Figure 83-1.



Note: Control field support can be restricted to a subset of outputs. CTRLn represents the memory mapped control fields in the XBAR_CTRL registers for each corresponding XBAR_OUT output. Control fields provide edge status, edge selection, and interrupt/DMA generation controls.

Figure 83-1. XBAR Block Diagram

83.2.2 Features

The XBAR module design includes these distinctive features:

- M identical N-input muxes with individual select fields.

- Edge detection with associated interrupt or DMA request generation for a subset of mux outputs.
- Memory mapped registers with IPBus interface for select and control fields.
- Register write protection input signal.

83.2.3 Modes of Operation

The XBAR module design operates in only a single mode of operation: Functional Mode. The various operation modes are detailed in Functional Mode.

83.3 Functional Description

83.3.1 General

The XBAR module has only one mode of operation, functional mode.

83.3.2 Functional Mode

The value of each mux output is XBAR_OUT[n] = XBAR_IN[SELn]. The SELn select values are configured in the XBAR_SEL registers. All muxes share the same inputs in the same order.

The following is an example to show how to specify the specific input for the specific output:

To select XBAR_IN03 to connect with XBAR_OUT07, according to the index of XBAR_OUT07, select the SEL07 (same number with the selected XBAR_OUT index) fields in the MSBs of the Select Register 3 XBAR_SEL3, and assign the index (0x03, Hexadecimal) of the selected XBAR_IN in it.

```
XBAR_SEL3 &= 0x0011; /*Clear the SEL7 fields
XBAR_SEL3 |= 0x0300; /*Assign the 0x03 to the SEL7 fields
```

A subset of XBAR_OUT[*] outputs has dedicated control fields in a Crossbar Control (XBAR_CTRL) register. Control fields provide the ability to perform edge detection on the corresponding XBAR_OUT output. Edge detection in turn can optionally be used to trigger an interrupt or DMA request. The intention is that, by detecting specified edges on signals propagating through the Crossbar, interrupts or DMA requests can be triggered to perform data transfers to or from other system components.

Control fields include an edge status field (STS), an detected edge type field (EDGE), and interrupt and DMA enable fields (IEN and DEN). STSn is set to 1 when an edge consistent with EDGEn occurs on XBAR_OUT[n]. STSn is cleared by writing 1 to it. Writing 0 as no effect. See Interrupts and DMA Requests for details on the use of STSn for DMA and interrupt request generation.

83.3.3 Clocks

All sequential functionality is controlled by the Bus Clock.

83.3.4 Resets

The XBAR module can be reset by only a hard reset, which forces all registers to their reset state.

83.3.5 Interrupts and DMA Requests

For each XBAR_OUT[*] output with XBAR_CTRL register support, DMA or interrupt functionality can be enabled by setting the corresponding XBAR_CTRL register bit DENn or IENn to 1. DENn and IENn should not be set to 1 at the same time for the same output XBAR_OUT[n].

Setting DENn to 1 enables DMA functionality for XBAR_OUT[n]. When DMA functionality is enabled, the output DMA_REQ[n] reflects the value of STSn. Thus the DMA request asserts when the edge specified by EDGEn is detected on XBAR_OUT[n]. Also, a rising edge on DMA_ACK[n] sets STSn to zero and thus clears the DMA request. When DEN is 0, DMA_REQ[n] is held low and DMA_ACK[n] is ignored.

Setting IENn to 1 enables interrupt functionality for XBAR_OUT[n]. When interrupt functionality is enabled, the output INT_REQ[n] reflects the value of STSn. Thus the interrupt request asserts when the edge specified by EDGEn is detected on XBAR_OUT[n]. The interrupt request is cleared by writing a 1 to STSn. When IENn is 0, INT_REQ[n] is held low.

83.4 External Signals

The following table summarizes the module's external signals.

Table 83-2. Control Signal Properties

Name	I/O Type	Function	Reset State	Notes
XBAR_OUT [0:NUMOUT-1]	O	Mux Outputs with configurable width	XBAR_IN[0]	
XBAR_IN [0:NUMIN-1]	I	Mux Inputs with configurable width	*	
DMA_REQ	O	DMA request	0	
INT_REQ	O	Interrupt request	0	
DMA_ACK	I	DMA acknowledge	0	

At reset, each output XBAR_OUT[*] contains the reset value of the signal driving XBAR_IN[0].

83.4.1 XBAR_OUT[0:NUM_OUT-1] - MUX Outputs

This is a one-dimensional array of the mux outputs. The value on each output XBAR_OUT[n] is determined by the setting of the corresponding memory mapped register SELn such that XBAR_OUT[n] = XBAR_IN[SELn].

83.4.2 XBAR_IN[0:NUM_IN-1] - MUX Inputs

This is a one-dimensional array consisting of the inputs shared by each mux. All muxes share the same inputs in the same order.

83.4.3 DMA_REQ[n] - DMA Request Output(s)

DMA_REQ[n] is a DMA request to the DMA controller.

83.4.4 DMA_ACK[n] - DMA Acknowledge Input(s)

DMA_ACK[n] is a DMA acknowledge input from the DMA controller.

83.4.5 INT_REQ[n] - Interrupt Request Output(s)

INT_REQ[n] is an interrupt request output to the interrupt controller.

83.5 Memory Map and Register Descriptions

The XBAR module has select registers and control registers.

In the XBAR select registers, the SELn fields select which of the shared inputs (XBAR_IN[*]) is muxed to each mux output (XBAR_OUT[*]). There is one SELn field per mux and therefore one per XBAR_OUT output. Crossbar output XBAR_OUT[n] presents the value of XBAR_IN[SELn]. Each select register contains two SELn fields. In the first select register, the LSBs contain the select field for mux 0, and the MSBs contain the select field for mux 1. The pattern repeats in subsequent select registers.

The actual signals connected to XBAR_IN and XBAR_OUT are application specific and are described in the Interrupts, DMA Events, and XBAR Assignments chapter details.

The XBAR control registers configure edge detection, interrupt, and DMA features for a subset of the XBAR_OUT[*] outputs.

XBARA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_C000	Crossbar A Select Register 0 (XBARA1_SEL0)	16	R/W	0000h	83.5.1/6022
4003_C002	Crossbar A Select Register 1 (XBARA1_SEL1)	16	R/W	0000h	83.5.2/6022
4003_C004	Crossbar A Select Register 2 (XBARA1_SEL2)	16	R/W	0000h	83.5.3/6023
4003_C006	Crossbar A Select Register 3 (XBARA1_SEL3)	16	R/W	0000h	83.5.4/6023
4003_C008	Crossbar A Select Register 4 (XBARA1_SEL4)	16	R/W	0000h	83.5.5/6024
4003_C00A	Crossbar A Select Register 5 (XBARA1_SEL5)	16	R/W	0000h	83.5.6/6024
4003_C00C	Crossbar A Select Register 6 (XBARA1_SEL6)	16	R/W	0000h	83.5.7/6024
4003_C00E	Crossbar A Select Register 7 (XBARA1_SEL7)	16	R/W	0000h	83.5.8/6025
4003_C010	Crossbar A Select Register 8 (XBARA1_SEL8)	16	R/W	0000h	83.5.9/6025
4003_C012	Crossbar A Select Register 9 (XBARA1_SEL9)	16	R/W	0000h	83.5.10/ 6026
4003_C014	Crossbar A Select Register 10 (XBARA1_SEL10)	16	R/W	0000h	83.5.11/ 6026
4003_C016	Crossbar A Select Register 11 (XBARA1_SEL11)	16	R/W	0000h	83.5.12/ 6026
4003_C018	Crossbar A Select Register 12 (XBARA1_SEL12)	16	R/W	0000h	83.5.13/ 6027
4003_C01A	Crossbar A Select Register 13 (XBARA1_SEL13)	16	R/W	0000h	83.5.14/ 6027
4003_C01C	Crossbar A Select Register 14 (XBARA1_SEL14)	16	R/W	0000h	83.5.15/ 6028

Table continues on the next page...

XBARA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_C01E	Crossbar A Select Register 15 (XBARA1_SEL15)	16	R/W	0000h	83.5.16/ 6028
4003_C020	Crossbar A Select Register 16 (XBARA1_SEL16)	16	R/W	0000h	83.5.17/ 6028
4003_C022	Crossbar A Select Register 17 (XBARA1_SEL17)	16	R/W	0000h	83.5.18/ 6029
4003_C024	Crossbar A Select Register 18 (XBARA1_SEL18)	16	R/W	0000h	83.5.19/ 6029
4003_C026	Crossbar A Select Register 19 (XBARA1_SEL19)	16	R/W	0000h	83.5.20/ 6030
4003_C028	Crossbar A Select Register 20 (XBARA1_SEL20)	16	R/W	0000h	83.5.21/ 6030
4003_C02A	Crossbar A Select Register 21 (XBARA1_SEL21)	16	R/W	0000h	83.5.22/ 6030
4003_C02C	Crossbar A Select Register 22 (XBARA1_SEL22)	16	R/W	0000h	83.5.23/ 6031
4003_C02E	Crossbar A Select Register 23 (XBARA1_SEL23)	16	R/W	0000h	83.5.24/ 6031
4003_C030	Crossbar A Select Register 24 (XBARA1_SEL24)	16	R/W	0000h	83.5.25/ 6032
4003_C032	Crossbar A Select Register 25 (XBARA1_SEL25)	16	R/W	0000h	83.5.26/ 6032
4003_C034	Crossbar A Select Register 26 (XBARA1_SEL26)	16	R/W	0000h	83.5.27/ 6032
4003_C036	Crossbar A Select Register 27 (XBARA1_SEL27)	16	R/W	0000h	83.5.28/ 6033
4003_C038	Crossbar A Select Register 28 (XBARA1_SEL28)	16	R/W	0000h	83.5.29/ 6033
4003_C03A	Crossbar A Select Register 29 (XBARA1_SEL29)	16	R/W	0000h	83.5.30/ 6034
4003_C03C	Crossbar A Select Register 30 (XBARA1_SEL30)	16	R/W	0000h	83.5.31/ 6034
4003_C03E	Crossbar A Select Register 31 (XBARA1_SEL31)	16	R/W	0000h	83.5.32/ 6034
4003_C040	Crossbar A Select Register 32 (XBARA1_SEL32)	16	R/W	0000h	83.5.33/ 6035
4003_C042	Crossbar A Select Register 33 (XBARA1_SEL33)	16	R/W	0000h	83.5.34/ 6035
4003_C044	Crossbar A Select Register 34 (XBARA1_SEL34)	16	R/W	0000h	83.5.35/ 6036
4003_C046	Crossbar A Select Register 35 (XBARA1_SEL35)	16	R/W	0000h	83.5.36/ 6036
4003_C048	Crossbar A Select Register 36 (XBARA1_SEL36)	16	R/W	0000h	83.5.37/ 6036

Table continues on the next page...

XBARA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_C04A	Crossbar A Select Register 37 (XBARA1_SEL37)	16	R/W	0000h	83.5.38/ 6037
4003_C04C	Crossbar A Select Register 38 (XBARA1_SEL38)	16	R/W	0000h	83.5.39/ 6037
4003_C04E	Crossbar A Select Register 39 (XBARA1_SEL39)	16	R/W	0000h	83.5.40/ 6038
4003_C050	Crossbar A Select Register 40 (XBARA1_SEL40)	16	R/W	0000h	83.5.41/ 6038
4003_C052	Crossbar A Select Register 41 (XBARA1_SEL41)	16	R/W	0000h	83.5.42/ 6038
4003_C054	Crossbar A Select Register 42 (XBARA1_SEL42)	16	R/W	0000h	83.5.43/ 6039
4003_C056	Crossbar A Select Register 43 (XBARA1_SEL43)	16	R/W	0000h	83.5.44/ 6039
4003_C058	Crossbar A Select Register 44 (XBARA1_SEL44)	16	R/W	0000h	83.5.45/ 6040
4003_C05A	Crossbar A Select Register 45 (XBARA1_SEL45)	16	R/W	0000h	83.5.46/ 6040
4003_C05C	Crossbar A Select Register 46 (XBARA1_SEL46)	16	R/W	0000h	83.5.47/ 6040
4003_C05E	Crossbar A Select Register 47 (XBARA1_SEL47)	16	R/W	0000h	83.5.48/ 6041
4003_C060	Crossbar A Select Register 48 (XBARA1_SEL48)	16	R/W	0000h	83.5.49/ 6041
4003_C062	Crossbar A Select Register 49 (XBARA1_SEL49)	16	R/W	0000h	83.5.50/ 6042
4003_C064	Crossbar A Select Register 50 (XBARA1_SEL50)	16	R/W	0000h	83.5.51/ 6042
4003_C066	Crossbar A Select Register 51 (XBARA1_SEL51)	16	R/W	0000h	83.5.52/ 6042
4003_C068	Crossbar A Select Register 52 (XBARA1_SEL52)	16	R/W	0000h	83.5.53/ 6043
4003_C06A	Crossbar A Select Register 53 (XBARA1_SEL53)	16	R/W	0000h	83.5.54/ 6043
4003_C06C	Crossbar A Select Register 54 (XBARA1_SEL54)	16	R/W	0000h	83.5.55/ 6044
4003_C06E	Crossbar A Select Register 55 (XBARA1_SEL55)	16	R/W	0000h	83.5.56/ 6044
4003_C070	Crossbar A Select Register 56 (XBARA1_SEL56)	16	R/W	0000h	83.5.57/ 6044
4003_C072	Crossbar A Select Register 57 (XBARA1_SEL57)	16	R/W	0000h	83.5.58/ 6045
4003_C074	Crossbar A Select Register 58 (XBARA1_SEL58)	16	R/W	0000h	83.5.59/ 6045

Table continues on the next page...

XBARA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_C076	Crossbar A Select Register 59 (XBARA1_SEL59)	16	R/W	0000h	83.5.60/ 6046
4003_C078	Crossbar A Select Register 60 (XBARA1_SEL60)	16	R/W	0000h	83.5.61/ 6046
4003_C07A	Crossbar A Select Register 61 (XBARA1_SEL61)	16	R/W	0000h	83.5.62/ 6046
4003_C07C	Crossbar A Select Register 62 (XBARA1_SEL62)	16	R/W	0000h	83.5.63/ 6047
4003_C07E	Crossbar A Select Register 63 (XBARA1_SEL63)	16	R/W	0000h	83.5.64/ 6047
4003_C080	Crossbar A Select Register 64 (XBARA1_SEL64)	16	R/W	0000h	83.5.65/ 6048
4003_C082	Crossbar A Select Register 65 (XBARA1_SEL65)	16	R/W	0000h	83.5.66/ 6048
4003_C084	Crossbar A Select Register 66 (XBARA1_SEL66)	16	R/W	0000h	83.5.67/ 6048
4003_C086	Crossbar A Select Register 67 (XBARA1_SEL67)	16	R/W	0000h	83.5.68/ 6049
4003_C088	Crossbar A Select Register 68 (XBARA1_SEL68)	16	R/W	0000h	83.5.69/ 6049
4003_C08A	Crossbar A Select Register 69 (XBARA1_SEL69)	16	R/W	0000h	83.5.70/ 6050
4003_C08C	Crossbar A Select Register 70 (XBARA1_SEL70)	16	R/W	0000h	83.5.71/ 6050
4003_C08E	Crossbar A Select Register 71 (XBARA1_SEL71)	16	R/W	0000h	83.5.72/ 6050
4003_C090	Crossbar A Select Register 72 (XBARA1_SEL72)	16	R/W	0000h	83.5.73/ 6051
4003_C092	Crossbar A Select Register 73 (XBARA1_SEL73)	16	R/W	0000h	83.5.74/ 6051
4003_C094	Crossbar A Select Register 74 (XBARA1_SEL74)	16	R/W	0000h	83.5.75/ 6052
4003_C096	Crossbar A Select Register 75 (XBARA1_SEL75)	16	R/W	0000h	83.5.76/ 6052
4003_C098	Crossbar A Select Register 76 (XBARA1_SEL76)	16	R/W	0000h	83.5.77/ 6052
4003_C09A	Crossbar A Select Register 77 (XBARA1_SEL77)	16	R/W	0000h	83.5.78/ 6053
4003_C09C	Crossbar A Select Register 78 (XBARA1_SEL78)	16	R/W	0000h	83.5.79/ 6053
4003_C09E	Crossbar A Select Register 79 (XBARA1_SEL79)	16	R/W	0000h	83.5.80/ 6054
4003_C0A0	Crossbar A Select Register 80 (XBARA1_SEL80)	16	R/W	0000h	83.5.81/ 6054

Table continues on the next page...

XBARA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_C0A2	Crossbar A Select Register 81 (XBARA1_SEL81)	16	R/W	0000h	83.5.82/ 6054
4003_C0A4	Crossbar A Select Register 82 (XBARA1_SEL82)	16	R/W	0000h	83.5.83/ 6055
4003_C0A6	Crossbar A Select Register 83 (XBARA1_SEL83)	16	R/W	0000h	83.5.84/ 6055
4003_C0A8	Crossbar A Select Register 84 (XBARA1_SEL84)	16	R/W	0000h	83.5.85/ 6056
4003_C0AA	Crossbar A Select Register 85 (XBARA1_SEL85)	16	R/W	0000h	83.5.86/ 6056
4003_C0AC	Crossbar A Select Register 86 (XBARA1_SEL86)	16	R/W	0000h	83.5.87/ 6056
4003_C0AE	Crossbar A Select Register 87 (XBARA1_SEL87)	16	R/W	0000h	83.5.88/ 6057
4003_C0B0	Crossbar A Control Register 0 (XBARA1_CTRL0)	16	R/W	0000h	83.5.89/ 6057
4003_C0B2	Crossbar A Control Register 1 (XBARA1_CTRL1)	16	R/W	0000h	83.5.90/ 6059

83.5.1 Crossbar A Select Register 0 (XBARAx_SEL0)

Address: 4003_C000h base + 0h offset = 4003_C000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL1								SEL0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL0 field descriptions

Field	Description
15–8 SEL1	Input (XBARA_INn) to be muxed to XBARA_OUT1 (refer to Functional Description section for input/output assignment)
SEL0	Input (XBARA_INn) to be muxed to XBARA_OUT0 (refer to Functional Description section for input/output assignment)

83.5.2 Crossbar A Select Register 1 (XBARAx_SEL1)

Address: 4003_C000h base + 2h offset = 4003_C002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL3								SEL2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL1 field descriptions

Field	Description
15–8 SEL3	Input (XBARA_INn) to be muxed to XBARA_OUT3 (refer to Functional Description section for input/output assignment)
SEL2	Input (XBARA_INn) to be muxed to XBARA_OUT2 (refer to Functional Description section for input/output assignment)

83.5.3 Crossbar A Select Register 2 (XBARAx_SEL2)

Address: 4003_C000h base + 4h offset = 4003_C004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL5								SEL4							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL2 field descriptions

Field	Description
15–8 SEL5	Input (XBARA_INn) to be muxed to XBARA_OUT5 (refer to Functional Description section for input/output assignment)
SEL4	Input (XBARA_INn) to be muxed to XBARA_OUT4 (refer to Functional Description section for input/output assignment)

83.5.4 Crossbar A Select Register 3 (XBARAx_SEL3)

Address: 4003_C000h base + 6h offset = 4003_C006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL7								SEL6							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL3 field descriptions

Field	Description
15–8 SEL7	Input (XBARA_INn) to be muxed to XBARA_OUT7 (refer to Functional Description section for input/output assignment)
SEL6	Input (XBARA_INn) to be muxed to XBARA_OUT6 (refer to Functional Description section for input/output assignment)

83.5.5 Crossbar A Select Register 4 (XBARAx_SEL4)

Address: 4003_C000h base + 8h offset = 4003_C008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL9								SEL8							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL4 field descriptions

Field	Description
15–8 SEL9	Input (XBARA_INn) to be muxed to XBARA_OUT9 (refer to Functional Description section for input/output assignment)
SEL8	Input (XBARA_INn) to be muxed to XBARA_OUT8 (refer to Functional Description section for input/output assignment)

83.5.6 Crossbar A Select Register 5 (XBARAx_SEL5)

Address: 4003_C000h base + Ah offset = 4003_C00Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL11								SEL10							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL5 field descriptions

Field	Description
15–8 SEL11	Input (XBARA_INn) to be muxed to XBARA_OUT11 (refer to Functional Description section for input/output assignment)
SEL10	Input (XBARA_INn) to be muxed to XBARA_OUT10 (refer to Functional Description section for input/output assignment)

83.5.7 Crossbar A Select Register 6 (XBARAx_SEL6)

Address: 4003_C000h base + Ch offset = 4003_C00Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL13								SEL12							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL6 field descriptions

Field	Description
15–8 SEL13	Input (XBARA_INn) to be muxed to XBARA_OUT13 (refer to Functional Description section for input/output assignment)
SEL12	Input (XBARA_INn) to be muxed to XBARA_OUT12 (refer to Functional Description section for input/output assignment)

83.5.8 Crossbar A Select Register 7 (XBARAx_SEL7)

Address: 4003_C000h base + Eh offset = 4003_C00Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL15								SEL14							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL7 field descriptions

Field	Description
15–8 SEL15	Input (XBARA_INn) to be muxed to XBARA_OUT15 (refer to Functional Description section for input/output assignment)
SEL14	Input (XBARA_INn) to be muxed to XBARA_OUT14 (refer to Functional Description section for input/output assignment)

83.5.9 Crossbar A Select Register 8 (XBARAx_SEL8)

Address: 4003_C000h base + 10h offset = 4003_C010h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL17								SEL16							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL8 field descriptions

Field	Description
15–8 SEL17	Input (XBARA_INn) to be muxed to XBARA_OUT17 (refer to Functional Description section for input/output assignment)
SEL16	Input (XBARA_INn) to be muxed to XBARA_OUT16 (refer to Functional Description section for input/output assignment)

83.5.10 Crossbar A Select Register 9 (XBARAx_SEL9)

Address: 4003_C000h base + 12h offset = 4003_C012h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL19								SEL18							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL9 field descriptions

Field	Description
15–8 SEL19	Input (XBARA_INn) to be muxed to XBARA_OUT19 (refer to Functional Description section for input/ output assignment)
SEL18	Input (XBARA_INn) to be muxed to XBARA_OUT18 (refer to Functional Description section for input/ output assignment)

83.5.11 Crossbar A Select Register 10 (XBARAx_SEL10)

Address: 4003_C000h base + 14h offset = 4003_C014h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL21								SEL20							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL10 field descriptions

Field	Description
15–8 SEL21	Input (XBARA_INn) to be muxed to XBARA_OUT21 (refer to Functional Description section for input/ output assignment)
SEL20	Input (XBARA_INn) to be muxed to XBARA_OUT20 (refer to Functional Description section for input/ output assignment)

83.5.12 Crossbar A Select Register 11 (XBARAx_SEL11)

Address: 4003_C000h base + 16h offset = 4003_C016h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL23								SEL22							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL11 field descriptions

Field	Description
15–8 SEL23	Input (XBARA_INn) to be muxed to XBARA_OUT23 (refer to Functional Description section for input/ output assignment)
SEL22	Input (XBARA_INn) to be muxed to XBARA_OUT22 (refer to Functional Description section for input/ output assignment)

83.5.13 Crossbar A Select Register 12 (XBARAx_SEL12)

Address: 4003_C000h base + 18h offset = 4003_C018h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL25								SEL24							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL12 field descriptions

Field	Description
15–8 SEL25	Input (XBARA_INn) to be muxed to XBARA_OUT25 (refer to Functional Description section for input/ output assignment)
SEL24	Input (XBARA_INn) to be muxed to XBARA_OUT24 (refer to Functional Description section for input/ output assignment)

83.5.14 Crossbar A Select Register 13 (XBARAx_SEL13)

Address: 4003_C000h base + 1Ah offset = 4003_C01Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL27								SEL26							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL13 field descriptions

Field	Description
15–8 SEL27	Input (XBARA_INn) to be muxed to XBARA_OUT27 (refer to Functional Description section for input/ output assignment)
SEL26	Input (XBARA_INn) to be muxed to XBARA_OUT26 (refer to Functional Description section for input/ output assignment)

83.5.15 Crossbar A Select Register 14 (XBARAx_SEL14)

Address: 4003_C000h base + 1Ch offset = 4003_C01Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL29								SEL28							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL14 field descriptions

Field	Description
15–8 SEL29	Input (XBARA_INn) to be muxed to XBARA_OUT29 (refer to Functional Description section for input/ output assignment)
SEL28	Input (XBARA_INn) to be muxed to XBARA_OUT28 (refer to Functional Description section for input/ output assignment)

83.5.16 Crossbar A Select Register 15 (XBARAx_SEL15)

Address: 4003_C000h base + 1Eh offset = 4003_C01Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL31								SEL30							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL15 field descriptions

Field	Description
15–8 SEL31	Input (XBARA_INn) to be muxed to XBARA_OUT31 (refer to Functional Description section for input/ output assignment)
SEL30	Input (XBARA_INn) to be muxed to XBARA_OUT30 (refer to Functional Description section for input/ output assignment)

83.5.17 Crossbar A Select Register 16 (XBARAx_SEL16)

Address: 4003_C000h base + 20h offset = 4003_C020h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL33								SEL32							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL16 field descriptions

Field	Description
15–8 SEL33	Input (XBARA_INn) to be muxed to XBARA_OUT33 (refer to Functional Description section for input/ output assignment)
SEL32	Input (XBARA_INn) to be muxed to XBARA_OUT32 (refer to Functional Description section for input/ output assignment)

83.5.18 Crossbar A Select Register 17 (XBARAx_SEL17)

Address: 4003_C000h base + 22h offset = 4003_C022h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL35								SEL34							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL17 field descriptions

Field	Description
15–8 SEL35	Input (XBARA_INn) to be muxed to XBARA_OUT35 (refer to Functional Description section for input/ output assignment)
SEL34	Input (XBARA_INn) to be muxed to XBARA_OUT34 (refer to Functional Description section for input/ output assignment)

83.5.19 Crossbar A Select Register 18 (XBARAx_SEL18)

Address: 4003_C000h base + 24h offset = 4003_C024h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL37								SEL36							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL18 field descriptions

Field	Description
15–8 SEL37	Input (XBARA_INn) to be muxed to XBARA_OUT37 (refer to Functional Description section for input/ output assignment)
SEL36	Input (XBARA_INn) to be muxed to XBARA_OUT36 (refer to Functional Description section for input/ output assignment)

83.5.20 Crossbar A Select Register 19 (XBARAx_SEL19)

Address: 4003_C000h base + 26h offset = 4003_C026h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL39								SEL38							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL19 field descriptions

Field	Description
15–8 SEL39	Input (XBARA_INn) to be muxed to XBARA_OUT39 (refer to Functional Description section for input/output assignment)
SEL38	Input (XBARA_INn) to be muxed to XBARA_OUT38 (refer to Functional Description section for input/output assignment)

83.5.21 Crossbar A Select Register 20 (XBARAx_SEL20)

Address: 4003_C000h base + 28h offset = 4003_C028h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL41								SEL40							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL20 field descriptions

Field	Description
15–8 SEL41	Input (XBARA_INn) to be muxed to XBARA_OUT41 (refer to Functional Description section for input/output assignment)
SEL40	Input (XBARA_INn) to be muxed to XBARA_OUT40 (refer to Functional Description section for input/output assignment)

83.5.22 Crossbar A Select Register 21 (XBARAx_SEL21)

Address: 4003_C000h base + 2Ah offset = 4003_C02Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL43								SEL42							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL21 field descriptions

Field	Description
15–8 SEL43	Input (XBARA_INn) to be muxed to XBARA_OUT43 (refer to Functional Description section for input/ output assignment)
SEL42	Input (XBARA_INn) to be muxed to XBARA_OUT42 (refer to Functional Description section for input/ output assignment)

83.5.23 Crossbar A Select Register 22 (XBARAx_SEL22)

Address: 4003_C000h base + 2Ch offset = 4003_C02Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL45								SEL44							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL22 field descriptions

Field	Description
15–8 SEL45	Input (XBARA_INn) to be muxed to XBARA_OUT45 (refer to Functional Description section for input/ output assignment)
SEL44	Input (XBARA_INn) to be muxed to XBARA_OUT44 (refer to Functional Description section for input/ output assignment)

83.5.24 Crossbar A Select Register 23 (XBARAx_SEL23)

Address: 4003_C000h base + 2Eh offset = 4003_C02Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL47								SEL46							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL23 field descriptions

Field	Description
15–8 SEL47	Input (XBARA_INn) to be muxed to XBARA_OUT47 (refer to Functional Description section for input/ output assignment)
SEL46	Input (XBARA_INn) to be muxed to XBARA_OUT46 (refer to Functional Description section for input/ output assignment)

83.5.25 Crossbar A Select Register 24 (XBARAx_SEL24)

Address: 4003_C000h base + 30h offset = 4003_C030h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL49								SEL48							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL24 field descriptions

Field	Description
15–8 SEL49	Input (XBARA_INn) to be muxed to XBARA_OUT49 (refer to Functional Description section for input/output assignment)
SEL48	Input (XBARA_INn) to be muxed to XBARA_OUT48 (refer to Functional Description section for input/output assignment)

83.5.26 Crossbar A Select Register 25 (XBARAx_SEL25)

Address: 4003_C000h base + 32h offset = 4003_C032h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL51								SEL50							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL25 field descriptions

Field	Description
15–8 SEL51	Input (XBARA_INn) to be muxed to XBARA_OUT51 (refer to Functional Description section for input/output assignment)
SEL50	Input (XBARA_INn) to be muxed to XBARA_OUT50 (refer to Functional Description section for input/output assignment)

83.5.27 Crossbar A Select Register 26 (XBARAx_SEL26)

Address: 4003_C000h base + 34h offset = 4003_C034h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL53								SEL52							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL26 field descriptions

Field	Description
15–8 SEL53	Input (XBARA_INn) to be muxed to XBARA_OUT53 (refer to Functional Description section for input/ output assignment)
SEL52	Input (XBARA_INn) to be muxed to XBARA_OUT52 (refer to Functional Description section for input/ output assignment)

83.5.28 Crossbar A Select Register 27 (XBARAx_SEL27)

Address: 4003_C000h base + 36h offset = 4003_C036h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL55								SEL54							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL27 field descriptions

Field	Description
15–8 SEL55	Input (XBARA_INn) to be muxed to XBARA_OUT55 (refer to Functional Description section for input/ output assignment)
SEL54	Input (XBARA_INn) to be muxed to XBARA_OUT54 (refer to Functional Description section for input/ output assignment)

83.5.29 Crossbar A Select Register 28 (XBARAx_SEL28)

Address: 4003_C000h base + 38h offset = 4003_C038h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL57								SEL56							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL28 field descriptions

Field	Description
15–8 SEL57	Input (XBARA_INn) to be muxed to XBARA_OUT57 (refer to Functional Description section for input/ output assignment)
SEL56	Input (XBARA_INn) to be muxed to XBARA_OUT56 (refer to Functional Description section for input/ output assignment)

83.5.30 Crossbar A Select Register 29 (XBARAx_SEL29)

Address: 4003_C000h base + 3Ah offset = 4003_C03Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL59								SEL58							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL29 field descriptions

Field	Description
15–8 SEL59	Input (XBARA_INn) to be muxed to XBARA_OUT59 (refer to Functional Description section for input/output assignment)
SEL58	Input (XBARA_INn) to be muxed to XBARA_OUT58 (refer to Functional Description section for input/output assignment)

83.5.31 Crossbar A Select Register 30 (XBARAx_SEL30)

Address: 4003_C000h base + 3Ch offset = 4003_C03Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL61								SEL60							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL30 field descriptions

Field	Description
15–8 SEL61	Input (XBARA_INn) to be muxed to XBARA_OUT61 (refer to Functional Description section for input/output assignment)
SEL60	Input (XBARA_INn) to be muxed to XBARA_OUT60 (refer to Functional Description section for input/output assignment)

83.5.32 Crossbar A Select Register 31 (XBARAx_SEL31)

Address: 4003_C000h base + 3Eh offset = 4003_C03Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL63								SEL62							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL31 field descriptions

Field	Description
15–8 SEL63	Input (XBARA_INn) to be muxed to XBARA_OUT63 (refer to Functional Description section for input/ output assignment)
SEL62	Input (XBARA_INn) to be muxed to XBARA_OUT62 (refer to Functional Description section for input/ output assignment)

83.5.33 Crossbar A Select Register 32 (XBARAx_SEL32)

Address: 4003_C000h base + 40h offset = 4003_C040h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL65								SEL64							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL32 field descriptions

Field	Description
15–8 SEL65	Input (XBARA_INn) to be muxed to XBARA_OUT65 (refer to Functional Description section for input/ output assignment)
SEL64	Input (XBARA_INn) to be muxed to XBARA_OUT64 (refer to Functional Description section for input/ output assignment)

83.5.34 Crossbar A Select Register 33 (XBARAx_SEL33)

Address: 4003_C000h base + 42h offset = 4003_C042h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL67								SEL66							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL33 field descriptions

Field	Description
15–8 SEL67	Input (XBARA_INn) to be muxed to XBARA_OUT67 (refer to Functional Description section for input/ output assignment)
SEL66	Input (XBARA_INn) to be muxed to XBARA_OUT66 (refer to Functional Description section for input/ output assignment)

83.5.35 Crossbar A Select Register 34 (XBARAx_SEL34)

Address: 4003_C000h base + 44h offset = 4003_C044h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL69								SEL68							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL34 field descriptions

Field	Description
15–8 SEL69	Input (XBARA_INn) to be muxed to XBARA_OUT69 (refer to Functional Description section for input/output assignment)
SEL68	Input (XBARA_INn) to be muxed to XBARA_OUT68 (refer to Functional Description section for input/output assignment)

83.5.36 Crossbar A Select Register 35 (XBARAx_SEL35)

Address: 4003_C000h base + 46h offset = 4003_C046h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL71								SEL70							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL35 field descriptions

Field	Description
15–8 SEL71	Input (XBARA_INn) to be muxed to XBARA_OUT71 (refer to Functional Description section for input/output assignment)
SEL70	Input (XBARA_INn) to be muxed to XBARA_OUT70 (refer to Functional Description section for input/output assignment)

83.5.37 Crossbar A Select Register 36 (XBARAx_SEL36)

Address: 4003_C000h base + 48h offset = 4003_C048h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL73								SEL72							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL36 field descriptions

Field	Description
15–8 SEL73	Input (XBARA_INn) to be muxed to XBARA_OUT73 (refer to Functional Description section for input/ output assignment)
SEL72	Input (XBARA_INn) to be muxed to XBARA_OUT72 (refer to Functional Description section for input/ output assignment)

83.5.38 Crossbar A Select Register 37 (XBARAx_SEL37)

Address: 4003_C000h base + 4Ah offset = 4003_C04Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL75								SEL74							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL37 field descriptions

Field	Description
15–8 SEL75	Input (XBARA_INn) to be muxed to XBARA_OUT75 (refer to Functional Description section for input/ output assignment)
SEL74	Input (XBARA_INn) to be muxed to XBARA_OUT74 (refer to Functional Description section for input/ output assignment)

83.5.39 Crossbar A Select Register 38 (XBARAx_SEL38)

Address: 4003_C000h base + 4Ch offset = 4003_C04Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL77								SEL76							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL38 field descriptions

Field	Description
15–8 SEL77	Input (XBARA_INn) to be muxed to XBARA_OUT77 (refer to Functional Description section for input/ output assignment)
SEL76	Input (XBARA_INn) to be muxed to XBARA_OUT76 (refer to Functional Description section for input/ output assignment)

83.5.40 Crossbar A Select Register 39 (XBARAx_SEL39)

Address: 4003_C000h base + 4Eh offset = 4003_C04Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL79								SEL78							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL39 field descriptions

Field	Description
15–8 SEL79	Input (XBARA_INn) to be muxed to XBARA_OUT79 (refer to Functional Description section for input/output assignment)
SEL78	Input (XBARA_INn) to be muxed to XBARA_OUT78 (refer to Functional Description section for input/output assignment)

83.5.41 Crossbar A Select Register 40 (XBARAx_SEL40)

Address: 4003_C000h base + 50h offset = 4003_C050h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL81								SEL80							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL40 field descriptions

Field	Description
15–8 SEL81	Input (XBARA_INn) to be muxed to XBARA_OUT81 (refer to Functional Description section for input/output assignment)
SEL80	Input (XBARA_INn) to be muxed to XBARA_OUT80 (refer to Functional Description section for input/output assignment)

83.5.42 Crossbar A Select Register 41 (XBARAx_SEL41)

Address: 4003_C000h base + 52h offset = 4003_C052h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL83								SEL82							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL41 field descriptions

Field	Description
15–8 SEL83	Input (XBARA_INn) to be muxed to XBARA_OUT83 (refer to Functional Description section for input/ output assignment)
SEL82	Input (XBARA_INn) to be muxed to XBARA_OUT82 (refer to Functional Description section for input/ output assignment)

83.5.43 Crossbar A Select Register 42 (XBARAx_SEL42)

Address: 4003_C000h base + 54h offset = 4003_C054h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL85								SEL84							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL42 field descriptions

Field	Description
15–8 SEL85	Input (XBARA_INn) to be muxed to XBARA_OUT85 (refer to Functional Description section for input/ output assignment)
SEL84	Input (XBARA_INn) to be muxed to XBARA_OUT84 (refer to Functional Description section for input/ output assignment)

83.5.44 Crossbar A Select Register 43 (XBARAx_SEL43)

Address: 4003_C000h base + 56h offset = 4003_C056h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL87								SEL86							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL43 field descriptions

Field	Description
15–8 SEL87	Input (XBARA_INn) to be muxed to XBARA_OUT87 (refer to Functional Description section for input/ output assignment)
SEL86	Input (XBARA_INn) to be muxed to XBARA_OUT86 (refer to Functional Description section for input/ output assignment)

83.5.45 Crossbar A Select Register 44 (XBARAx_SEL44)

Address: 4003_C000h base + 58h offset = 4003_C058h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL89								SEL88							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL44 field descriptions

Field	Description
15–8 SEL89	Input (XBARA_INn) to be muxed to XBARA_OUT89 (refer to Functional Description section for input/output assignment)
SEL88	Input (XBARA_INn) to be muxed to XBARA_OUT88 (refer to Functional Description section for input/output assignment)

83.5.46 Crossbar A Select Register 45 (XBARAx_SEL45)

Address: 4003_C000h base + 5Ah offset = 4003_C05Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL91								SEL90							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL45 field descriptions

Field	Description
15–8 SEL91	Input (XBARA_INn) to be muxed to XBARA_OUT91 (refer to Functional Description section for input/output assignment)
SEL90	Input (XBARA_INn) to be muxed to XBARA_OUT90 (refer to Functional Description section for input/output assignment)

83.5.47 Crossbar A Select Register 46 (XBARAx_SEL46)

Address: 4003_C000h base + 5Ch offset = 4003_C05Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL93								SEL92							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL46 field descriptions

Field	Description
15–8 SEL93	Input (XBARA_INn) to be muxed to XBARA_OUT93 (refer to Functional Description section for input/output assignment)
SEL92	Input (XBARA_INn) to be muxed to XBARA_OUT92 (refer to Functional Description section for input/output assignment)

83.5.48 Crossbar A Select Register 47 (XBARAx_SEL47)

Address: 4003_C000h base + 5Eh offset = 4003_C05Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL95								SEL94							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL47 field descriptions

Field	Description
15–8 SEL95	Input (XBARA_INn) to be muxed to XBARA_OUT95 (refer to Functional Description section for input/output assignment)
SEL94	Input (XBARA_INn) to be muxed to XBARA_OUT94 (refer to Functional Description section for input/output assignment)

83.5.49 Crossbar A Select Register 48 (XBARAx_SEL48)

Address: 4003_C000h base + 60h offset = 4003_C060h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL97								SEL96							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL48 field descriptions

Field	Description
15–8 SEL97	Input (XBARA_INn) to be muxed to XBARA_OUT97 (refer to Functional Description section for input/output assignment)
SEL96	Input (XBARA_INn) to be muxed to XBARA_OUT96 (refer to Functional Description section for input/output assignment)

83.5.50 Crossbar A Select Register 49 (XBARAx_SEL49)

Address: 4003_C000h base + 62h offset = 4003_C062h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL99								SEL98							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL49 field descriptions

Field	Description
15–8 SEL99	Input (XBARA_INn) to be muxed to XBARA_OUT99 (refer to Functional Description section for input/output assignment)
SEL98	Input (XBARA_INn) to be muxed to XBARA_OUT98 (refer to Functional Description section for input/output assignment)

83.5.51 Crossbar A Select Register 50 (XBARAx_SEL50)

Address: 4003_C000h base + 64h offset = 4003_C064h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL101								SEL100							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL50 field descriptions

Field	Description
15–8 SEL101	Input (XBARA_INn) to be muxed to XBARA_OUT101 (refer to Functional Description section for input/output assignment)
SEL100	Input (XBARA_INn) to be muxed to XBARA_OUT100 (refer to Functional Description section for input/output assignment)

83.5.52 Crossbar A Select Register 51 (XBARAx_SEL51)

Address: 4003_C000h base + 66h offset = 4003_C066h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL103								SEL102							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL51 field descriptions

Field	Description
15–8 SEL103	Input (XBARA_INn) to be muxed to XBARA_OUT103 (refer to Functional Description section for input/output assignment)
SEL102	Input (XBARA_INn) to be muxed to XBARA_OUT102 (refer to Functional Description section for input/output assignment)

83.5.53 Crossbar A Select Register 52 (XBARAx_SEL52)

Address: 4003_C000h base + 68h offset = 4003_C068h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL105								SEL104							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL52 field descriptions

Field	Description
15–8 SEL105	Input (XBARA_INn) to be muxed to XBARA_OUT105 (refer to Functional Description section for input/output assignment)
SEL104	Input (XBARA_INn) to be muxed to XBARA_OUT104 (refer to Functional Description section for input/output assignment)

83.5.54 Crossbar A Select Register 53 (XBARAx_SEL53)

Address: 4003_C000h base + 6Ah offset = 4003_C06Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL107								SEL106							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL53 field descriptions

Field	Description
15–8 SEL107	Input (XBARA_INn) to be muxed to XBARA_OUT107 (refer to Functional Description section for input/output assignment)
SEL106	Input (XBARA_INn) to be muxed to XBARA_OUT106 (refer to Functional Description section for input/output assignment)

83.5.55 Crossbar A Select Register 54 (XBARAx_SEL54)

Address: 4003_C000h base + 6Ch offset = 4003_C06Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL109								SEL108							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL54 field descriptions

Field	Description
15–8 SEL109	Input (XBARA_INn) to be muxed to XBARA_OUT109 (refer to Functional Description section for input/ output assignment)
SEL108	Input (XBARA_INn) to be muxed to XBARA_OUT108 (refer to Functional Description section for input/ output assignment)

83.5.56 Crossbar A Select Register 55 (XBARAx_SEL55)

Address: 4003_C000h base + 6Eh offset = 4003_C06Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL111								SEL110							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL55 field descriptions

Field	Description
15–8 SEL111	Input (XBARA_INn) to be muxed to XBARA_OUT111 (refer to Functional Description section for input/ output assignment)
SEL110	Input (XBARA_INn) to be muxed to XBARA_OUT110 (refer to Functional Description section for input/ output assignment)

83.5.57 Crossbar A Select Register 56 (XBARAx_SEL56)

Address: 4003_C000h base + 70h offset = 4003_C070h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL113								SEL112							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL56 field descriptions

Field	Description
15–8 SEL113	Input (XBARA_INn) to be muxed to XBARA_OUT113 (refer to Functional Description section for input/output assignment)
SEL112	Input (XBARA_INn) to be muxed to XBARA_OUT112 (refer to Functional Description section for input/output assignment)

83.5.58 Crossbar A Select Register 57 (XBARAx_SEL57)

Address: 4003_C000h base + 72h offset = 4003_C072h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL115								SEL114							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL57 field descriptions

Field	Description
15–8 SEL115	Input (XBARA_INn) to be muxed to XBARA_OUT115 (refer to Functional Description section for input/output assignment)
SEL114	Input (XBARA_INn) to be muxed to XBARA_OUT114 (refer to Functional Description section for input/output assignment)

83.5.59 Crossbar A Select Register 58 (XBARAx_SEL58)

Address: 4003_C000h base + 74h offset = 4003_C074h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL117								SEL116							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL58 field descriptions

Field	Description
15–8 SEL117	Input (XBARA_INn) to be muxed to XBARA_OUT117 (refer to Functional Description section for input/output assignment)
SEL116	Input (XBARA_INn) to be muxed to XBARA_OUT116 (refer to Functional Description section for input/output assignment)

83.5.60 Crossbar A Select Register 59 (XBARAx_SEL59)

Address: 4003_C000h base + 76h offset = 4003_C076h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL119								SEL118							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL59 field descriptions

Field	Description
15–8 SEL119	Input (XBARA_INn) to be muxed to XBARA_OUT119 (refer to Functional Description section for input/output assignment)
SEL118	Input (XBARA_INn) to be muxed to XBARA_OUT118 (refer to Functional Description section for input/output assignment)

83.5.61 Crossbar A Select Register 60 (XBARAx_SEL60)

Address: 4003_C000h base + 78h offset = 4003_C078h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL121								SEL120							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL60 field descriptions

Field	Description
15–8 SEL121	Input (XBARA_INn) to be muxed to XBARA_OUT121 (refer to Functional Description section for input/output assignment)
SEL120	Input (XBARA_INn) to be muxed to XBARA_OUT120 (refer to Functional Description section for input/output assignment)

83.5.62 Crossbar A Select Register 61 (XBARAx_SEL61)

Address: 4003_C000h base + 7Ah offset = 4003_C07Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL123								SEL122							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL61 field descriptions

Field	Description
15–8 SEL123	Input (XBARA_INn) to be muxed to XBARA_OUT123 (refer to Functional Description section for input/output assignment)
SEL122	Input (XBARA_INn) to be muxed to XBARA_OUT122 (refer to Functional Description section for input/output assignment)

83.5.63 Crossbar A Select Register 62 (XBARAx_SEL62)

Address: 4003_C000h base + 7Ch offset = 4003_C07Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL125								SEL124							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL62 field descriptions

Field	Description
15–8 SEL125	Input (XBARA_INn) to be muxed to XBARA_OUT125 (refer to Functional Description section for input/output assignment)
SEL124	Input (XBARA_INn) to be muxed to XBARA_OUT124 (refer to Functional Description section for input/output assignment)

83.5.64 Crossbar A Select Register 63 (XBARAx_SEL63)

Address: 4003_C000h base + 7Eh offset = 4003_C07Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL127								SEL126							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL63 field descriptions

Field	Description
15–8 SEL127	Input (XBARA_INn) to be muxed to XBARA_OUT127 (refer to Functional Description section for input/output assignment)
SEL126	Input (XBARA_INn) to be muxed to XBARA_OUT126 (refer to Functional Description section for input/output assignment)

83.5.65 Crossbar A Select Register 64 (XBARAx_SEL64)

Address: 4003_C000h base + 80h offset = 4003_C080h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL129								SEL128							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL64 field descriptions

Field	Description
15–8 SEL129	Input (XBARA_INn) to be muxed to XBARA_OUT129 (refer to Functional Description section for input/output assignment)
SEL128	Input (XBARA_INn) to be muxed to XBARA_OUT128 (refer to Functional Description section for input/output assignment)

83.5.66 Crossbar A Select Register 65 (XBARAx_SEL65)

Address: 4003_C000h base + 82h offset = 4003_C082h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL131								SEL130							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL65 field descriptions

Field	Description
15–8 SEL131	Input (XBARA_INn) to be muxed to XBARA_OUT131 (refer to Functional Description section for input/output assignment)
SEL130	Input (XBARA_INn) to be muxed to XBARA_OUT130 (refer to Functional Description section for input/output assignment)

83.5.67 Crossbar A Select Register 66 (XBARAx_SEL66)

Address: 4003_C000h base + 84h offset = 4003_C084h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL133								SEL132							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL66 field descriptions

Field	Description
15–8 SEL133	Input (XBARA_INn) to be muxed to XBARA_OUT133 (refer to Functional Description section for input/output assignment)
SEL132	Input (XBARA_INn) to be muxed to XBARA_OUT132 (refer to Functional Description section for input/output assignment)

83.5.68 Crossbar A Select Register 67 (XBARAx_SEL67)

Address: 4003_C000h base + 86h offset = 4003_C086h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL135								SEL134							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL67 field descriptions

Field	Description
15–8 SEL135	Input (XBARA_INn) to be muxed to XBARA_OUT135 (refer to Functional Description section for input/output assignment)
SEL134	Input (XBARA_INn) to be muxed to XBARA_OUT134 (refer to Functional Description section for input/output assignment)

83.5.69 Crossbar A Select Register 68 (XBARAx_SEL68)

Address: 4003_C000h base + 88h offset = 4003_C088h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL137								SEL136							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL68 field descriptions

Field	Description
15–8 SEL137	Input (XBARA_INn) to be muxed to XBARA_OUT137 (refer to Functional Description section for input/output assignment)
SEL136	Input (XBARA_INn) to be muxed to XBARA_OUT136 (refer to Functional Description section for input/output assignment)

83.5.70 Crossbar A Select Register 69 (XBARAx_SEL69)

Address: 4003_C000h base + 8Ah offset = 4003_C08Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL139								SEL138							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL69 field descriptions

Field	Description
15–8 SEL139	Input (XBARA_INn) to be muxed to XBARA_OUT139 (refer to Functional Description section for input/output assignment)
SEL138	Input (XBARA_INn) to be muxed to XBARA_OUT138 (refer to Functional Description section for input/output assignment)

83.5.71 Crossbar A Select Register 70 (XBARAx_SEL70)

Address: 4003_C000h base + 8Ch offset = 4003_C08Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL141								SEL140							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL70 field descriptions

Field	Description
15–8 SEL141	Input (XBARA_INn) to be muxed to XBARA_OUT141 (refer to Functional Description section for input/output assignment)
SEL140	Input (XBARA_INn) to be muxed to XBARA_OUT140 (refer to Functional Description section for input/output assignment)

83.5.72 Crossbar A Select Register 71 (XBARAx_SEL71)

Address: 4003_C000h base + 8Eh offset = 4003_C08Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL143								SEL142							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL71 field descriptions

Field	Description
15–8 SEL143	Input (XBARA_INn) to be muxed to XBARA_OUT143 (refer to Functional Description section for input/output assignment)
SEL142	Input (XBARA_INn) to be muxed to XBARA_OUT142 (refer to Functional Description section for input/output assignment)

83.5.73 Crossbar A Select Register 72 (XBARAx_SEL72)

Address: 4003_C000h base + 90h offset = 4003_C090h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL145								SEL144							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL72 field descriptions

Field	Description
15–8 SEL145	Input (XBARA_INn) to be muxed to XBARA_OUT145 (refer to Functional Description section for input/output assignment)
SEL144	Input (XBARA_INn) to be muxed to XBARA_OUT144 (refer to Functional Description section for input/output assignment)

83.5.74 Crossbar A Select Register 73 (XBARAx_SEL73)

Address: 4003_C000h base + 92h offset = 4003_C092h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL147								SEL146							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL73 field descriptions

Field	Description
15–8 SEL147	Input (XBARA_INn) to be muxed to XBARA_OUT147 (refer to Functional Description section for input/output assignment)
SEL146	Input (XBARA_INn) to be muxed to XBARA_OUT146 (refer to Functional Description section for input/output assignment)

83.5.75 Crossbar A Select Register 74 (XBARAx_SEL74)

Address: 4003_C000h base + 94h offset = 4003_C094h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL149								SEL148							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL74 field descriptions

Field	Description
15–8 SEL149	Input (XBARA_INn) to be muxed to XBARA_OUT149 (refer to Functional Description section for input/output assignment)
SEL148	Input (XBARA_INn) to be muxed to XBARA_OUT148 (refer to Functional Description section for input/output assignment)

83.5.76 Crossbar A Select Register 75 (XBARAx_SEL75)

Address: 4003_C000h base + 96h offset = 4003_C096h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL151								SEL150							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL75 field descriptions

Field	Description
15–8 SEL151	Input (XBARA_INn) to be muxed to XBARA_OUT151 (refer to Functional Description section for input/output assignment)
SEL150	Input (XBARA_INn) to be muxed to XBARA_OUT150 (refer to Functional Description section for input/output assignment)

83.5.77 Crossbar A Select Register 76 (XBARAx_SEL76)

Address: 4003_C000h base + 98h offset = 4003_C098h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL153								SEL152							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL76 field descriptions

Field	Description
15–8 SEL153	Input (XBARA_INn) to be muxed to XBARA_OUT153 (refer to Functional Description section for input/output assignment)
SEL152	Input (XBARA_INn) to be muxed to XBARA_OUT152 (refer to Functional Description section for input/output assignment)

83.5.78 Crossbar A Select Register 77 (XBARAx_SEL77)

Address: 4003_C000h base + 9Ah offset = 4003_C09Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL155								SEL154							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL77 field descriptions

Field	Description
15–8 SEL155	Input (XBARA_INn) to be muxed to XBARA_OUT155 (refer to Functional Description section for input/output assignment)
SEL154	Input (XBARA_INn) to be muxed to XBARA_OUT154 (refer to Functional Description section for input/output assignment)

83.5.79 Crossbar A Select Register 78 (XBARAx_SEL78)

Address: 4003_C000h base + 9Ch offset = 4003_C09Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL157								SEL156							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL78 field descriptions

Field	Description
15–8 SEL157	Input (XBARA_INn) to be muxed to XBARA_OUT157 (refer to Functional Description section for input/output assignment)
SEL156	Input (XBARA_INn) to be muxed to XBARA_OUT156 (refer to Functional Description section for input/output assignment)

83.5.80 Crossbar A Select Register 79 (XBARAx_SEL79)

Address: 4003_C000h base + 9Eh offset = 4003_C09Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL159								SEL158							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL79 field descriptions

Field	Description
15–8 SEL159	Input (XBARA_INn) to be muxed to XBARA_OUT159 (refer to Functional Description section for input/output assignment)
SEL158	Input (XBARA_INn) to be muxed to XBARA_OUT158 (refer to Functional Description section for input/output assignment)

83.5.81 Crossbar A Select Register 80 (XBARAx_SEL80)

Address: 4003_C000h base + A0h offset = 4003_C0A0h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL161								SEL160							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL80 field descriptions

Field	Description
15–8 SEL161	Input (XBARA_INn) to be muxed to XBARA_OUT161 (refer to Functional Description section for input/output assignment)
SEL160	Input (XBARA_INn) to be muxed to XBARA_OUT160 (refer to Functional Description section for input/output assignment)

83.5.82 Crossbar A Select Register 81 (XBARAx_SEL81)

Address: 4003_C000h base + A2h offset = 4003_C0A2h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL163								SEL162							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL81 field descriptions

Field	Description
15–8 SEL163	Input (XBARA_INn) to be muxed to XBARA_OUT163 (refer to Functional Description section for input/output assignment)
SEL162	Input (XBARA_INn) to be muxed to XBARA_OUT162 (refer to Functional Description section for input/output assignment)

83.5.83 Crossbar A Select Register 82 (XBARAx_SEL82)

Address: 4003_C000h base + A4h offset = 4003_C0A4h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL165								SEL164							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL82 field descriptions

Field	Description
15–8 SEL165	Input (XBARA_INn) to be muxed to XBARA_OUT165 (refer to Functional Description section for input/output assignment)
SEL164	Input (XBARA_INn) to be muxed to XBARA_OUT164 (refer to Functional Description section for input/output assignment)

83.5.84 Crossbar A Select Register 83 (XBARAx_SEL83)

Address: 4003_C000h base + A6h offset = 4003_C0A6h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL167								SEL166							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL83 field descriptions

Field	Description
15–8 SEL167	Input (XBARA_INn) to be muxed to XBARA_OUT167 (refer to Functional Description section for input/output assignment)
SEL166	Input (XBARA_INn) to be muxed to XBARA_OUT166 (refer to Functional Description section for input/output assignment)

83.5.85 Crossbar A Select Register 84 (XBARAx_SEL84)

Address: 4003_C000h base + A8h offset = 4003_C0A8h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL169								SEL168							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL84 field descriptions

Field	Description
15–8 SEL169	Input (XBARA_INn) to be muxed to XBARA_OUT169 (refer to Functional Description section for input/output assignment)
SEL168	Input (XBARA_INn) to be muxed to XBARA_OUT168 (refer to Functional Description section for input/output assignment)

83.5.86 Crossbar A Select Register 85 (XBARAx_SEL85)

Address: 4003_C000h base + AAh offset = 4003_C0AAh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL171								SEL170							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL85 field descriptions

Field	Description
15–8 SEL171	Input (XBARA_INn) to be muxed to XBARA_OUT171 (refer to Functional Description section for input/output assignment)
SEL170	Input (XBARA_INn) to be muxed to XBARA_OUT170 (refer to Functional Description section for input/output assignment)

83.5.87 Crossbar A Select Register 86 (XBARAx_SEL86)

Address: 4003_C000h base + ACh offset = 4003_C0ACh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL173								SEL172							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL86 field descriptions

Field	Description
15–8 SEL173	Input (XBARA_INn) to be muxed to XBARA_OUT173 (refer to Functional Description section for input/output assignment)
SEL172	Input (XBARA_INn) to be muxed to XBARA_OUT172 (refer to Functional Description section for input/output assignment)

83.5.88 Crossbar A Select Register 87 (XBARAx_SEL87)

Address: 4003_C000h base + AEh offset = 4003_C0AEh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SEL175								SEL174							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARAx_SEL87 field descriptions

Field	Description
15–8 SEL175	Input (XBARA_INn) to be muxed to XBARA_OUT175 (refer to Functional Description section for input/output assignment)
SEL174	Input (XBARA_INn) to be muxed to XBARA_OUT174 (refer to Functional Description section for input/output assignment)

83.5.89 Crossbar A Control Register 0 (XBARAx_CTRL0)

Use this register to configure edge detection, interrupt, and DMA features for the XBAR_OUT0 and XBAR_OUT1 outputs.

The XBAR_CTRL registers are organized similarly to the XBAR_SEL registers, with control fields for two XBAR_OUT outputs in each register. In control register 0, the LSBs contain the control fields for XBAR_OUT0, and the MSBs contain the control fields for XBAR_OUT1.

Address: 4003_C000h base + B0h offset = 4003_C0B0h

Bit	15	14	13	12	11	10	9	8
Read	0			STS1	EDGE1		IEN1	DEN1
Write				w1c				
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			STS0	EDGE0		IEN0	DEN0
Write				w1c				
Reset	0	0	0	0	0	0	0	0

XBARAx_CTRL0 field descriptions

Field	Description
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 STS1	Edge detection status for XBAR_OUT1 This bit reflects the results of edge detection for XBAR_OUT1. This field is set to 1 when an edge consistent with the current setting of EDGE1 is detected on XBAR_OUT1. This field is cleared by writing 1 to it or by a DMA_ACK1 reception when DEN1 is set. Writing 0 to the field has no effect. When interrupt or DMA functionality is enabled for XBAR_OUT1, this field is 1 when the interrupt or DMA request is asserted and 0 when the interrupt or DMA request has been cleared. 0 Active edge not yet detected on XBAR_OUT1 1 Active edge detected on XBAR_OUT1
11–10 EDGE1	Active edge for edge detection on XBAR_OUT1 This field selects which edges on XBAR_OUT1 cause STS1 to assert. 00 STS1 never asserts 01 STS1 asserts on rising edges of XBAR_OUT1 10 STS1 asserts on falling edges of XBAR_OUT1 11 STS1 asserts on rising and falling edges of XBAR_OUT1
9 IEN1	Interrupt Enable for XBAR_OUT1 This bit enables the interrupt function on the corresponding XBAR_OUT1 output. When the interrupt is enabled, the output INT_REQ1 reflects the value STS1. When the interrupt is disabled, INT_REQ1 remains low. The interrupt request is cleared by writing a 1 to STS1. **Restriction:** IEN1 and DEN1 should not both be set to 1. 0 Interrupt disabled 1 Interrupt enabled
8 DEN1	DMA Enable for XBAR_OUT1 This bit enables the DMA function on the corresponding XBAR_OUT1 output. When enabled, DMA_REQ1 presents the value STS1. When disabled, the DMA_REQ1 output remains low. **Restriction:** IEN1 and DEN1 should not both be set to 1. 0 DMA disabled 1 DMA enabled
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 STS0	Edge detection status for XBAR_OUT0 This bit reflects the results of edge detection for XBAR_OUT0. This field is set to 1 when an edge consistent with the current setting of EDGE0 is detected on XBAR_OUT0. This field is cleared by writing 1 to it or by a DMA_ACK0 reception when DEN0 is set. Writing 0 to the field has no effect. When interrupt or DMA functionality is enabled for XBAR_OUT0, this field is 1 when the interrupt or DMA request is asserted and 0 when the interrupt or DMA request has been cleared.

Table continues on the next page...

XBARAx_CTRL0 field descriptions (continued)

Field	Description
	0 Active edge not yet detected on XBAR_OUT0 1 Active edge detected on XBAR_OUT0
3–2 EDGE0	Active edge for edge detection on XBAR_OUT0 This field selects which edges on XBAR_OUT0 cause STS0 to assert. 00 STS0 never asserts 01 STS0 asserts on rising edges of XBAR_OUT0 10 STS0 asserts on falling edges of XBAR_OUT0 11 STS0 asserts on rising and falling edges of XBAR_OUT0
1 IEN0	Interrupt Enable for XBAR_OUT0 This bit enables the interrupt function on the corresponding XBAR_OUT0 output. When the interrupt is enabled, the output INT_REQ0 reflects the value STS0. When the interrupt is disabled, INT_REQ0 remains low. The interrupt request is cleared by writing a 1 to STS0. **Restriction:** IEN0 and DEN0 should not both be set to 1. 0 Interrupt disabled 1 Interrupt enabled
0 DEN0	DMA Enable for XBAR_OUT0 This bit enables the DMA function on the corresponding XBAR_OUT0 output. When enabled, DMA_REQ0 presents the value STS0. When disabled, the DMA_REQ0 output remains low. **Restriction:** IEN0 and DEN0 should not both be set to 1. 0 DMA disabled 1 DMA enabled

83.5.90 Crossbar A Control Register 1 (XBARAx_CTRL1)

Use this register to configure edge detection, interrupt, and DMA features for the XBAR_OUT2 and XBAR_OUT3 outputs.

The XBAR_CTRL registers are organized similarly to the XBAR_SEL registers, with control fields for two XBAR_OUT outputs in each register. In control register 1, the LSBs contain the control fields for XBAR_OUT2, and the MSBs contain the control fields for XBAR_OUT3.

Address: 4003_C000h base + B2h offset = 4003_C0B2h

Bit	15	14	13	12	11	10	9	8
Read	0			STS3	EDGE3		IEN3	DEN3
Write				w1c				
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			STS2	EDGE2		IEN2	DEN2
Write				w1c				
Reset	0	0	0	0	0	0	0	0

XBARAx_CTRL1 field descriptions

Field	Description
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 STS3	Edge detection status for XBAR_OUT3 This bit reflects the results of edge detection for XBAR_OUT3. This field is set to 1 when an edge consistent with the current setting of EDGE3 is detected on XBAR_OUT3. This field is cleared by writing 1 to it or by a DMA_ACK3 reception when DEN3 is set. Writing 0 to the field has no effect. When interrupt or DMA functionality is enabled for XBAR_OUT3, this field is 1 when the interrupt or DMA request is asserted and 0 when the interrupt or DMA request has been cleared. 0 Active edge not yet detected on XBAR_OUT3 1 Active edge detected on XBAR_OUT3
11–10 EDGE3	Active edge for edge detection on XBAR_OUT3 This field selects which edges on XBAR_OUT3 cause STS3 to assert. 00 STS3 never asserts 01 STS3 asserts on rising edges of XBAR_OUT3 10 STS3 asserts on falling edges of XBAR_OUT3 11 STS3 asserts on rising and falling edges of XBAR_OUT3
9 IEN3	Interrupt Enable for XBAR_OUT3 This bit enables the interrupt function on the corresponding XBAR_OUT3 output. When the interrupt is enabled, the output INT_REQ3 reflects the value STS3. When the interrupt is disabled, INT_REQ3 remains low. The interrupt request is cleared by writing a 1 to STS3. **Restriction:** IEN3 and DEN3 should not both be set to 1. 0 Interrupt disabled 1 Interrupt enabled
8 DEN3	DMA Enable for XBAR_OUT3 This bit enables the DMA function on the corresponding XBAR_OUT3 output. When enabled, DMA_REQ3 presents the value STS3. When disabled, the DMA_REQ3 output remains low. **Restriction:** IEN3 and DEN3 should not both be set to 1. 0 DMA disabled 1 DMA enabled
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 STS2	Edge detection status for XBAR_OUT2 This bit reflects the results of edge detection for XBAR_OUT2.

Table continues on the next page...

XBARAx_CTRL1 field descriptions (continued)

Field	Description
	This field is set to 1 when an edge consistent with the current setting of EDGE2 is detected on XBAR_OUT2. This field is cleared by writing 1 to it or by a DMA_ACK2 reception when DEN2 is set. Writing 0 to the field has no effect. When interrupt or DMA functionality is enabled for XBAR_OUT2, this field is 1 when the interrupt or DMA request is asserted and 0 when the interrupt or DMA request has been cleared. 0 Active edge not yet detected on XBAR_OUT2 1 Active edge detected on XBAR_OUT2
3–2 EDGE2	Active edge for edge detection on XBAR_OUT2 This field selects which edges on XBAR_OUT2 cause STS2 to assert. 00 STS2 never asserts 01 STS2 asserts on rising edges of XBAR_OUT2 10 STS2 asserts on falling edges of XBAR_OUT2 11 STS2 asserts on rising and falling edges of XBAR_OUT2
1 IEN2	Interrupt Enable for XBAR_OUT2 This bit enables the interrupt function on the corresponding XBAR_OUT2 output. When the interrupt is enabled, the output INT_REQ2 reflects the value STS2. When the interrupt is disabled, INT_REQ2 remains low. The interrupt request is cleared by writing a 1 to STS2. **Restriction:** IEN2 and DEN2 should not both be set to 1. 0 Interrupt disabled 1 Interrupt enabled
0 DEN2	DMA Enable for XBAR_OUT2 This bit enables the DMA function on the corresponding XBAR_OUT2 output. When enabled, DMA_REQ2 presents the value STS2. When disabled, the DMA_REQ2 output remains low. **Restriction:** IEN2 and DEN2 should not both be set to 1. 0 DMA disabled 1 DMA enabled

Chapter 84
Inter-Peripheral Crossbar Switch B (XBARB)

84.1 Chip-specific XBAR information

Table 84-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

On this device, for XBAR1 (see the XBARA chapter), the number of inputs is 144, and the number of outputs is 175. For XBAR2 and XBAR3 (see the XBARB chapter), the number of inputs is 98 , and the number of outputs is 16.

84.2 Overview

For a description of the general features and functionality of this module, refer to the XBARA description.

84.3 Functional Description

For a functional description of this module, refer to the XBARA functional description.

84.4 External Signals

The following table summarizes the module's external signals.

Table 84-2. Control Signal Properties

Name	I/O Type	Function	Reset State	Notes
XBAR_OUT [0:NUMOUT-1]	O	Mux Outputs with configurable width	XBAR_IN[0]	
XBAR_IN [0:NUMIN-1]	I	Mux Inputs with configurable width	*	

At reset, each output XBAR_OUT[*] contains the reset value of the signal driving XBAR_IN[0].

84.4.1 XBAR_OUT[0:NUM_OUT-1] - MUX Outputs

This is a one-dimensional array of the mux outputs. The value on each output XBAR_OUT[n] is determined by the setting of the corresponding memory mapped register SELn such that XBAR_OUT[n] = XBAR_IN[SELn].

84.4.2 XBAR_IN[0:NUM_IN-1] - MUX Inputs

This is a one-dimensional array consisting of the inputs shared by each mux. All muxes share the same inputs in the same order.

84.5 Memory Map and Register Descriptions

This section provides information about the XBARB instance of the inter-peripheral crossbar switch.

This XBAR module has only select registers.

In the XBAR select registers, the SELn fields select which of the shared inputs (XBAR_IN[*]) is muxed to each mux output (XBAR_OUT[*]). There is one SELn field per mux and therefore one per XBAR_OUT output. Crossbar output XBAR_OUT[n]

presents the value of XBAR_IN[SELn]. Each select register contains two SELn fields. In the first select register, the LSBs contain the select field for mux 0, and the MSBs contain the select field for mux 1. The pattern repeats in subsequent select registers.

The actual signals connected to XBAR_IN and XBAR_OUT are application specific and are described in the Interrupts, DMA Events, and XBAR Assignments chapter details.

XBARB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Crossbar B Select Register 0 (XBARB2_SEL0)	16	R/W	0000h	84.5.1/6065
4004_0002	Crossbar B Select Register 1 (XBARB2_SEL1)	16	R/W	0000h	84.5.2/6066
4004_0004	Crossbar B Select Register 2 (XBARB2_SEL2)	16	R/W	0000h	84.5.3/6066
4004_0006	Crossbar B Select Register 3 (XBARB2_SEL3)	16	R/W	0000h	84.5.4/6067
4004_0008	Crossbar B Select Register 4 (XBARB2_SEL4)	16	R/W	0000h	84.5.5/6067
4004_000A	Crossbar B Select Register 5 (XBARB2_SEL5)	16	R/W	0000h	84.5.6/6068
4004_000C	Crossbar B Select Register 6 (XBARB2_SEL6)	16	R/W	0000h	84.5.7/6068
4004_000E	Crossbar B Select Register 7 (XBARB2_SEL7)	16	R/W	0000h	84.5.8/6069
4004_4000	Crossbar B Select Register 0 (XBARB3_SEL0)	16	R/W	0000h	84.5.1/6065
4004_4002	Crossbar B Select Register 1 (XBARB3_SEL1)	16	R/W	0000h	84.5.2/6066
4004_4004	Crossbar B Select Register 2 (XBARB3_SEL2)	16	R/W	0000h	84.5.3/6066
4004_4006	Crossbar B Select Register 3 (XBARB3_SEL3)	16	R/W	0000h	84.5.4/6067
4004_4008	Crossbar B Select Register 4 (XBARB3_SEL4)	16	R/W	0000h	84.5.5/6067
4004_400A	Crossbar B Select Register 5 (XBARB3_SEL5)	16	R/W	0000h	84.5.6/6068
4004_400C	Crossbar B Select Register 6 (XBARB3_SEL6)	16	R/W	0000h	84.5.7/6068
4004_400E	Crossbar B Select Register 7 (XBARB3_SEL7)	16	R/W	0000h	84.5.8/6069

84.5.1 Crossbar B Select Register 0 (XBARBx_SEL0)

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL1							0	SEL0						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL0 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL1	Input (XBARB_INn) to be muxed to XBARB_OUT1 (refer to Functional Description section for input/output assignment)

Table continues on the next page...

XBARBx_SEL0 field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL0	Input (XBARB_INn) to be muxed to XBARB_OUT0 (refer to Functional Description section for input/output assignment)

84.5.2 Crossbar B Select Register 1 (XBARBx_SEL1)

Address: Base address + 2h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL3							0	SEL2						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL1 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL3	Input (XBARB_INn) to be muxed to XBARB_OUT3 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL2	Input (XBARB_INn) to be muxed to XBARB_OUT2 (refer to Functional Description section for input/output assignment)

84.5.3 Crossbar B Select Register 2 (XBARBx_SEL2)

Address: Base address + 4h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL5							0	SEL4						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL2 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL5	Input (XBARB_INn) to be muxed to XBARB_OUT5 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

XBARBx_SEL2 field descriptions (continued)

Field	Description
SEL4	Input (XBARB_INn) to be muxed to XBARB_OUT4 (refer to Functional Description section for input/output assignment)

84.5.4 Crossbar B Select Register 3 (XBARBx_SEL3)

Address: Base address + 6h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL7							0	SEL6						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL3 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL7	Input (XBARB_INn) to be muxed to XBARB_OUT7 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL6	Input (XBARB_INn) to be muxed to XBARB_OUT6 (refer to Functional Description section for input/output assignment)

84.5.5 Crossbar B Select Register 4 (XBARBx_SEL4)

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL9							0	SEL8						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL4 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL9	Input (XBARB_INn) to be muxed to XBARB_OUT9 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL8	Input (XBARB_INn) to be muxed to XBARB_OUT8 (refer to Functional Description section for input/output assignment)

84.5.6 Crossbar B Select Register 5 (XBARBx_SEL5)

Address: Base address + Ah offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL11							0	SEL10						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL5 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL11	Input (XBARB_INn) to be muxed to XBARB_OUT11 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL10	Input (XBARB_INn) to be muxed to XBARB_OUT10 (refer to Functional Description section for input/output assignment)

84.5.7 Crossbar B Select Register 6 (XBARBx_SEL6)

Address: Base address + Ch offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL13							0	SEL12						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL6 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL13	Input (XBARB_INn) to be muxed to XBARB_OUT13 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL12	Input (XBARB_INn) to be muxed to XBARB_OUT12 (refer to Functional Description section for input/output assignment)

84.5.8 Crossbar B Select Register 7 (XBARBx_SEL7)

Address: Base address + Eh offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	SEL15							0	SEL14						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XBARBx_SEL7 field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 SEL15	Input (XBARB_INn) to be muxed to XBARB_OUT15 (refer to Functional Description section for input/output assignment)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SEL14	Input (XBARB_INn) to be muxed to XBARB_OUT14 (refer to Functional Description section for input/output assignment)

Chapter 85
Analog Overview

85.1 Overview

The following analog modules are integrated in this chip:

- Analog-Digital-Converter (ADC) - Supports up to 4.2MS/s sampling rate and up to 20 external analog inputs
- ADC External Trigger Control (ADC_ETC) - Enables multiple users to share a ADC module in a Time-Division-Multiplexing (TDM) method
- Comparator (CMP) - Operational over the entire supply range and features a integrated 8-bit DAC and a integrated 8 to 1 channel mux
- Digital-Analog-Converter (DAC) - 12-bit resolution with on-chip programmable reference generator output

Below is the simplified block diagram:

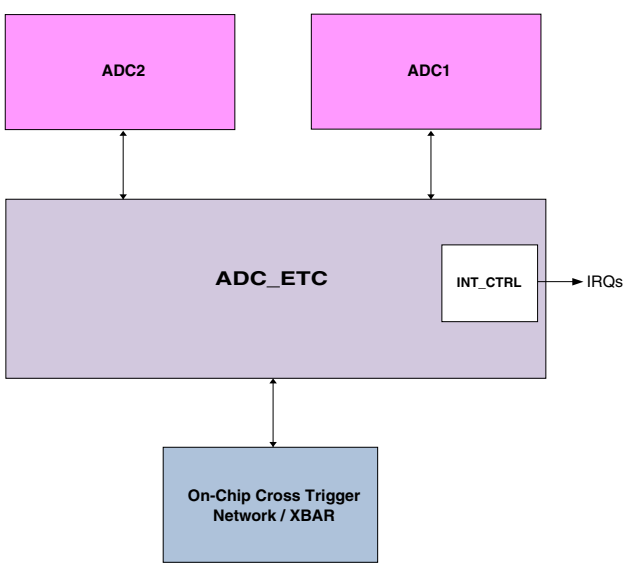


Figure 85-1. Block Diagram (Simplified)

85.1.1 Analog-to-Digital Converter (ADC)

The ADC module features a linear successive approximation algorithm with up to 12-bit singled-ended/13-bit differential resolution. ADC allows use of up to 8 trigger sources with priority level configuration, software or hardware trigger option for each, and 15 command buffers with independent options selection and channel sequence scanning. ADC also allows automatic compare function on results (less-than, greater-than, within range, or out of range). Conversion results are buffered in 16-entry FIFO with configurable watermark and overflow detection.

85.1.2 ADC External Trigger Control (ADC_ETC)

The ADC_ETC module enables multiple trigger sources to share the ADC modules in a TIME-Division-Multiplexing (TDM) way. It manages the external trigger mode (HWT) of the ADCs and supports up to x8 HWT for each ADC. It is capable of triggering dual ADCs in SyncMode (controlled by the same trigger source) or AsyncMode (controlled by separate trigger source). Every ADC HWT can be configured with a Back-to-Back (B2B) and Interrupt Enable (IE) settings and each external trigger can be configured with a fixed priority.

85.1.3 Comparator (CMP)

The analog comparator (CMP) allows for comparing two analog input voltages while the Analog MUX allows for selecting an analog input signal from eight channels. The CMP is operational across the full range of the supply voltage. The integrated 8-bit DAC provides a selectable voltage reference for applications where voltage reference is needed.

The CMPs share the same 1.8V reference voltage with the ADCs.

85.1.4 Digital-Analog Converter (DAC)

The DAC module is 12-bit resolution with on-chip programmable reference generator output. The voltage output range is from 1/4096 Vin to Vin, and the step is 1/4096 Vin, where Vin is the input voltage from ADC_VREFH.

Chapter 86
Analog Comparator (ACMP)

86.1 Chip-specific CMP information

Table 86-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

The CMP module clock inputs for the bus clock and fast clock can be found in the Clock Control Module (CCM) chapter. The slow clock is the OSC_32K clock, and the round robin clock is not used on this chip. The DAC module clock input is the bus clock, ipg_clk.

NOTE

Vin1 is not used and tied to ground on this chip. Vin2 is from VDDA_1P8_IN.

NOTE

Continuous mode is not supported on RT1170.

The following table displays input signal source selections.

Table 86-2. Input signal source selections

		ACMP1	ACMP2	ACMP3	ACMP4
CMP Inputs	IN0	Reserved	Reserved	Reserved	Reserved

Table continues on the next page...

Table 86-2. Input signal source selections (continued)

		ACMP1	ACMP2	ACMP3	ACMP4
	IN1	GPIO_AD_00	GPIO_AD_04	GPIO_AD_28	GPIO_AD_32
	IN2	GPIO_AD_01	GPIO_AD_05	GPIO_AD_29	GPIO_AD_33
	IN3	GPIO_AD_02	GPIO_AD_26	GPIO_AD_30	GPIO_AD_34
	IN4	GPIO_AD_03	GPIO_AD_27	GPIO_AD_31	GPIO_AD_35
	IN5	Reserved	Reserved	Reserved	Reserved
	IN6	Reserved	Reserved	Reserved	Reserved
	IN7	8b DAC output	8b DAC output	8b DAC output	8b DAC output
CMP Output		GPIO_AD_17	GPIO_AD_18	GPIO_AD_19	GPIO_AD_20

Special considerations to use the ACMP for wakeup from system STANDBY mode

In order to use the ACMP module as a wakeup source when the chip is in system STANDBY mode, the clock for the ACMP must be configured to be OFF for the corresponding core mode that will be used with the system STANDBY request. The following table shows which combinations can be used for a STANDBY mode wakeup:

Table 86-3. STANDBY mode wakeup combinations

Clock Gate Level	WAIT STBY	STOP STBY	SUSPEND STBY
Clock needed in Run mode	N	Y	Y
Clock needed in Run and Wait	N	Y	Y
Clock needed in Run, Wait, and Stop	N	N	Y
Clock on in any mode	N	N	N

86.2 Overview

The comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 8-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The DAC is a 256-tap resistor ladder network that provides a selectable voltage reference for applications requiring a voltage reference. The 256-tap resistor ladder network divides the supply reference V_{in} into 256 voltage levels. A 8-bit digital register input selects the output voltage level, which varies from V_{in} to $V_{in}/256$. V_{in} can be selected from two voltage sources, V_{in1} or V_{in2}.

86.2.1 CMP, DAC, and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

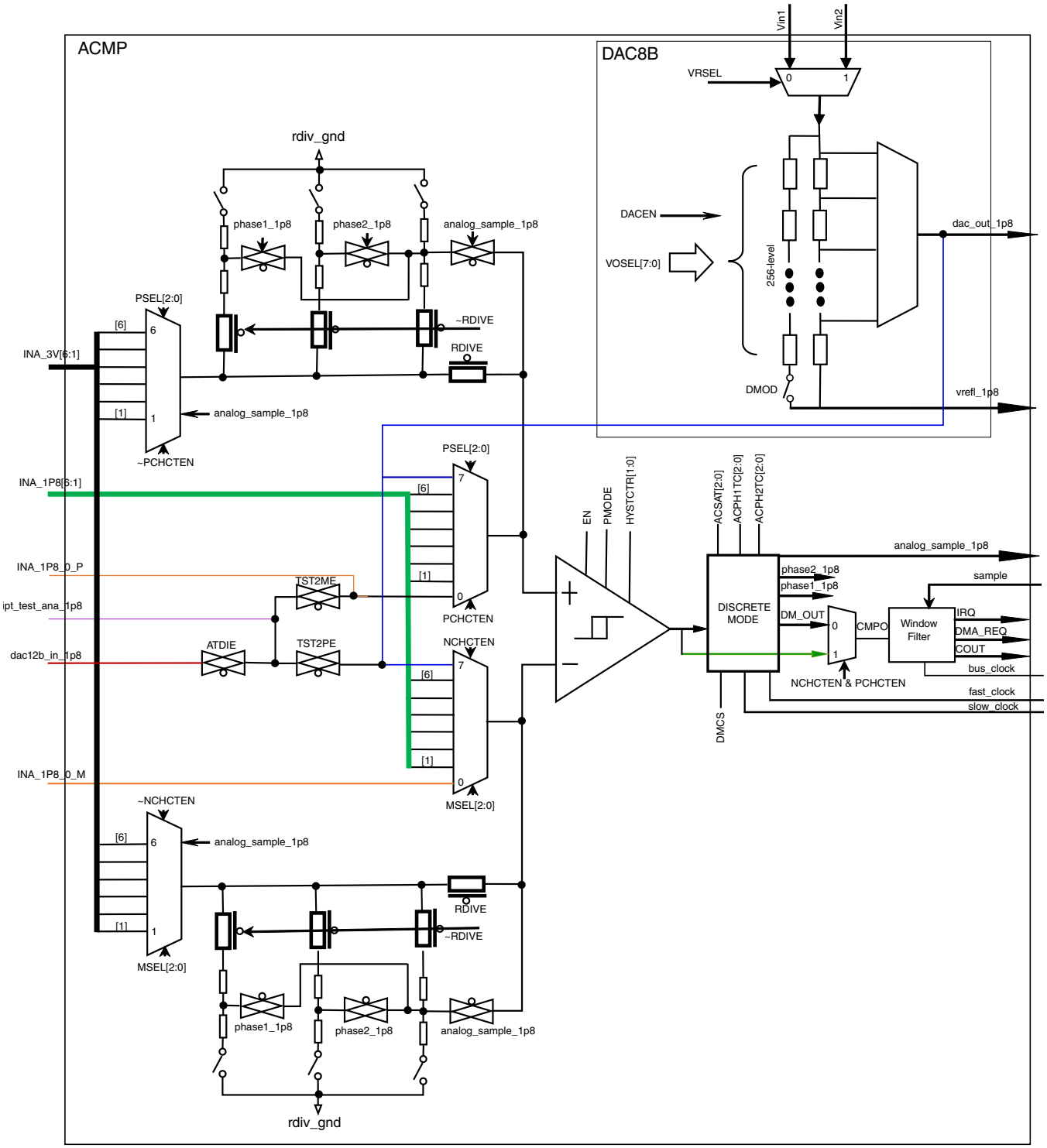


Figure 86-1. CMP high level diagram

86.2.2 CMP block diagram

The following figure shows the block diagram for the CMP module.

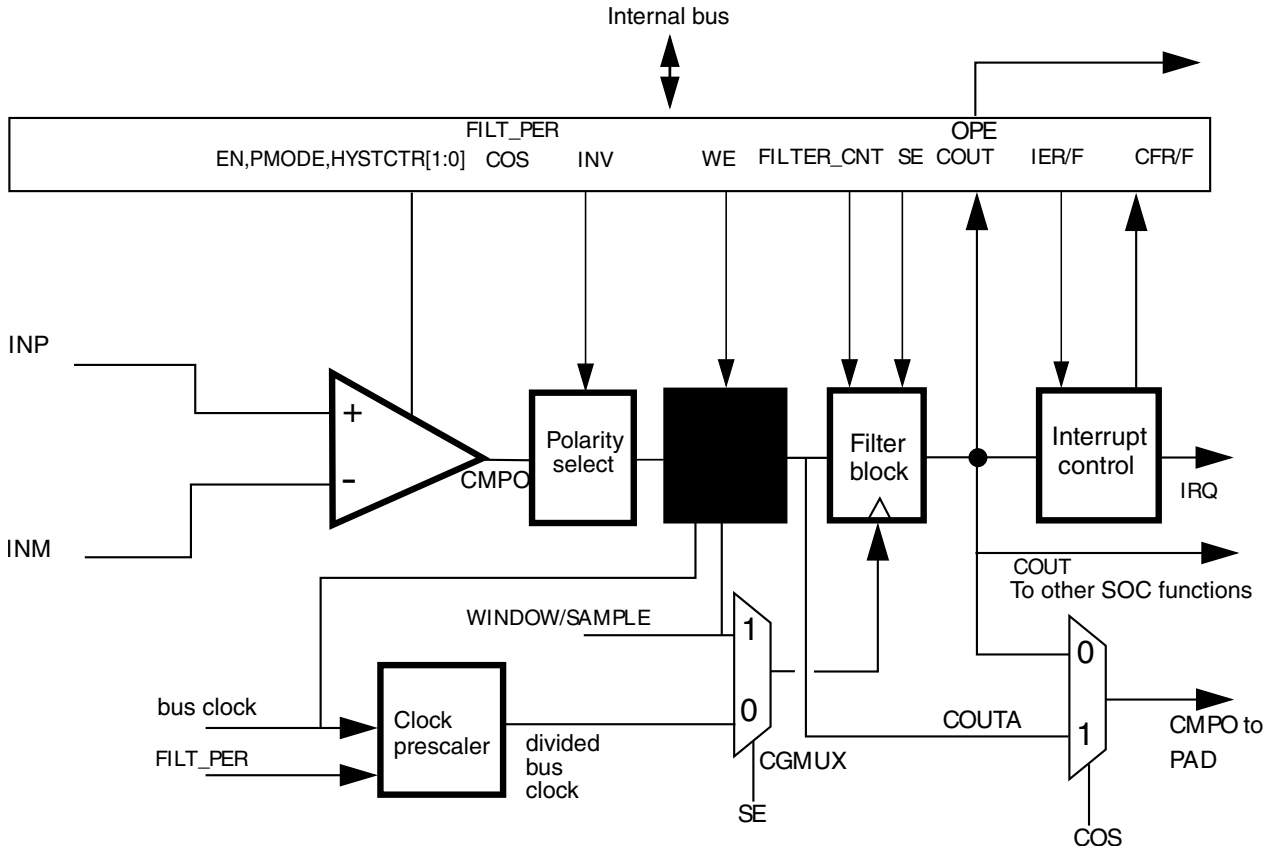


Figure 86-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when C0[WE] = 0.
- If C0[WE] = 1, the comparator output is sampled on every bus clock when WINDOW=1 to generate COUTA. Sampling does NOT occur when WINDOW = 0.
- The Filter block is bypassed when not in use.

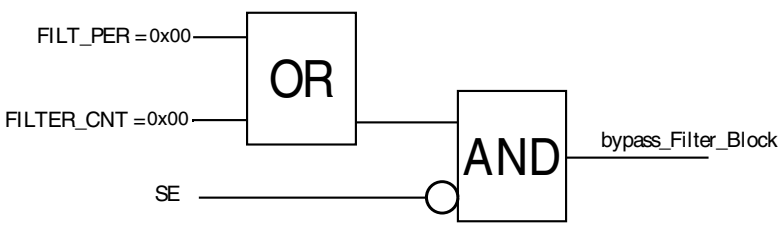


Figure 86-3. Filter block bypass logic

- The Filter block acts as a simple sampler if the filter is bypassed and C0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and C0[FILTER_CNT] is set greater than 0x01.

- If C0[SE] = 1, the external SAMPLE input is used as the sampling clock.
- IF C0[SE] = 0, the divided bus clock is used as the sampling clock.

- If enabled, the Filter block will incur up to one bus clock additional latency penalty on COUT due to the fact that COUT, which crosses clock domain boundaries, must be resynchronized to the bus clock.
- C0[WE] and C0[SE] are mutually exclusive.
- If enabled, the filter clock and the sample period must be at least 4 times slower than the system clock to the comparator.

86.2.3 Features

The following subsections list the features of the CMP, the 8-bit DAC, and the ANMUX.

86.2.3.1 CMP features

The CMP has the following features:

- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control
- Selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output
- Selectable inversion on comparator output
- Capability to produce a wide range of outputs such as:
  - Sampled
  - Windowed, which is ideal for certain PWM zero-crossing-detection applications
  - Digitally filtered:
    - Filter can be bypassed
    - Can be clocked via external SAMPLE signal or scaled bus clock
- External hysteresis can be used at the same time that the output filter is used for internal functions

- Two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with longer propagation delay
- DMA transfer support
  - A comparison event can be selected to trigger a DMA transfer
- Functional in all MCU power modes
- The window and filter functions are not available in STOP modes
- The comparator can be triggered by other peripherals

86.2.3.2 8-bit DAC key features

The 8-bit DAC has the following features:

- 8-bit resolution
- Selectable supply reference source
- Configurable low power mode or high speed mode
- Power Down mode to conserve power when not in use
- Option to route the output of 8-bit DAC to the internal comparator input, where it can be selected as one of the CMP inputs as reference for comparison

86.2.3.3 ANMUX key features

The ANMUX has the following features:

- Two 8-to-1 channel MUXes for selection of the source for comparator input
- Operational over the entire supply range

86.3 CMP functional description

The CMP module can be used to compare two analog input voltages applied to INP and INM. CMPO is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. This signal can be selectively inverted by setting CMP_C0[INVT] = 1.

CMP_C0[IER] and CMP_C0[IEF] are used to select the condition that causes the CMP module to assert an interrupt to the processor. CMP_C0[CFF] is set on a falling edge, and CMP_C0[CFR] is set on a rising edge of the comparator output. The optionally filtered CMPO can be read directly through CMP_C0[COUT].

86.3.1 CMP functional modes

There are three main sub-blocks to the CMP module:

- The comparator itself
- The window function
- The filter function

The filter, CMP_C0[FILTER_CNT], can be clocked from an internal or external clock source. The filter is programmable with respect to the number of samples that must agree before a change in the output is registered. In the simplest case, only one sample must agree. In this case, the filter acts as a simple sampler.

The external sample input is enabled using CMP_C0[SE]. When set, the output of the comparator is sampled only on rising edges of the sample input.

The "windowing mode" is enabled by setting CMP_C0[WE]. When set, the comparator output is sampled only when WINDOW=1. This feature can be used to ignore the comparator output during time periods in which the input voltages are not valid. This is especially useful when implementing zero-crossing-detection for certain PWM applications.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

Table 86-4. Comparator sample/filter controls

Mode #	CMP_C0[EN]	CMP_C0[WE]	CMP_C0[SE]	CMP_C0[FILTER_CNT]	CMP_C0[FPR]	Operation
1	0	X	X	X	X	**Disabled** See the Disabled mode (# 1).
2A	1	0	0	0x00	X	**Continuous Mode** See the Continuous mode (#s 2A & 2B).
2B	1	0	0	X	0x00	
3A	1	0	1	0x01	X	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3A & 3B).
3B	1	0	0	0x01	> 0x00	
4A	1	0	1	> 0x01	X	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4A & 4B).
4B	1	0	0	> 0x01	> 0x00	
5A	1	1	0	0x00	X	**Windowed mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA. See the Windowed mode (#s 5A & 5B).
5B	1	1	0	X	0x00	
6	1	1	0	0x01	0x01–0xFF	**Windowed/Resampled mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled on an interval determined by CMP_C0[FPR] to generate COUT. See the Windowed/Resampled mode (# 6).
7	1	1	0	> 0x01	0x01–0xFF	**Windowed/Filtered mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled and filtered to generate COUT. See the Windowed/Filtered mode (#7).
8	1	1	1	X	X	**Trigger mode** In RUN mode Comparator is configured to the modes same with these Modes #5A, #5B, #6 and #7 as elaborated in above rows depended on the CMP_C0[FILT_CNT] and CMP_C0[FPR] settings. In STOP mode, the comparator and possible the 8bit DAC is triggered work

Table continues on the next page...

Table 86-4. Comparator sample/filter controls (continued)

Mode #	CMP_C0[EN]	CMP_C0[WE]	CMP_C0[SE]	CMP_C0[FILTER_CNT]	CMP_C0[FPR]	Operation
						periodically and the active channels follow the round robin cycling scheme. See the Trigger mode.
All other combinations of CMP_C0[EN], CMP_C0[WE], CMP_C0[SE], CMP_C0[FILTER_CNT], and CMP_C0[FPR] are illegal.						

For cases where a comparator is used to drive a fault input, for example, for a motor-control module such as FTM or FlexPWM, it must be configured to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

Note

Filtering and sampling settings must be changed only after setting CMP_C0[SE]=0, CMP_C0[FPR] =0 and CMP_C0[FILTER_CNT]=0x00. This resets the filter to a known state.

86.3.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

86.3.1.2 Continuous mode (#s 2A & 2B)

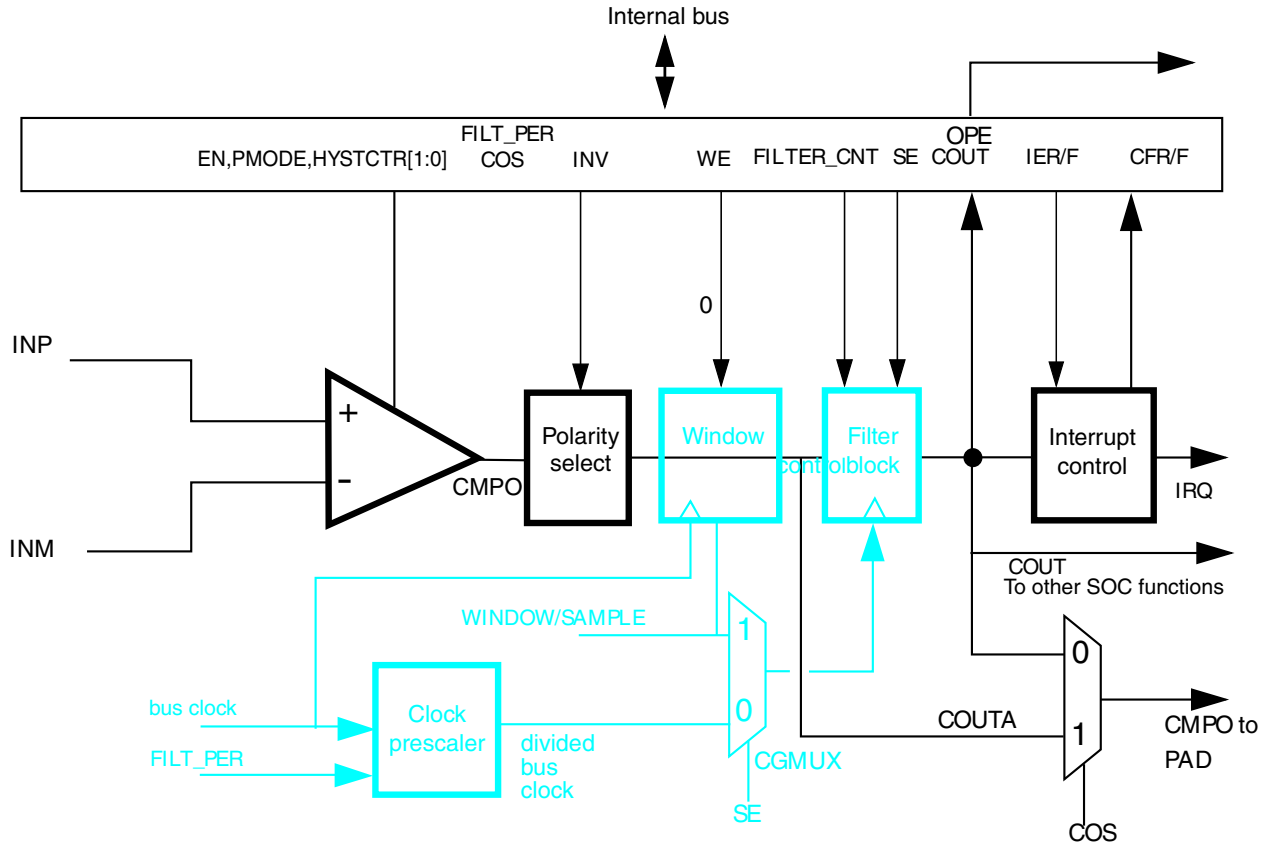


Figure 86-4. Comparator operation in Continuous mode

NOTE

See the chip configuration section for the source of sample/ window input.

The analog comparator block is powered and active. CMPO may be optionally inverted, but is not subject to external sampling or filtering. Both window control and filter blocks are completely bypassed. CMP_C0[COUT] is updated continuously. The path from comparator input pins to output pin is operating in combinational unclocked mode. COUT and COUTA are identical.

For control configurations that result in disabling the filter block, seeFigure 86-3 .

86.3.1.3 Sampled, Non-Filtered mode (#s 3A & 3B)

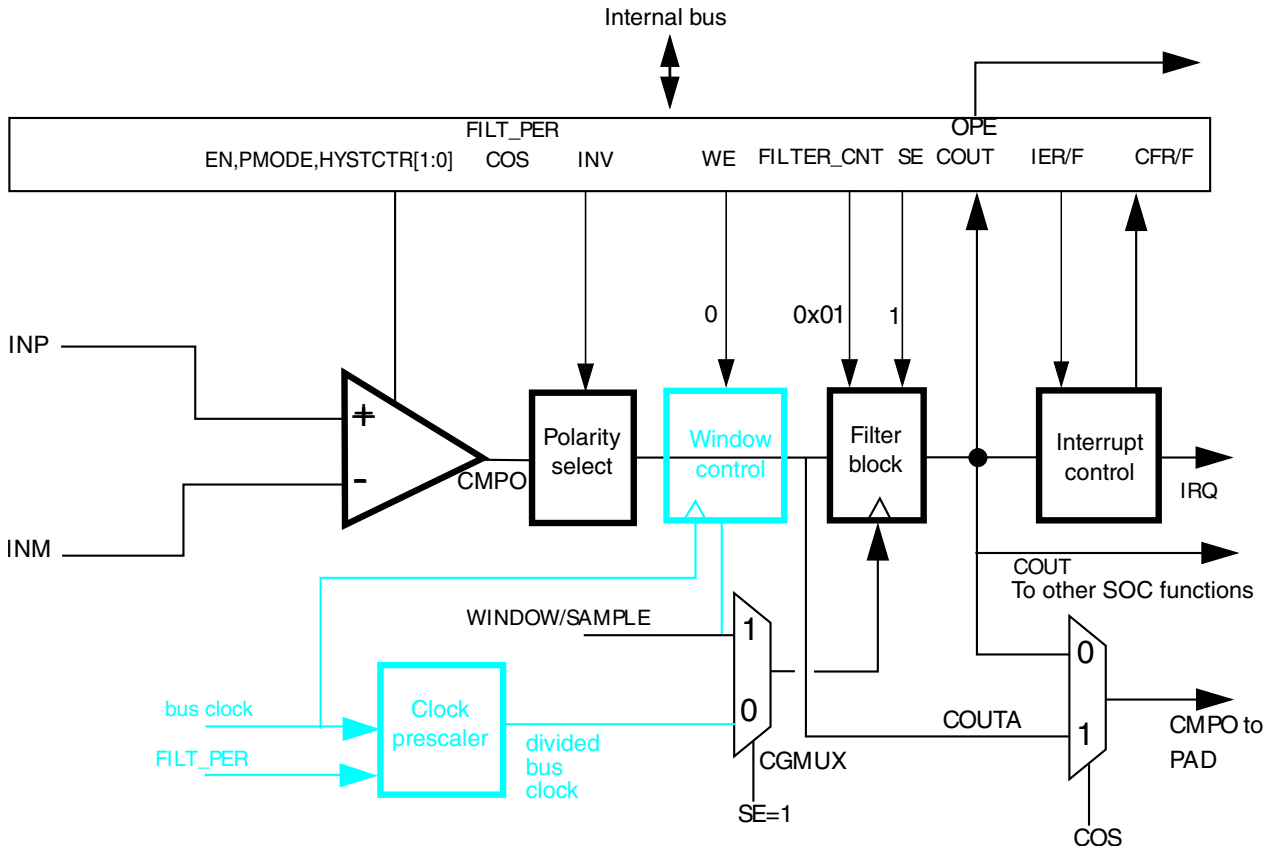


Figure 86-5. Sampled, Non-Filtered (# 3A): sampling point externally driven

In Sampled, Non-Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Non-Filtered (# 3B) is in how the clock to the filter block is derived. In #3A, the clock to filter block is externally derived while in #3B, the clock to filter block is internally derived.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

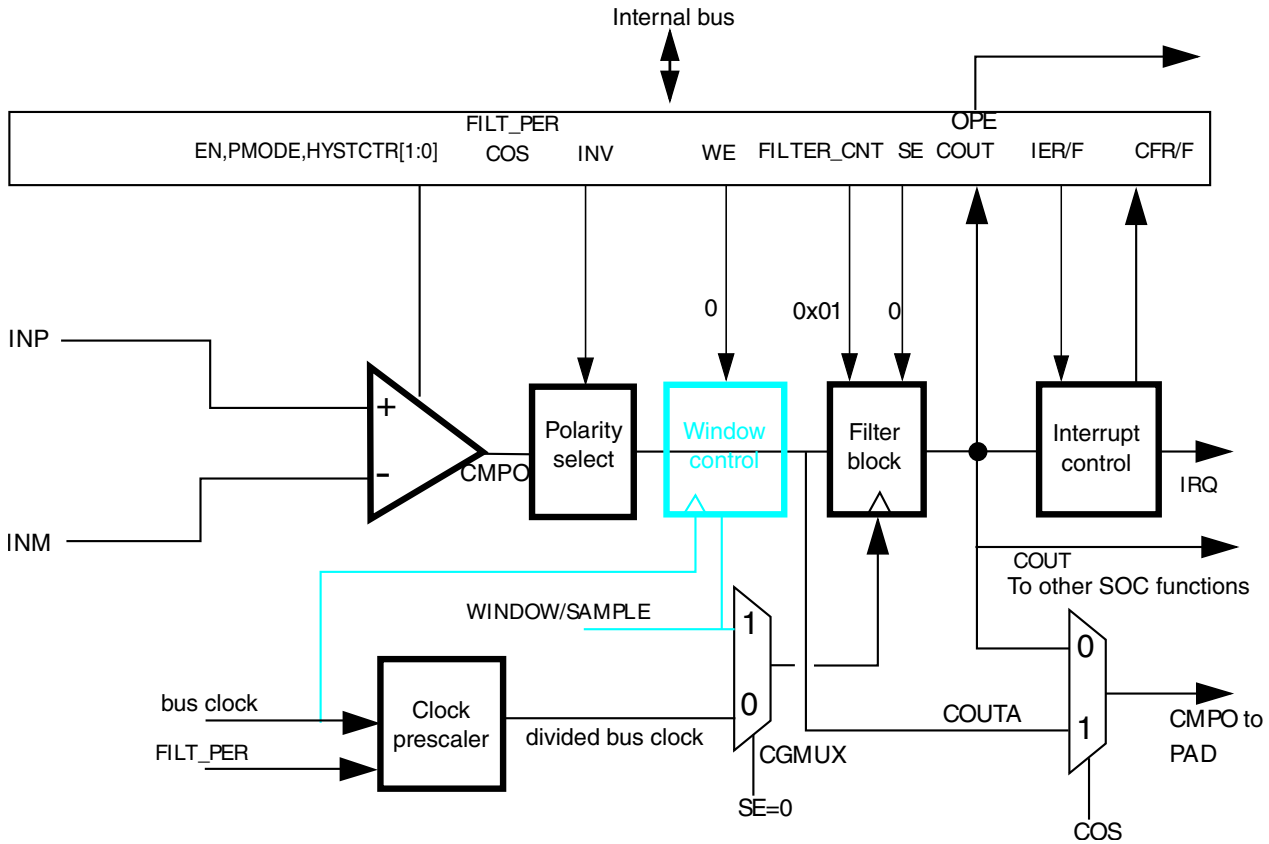


Figure 86-6. Sampled, Non-Filtered (# 3B): sampling interval internally derived

86.3.1.4 Sampled, Filtered mode (#s 4A & 4B)

In Sampled, Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Filtered (# 4A) is that, now, CMP_C0[FILTER_CNT]>1, which activates filter operation.

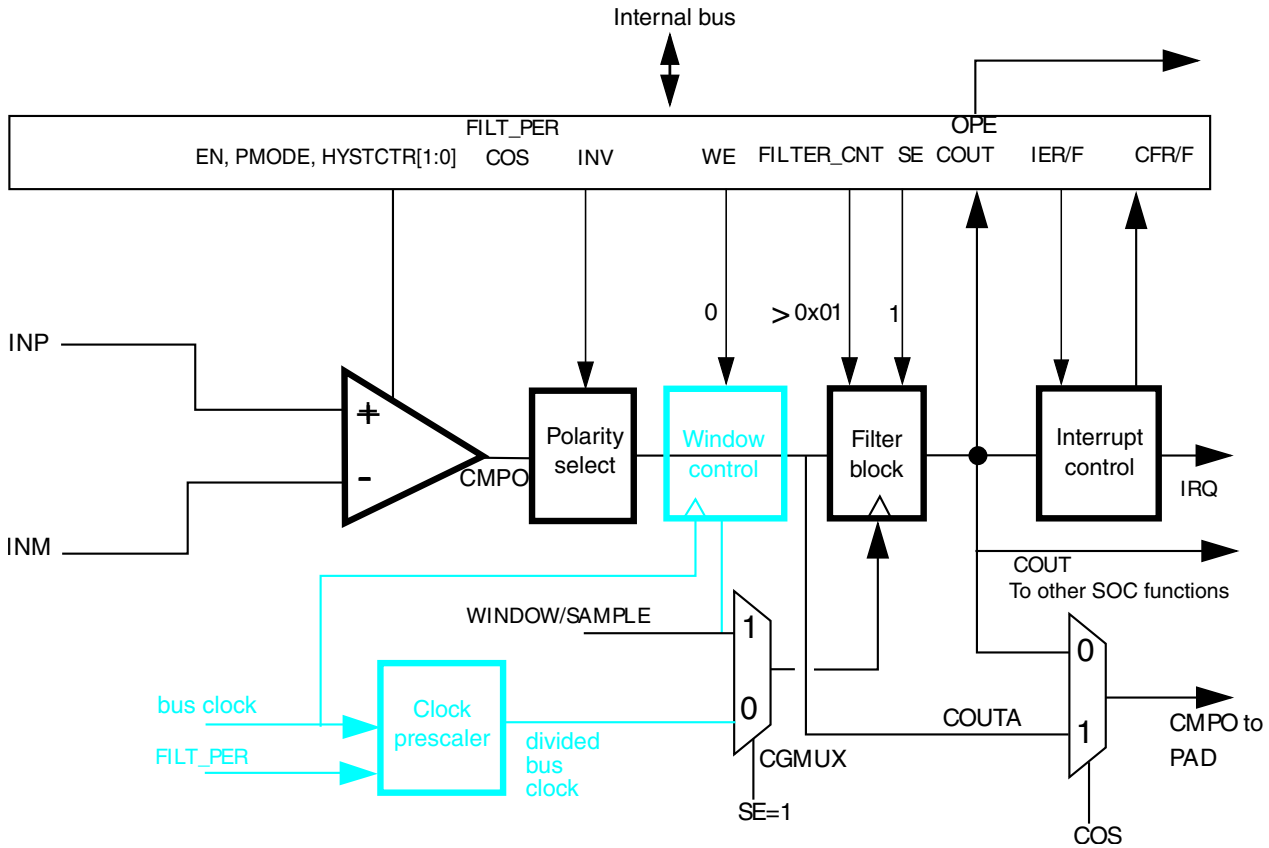


Figure 86-7. Sampled, Filtered (# 4A): sampling point externally driven

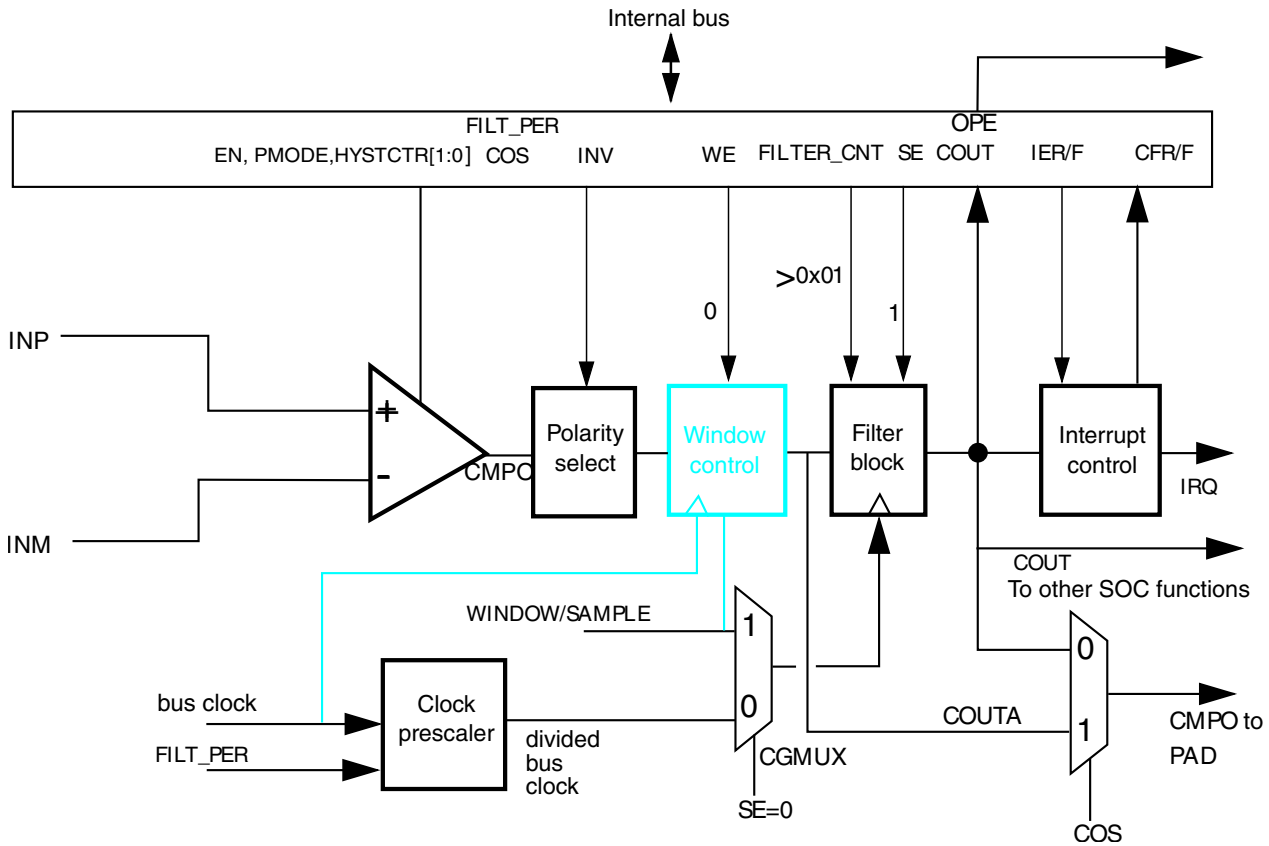


Figure 86-8. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, CMP_C0[FILTER_CNT]>1, which activates filter operation.

86.3.1.5 Windowed mode (#s 5A & 5B)

The following figure illustrates comparator operation in the Windowed mode, ignoring latency of the analog comparator, polarity select, and window control block. It also assumes that the polarity select is set to non-inverting state.

NOTE

The analog comparator output is passed to COUTA only when the WINDOW signal is high.

In actual operation, COUTA may lag the analog inputs by up to one bus clock cycle plus the combinational path delay through the comparator and polarity select logic.

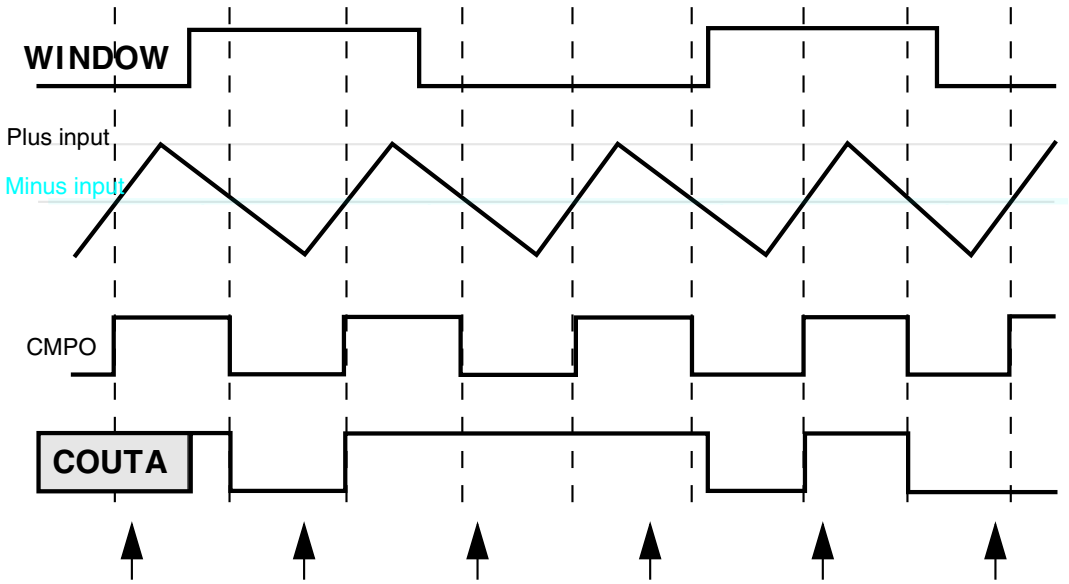


Figure 86-9. Windowed mode timing diagram

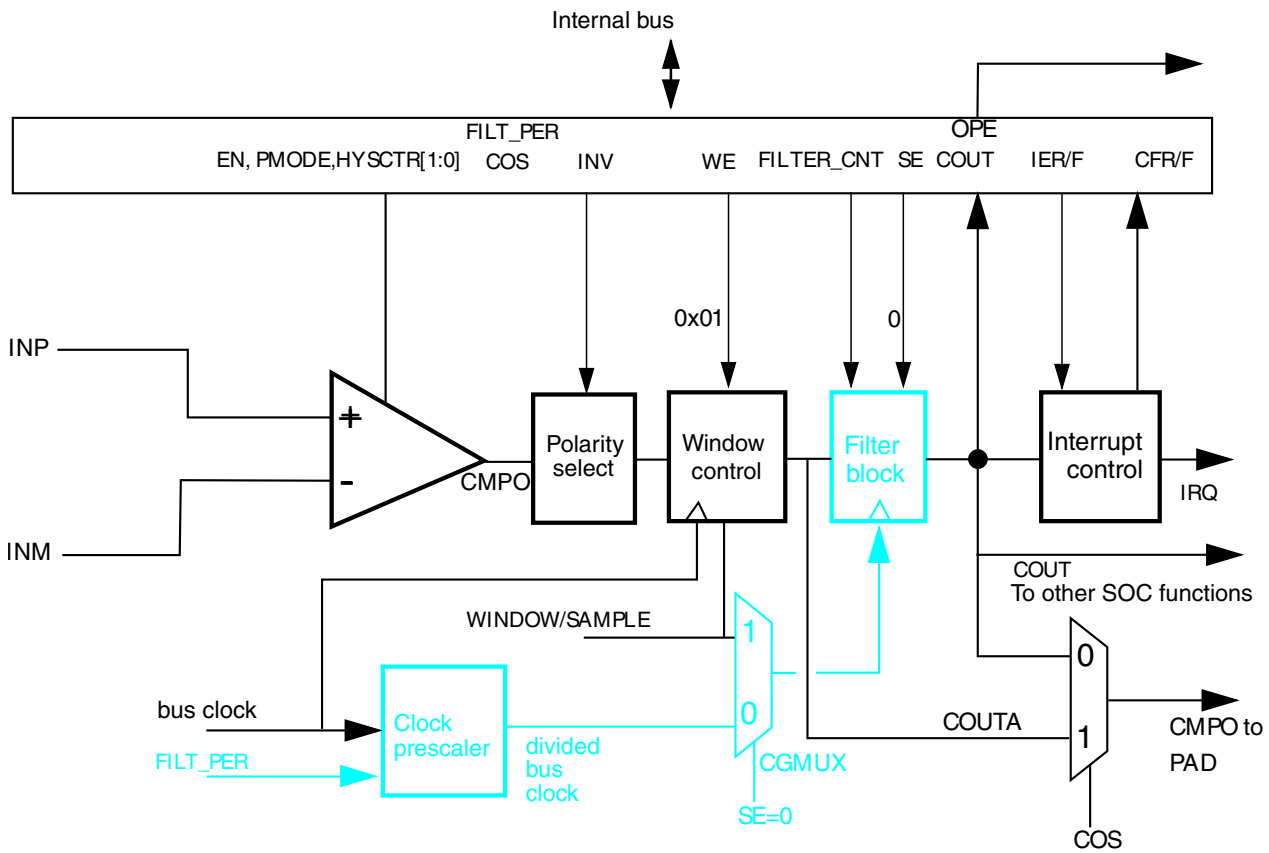


Figure 86-10. Windowed mode

For control configurations which result in disabling the filter block, seeFigure 86-3 .

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

86.3.1.6 Windowed/Resampled mode (# 6)

The following figure uses the same input stimulus shown in Figure 86-9, and adds resampling of COUTA to generate COUT. Samples are taken at the time points indicated by the arrows in the figure. Again, prop delays and latency are ignored for the sake of clarity.

This example was generated solely to demonstrate operation of the comparator in windowed/resampled mode, and does not reflect any specific application. Depending upon the sampling rate and window placement, COUT may not see zero-crossing events detected by the analog comparator. Sampling period and/or window placement must be carefully considered for a given application.

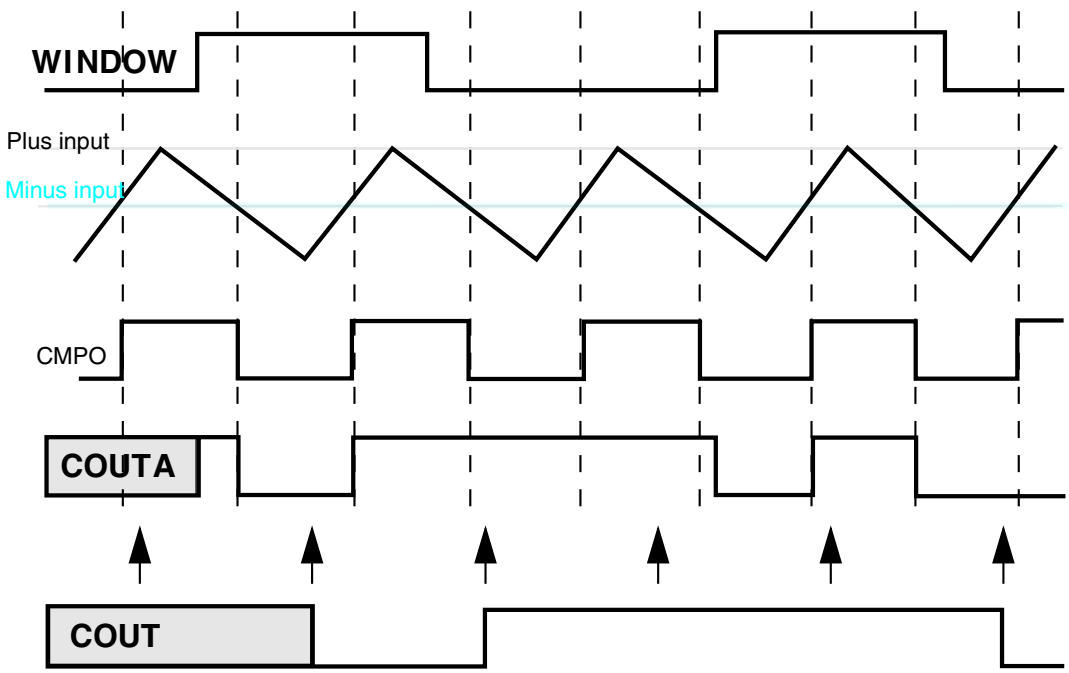


Figure 86-11. Windowed/resampled mode operation

This mode of operation results in an unfiltered string of comparator samples where the interval between the samples is determined by FPR[FILT_PER] and the bus clock rate. Configuration for this mode is virtually identical to that for the Windowed/Filtered Mode shown in the next section. The only difference is that the value of C0[FILTER_CNT] must be 1.

86.3.1.7 Windowed/Filtered mode (#7)

This is the most complex mode of operation for the comparator block, as it uses both windowing and filtering features. It also has the highest latency of any of the modes. This can be approximated: up to 1 bus clock synchronization in the window function + ((C0[FILTER_CNT] * C0[FPR]) + 1) * bus clock for the filter function.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

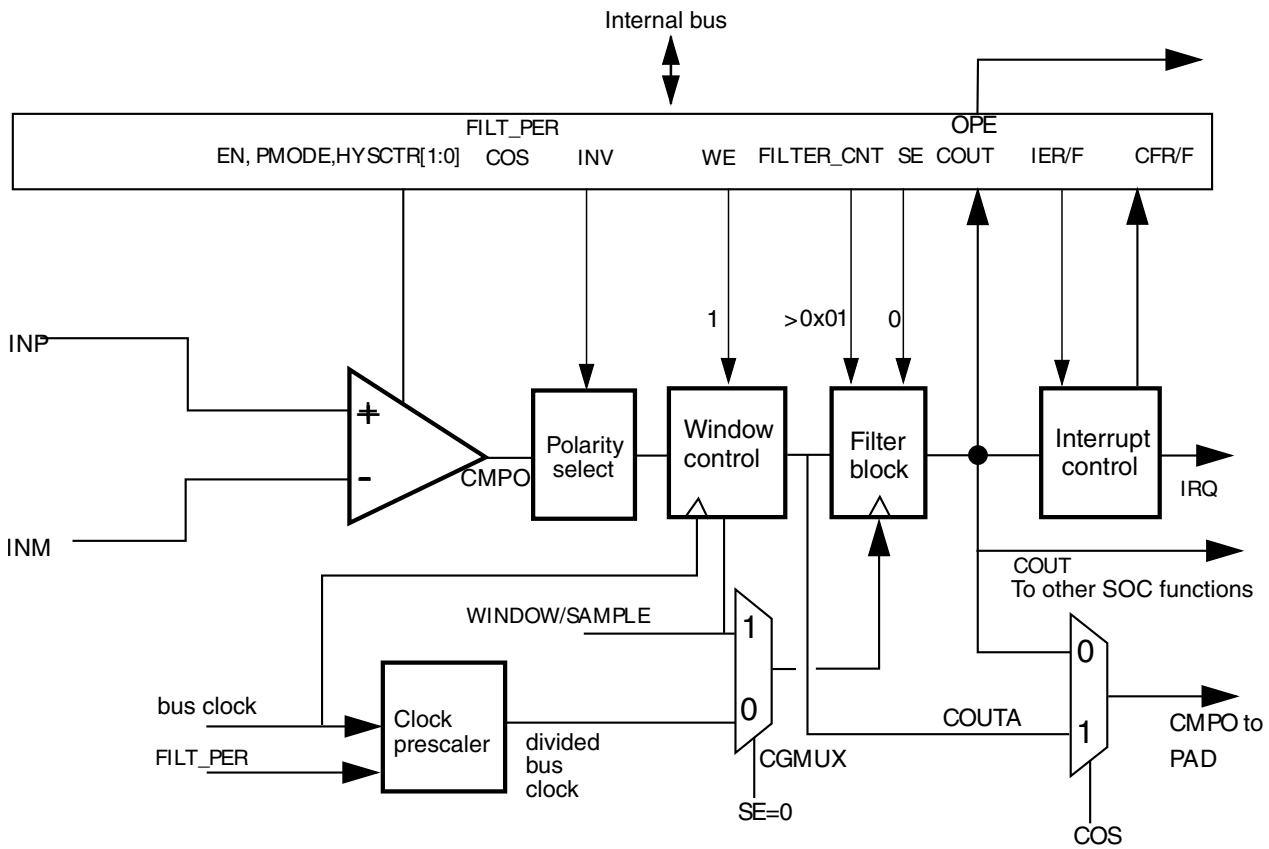


Figure 86-12. Windowed/Filtered mode

86.3.2 Low-pass filter

The low-pass filter operates on the unfiltered and unsynchronized and optionally inverted comparator output COUTA and generates the filtered and synchronized output COUT. Both COUTA and COUT can be configured as module outputs and are used for different purposes within the system.

Synchronization and edge detection are always used to determine status register bit values. They also apply to COUT for all sampling and windowed modes. Filtering can be performed using an internal timebase defined by CMP_C0[FPR], or using an external SAMPLE input to determine sample time.

The need for digital filtering and the amount of filtering is dependent on user requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, high-frequency oscillations can be generated at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

86.3.2.1 Enabling filter modes

Filter modes can be enabled by:

- Setting CMP_C0[FILTER_CNT] > 0x01 and
- Setting CMP_C0[FPR] to a nonzero value or setting C0[SE]=1

If using the divided bus clock to drive the filter, it samples COUTA every CMP_C0[FPR] bus clock cycles.

The filter output is at logic 0 when first initialized, and subsequently changes when all the consecutive CMP_C0[FILTER_CNT] samples agree that the output value has changed. In other words, CMP_C0[COUT] is 0 for some initial period, even when COUTA is at logic 1.

Setting all of CMP_C0[SE], CMP_C0[FPR] and CMP_C0[FILTER_CNT] to 0 disables the filter and eliminates switching current associated with the filtering process.

Note

Always switch to this setting prior to making any changes in filter parameters. This resets the filter to a known state. Switching CMP_C0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

If CMP_C0[SE]=1, the filter samples COUTA on each positive transition of the sample input. The output state of the filter changes when all the consecutive CMP_C0[FILTER_CNT] samples agree that the output value has changed.

86.3.2.2 Latency issues

The value of CMP_C0[FPR] or SAMPLE period must be set such that the sampling period is just longer than the period of the expected noise. This way a noise spike will corrupt only one sample. The value of CMP_C0[FILTER_CNT] must be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CMP_C0[FILTER_CNT].

The values of CMP_C0[FPR] or SAMPLE period and CMP_C0[FILTER_CNT] must also be traded off against the desire for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CMP_C0[FILTER_CNT].

The following table summarizes maximum latency values for the various modes of operation *in the absence of noise*. Filtering latency is restarted each time an actual output transition is masked by noise.

Table 86-5. Comparator sample/filter maximum latencies

Mode #	CMP_C0[EN]	CMP_C0[WE]	CMP_C0[SE]	CMP_C0[FILTER_CNT]	Co[FPR]	Operation	Maximum latency[1]
1	0	X	X	X	X	Disabled	N/A
2A	1	0	0	0x00	X	Continuous Mode	T_{PD}
2B	1	0	0	X	0x00		
3A	1	0	1	0x01	X	Sampled, Non-Filtered mode	$T_{PD} + T_{SAMPLE} + T_{per}$
3B	1	0	0	0x01	> 0x00		T_{PD} + (CMP_C0[FPR] * T_{per}) + T_{per}
4A	1	0	1	> 0x01	X	Sampled, Filtered mode	T_{PD} + (CMP_C0[FILTER_CNT] * T_{SAMPLE}) + T_{per}
4B	1	0	0	> 0x01	> 0x00		T_{PD} + (CMP_C0[FILTER_CNT] * CMP_C0[FPR] x T_{per}) + T_{per}
5A	1	1	0	0x00	X	Windowed mode	$T_{PD} + T_{per}$
5B	1	1	0	X	0x00		$T_{PD} + T_{per}$
6	1	1	0	0x01	0x01 - 0xFF	Windowed / Resampled mode	T_{PD} + (CMP_C0[FPR] * T_{per}) + $2T_{per}$
7	1	1	0	> 0x01	0x01 - 0xFF	Windowed / Filtered mode	T_{PD} + (CMP_C0[FILTER_CNT] * CMP_C0[FPR] x T_{per}) + $2T_{per}$

1. T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{SAMPLE} is the clock period of the external sample clock. T_{per} is the period of the bus clock.

NOTE

The delay in this table does not include the delay caused by the Discrete Mode.

86.3.3 CMP Discrete mode timing sequence

In order to support the 3 V domain input the analog comparator should be configured to work in Discrete Mode since the internal transistors are not 3 V tolerance. This section summarizes the possible configurations depending on the input range.

Note that the sample signal below means that the analog comparator samples the input analog input. It is different from the sample mode that occurs on the digital process to the final comparison result. Also, note that the cross domain round robin cycling (see Trigger mode) is not allowed.

1. When CMP_C3[PCHCTEN] and CMP_C3[NCHCTEN] are set to 1, the CMP is in continuous time operation. No special timing is required.
2. When either CMP_C3[PCHCTEN] or CMP_C3[NCHCTEN] is 0, it means at least one input comes from the 3V PAD.
   a. If CMP_C3[RDIVE] is zero, it means the input signal comes from 3V PAD, but its range still is in the range of 0 to 1.8V, no timing requirement for the generation of Phase1 and Phase2.

      In this condition, there are two modes for this comparator, high speed mode and low speed mode. Clock source may come from fast clock(16-20M)

      In Figure 86-13, the Analog ready indicates the analog settling time is reached and is ready for comparison. Tc is set by CMP_C3[ACSAT], in high speed mode, it is normally used as 1*T, 2*T, 4*T and 8*T where T is CLOCK period, and in low speed mode (CMP_C0[PMODE] = 0), CMP_C3[ACSAT] is normally used as 16*T, 32*T, 64*T and 256*T.

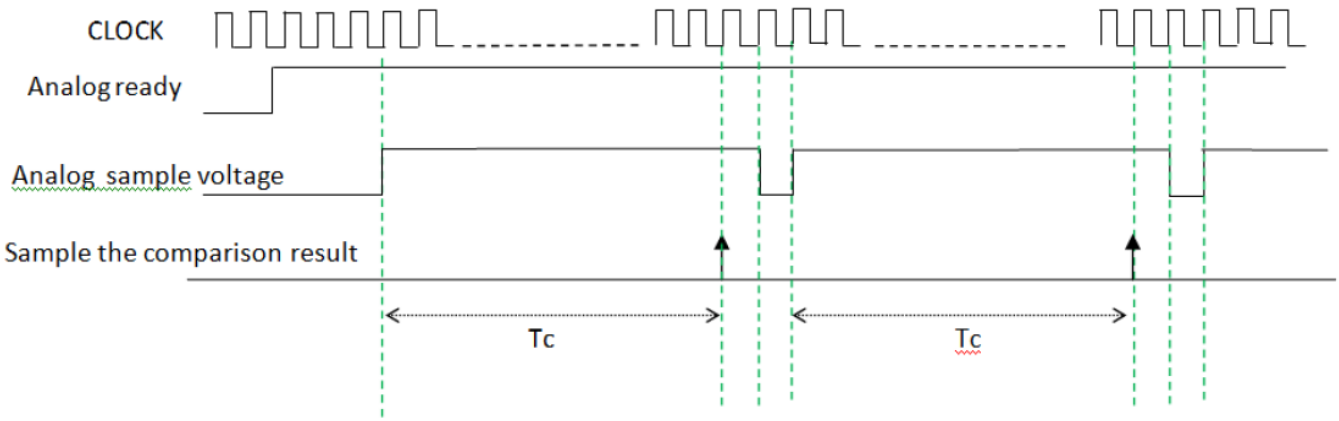


Figure 86-13. CMP_C3[RDIVE] = 0

b. If CMP_C3[RDIVE] == 1, it means the signals come from the 3V PAD and it could over 1.8 V, Phase1, Phase2 will be generated.

   If CMP is in high speed mode (CMP_C0[PMODE] = 1) , Tc, Phase1, and Phase2 can be set based on the possible combinations in Table 86-6.

   If CMP works in low speed mode, Phase1 is suggested to set to 1*T, Phase2 is suggested to set to 8*T, and Tc is normally set to 16*T, 32*T, 64*T and 256*T.

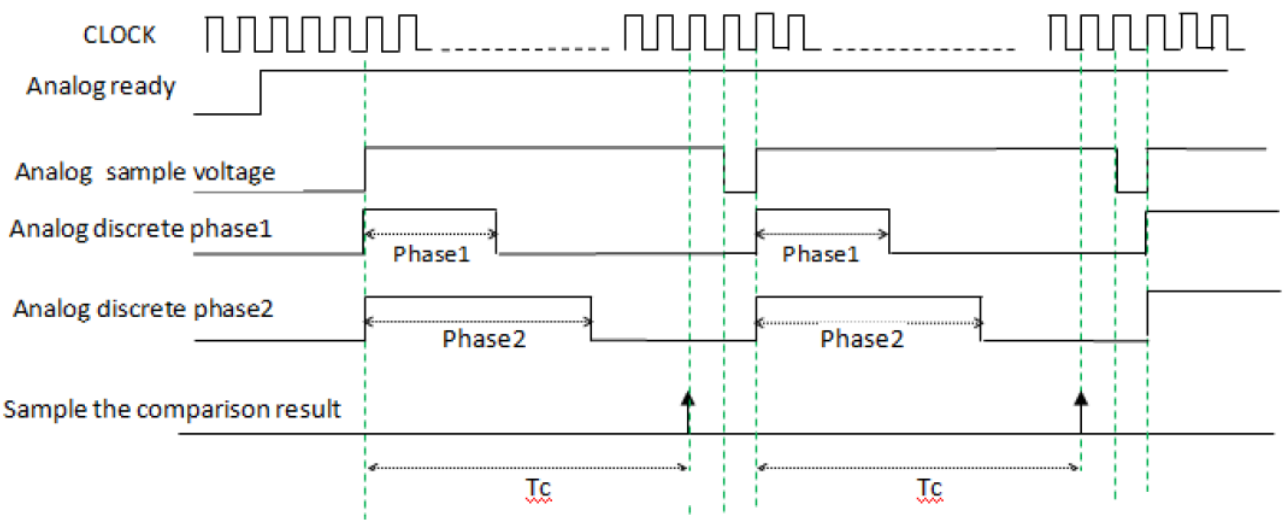


Figure 86-14. CMP_C3[RDIVE] = 1

Table 86-6. Possible Combinations of CMP_C3[ACSAT], CMP_C3[ACPH1TC] and CMP_C3[ACPH2TC]

Tc	Phase1	Phase2
1*T	1*T	1*T
2*T	2*T	2*T
4*T	4*T	4*T
8*T	8*T	8*T

Table continues on the next page...

Table 86-6. Possible Combinations of CMP_C3[ACSAT], CMP_C3[ACPH1TC] and CMP_C3[ACPH2TC] (continued)

Tc	Phase1	Phase2
16*T	1*T	16*T
32*T	1*T	32*T
64*T	1*T	64*T
256*T	0	16*T

If the clock comes from 32KHz slow clock(CMP_C3[DMCLKSEL] = 0), both analog phase 1 and phase 2 will be driven to 0 and Tc is limited to T as shown in the figure below.

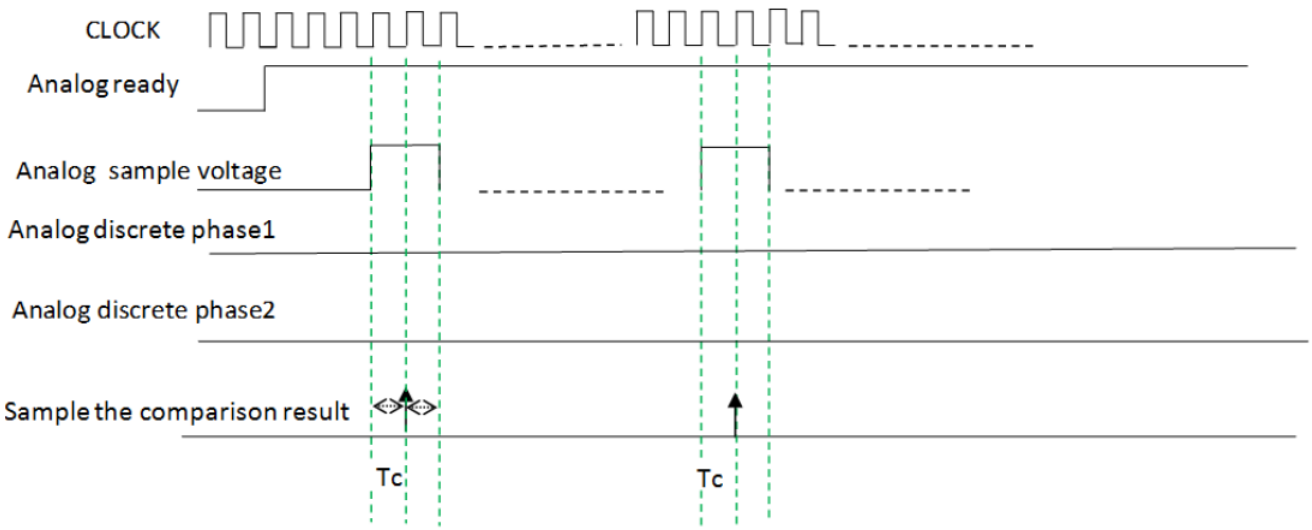


Figure 86-15. Slow (32 kHz) clock timing

86.3.4 CMP clocks

This module has the following clocks:

- Bus clock for register access and window/filter operation
- Fast clock for the discrete mode
- Slow clock for the discrete mode
- Round robin clock for the trigger mode

86.3.5 Interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both. Assuming the CMP DMA enable bit is not set, the following table gives the conditions in which the interrupt request is asserted and deasserted.

Table 86-7. CMP interrupt generations

When	Then
CMP_C0[IER] and CMP_C0[CFR] are set	The interrupt request is asserted
CMP_C0[IEF] and CMP_C0[CFF] are set	The interrupt request is asserted
CMP_C0[IER] and CMP_C0[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
CMP_C0[IEF] and CMP_C0[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

86.3.6 DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting CMP_C0[DMAEN] and the interrupt is enabled by setting CMP_C0[IER], CMP_C0[IEF], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt instead. When the DMA has completed the transfer, it sends a transfer completing indicator signal that deasserts the DMA transfer request and clears the flags (CMP_C0[CFR] and CMP_C0[CFF]) to allow a subsequent change on comparator output to occur and force another DMA request.

The comparator can remain functional in STOP modes. When DMA support is enabled by setting CMP_C0[DMAEN] and the interrupt is enabled by setting CMP_C0[IER], CMP_C0[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, the system will go back to STOP modes. Refer to the DMA chapters in the device reference manual for the asynchronous DMA function for details.

86.4 DAC functional description

This section provides DAC functional description.

86.4.1 Digital-to-analog converter block diagram

The following figure shows the block diagram of the DAC module. It contains a 256-tap resistor ladder network and a 256-to-1 multiplexer, which selects an output voltage from one of 256 distinct levels that outputs from DACO. It is controlled through the control register2(CMP_C1). Its supply reference source can be selected from two sources V_{in1} and V_{in2}. The module can be powered down or disabled when not in use. When in Disabled mode, DACO is connected to the analog ground.

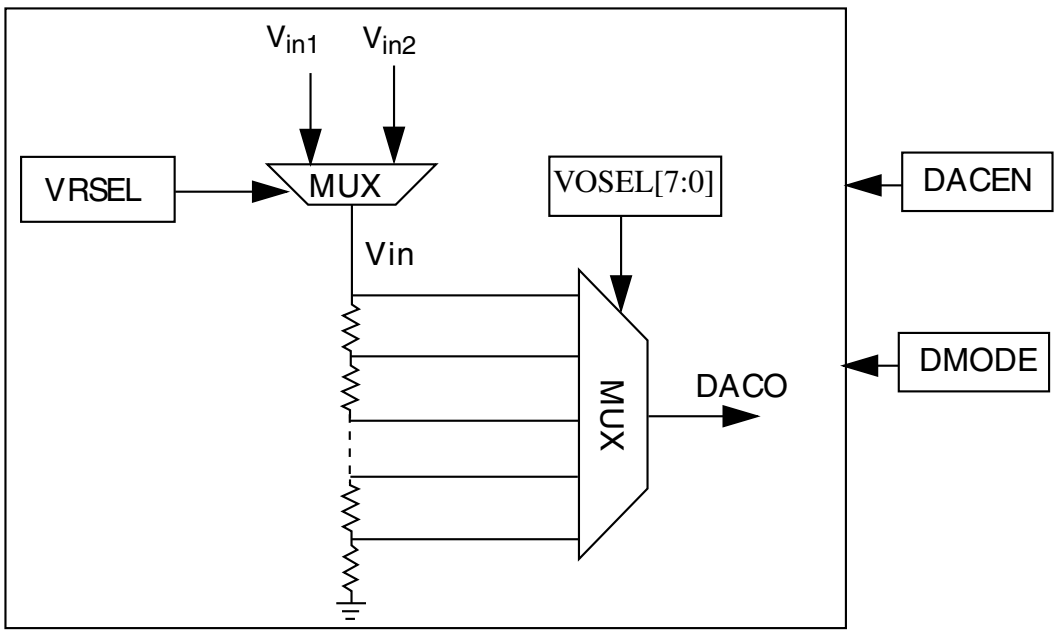


Figure 86-16. 8-bit DAC block diagram

86.4.2 Voltage reference source select

- V_{in1} must be used to connect to the primary voltage source as supply reference of 256 tap resistor ladder
- V_{in2} must be used to connect to an alternate voltage source, or primary source, if an alternate voltage source is not available

86.4.3 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

86.4.4 DAC clocks

This module has a single clock input, the bus clock.

86.4.5 DAC interrupts

This module has no interrupts.

86.4.6 Trigger mode

The CMP and the 8-bit DAC are designed to support the trigger mode operation, which is enabled when the MCU enters STOP modes with CMP_C0[WE], CMP_C0[SE] and CMP_C0[EN] are set.

With this mode enabled, the trigger events that include the operation clock and a trigger start signal will initiate a compare sequence that must first enable the CMP and DAC prior to performing a CMP operation and capturing the output. A fixed channel for either the plus side mux or the minus side mux is selected by software with CMP_C2[FXMP] and CMP_C2[FXMXCH]. It is a mandatory request that the round robin cycling period must be set longer than the time that all the active channels complete the specified comparison cycles specified by CMP_C2[NSAM].

The active channels selected by CMP_C1[CHN*n*] are then routed to the non-fixed channel mux and compared with the reference input in a round-robin manner. In order to meet the comparator stabilization time, after the configurable number of operation clocks defined by CMP_C2[NSAM], the comparison result is sampled for the selected channel. A software pre-programmed state for each channel is configured by writing to CMP_C2[ACO*n*] field. After all the active channels are sampled, if the comparison result changes from its pre-programmed state, the corresponding flag in CMP_C2[CH*n*F] is set. If CMP_C2[RRIE] is set, an asynchronous reset is asserted to bring the MCU out of STOP mode.

Note that these flags do not support generating a DMA transfer event.

This mode is active when the MCU is in STOP mode, so none of the window/filter functions are available. A basic assumption of this mode is that the selected inputs are changing at a much slower rate than the operation clock. It is suggested to configure the comparator in low power comparison mode as well. In programming the CMP_C2[INITMOD] registers it is need to make sure the INITMOD*round robin clock period must be longer than the initialization delay which can be referred from the chip datasheet.

The following diagram shows the basic flow of this mode. In the diagram, CMP_C1[CHN1], CMP_C1[CHN3], and CMP_C1[CHN4] are set, so channels #1, #3, and #4 are selected for round robin depended on their polarity setting. CMP_C2[NSAM] is set to 2'b01, so one clock later the comparison result of the selected channel is sampled. When channel #3 is compared, the result is sampled, and round robin ends. If any of the comparison results from channel #1, #3, or #4 changed from their programmed value (written to CMP_C2[ACO1], CMP_C2[ACO3], and CMP_C2[ACO4]), an interrupt is generated to wake up the MCU from the STOP mode. Software can then poll the CMP_C2[CH*n*F] to see which channel input(s) changed value during the STOP mode.

NOTE

In round robin mode, it should be ensured that the RTC_CLK period is greater than the comparison time corresponding to the value of CMPx_C0[PMODE].

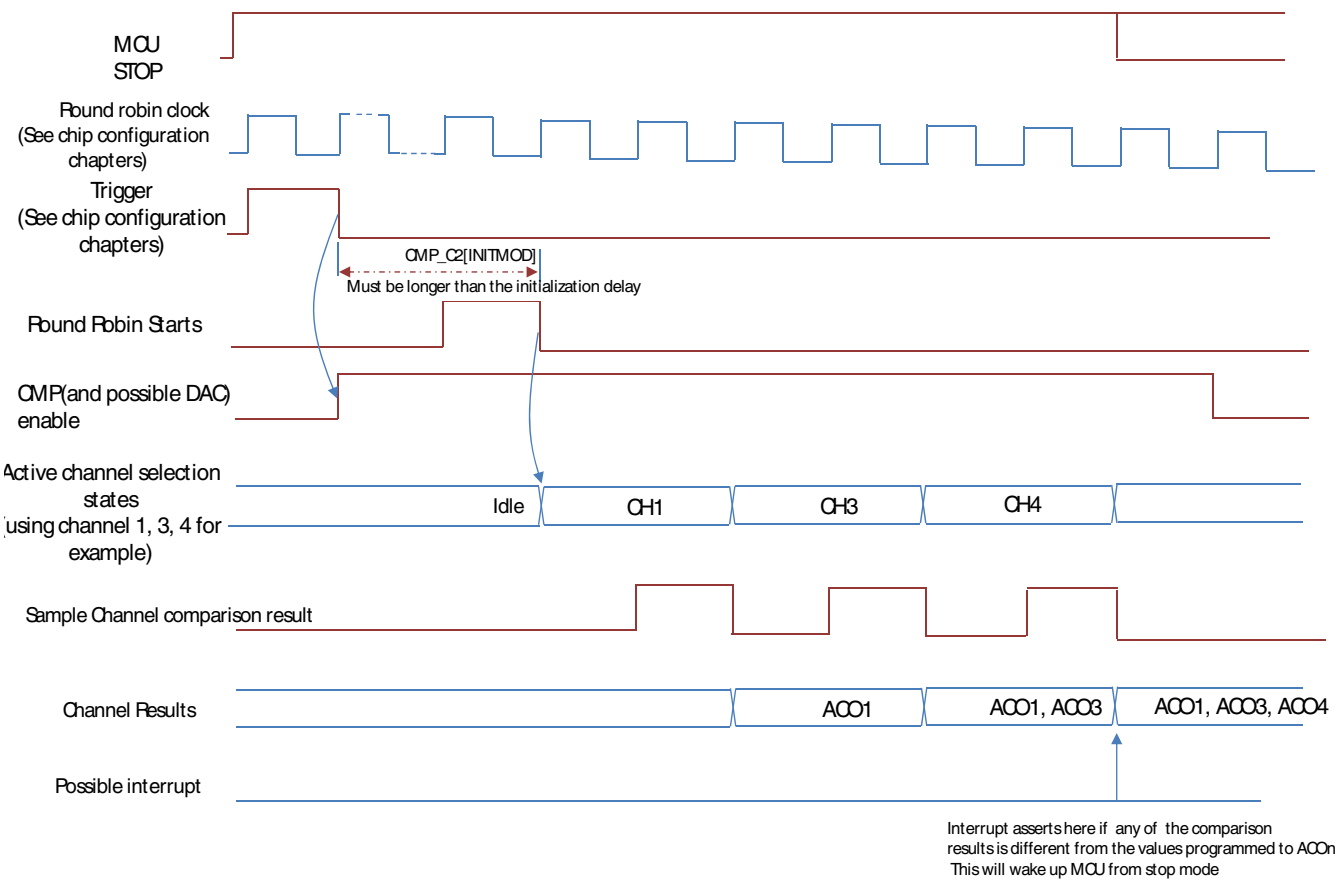


Figure 86-17. Trigger mode

86.5 CMP external signals

This section provides the comparator pin descriptions. The external 3V domain inputs INA_3V[6:1] and 1.8V domain INA_1P8[6:1], INA_1P8_0_P, INA_1P8_0_M are muxed by CMP_C1[PSEL] and CMP_C1[MSEL] before going to the positive and negative ports of the comparator respectively.

Table 86-8. CMP signal descriptions

Signal	Description	I/O
INA_3V[6:1]	The 3V domain analog inputs[6:1]	I
INA_1P8[6:1]	Analog 1.8V input channels [6:1].	I
INA_1P8_0_P	Analog 1.8V input channel 0 positive.	I
INA_1P8_0_m	Analog 1.8V input channel 0 minus.	I

NOTE

It is not recommended to compare 3V domain channel with 1.8V domain channel. If the 3V domain channel's voltage is smaller than 1.8V, in this case, it can be compared with 1.8V domain channel by setting C3[RDIVE] to 0.

86.5.1 External pins

The CMP has two analog inputs: INP and INM. Each of these pins can accept an input voltage that varies across the full operating range of the MCU. If the module is not enabled, each pin can be used as a digital input or output. Consult the specific MCU documentation to determine what functions are shared with these analog inputs.

The user can select either filtered or unfiltered comparator outputs for use on an external I/O pad.

86.6 Initialization

A typical startup sequence is as follows.

The time required to stabilize COUT is the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator, windowing function, and filter. See the datasheet for power-on delays of the comparators. The windowing function has a maximum of one bus clock period delay. The filter delay is specified in the Low-pass filter section.

During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and CMP_C0[CFR]/ CMP_C0[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.

When programmed for filtering modes, COUT initially equals 0 until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

86.7 Memory map/register definitions

86.7.1 CMP register descriptions

86.7.1.1 CMP memory map

CMP1 base address: 401A_4000h

CMP2 base address: 401A_8000h

CMP3 base address: 401A_C000h

CMP4 base address: 401B_0000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version ID Register (VERID)	32	RO	0100_0000
4	Parameter Register (PARAM)	32	RO	0000_0000
8	CMP Control Register 0 (C0)	32	RW	0000_0000
C	CMP Control Register 1 (C1)	32	RW	0000_0000
10	CMP Control Register 2 (C2)	32	RW	0000_0000
14	CMP Control Register 3 (C3)	32	RW	1100_0000

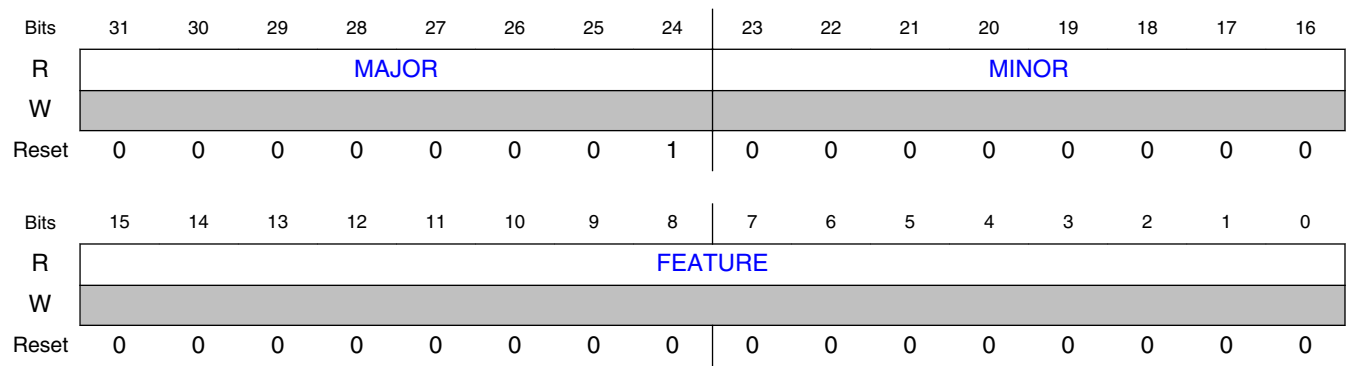
86.7.1.2 Version ID Register (VERID)

The Version ID register indicates the version integrated for this instance on the device.

86.7.1.2.1 Offset

Register	Offset
VERID	0h

86.7.1.2.2 Diagram



86.7.1.2.3 Fields

Field	Description
31-24 MAJOR	Major Version Number. This read only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number. This read only field returns the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number. This read only filed returns the feature set number.

86.7.1.3 Parameter Register (PARAM)

The Parameter register indicates the feature parameters for this instance on the device.

86.7.1.3.1 Offset

Register	Offset
PARAM	4h

86.7.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PARAM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PARAM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

86.7.1.3.3 Fields

Field	Description
31-0 PARAM	Parameter Registers. This read only filed returns the feature parameters implemented along with the Version ID register.

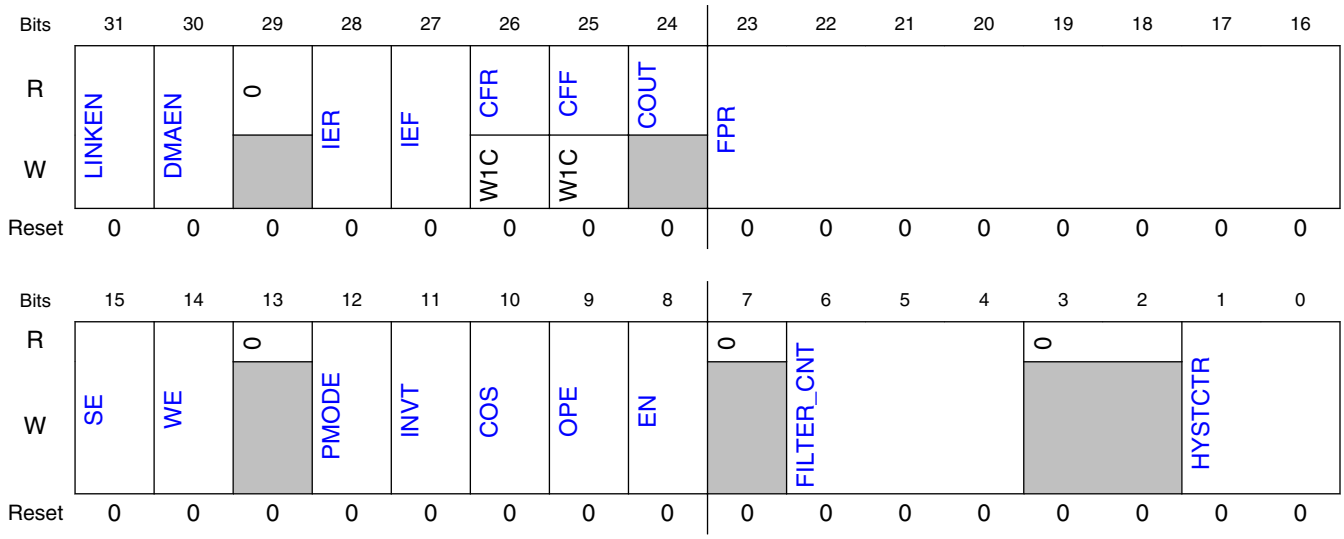
86.7.1.4 CMP Control Register 0 (C0)

The CMP Control Register 0 (C0) contains configuration options for CMP function, such as DMA and interrupt control, window/filter function. It also includes the hysteresis control for the analog CMP.

86.7.1.4.1 Offset

Register	Offset
C0	8h

86.7.1.4.2 Diagram



86.7.1.4.3 Fields

Field	Description
31 LINKEN	CMP to DAC link enable. This bit is used to enable the link from CMP to DAC enables. When this bit is set, the DAC enable/disable is controlled by CMP_C0[EN] bit instead of CMP_C1[DACEN]. 0 - CMP to DAC link is disabled 1 - CMP to DAC link is enabled.
30 DMAEN	DMA Enable Enables the DMA transfer triggered from the CMP module. When this field and the corresponding interrupt enable bit are set, a DMA request is asserted when CFR or CFF is set. 0 - DMA is disabled. 1 - DMA is enabled.
29 —	Reserved
28 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from the CMP. When this field is set, an interrupt will be asserted when CFR is set. 0 - Interrupt is disabled. 1 - Interrupt is enabled.
27 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from the CMP. When this field is set, an interrupt will be asserted when CFF is set. 0 - Interrupt is disabled. 1 - Interrupt is enabled.

Table continues on the next page...

Field	Description
26 CFR	Analog Comparator Flag Rising Detects a rising-edge on COUT, when set, during normal operation. CFR is cleared by writing 1 to it. During Stop modes, CFR is level sensitive 0 - A rising edge has not been detected on COUT. 1 - A rising edge on COUT has occurred.
25 CFF	Analog Comparator Flag Falling Detects a falling-edge on COUT, when set, during normal operation. CFF is cleared by writing 1 to it. During Stop modes, CFF is level senstive . 0 - A falling edge has not been detected on COUT. 1 - A falling edge on COUT has occurred.
24 COUT	Analog Comparator Output Returns the current value of the Analog Comparator output, when read. The field is reset to 0 and will read as C0[INVT] when the Analog Comparator module is disabled, that is, when C0[EN] = 0. Writes to this field are ignored.
23-16 FPR	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when C1[SE] = 0. Setting FPR to 0x0 disables the filter. Filter programming and latency details are provided in the CMP functional description. This field has no effect when C0[SE]= 1. In that case, the external SAMPLE signal is used to determine the sampling period.
15 SE	Sample Enable If a write to this register attempts to set both SE and WE, then only Sampling mode is taking effect in MCU Run mode and the Round Robin Cycling(Trigger Mode) is enabled in MCU STOP mode. 0 - Sampling mode is not selected. 1 - Sampling mode is selected.
14 WE	Windowing Enable If a write to this register attempts to set both SE and WE, then only Sampling mode is taking effect in MCU Run mode and the Round Robin Cycling(Trigger Mode) is enabled in MCU STOP mode. 0 - Windowing mode is not selected. 1 - Windowing mode is selected.
13 —	Reserved
12 PMODE	Power Mode Select 0 - Low Speed (LS) comparison mode is selected. 1 - High Speed (HS) comparison mode is selected.
11 INVT	Comparator invert This bit allows selecting the polarity of the analog comparator function. It is also driven to the COUT output (on both the device pin and as C0[COUT]) when CR1[OPE]=0. 0 - Does not invert the comparator output. 1 - Inverts the comparator output.
10 COS	Comparator Output Select 0 - Set CMPO to equal COUT (filtered comparator output). 1 - Set CMPO to equal COUTA (unfiltered comparator output).

Table continues on the next page...

Field	Description
9 OPE	Comparator Output Pin Enable The OPE bit enables the path from the comparator output to a selected pin. 0 - When OPE is 0, the comparator output (after window/filter settings dependent on software configuration) is not available to a packaged pin. 1 - When OPE is 1, and if the software has configured the comparator to own a packaged pin, the comparator is available in a packaged pin.
8 EN	Comparator Module Enable The EN bit enables the Analog Comparator Module. When the module is not enabled, the analog part remains in the off state, and consumes no power. When the same input is selected from analog mux to the positive and negative port, the comparator is disabled automatically. 0 - Analog Comparator is disabled. 1 - Analog Comparator is enabled.
7 —	Reserved
6-4 FILTER_CNT	Filter Sample Count This field specifies the number of consecutive samples that must agree prior to the comparator ouput filter accepting a new output state. For information regarding filter programming and latency, please see the Functional Description. 000 - Filter is disabled. If SE = 1, then COUT is a logic zero (this is not a legal state, and is not recommended). If SE = 0, COUT = COUTA. 001 - 1 consecutive sample must agree (comparator output is simply sampled). 010 - 2 consecutive samples must agree. 011 - 3 consecutive samples must agree. 100 - 4 consecutive samples must agree. 101 - 5 consecutive samples must agree. 110 - 6 consecutive samples must agree. 111 - 7 consecutive samples must agree.
3-2 —	Reserved
1-0 HYSTCTR	Comparator hard block hysteresis control. See chip data sheet to get the actual hystersis value with each level 00 - The hard block output has level 0 hysteresis internally. 01 - The hard block output has level 1 hysteresis internally. 10 - The hard block output has level 2 hysteresis internally. 11 - The hard block output has level 3 hysteresis internally.

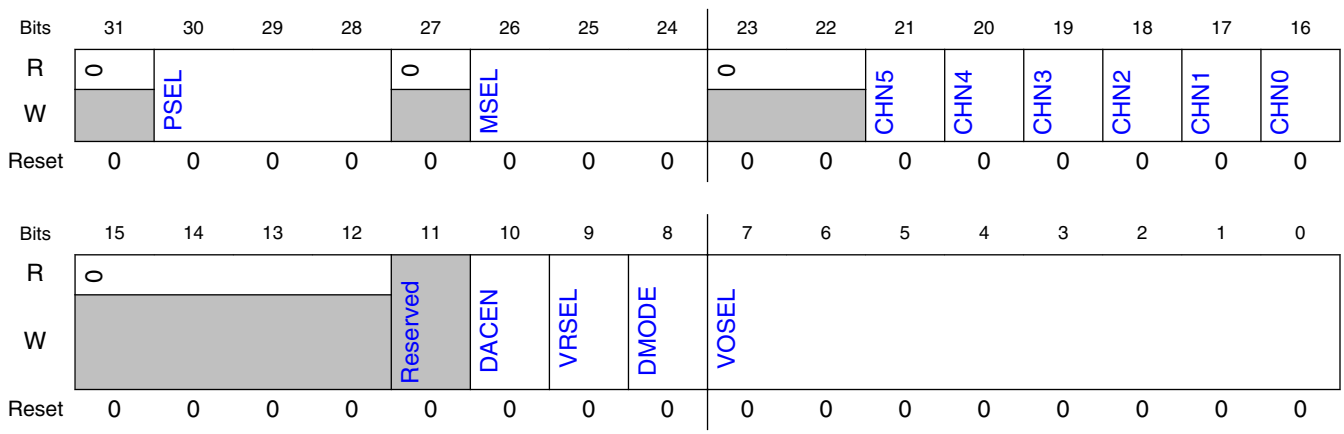
86.7.1.5 CMP Control Register 1 (C1)

The CMP Control Register 1 (C1) contains configuration options for CMP channel selection. It also includes the control for the analog DAC.

86.7.1.5.1 Offset

Register	Offset
C1	Ch

86.7.1.5.2 Diagram



86.7.1.5.3 Fields

Field	Description
31 —	Reserved
30-28 PSEL	Plus Input MUX Control These bits determines which input is selected for the positive mux. Refer to the chip configuation chapters to get the detailed connections about Reference Input{0,1,2,3,4,5} and Internal Positive Input 0. The 8b DAC output has been connected internally. Note: For the round robin mode of operation, Internal Posivitive input 0 should be excluded from the activie channels and the fixed channel ports, meanwhile, the MSEL and PSEL bitfields in CMP_C1 register must have different values. 000 - Internal Positive Input 0 for Plus Channel -- Internal Plus Input 001 - External Input 1 for Plus Channel -- Reference Input 0 010 - External Input 2 for Plus Channel -- Reference Input 1 011 - External Input 3 for Plus Channel -- Reference Input 2 100 - External Input 4 for Plus Channel -- Reference Input 3 101 - External Input 5 for Plus Channel -- Reference Input 4 110 - External Input 6 for Plus Channel -- Reference Input 5 111 - Internal 8b DAC output

Table continues on the next page...

Field	Description
27 —	Reserved
26-24 MSEL	Minus Input MUX Control These bits determine which input is selected for the negative mux. Refer to the chip configuation chapters to get the detailed connections about Reference Input{0,1,2,3,4,5} and Internal Negative Input 0. The 8b DAC output has been connected internally. Note: For the round robin mode of operation, Internal negative input 0 should be excluded from the activie channels and the fixed channel ports, meanwhile, the MSEL and PSEL bitfields in CMP_C1 register must have different values. 000 - Internal Negative Input 0 for Minus Channel -- Internal Minus Input 001 - External Input 1 for Minus Channel -- Reference Input 0 010 - External Input 2 for Minus Channel -- Reference Input 1 011 - External Input 3 for Minus Channel -- Reference Input 2 100 - External Input 4 for Minus Channel -- Reference Input 3 101 - External Input 5 for Minus Channel -- Reference Input 4 110 - External Input 6 for Minus Channel -- Reference Input 5 111 - Internal 8b DAC output
23-22 —	Reserved
21 CHN5	Channel 5 input enable Channel 5 of the input enable for the round-robin checker. If CHN5 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
20 CHN4	Channel 4 input enable Channel 4 of the input enable for the round-robin checker. If CHN4 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
19 CHN3	Channel 3 input enable Channel 3 of the input enable for the round-robin checker. If CHN3 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
18 CHN2	Channel 2 input enable Channel 2 of the input enable for the round-robin checker. If CHN2 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
17 CHN1	Channel 1 input enable Channel 1 of the input enable for the round-robin checker. If CHN1 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
16 CHN0	Channel 0 input enable Channel 0 of the input enable for the round-robin checker. If CHN0 is set, then the corresponding channel to the non-fixed mux port is enabled to check its voltage value in the round-robin mode. If the same channel is selected as the reference voltage, this bit has no effect.
15-12	Reserved

Table continues on the next page...

Field	Description
—	
11 —	Reserved
10 DACEN	DAC Enable This bit is used to enable the DAC. When the DAC is disabled, it is powered down to conserve power. 0 - DAC is disabled. 1 - DAC is enabled.
9 VRSEL	Supply Voltage Reference Source Select 0 - Vin1 is selected as resistor ladder network supply reference Vin. Vin1 is from internal PMC. 1 - Vin2 is selected as resistor ladder network supply reference Vin. Vin2 is from PAD.
8 DMODE	DAC Mode Selection 0 - DAC is selected to work in low speed and low power mode. 1 - DAC is selected to work in high speed high power mode.
7-0 VOSEL	DAC Output Voltage Select This bit selects an output voltage from one of 256 distinct levels. DACO = (Vin/256) * (VOSEL[7:0] + 1), so the DACO range is from Vin/256 to Vin.

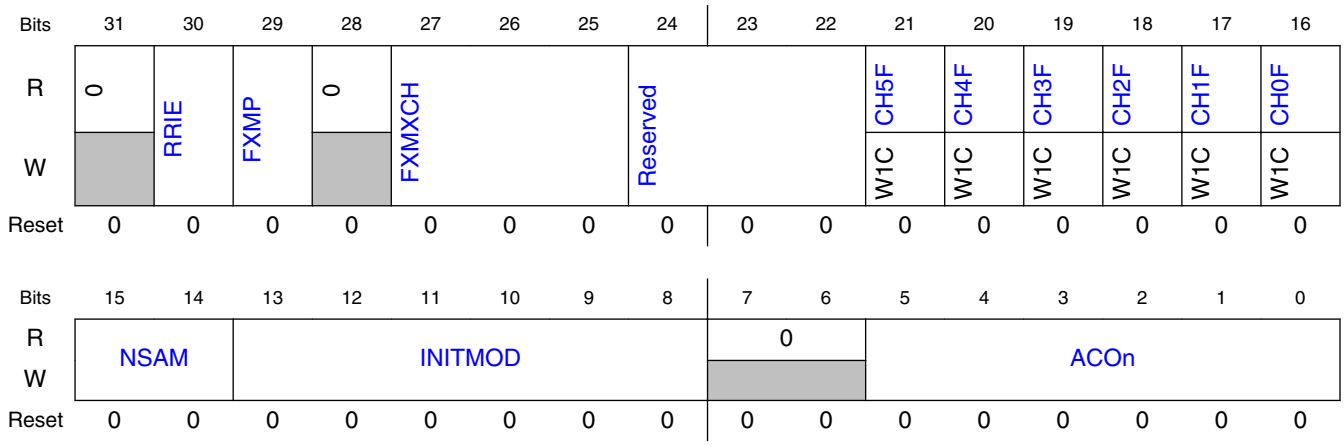
86.7.1.6 CMP Control Register 2 (C2)

The CMP Control Register 2 (C2) contains configuration options for the trigger mode.

86.7.1.6.1 Offset

Register	Offset
C2	10h

86.7.1.6.2 Diagram



86.7.1.6.3 Fields

Field	Description
31 —	Reserved
30 RRIE	Round-Robin interrupt enable This bit enables the interrupt/wake-up when the comparison result changes for a given channel. 0 - The round-robin interrupt is disabled. 1 - The round-robin interrupt is enabled when a comparison result changes from the last sample.
29 FXMP	Fixed MUX Port This bit is used to fix the analog mux port for the round-robin mode. 0 - The Plus port is fixed. Only the inputs to the Minus port are swept in each round. 1 - The Minus port is fixed. Only the inputs to the Plus port are swept in each round.
28 —	Reserved
27-25 FXMXCH	Fixed channel selection This field indicates which channel in the mux port is fixed in a given round-robin mode. If FXDACI is set, FXMXCH has no effect. 000 - External Reference Input 0 is selected as the fixed reference input for the fixed mux port. 001 - External Reference Input 1 is selected as the fixed reference input for the fixed mux port. 010 - External Reference Input 2 is selected as the fixed reference input for the fixed mux port. 011 - External Reference Input 3 is selected as the fixed reference input for the fixed mux port. 100 - External Reference Input 4 is selected as the fixed reference input for the fixed mux port. 101 - External Reference Input 5 is selected as the fixed reference input for the fixed mux port. 110 - Reserved. 111 - The 8bit DAC is selected as the fixed reference input for the fixed mux port.

Table continues on the next page...

Field	Description
24-22 —	Reserved
21 CH5F	CH5F External Channel 5 input changed flag. This bit is set If the channel 5 input changed from the last comparison with the fixed mux port.
20 CH4F	CH4F External Channel 4 input changed flag. This bit is set If the channel 4 input changed from the last comparison with the fixed mux port.
19 CH3F	CH3F External Channel 3 input changed flag. This bit is set If the channel 3 input changed from the last comparison with the fixed mux port.
18 CH2F	CH2F External Channel 2 input changed flag. This bit is set If the channel 2 input changed from the last comparison with the fixed mux port.
17 CH1F	CH1F External Channel 1 input changed flag. This bit is set If the channel 1 input changed from the last comparison with the fixed mux port.
16 CH0F	CH0F External Channel 0 input changed flag. This bit is set If the channel 0 input changed from the last comparison with the fixed mux port.
15-14 NSAM	Number of sample clocks For a given channel, this field specifies how many round-robin clock cycles later the sample takes place. 00 - The comparison result is sampled as soon as the active channel is scanned in one round-robin clock. 01 - The sampling takes place 1 round-robin clock cycle after the next cycle of the round-robin clock. 10 - The sampling takes place 2 round-robin clock cycles after the next cycle of the round-robin clock. 11 - The sampling takes place 3 round-robin clock cycles after the next cycle of the round-robin clock.
13-8 INITMOD	Comparator and DAC initialization delay modulus. These values specify the round robin clock cycles used to determine the comparator and DAC initialization delays specified by the datasheet. For example the initialization delay is 80us and the round robin clock is 100kHz, then INITMOD should be set to 80us/10us = 8. 000000 - The modulus is set to 64(same with 111111). other values - Inilization delay is set to INITMOD * round robin clock period.
7-6 —	Reserved
5-0 ACOn	ACOn The result of the input comparison for channel *n* . This field stores the latest comparison result of the input channel *n* with the fixed mux port. Reading this bit returns the latest comparison result. Writing this field defines the pre-set state of channel *n*.

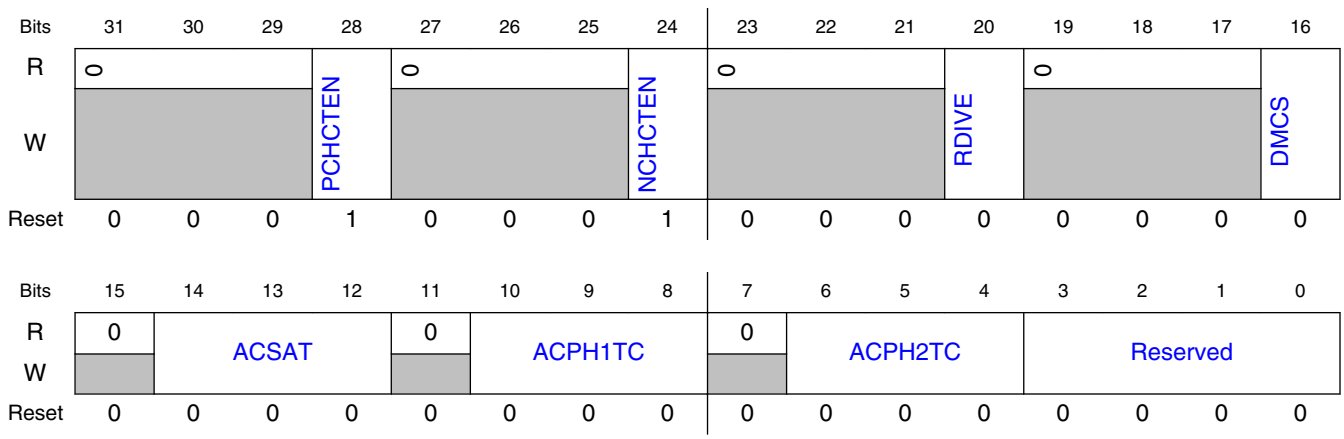
86.7.1.7 CMP Control Register 3 (C3)

The CMP Control Register 3 (C3) contains configuration options for the discrete mode.

86.7.1.7.1 Offset

Register	Offset
C3	14h

86.7.1.7.2 Diagram



86.7.1.7.3 Fields

Field	Description
31-29 —	Reserved
28 PCHCTEN	Positive Channel Continuous Mode Enable. This bit is used to enable the positive channel working in continuous mode. When this bit is set to 1, it means all the inputs to positive channels are coming from 1.8v domain and no discrete timing is required for the positive channel mux. Otherwise, when it is cleared to 0, positive channel will work in discrete mode, means the positive inputs are coming from 3v domain. 0 - Positive channel is in Discrete Mode and special timing needs to be configured. 1 - Positive channel is in Continuous Mode and no special timing is requried.
27-25 —	Reserved
24 NCHCTEN	Negative Channel Continuous Mode Enable. This bit is used to enable the negative channel working in continuous mode. When this bit is set to 1, it means all the inputs to negative channels are coming from 1.8v domain and no discrete timing is required

Table continues on the next page...

Field	Description
	for the positive channel mux. Otherwise, when it is cleared to 0, negative channel will work in discrete mode, means the negative inputs are coming from 3v domain. 0 - Negative channel is in Discrete Mode and special timing needs to be configured. 1 - Negative channel is in Continuous Mode and no special timing is requried.
23-21 —	Reserved
20 RDIVE	Resistor Divider Enable This bit is used to enable the resistor divider for the inputs when they come from 3v domain and their values are above 1.8v. 0 - The resistor is not enabled even when either NCHEN or PCHEN is set to1 but the actual input is in the range of 0 - 1.8v. 1 - The resistor is enabled because the inputs are above 1.8v.
19-17 —	Reserved
16 DMCS	Discrete Mode Clock Selection This bit is used to select the clock source in order to generate the requiried timing for comparator to work in discrete mode. There are generally two kinds of clocks coming from SoC, one is a fast clock(16 - 20MHz) to generate fast prop delay and another one is noramlly 32kHz to work in low power mode. 0 - Slow clock is selected for the timing generation. 1 - Fast clock is selected for the timing generation.
15 —	Reserved
14-12 ACSAT	Analog Comparator Sampling Time control. These values configures the analog comparator sampling timing(speicified by the discrete mode clock period T which is selected by DMCS) in discrete mode. 000 - The sampling time equals to T 001 - The sampling time equasl to 2*T 010 - The sampling time equasl to 4*T 011 - The sampling time equasl to 8*T 100 - The sampling time equasl to 16*T 101 - The sampling time equasl to 32*T 110 - The sampling time equasl to 64*T 111 - The sampling time equasl to 256*T
11 —	Reserved
10-8 ACPH1TC	Analog Comparator Phase1 Timing Control. These values configures the analog comparator phase1 timing when RDIV is set to 1 which means the input volage level is above 1.8v. T means the discrete mode clock period in the table. 000 - Phase1 active time in one sampling period equals to T 001 - Phase1 active time in one sampling period equals to 2*T 010 - Phase1 active time in one sampling period equals to 4*T

Table continues on the next page...

Field	Description
	011 - Phase1 active time in one sampling period equals to 8*T 100 - Phase1 active time in one sampling period equals to T 101 - Phase1 active time in one sampling period equals to T 110 - Phase1 active time in one sampling period equals to T 111 - Phase1 active time in one sampling period equals to 0
7 —	Reserved
6-4 ACPH2TC	Analog Comparator Phase2 Timing Control. These values configures the analog comparator phase2 timing when RDIV is set to 1 which means the input volage level is above 1.8v. T means the discrete mode clock period in the table. 000 - Phase2 active time in one sampling period equals to T 001 - Phase2 active time in one sampling period equals to 2*T 010 - Phase2 active time in one sampling period equals to 4*T 011 - Phase2 active time in one sampling period equals to 8*T 100 - Phase2 active time in one sampling period equals to 16*T 101 - Phase2 active time in one sampling period equals to 32*T 110 - Phase2 active time in one sampling period equals to 64*T 111 - Phase2 active time in one sampling period equals to 16*T
3-0 —	Reserved

Chapter 87
Analog-to-Digital Converter (LPADC)

87.1 Chip-specific LPADC information

Table 87-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

87.1.1 LPADC instantiation information

This device contains two ADCs with up to 4.2Msps rate.

87.1.2 Number of ADC channels and channel assignments

The number of ADC1 channels present on the device is 12 split into 6 A channels and 6 B channels. The number of ADC2 channels present on device is 14. There are 26 channels in total. However, 6 channels are shared between ADC1 and ADC2.

The channel assignments are as determined in following figure:

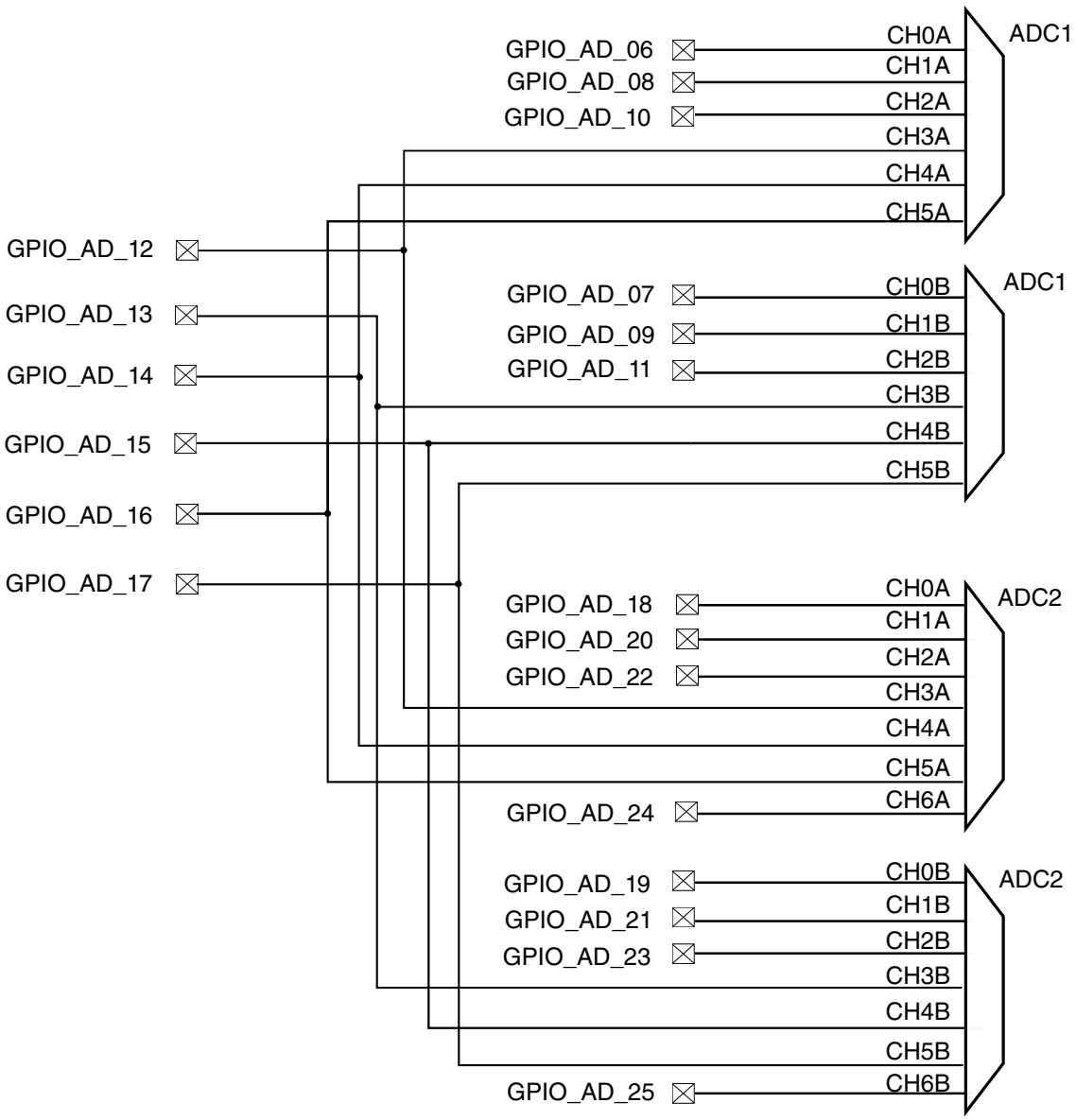


Figure 87-1. External sources channel assignments

Internal sources channel assignments:

- ADC1 CH6A – DAC output
- ADC2 CH7B – DAC output

Note

All external sources channel pairs can be used in differential mode.

87.1.3 LPADC low leakage stop mode

The low leakage stop mode is set by setting IOMUXC_GPR70[ADCx_IPG_STOP_MODE] = 1.

87.1.4 LPADC voltage reference options

The LPADC supports the following voltage reference options ADCx_CFG[REFSEL]:

- 00 - VREFH - connected as the primary reference option
- 01/10 - VDDA_ADC_1P8 - connected as the alternative reference option
- 11 - Reserved

NOTE

V_{SSA} and V_{REFL} is tied to common ground V_{SS} in package.

87.1.5 LPADC triggers

The LPADC supports both software and hardware triggers. The only hardware mechanism for triggering the LPADC is the ADC_ETC. The ADC_ETC itself can be triggered by other peripherals dedicated by interconnection via XBAR module. In total ADC_ETC can select one of the 8 ADC triggers (TRIG0-7) by configuring ADC_ETC->TRIGx_CHAIN_y_z[HWTSn] to initiate ADC conversion. Moreover, ADC_ETC can select also command executed with dedicated trigger by configuring ADC_ETC->TRIGx_CHAIN_y_z[CSELn] when ADCx->TCTRLx[CMD_SEL] = 1.

87.2 Overview

The 12-bit analog-to-digital converter (LPADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

For the chip specific modes of operation, see the power management information of the device.

87.2.1 Block diagram

The following figure is the LPADC module block diagram.

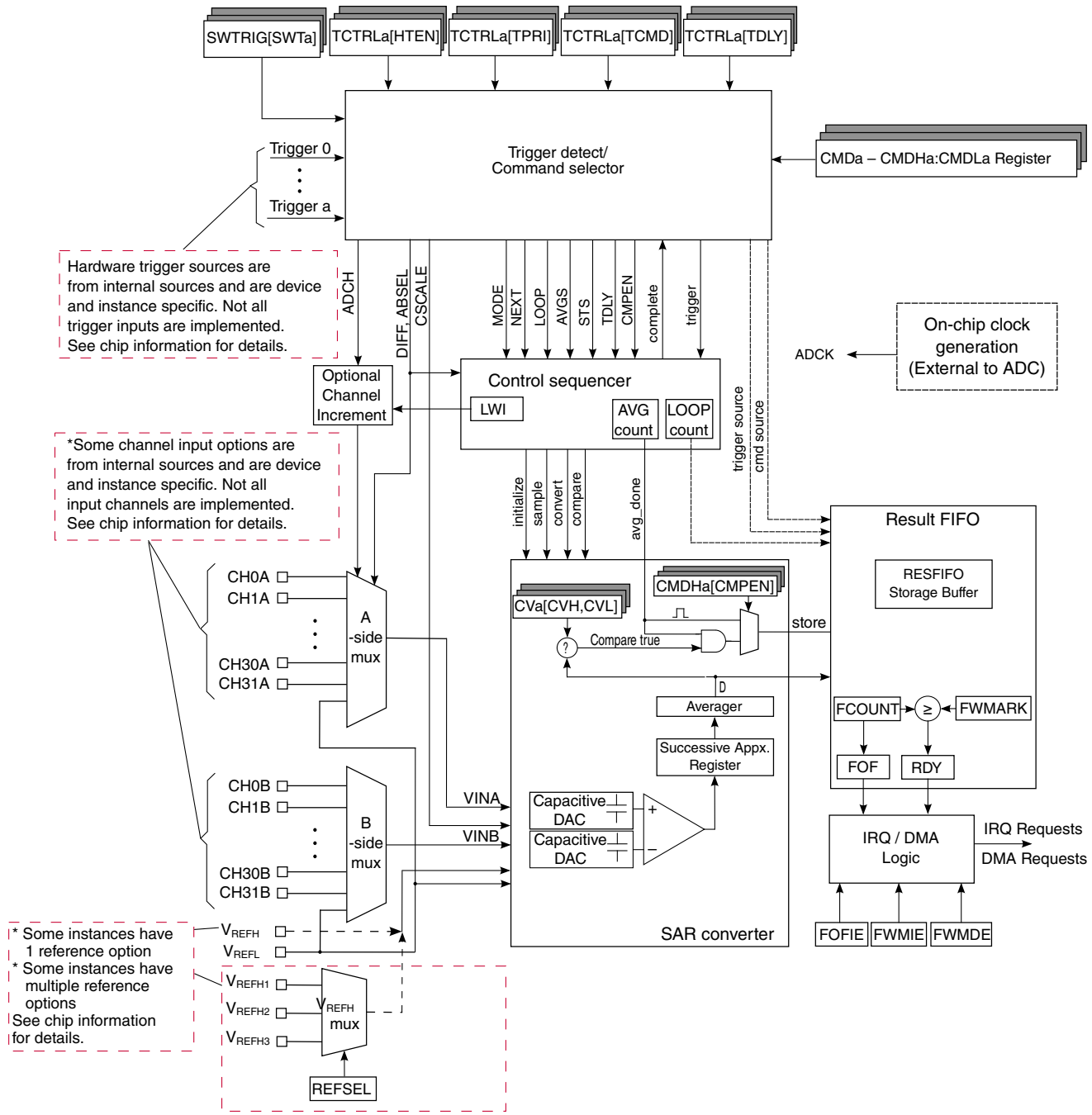


Figure 87-2. LPADC block diagram

87.2.2 Features

Following are the features of the LPADC module.

- Linear successive approximation algorithm

    - Differential operation with 13-bit resolution
    - Single-ended operation with 12-bit resolution
- Channel support for up to 64 analog input channels for conversion of external pins and from internal sources
    - Select external pin inputs paired for conversion as differential channel input
    - Measurement of on-chip analog sources such as DAC or temperature sensor
- Channel scaling allows input voltage levels higher than the ADC reference voltage
- Configurable analog input sample time
- Configurable speed options to accommodate operation in low power modes of the device.
- Trigger detect with up to 8 trigger sources with priority level configuration. Software or hardware trigger option for each.
- 15 command buffers allow independent options selection and channel sequence scanning.
- Automatic compare for less-than, greater-than, within range, or out-of-range with "store on true" and "repeat until true" options
- 16-entry conversion result data FIFO with configurable watermark and overflow detection
- Interrupt, DMA or polled operation

87.3 Functional description

The LPADC module performs analog-to-digital conversions on any of the software selectable analog input channels by a successive approximation algorithm. The module initializes to its lowest power state during reset and in Low-Power Stop mode. The LPADC analog circuits can optionally be pre-enabled for faster starts to conversions at the expense of higher idle currents. Conversions are initiated by selectable trigger events from software or hardware sources. The trigger detect logic includes a configurable enable and priority scheme for the available trigger sources. The module includes multiple command buffers that provide configurable flexibility for channel scanning and independent channel selections for different trigger sources. Multiple command buffers also allow variable option selection such as input scaling factor,differential vs. single-ended, sample time and averaging on a per-channel basis.

This module optionally averages the result of multiple conversions on a channel before storing the calculated result. The hardware average function is enabled by setting CMDH*a*[AVGS] bitfield to a non-zero value and operates in any of the conversion modes and configurations.

When the conversion and averaging loops are completed, the resulting data is placed in a FIFO data buffer along with other tag information associated with the result. A configurable watermark level supports interrupt or DMA requests when the number of stored datawords exceeds the setting.

The LPADC module optionally compares the result of a conversion with the contents of two value registers for less-than, greater-than, inside-range or outside-range detection. The compare function operates in any of the conversion modes and configurations.

The sequencing of an ADC command is summarized in the flow diagram shown below.

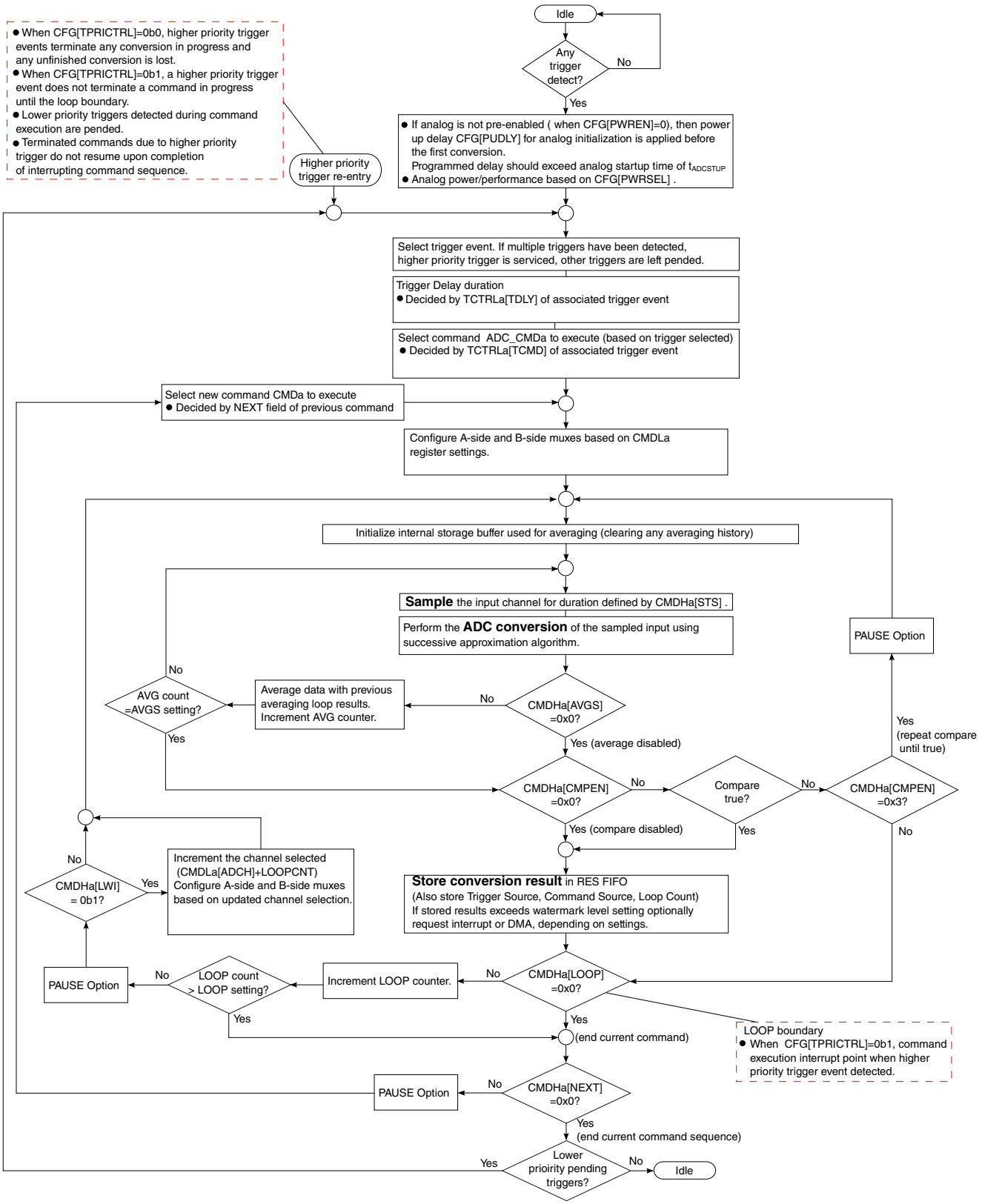


Figure 87-3. ADC command execution flow diagram

The figure below shows how LPADC interfaces with the ADC_ETC.

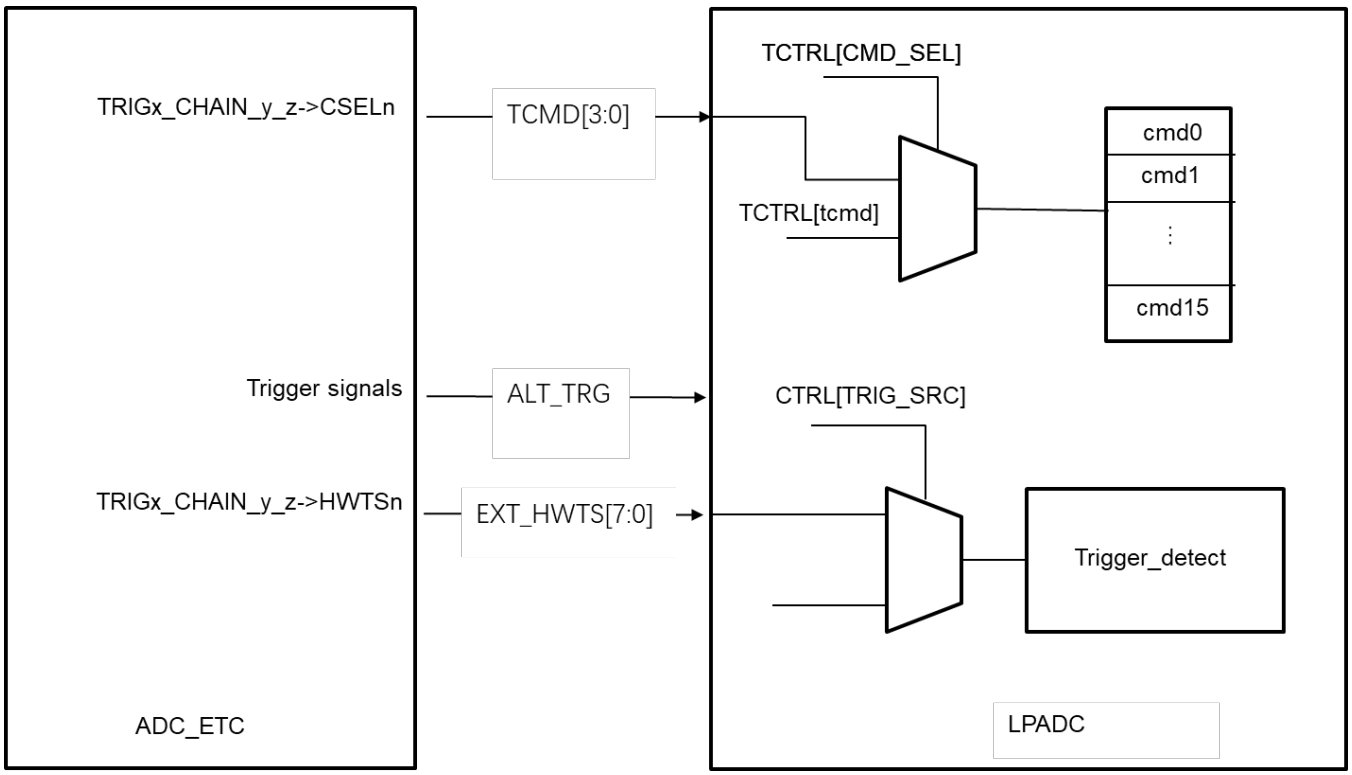


Figure 87-4. Trigger block diagram

The following is a description of the trigger signals that interface between the ADC_ETC and the LPADC:

- TCMD[3:0] is a hardware-controllable command selector, which could be sampled/ latched when EXT_HWTS[a] is high.
- EXT_HWTS is a hardware bus to select the trigger channel.
- ALT_TRIG is a hardware trigger signal. All channels share a same trigger signal, and need EXT_HWTS to indicate which channel it is from. EXT_HWTS needs to be at least one clock cycle ahead of trigger assertion.

TCMD, EXT_HWTS, and ALT_TRIG are synced with ipg_clk from SOC level.

87.3.1 Modes of operation

The table below shows the operation of LPADC module in various power modes.

Table 87-2. Chip modes supported by the ADC module

Chip mode	LPADC Operation
RUN	Normal operation

Table continues on the next page...

Table 87-2. Chip modes supported by the ADC module (continued)

Chip mode	LPADC Operation
STOP/WAIT	Can continue operating provided the Doze Enable bit (CTRL[DOZEN]) is clear and the module is using an external or internal clock source which remains operating during stop/wait modes. When the CTRL[DOZEN] bit is set, the module will wait for the current averaging iteration/FIFO storage to complete before acknowledging stop or wait mode entry.
LOW LEAKAGE STOP	The Doze Enable (CTRL[DOZEN]) bit is ignored and the ADC will wait for the current transfer to complete any pending operation before acknowledging low-leakage mode entry.

87.3.2 Voltage reference

The voltage reference high (V_{REFH}) used by the LPADC is supplied from either an on-chip voltage reference source or from an off-chip source supplied through external pins. V_{REFL} is always from an external pin and must be at the same voltage potential as V_{SSA}. See the chip configuration information on the voltage reference options specific to this packaged device.

This block supports a programmable selection of the voltage reference used for ADC conversions (via the CFG[REFSEL] field). See the chip configuration information on the voltage reference options specific to this packaged device.

NOTE

The external analog input pins can support higher input voltages than reference by scaling features implementation in ADCxCMDLy[CSCALE].

87.3.3 Power control

The default setting for the ADC analog circuits is disabled (CFG[PWREN] = 0) while the LPADC is in its Idle state. When a trigger is detected and ADC command processing is initiated, the analog circuits are enabled and require a period of initialization before the first conversion cycle. The CFG[PUDLY] should be programmed such that a delay duration longer than $t_{ADCSTUP}$ is incurred. Accuracy of the initial conversion(s) after activation is degraded if CFG[PUDLY] is set to too small a value.

Faster conversion startup times can be achieved by optionally setting the CFG[PWREN] field to pre-enable the analog circuits of the LPADC at the expense of power consumption even while the module is in an idle state. When CFG[PWREN] is set, the Power Enable timer is activated and enforces the minimum startup before detected triggers are allowed to initiate ADC conversions.

This module also has configuration options for controlling power and performance summarized in the table below. For example: To achieve the maximum sample rate, the highest power setting is required. See the device datasheet for specification of the available power modes.

Table 87-3. Power option settings

CFG[PWRSEL]	Description
0b00 (Default)	Level 1. Slowest speed/Lowest power setting.
0b01	Level 2.
0b10	Level 3.
0b11	Level 4. Fastest speed/highest power setting.

In the two highest power mode settings (CFG[PWRSEL]=0x2 or CFG[PWRSEL]=0x3) the LPADC will delay start of sampling for 16 ADC clock cycles on the initial conversion since being in its Idle state. This delay is in addition to any configured Power Up delay or trigger delay.

87.3.4 Clock operation

The LPADC operates from the ADCK clock input provided from an on-chip clock select block and is used by the SAR conversion control sequencing logic and the FIFO storage buffer. The ADCK frequency must fall within the specified frequency range for ADCK and will vary based on configuration of CFG[PWRSEL]. See the device datasheet for supported frequency ranges.

The LPADC continues operating in stop and wait modes provided the Doze Enable bit (CTRL[DOZEN]) is clear and the on-chip clock select block continues to supply an ADCK clock source. When the CTRL[DOZEN] bit is set, the module will wait for the current averaging iteration/FIFO storage to complete before acknowledging stop or wait mode entry.

87.3.5 Sample time, compare time, and total conversion time

Sample time is a time interval required to charge ADC sampling capacitance (bank of capacitances) by required ADC input channel voltage. Short sample time interval can cause the voltage across sampling capacitance at the end of sampling interval is not sufficient (lower that ADC input voltage) and consequentially it is used for ADC conversion which at the end results in inaccurate ADC results. Longer sample time interval affects resulting ADC sample rate. To get higher ADC sample rate (sample per second) with shorter sample time interval, it means to achieve best ADC performance analog source resistance RAS must be kept as low as possible, for example used operation amplifier out to be connected to ADC input. The sample time interval can be calculated based on STS field in ADC_CMDHx register and ADC clock:

Sample time interval = $(3+2^{STS})$*adc_clk

Compare (conversion) time is a time interval starting from sample time end (sample capacitance is charged by ADC input voltage) until conversion complete (ADC result is available in ADC result FIFO). Basically it represents a time during which ADC is executing successive approximation steps. This time interval is fixed to 17.5 ADC clock cycles (adc_clk).

Total conversion time represents a sum of sampling and compare time intervals. It is a time interval during which the conversion cannot be interrupted by another ADC trigger event to be able to finish successfully. The equation to calculate this is:

Total conversion time = sampling time + compare time

87.3.6 Trigger detect and command execution

See Figure 87-3 for a flow diagram of command execution sequencing.

LPADC command execution is initiated from up to 8 trigger sources. Each trigger can be software generated by writing 0b1 to the corresponding SWTRIG[SWT*n*] bitfield. Alternatively, hardware triggers can be generated from asynchronous input sources at the periphery of the module. The number and sources of hardware triggers implemented is device-specific. See the chip configuration information for description of available hardware trigger sources for this device.

Each hardware trigger source is enabled by setting the associated enable bit (TCTRL*a*[HTEN]). Each trigger source is assigned a priority via the associated priority control field (TCTRL*a*[TPRI]). Each of the trigger sources is configurably associated with a command in the command buffer via the associated command select field (TCTRL*a*[TCMD]).

When a hardware trigger input is enabled, hardware trigger events are detected on the rising-edge of the associated hardware trigger source.

If a trigger event occurs that is configured for a higher priority than the trigger source associated with the currently executing command, the current command is handled in one of two ways depending on the CFG[TPRICTRL] bitfield setting. When CFG[TPRICTRL]=0b0, a higher priority trigger causes an immediate command abort and the new command specified by the trigger is immediately started. When CFG[TPRICTRL]=0b1, a higher priority triggers allows the current conversion to complete (including averaging and compare functions) before the higher priority command is initiated.

If a lower priority trigger occurs (i.e., a trigger event occurs that is configured for a lower priority than the trigger source associated with the currently executing command), the trigger detect is pended until completion of the current command sequence.

At the completion of a command, the LPADC does not resume a previously interrupted command (or series of commands). The module begins command execution associated with the processing of lower priority pending triggers. If no trigger events are pending, the LPADC transitions to its idle state.

When a conversion is completed (including hardware averaging when AVGS is non-zero), the result is placed in the RESFIFO buffer. When an ADC command selects looping (when LOOP is non-zero) a command will store multiple conversion results to the FIFO during execution of that command.

At the end of command execution, the NEXT field of the command selects the next command to be executed. Multiple commands can be executed sequentially by configuration of each command's NEXT field. Setting the next command to 0x0 causes conversions to terminate at the completion of the command. Unending circular command execution is allowed by setting the NEXT field in the last command in a sequence to the first command in the sequence. It is also allowed for a command to set the next command to itself, resulting in a continuous conversion configuration. Lower priority trigger events cannot be serviced until a higher priority triggered command (or command sequence) completes.

Disabling the LPADC by writing 0b0 to the CTRL[ADCEN] bitfield terminates any active ADC command processing. Writing 0b0 to the CTRL[ADCEN] bitfield causes the current command (or command sequence) to terminate, clears any pending triggers, and sends the module to an Idle state.

87.3.7 Pause option

When the maximum conversion rate is not required by an application the effective conversion rate can be reduced by implementing periodic trigger events to initiate ADC conversions or by selecting a reduced frequency clock as the ADACK source. Both of these options are chip specific and are dependent on ADC triggering and clocking options external to the LPADC module. The latency associated with ADC analog power up delays results in a limit on the maximum conversion rate when using periodic triggering.

Another means of reducing conversion rates is by inserting a pause of a programmable duration between LOOP iterations, between commands in a sequence, and between conversions when command is executing in the "Compare Until True" configuration. When PAUSE[PAUSEEN] is set, the PAUSE[PAUSEDLY] field controls the duration of pausing during command execution sequencing. The pause delay is a count of (PAUSEDLY*4) ADCK cycles. The PAUSE register cannot be changed while the CTRL[ADCEN] bit is set. Writes to the PAUSE register while CTRL[ADCEN] is set are ignored.

See Figure 87-3 for the places during command execution sequencing where the pause is optionally inserted.

87.3.8 Result FIFO operation

The LPADC includes a 16 entry FIFO in which the result of ADC conversions are stored. In addition, a valid indicator bit, the trigger source, the source command and the loop count are also stored along with the data. FCTRL[FCOUNT] indicates how many valid datawords are stored in the RESFIFO.

A programmable watermark threshold supports configurable notification of data availability. When FCOUNT is greater than FCTRL[FWMARK], the STAT[RDY] flag is asserted. When IE[FWMIE] is set, a watermark interrupt request is issued. When DE[FWMDE] is set, a DMA request is issued. Reading RESFIFO provides the oldest unread dataword entry in the FIFO and decrements FCOUNT. When FCOUNT falls equal to or below FWMARK, the RDY flag is cleared.

The FIFO can be emptied by successive reads of RESFIFO. When the RESFIFO[VALID] bit is 1, the associated FIFO entry is valid. Reading RESFIFO when the FIFO is empty (when RESFIFO[VALID] is clear and FCOUNT=0x0) provides an undefined dataword. The FIFO is reset by writing 0b1 to the CTRL[RSTFIFO] bit.

If the LPADC attempts to store a dataword to the FIFO when the FIFO is full, the FIFO overflow flag (FCTRL[FOF]) is set. When IE[FOFIE] is set, an overflow interrupt request is issued. The STAT[FOF] flag is cleared by writing 1 to FOF. On overflow

events, no new data is stored and the data associated with the store that triggered the overflow is lost, the current ADC command (sequence) is aborted and all pending trigger events are discarded. No new triggers are detected until the overflow flag is cleared.

87.3.9 Compare function

After the input is sampled and converted and any averaging iterations are performed, the CMDH*a*[CMPEN] field guides operation of the automatic compare function to optionally only store when the compare operation is true. There are multiple options on command sequencing related to the compare function as summarized in the table below.

NOTE

Not all Command Buffers have an associated Compare Value register. The compare function is only available on Command Buffers that have a corresponding Compare Value register.

Table 87-4. Compare modes

CMDH*a*[CMPEN]	Compare Function	Description
0b00	Compare disabled	Do not perform compare operation. Always store the conversion result to the FIFO.
0b01	Reserved	
0b10	Store on true	Perform compare operation. Store conversion result to FIFO at end of comparison only if compare is true. If compare is false, do not store the result to the FIFO. In either the true or false condition, the LOOP setting is considered and increments the LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required.
0b11	Repeat compare until true	Perform compare operation. Store conversion result to FIFO at end of comparison only if compare is true. Once the true condition is found the LOOP setting is considered and increments the LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required. If the compare is false do not store the result to the FIFO. The conversion is repeated without consideration of LOOP setting and does not increment the LOOP counter.

Depending on CV*a*[CVH] and CV*a*[CVL] values programmed, the compare operation checks whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. The compare values are used as described in the following table.

Table 87-5. Compare operations

CV*a*[CVL] vs. CV*a*[CVH]	Operation	Description
set CVL < CVH	Outside range	Compare true if the result is less than CVL value or greater than CVH value.

Table continues on the next page...

Table 87-5. Compare operations (continued)

CV*a*[CVL] vs. CV*a*[CVH]	Operation	Description
	(General form)	
set CVH to max value set CVL to compare point	Less than	Compare true if the result is less than CVL value.
set CVL to min value set CVH to compare point	Greater than	Compare true if the result is greater than CVH value.
set CVL > CVH	Inside range	Compare true if the result is less than CVL value and greater than CVH value.

Note

In low power modes where the LPADC continues to operate, the compare function can monitor the voltage and only wake the device when the compare condition is met.

87.4 Signal descriptions

The LPADC module supports up to 64 analog channel inputs with differential and single-ended conversion options for all channels. The module also requires supply and ground connections.

Table 87-6. LPADC signal descriptions

Signal	Description	I/O
V_{REFH}	ADC Analog Voltage Reference High Input	I
V_{DDA}_ADC_3P3	ADC Analog Power Supply Input 3.3V	I
V_{DDA}_ADC_1P8	ADC Analog Power Supply Input 1.8V	I
V_{SSA}	Analog Ground, Shorted with common ground VSS in package	I
V_{REFL}	Analog Ground, Shorted with common ground VSS in package	I
CH30A-CH0A	A-side Analog Channel Inputs	I
CH30B-CH0B	B-side Analog Channel Inputs	I

NOTE

The voltage reference (V_{REFH}) used by the LPADC is supplied from either an on-chip voltage reference source or from an off-chip source supplied through external pins. V_{REFL} is always from an external pin and must be at the same voltage potential as V_{SSA}. This block supports a programmable selection of the Voltage Reference used for ADC conversions (via the

CFG[REFSEL] field). See the chip configuration information on the voltage reference options specific to this packaged device.

87.4.1 Analog Power (V_{DDA}_ADC_3P3 and V_{DDA}_ADC_1P8)

The ADC analog portion uses V_{DDA}_ADC_3P3 and V_{DDA}_ADC_1P8 as its power connection. If externally available, connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA}_ADC_3P3 and V_{DDA}_ADC_1P8 for good results.

87.4.2 Analog Ground (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground connection. In some packages, V_{SSA} is connected internally to V_{SS}. If externally available, connect the V_{SSA} pin to the same voltage potential as V_{SS}.

87.4.3 Analog Channel Inputs (CH*n*A and CH*n*B)

The CMDL*a*[ADCH], CMDL*a*[DIFF] and CMDL*a*[ABSEL] bitfields control selection of paired or individual input channels. Each ADC command independently makes a channel selection. Each ADCH channel selection has an associated A side and an associated B side input. Each ADCH pair can optionally be converted in a differential mode but only limited pairs are intended to be converted as differential channels (i.e., adjacent pins that have been designed with matched impedance). For the pin pairings available for differential conversions for your device, see the chip-specific information.

NOTE

Some of the input channels are from on-chip sources such as temperature sensors and reference voltage sources and may only be connected to individual instances of the ADC module. Some of the input channel options in the bitfield-setting descriptions may not be available for your device. Refer to the chip-specific information for the channels supported on this device.

87.5 Initialization

This section gives an example that provides some basic direction on how to initialize the LPADC module. The user can configure the module for 12-bit single-ended resolution or 13-bit differential resolution, single or continuous conversion, and a polled or interrupt approach, among many other options.

87.5.1 LPADC module initialization example

LPADC module should be properly initialized before it is configured according the application needs. It is important to get the LPADC module to work properly from its first conversion.

87.5.2 Initialization sequence

Before the LPADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is:

1. ADC clock needs to be enabled prior writing into the LPADC registers. Set the ADC clock considering target application sample and conversion time requirements then enable ADC clock in CCM module.
2. Reset LPADC module by writing 1 (keep module in reset) and then 0 (release from reset) into the RST field in control register CTRL.
3. Reset ADC results FIFO by writing 1 into RSTFIFO field in control register CTRL.
4. Disable LPADC module (if not already) by writing 0 ADCEN field in control register CTRL. Keep ADC module disabled until not fully initialized.
5. Select proper voltage reference by writing appropriate value into the REFSEL field in configuration register CFG.
6. Configure correct LPADC module start-up, power and performance according to application needs.
    a. To achieve lower power consumption disable pre-enabling of ADC analog circuitry by writing 0 into the PWREN field in ADC configuration register CFG. Then based of application requirement configure power up delay by writing appropriate value in PUDLY field in ADC configuration register. Set lowest power configuration by writing 0 into PWRSEL field in ADC configuration register CFG.
    b. To achieve high ADC conversion rate enable pre-enabling of ADC circuitry by writing 1 into the PWREN field in ADC configuration register CFG. Set the

highest power setting by writing 0x3 into the PWRSEL field in ADC configuration register ADC->CFG.

c. For balanced configuration between the low power configuration and faster ADC speed follow the configuration of PWREN, PUDLY and PWRSEL fields somewhere between previous descripted configurations (a., b.).

NOTE

Steps 5 and 6 can be done at one time.

7. Enable ADC module by writing 1 into the ADCEN field in control register CTRL.

NOTE

Do not initiate any trigger for ADC conversion start prior its proper full configuration.

87.6 Application information

This section contains information for using the LPADC module in applications. The LPADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an A/D converter.

NOTE

This part represents general application consideration. For chip specific features see chapter chip-specific information at the beginning of LPADC module chapter or dedicated device datasheet.

87.6.1 External pins and routing

Very important role regarding to accurate ADC measurement plays components connected to LPADC module externally, including ADC analog module power supply, ADC reference as well as voltage sources connected to ADC inputs. All these voltage sources have their capability and impedances which can significantly affect measurement accuracy. Moreover the total source impedance is affected also by routing on PCB. In general, the rule is to minimize parasitic impedances (especially resistive and inductive) as much as possible to achieve better ADC performance.

Any noise generated on ADC reference or ADC input affects resulting digital result directly as defined by:

ADC result = $(V_{ADIN}/V_{ADCREF})(2^N-1)$

Where V_{ADCIN} represents the voltage at ADC input pin and V_{ADCREF} represents analog voltage reference at reference pin. The noise generated on analog power supply can also affect ADC measurement accuracy indirectly as it represents analog peripheral modules power supply.

NOTE

On some devices, there is only one power ground available externally, usually as V_{SS}. In such case digital ground (V_{SS}), analog ground (V_{SSA}) and negative reference voltage (VREFL) are tied together inside package.

87.6.1.1 Analog supply pins

Depending on the device, the analog power and analog ground pads of the LPADC module can be available as:

- Separate pins - when available on a separate pin, analog ground must be connected to the same voltage potential as digital ground, and analog power supply is recommended to be impedance isolated from digital power supply by ferrite bead, for example. Both analog power and analog ground must be routed carefully for maximum noise immunity and bypass capacitors must be placed as near as possible to the pin package.
- Analog ground is shared on the same pin as the MCU digital ground (connected internally).
- Analog power and analog ground are both shared with the MCU digital power and ground pins (connected internally) - In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the analog power pin. This must be the only ground connection between these supplies, if possible. Analog ground makes a good single point ground location.

87.6.1.2 Analog voltage reference pins

In addition to the analog supplies, the LPADC module can have connections to selectable reference voltage inputs used by the converter:

- V_{REFH} is the high reference voltage for the converter.
- V_{REFL} is the low reference voltage for the converter.

The ADC can be configured to accept more than one voltage reference pairs. Each pair contains a positive reference and a ground reference and they can represents additional external pins or internal sources based on MCU configuration (if reference voltage peripheral module available on device). One of these positive references may be shared on the same pin as analog power supply. As well as one of these ground references may be shared on the same pin as analog ground. It is chip specific, see the chip-specific information on the voltage references specific to this device.

If externally available, the positive reference may be driven by an external source to a level between the minimum VREFH and analog power supply potential. The positive reference VREFH must never exceed analog power supply level (see also device datasheet). If externally available, the ground reference must be connected to the same voltage potential as analog power supply ground. The voltage reference pairs must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the VREFH and VREFL loop. The best external component value to meet this current demand can be found in corresponding device hardware development guide (HDUG). This capacitor is connected between VREFH and VREFL and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum, thus reduce parasitic inductance as much as possible.

87.6.1.3 Analog input pins

The external analog inputs are typically shared with digital I/O pins on MCU devices. Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used, they must be placed as near as possible to the package pins and be referenced to analog power ground. For proper conversion, the input voltage must fall between VREFH and VREFL. If corresponding device support scale factor (CSCALE) the maximum positive input voltage is defined as:

V_{INMAX} = (VREFH-VREFL)/Factor

For factor definition see CSCALE field description. If the input is equal to or exceeds VREFH, the converter circuit converts the signal to 0x7FFF (or 0xFFFF), which is full scale 12-bit single ended (or 13-bit differential) representation. If the input is equal to or less than VREFL, the converter circuit converts it to 0x000. Input voltages between

VREFH and VREFL are straight-line linear conversions. There is a brief current associated with VREFL when the sampling capacitor is charging. For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins must not be transitioning during conversions.

87.6.2 Sources of error

Several sources of error exist for A/D conversions. These are discussed in the following sections.

NOTE

In the next sections, the errors description does not consider the LPADC module data result format, meaning 12-bit (13-bit) ADC resolution is represented by 16-bit results. Hence, one step up/down in digital results does not represents 1 digit (0b1) but 8 digits (0b1000).

87.6.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Higher source resistances or higher-accuracy sampling is possible by setting STS field in LPADC command high buffer register CMDHx (to increase the sample window) or decreasing ADC clock frequency to increase sample time. Care must be taken to not exceed the maximum analog source resistance R_{AS} of 5kohm which is connected externally. To achieve best performance analog source resistance R_{AS} must be kept as low as possible and time constant of external components (R_{AS} and C_{AS}) must be kept < 1ns. For example: datasheet values are achieved with RAS < 15.

NOTE

For guidance on selecting optimum external component values and converter parameters see AN4373: Cookbook for SAR ADC Measurements.

87.6.2.2 Pin leakage error

Leakage on the I/O pins can cause conversion error if the external analog source resistance (R_{AS}) is high. If this error cannot be tolerated by the application, keep R_{AS} :

R_{AS} < (analog power supply voltage)/(2*N*$I_{leakage}$)

$I_{leakage}$ can be found in corresponding device datasheet and N represents ADC resolution 12-bit. With this value of R_{AS} it is possible to achieve less than 1/4LSB leakage error.

87.6.2.3 Noise-induced errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is dedicated value (see corresponding device hardware development user guide (HDUG)) high dynamic capable and low-ESR capacitor connected as close as possible to VREFH and VREFL (if available externally in package).
- There is dedicated value (see corresponding device hardware development user guide (HDUG)) high dynamic capable and low-ESR capacitor from analog power supply to analog power ground.
- If inductive isolation is used from the primary supply, an additional 1 µF capacitor is placed from analog power supply to analog power ground.
- Analog ground (and VREFL, if available externally) is connected to digital ground at a quiet point in the ground plane.
- Operate the MCU in wait or stop mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion. For software triggered conversions, immediately follow the write to software trigger register (SWTRIG) with a wait instruction or stop instruction.
- There is no I/O switching, input or output, on the MCU during the conversion. There are some situations where external system activity causes radiated or conducted noise emissions or excessive digital power supply noise is coupled into the LPADC. In these situations, or when the MCU cannot be placed in wait or stop or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:
  - Place a 0.01 µF capacitor (C_{AS}) between the selected ADC input and ADC voltage reference low VREFL (if available externally) or analog power ground. This improves noise issues, but affects the sample rate based on the external analog source resistance.
  - Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1LSB, one-time error.

87.6.2.4 Code width and quantization error

The LPADC quantizes the ideal straight-line transfer function into 4096 steps (in 12-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N bit converter (in this case N can be 12), defined as 1LSB, is:

$1LSB = (VREFH-VREFL)/(2^N)$

There is an inherent quantization error due to the digitization of the result. The code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be +/- 1/2 LSB (see figure below). As a consequence, however, the code width of the first (0x000) conversion is only 1/2 LSB and the code width of the last 0x7FFF (0xFFFF) is 1.5 LSB.

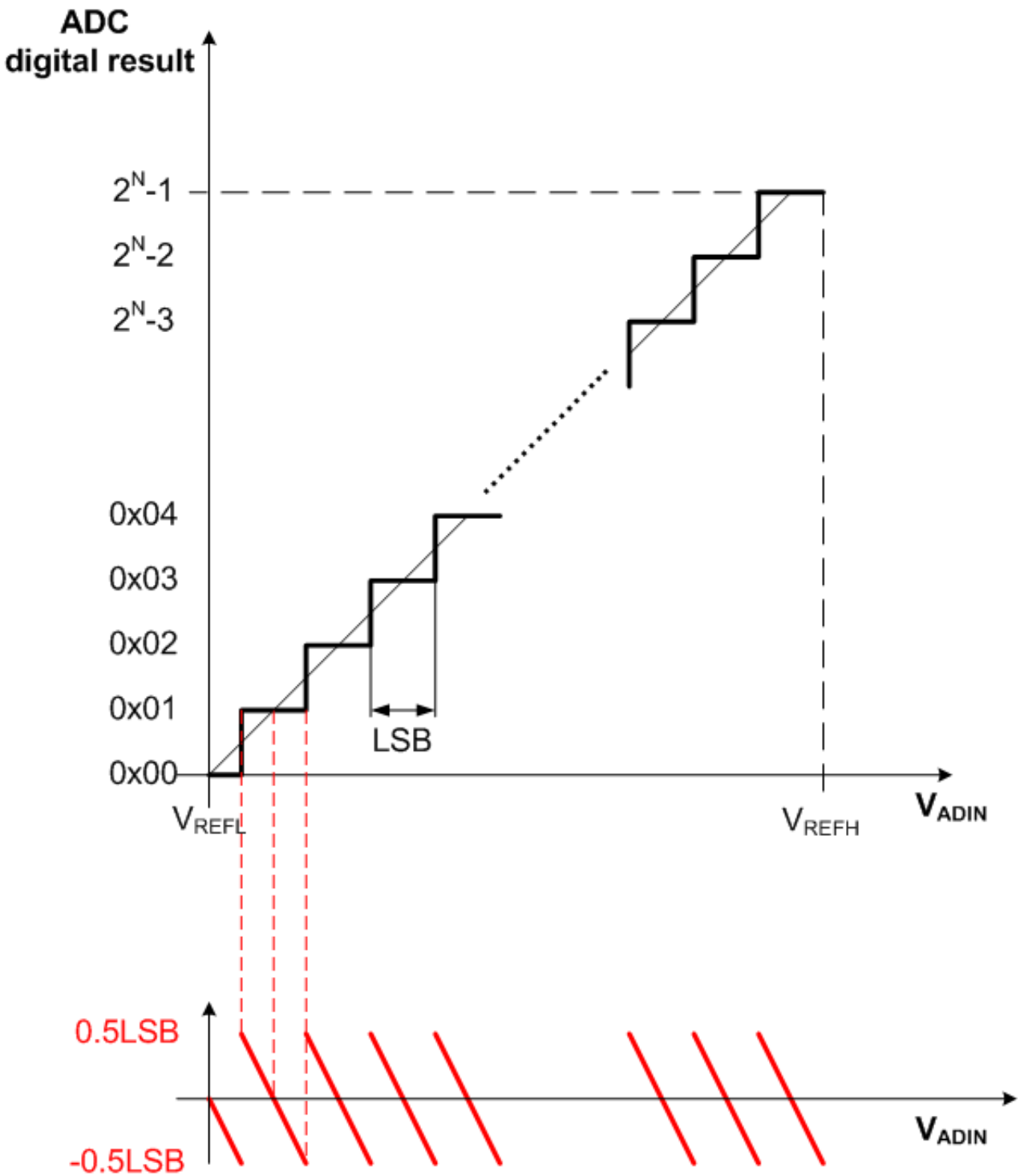


Figure 87-5. Quantization error

87.6.2.5 Linearity errors

The LPADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors but the system should be aware of them because they affect overall accuracy.

These errors are:

- Zero-scale error (EZS) (sometimes called offset) — This error is defined as the difference between the actual code width of the first conversion and the ideal code width. If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal is used.

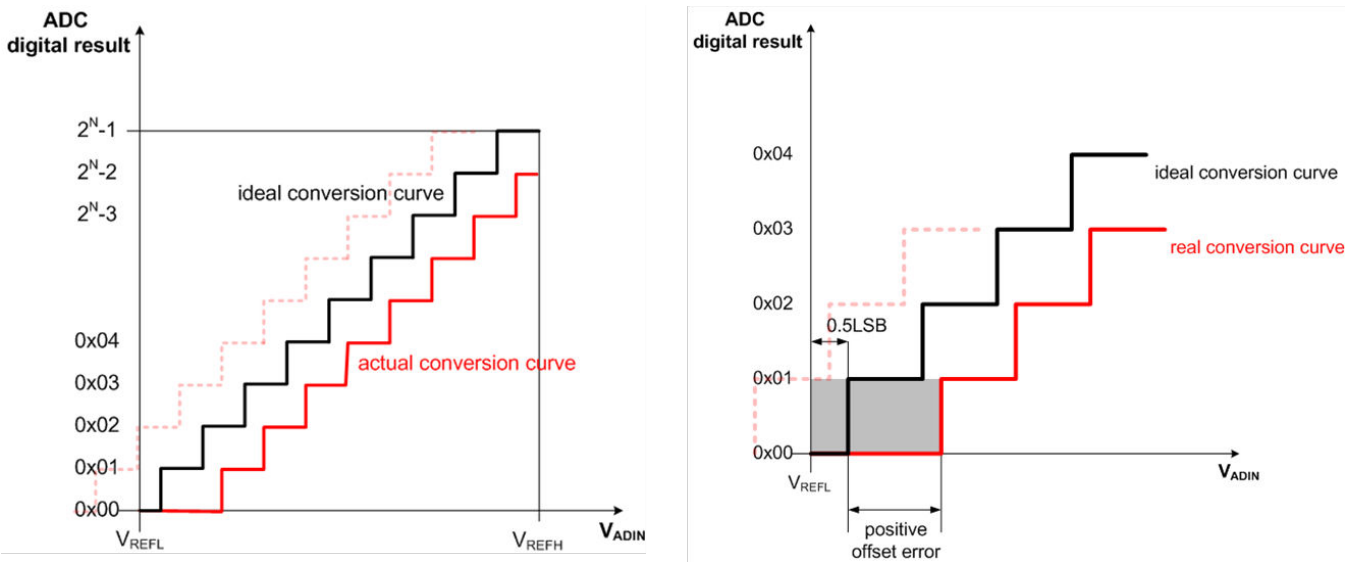


Figure 87-6. Zero-scale (offset) error

- Full-scale error (EFS) — This error is defined as the difference between the actual code width of the last conversion and the ideal code width. If the last conversion is 0x7FFF (or 0xFFFF), the difference between the actual 0x7FFF (or 0xFFFF) code width and its ideal is used.

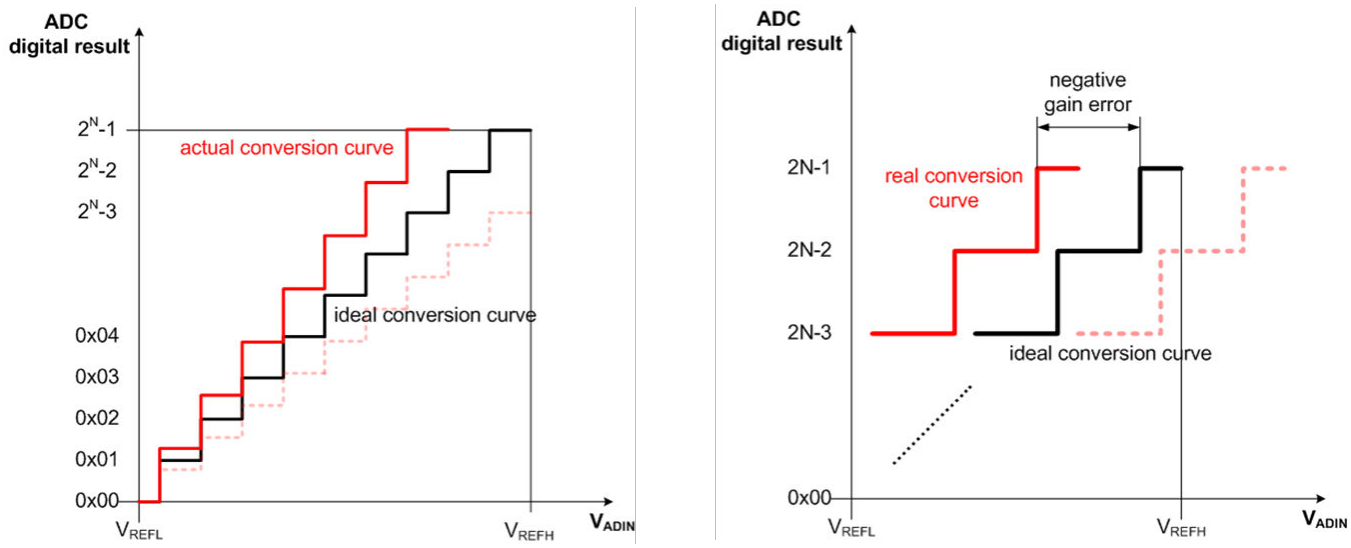


Figure 87-7. Full-scale (gain) error

- Differential non-linearity (DNL) — This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.

$$DNL = MAX\left\{\frac{width_i - 1LSB}{1LSB}\right\}$$

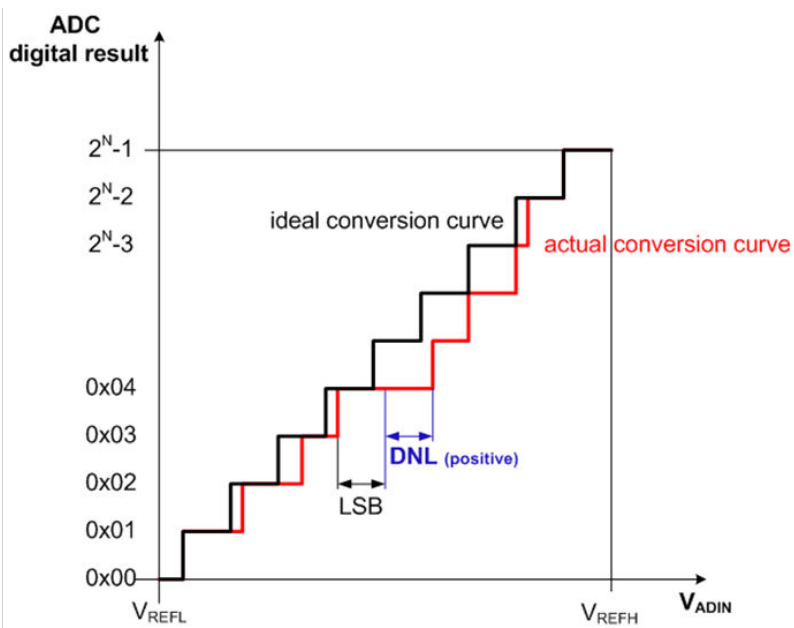


Figure 87-8. Differential non-linearity error

- Integral non-linearity (INL) — This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.

$$INL = MAX\left\{\frac{\sum_{i=0}^{N} width_i - N.LSB}{1LSB}\right\}$$

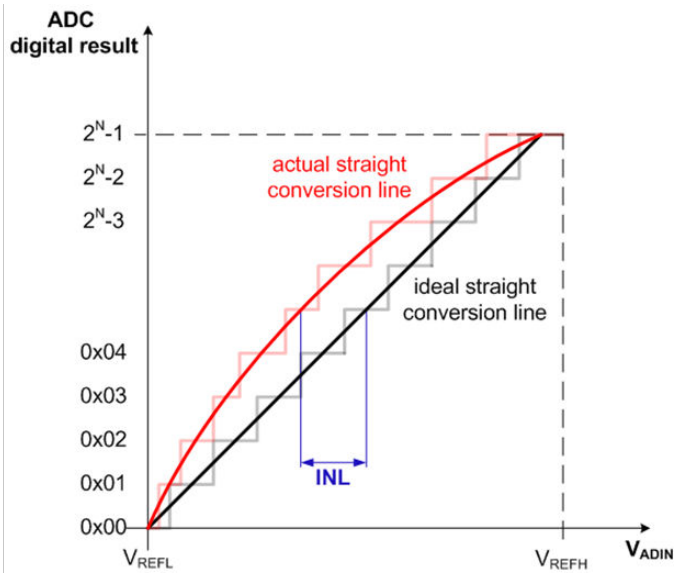


Figure 87-9. Integral non-linearity error

- Total unadjusted error (TUE) — This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

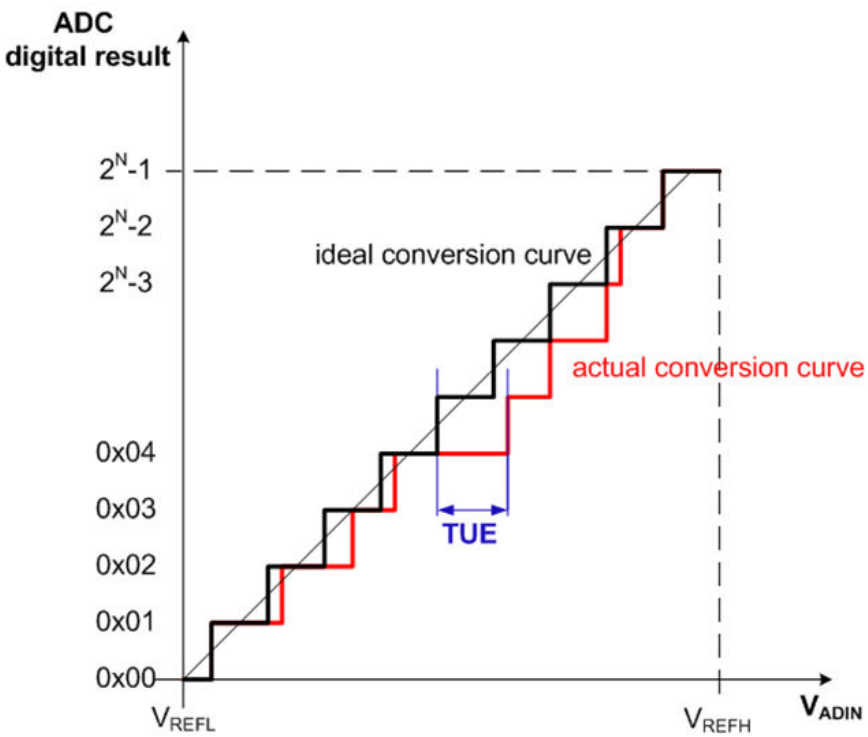


Figure 87-10. Total unadjusted error

87.6.2.6 Code jitter, non-monotonicity, and missing codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes. Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code (and vice-versa). However, even small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is around 2 LSB in 12-bit mode, and increases with noise. This error may be reduced by repeatedly sampling the input and averaging the result. Additionally the techniques discussed in Noise-Induced Errors reduces this error.

- Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage.
- Missing codes are those values never converted for any input value.

87.7 Memory map and register definition

This section includes the LPADC module memory map and detailed descriptions of all registers.

87.7.1 LPADC register descriptions

This section describes the LPADC registers.

87.7.1.1 LPADC memory map

LPADC1 base address: 4005_0000h

LPADC2 base address: 4005_4000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	Table 87-6	0100_001Ah

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Parameter Register (PARAM)	32	RO	0F04_1008h
10h	LPADC Control Register (CTRL)	32	Table 87-6	0000_0000h
14h	LPADC Status Register (STAT)	32	Table 87-6	0000_0000h
18h	Interrupt Enable Register (IE)	32	Table 87-6	0000_0000h
1Ch	DMA Enable Register (DE)	32	Table 87-6	0000_0000h
20h	LPADC Configuration Register (CFG)	32	Table 87-6	0080_0000h
24h	LPADC Pause Register (PAUSE)	32	Table 87-6	0000_0000h
30h	LPADC FIFO Control Register (FCTRL)	32	Table 87-6	0000_0000h
34h	Software Trigger Register (SWTRIG)	32	Table 87-6	0000_0000h
C0h - DCh	Trigger Control Register (TCTRL0 - TCTRL7)	32	Table 87-6	0000_0000h
100h	LPADC Command Low Buffer Register (CMDL1)	32	Table 87-6	0000_2000h
104h	LPADC Command High Buffer Register (CMDH1)	32	Table 87-6	0000_0000h
108h	LPADC Command Low Buffer Register (CMDL2)	32	Table 87-6	0000_2000h
10Ch	LPADC Command High Buffer Register (CMDH2)	32	Table 87-6	0000_0000h
110h	LPADC Command Low Buffer Register (CMDL3)	32	Table 87-6	0000_2000h
114h	LPADC Command High Buffer Register (CMDH3)	32	Table 87-6	0000_0000h
118h	LPADC Command Low Buffer Register (CMDL4)	32	Table 87-6	0000_2000h
11Ch	LPADC Command High Buffer Register (CMDH4)	32	Table 87-6	0000_0000h
120h	LPADC Command Low Buffer Register (CMDL5)	32	Table 87-6	0000_2000h
124h	LPADC Command High Buffer Register (CMDH5)	32	Table 87-6	0000_0000h
128h	LPADC Command Low Buffer Register (CMDL6)	32	Table 87-6	0000_2000h
12Ch	LPADC Command High Buffer Register (CMDH6)	32	Table 87-6	0000_0000h
130h	LPADC Command Low Buffer Register (CMDL7)	32	Table 87-6	0000_2000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
134h	LPADC Command High Buffer Register (CMDH7)	32	Table 87-6	0000_0000h
138h	LPADC Command Low Buffer Register (CMDL8)	32	Table 87-6	0000_2000h
13Ch	LPADC Command High Buffer Register (CMDH8)	32	Table 87-6	0000_0000h
140h	LPADC Command Low Buffer Register (CMDL9)	32	Table 87-6	0000_2000h
144h	LPADC Command High Buffer Register (CMDH9)	32	Table 87-6	0000_0000h
148h	LPADC Command Low Buffer Register (CMDL10)	32	Table 87-6	0000_2000h
14Ch	LPADC Command High Buffer Register (CMDH10)	32	Table 87-6	0000_0000h
150h	LPADC Command Low Buffer Register (CMDL11)	32	Table 87-6	0000_2000h
154h	LPADC Command High Buffer Register (CMDH11)	32	Table 87-6	0000_0000h
158h	LPADC Command Low Buffer Register (CMDL12)	32	Table 87-6	0000_2000h
15Ch	LPADC Command High Buffer Register (CMDH12)	32	Table 87-6	0000_0000h
160h	LPADC Command Low Buffer Register (CMDL13)	32	Table 87-6	0000_2000h
164h	LPADC Command High Buffer Register (CMDH13)	32	Table 87-6	0000_0000h
168h	LPADC Command Low Buffer Register (CMDL14)	32	Table 87-6	0000_2000h
16Ch	LPADC Command High Buffer Register (CMDH14)	32	Table 87-6	0000_0000h
170h	LPADC Command Low Buffer Register (CMDL15)	32	Table 87-6	0000_2000h
174h	LPADC Command High Buffer Register (CMDH15)	32	Table 87-6	0000_0000h
200h - 20Ch	Compare Value Register (CV1 - CV4)	32	RW	0000_0000h
300h	LPADC Data Result FIFO Register (RESFIFO)	32	Table 87-7	0000_0000h

87.7.1.2 Version ID Register (VERID)

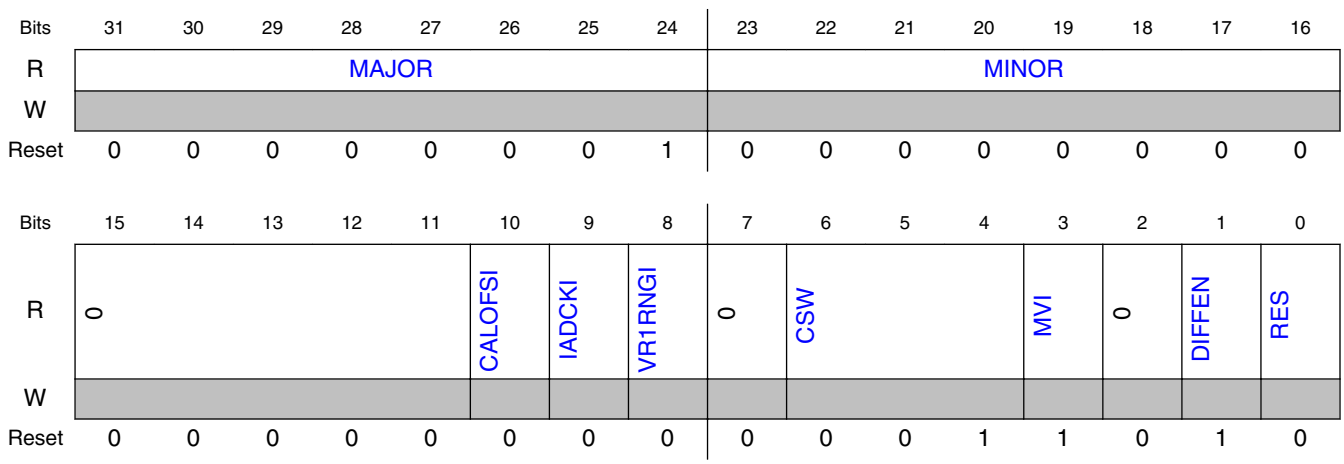
87.7.1.2.1 Offset

Register	Offset
VERID	0h

87.7.1.2.2 Function

The Version ID Register (VERID) indicates the version integrated for this instance on the device and also indicates inclusion/exclusion of several optional features.

87.7.1.2.3 Diagram



87.7.1.2.4 Fields

Field	Function
31-24 MAJOR	Major Version Number This read-only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number This read-only field returns the minor version number for the module specification.
15-11 —	Reserved This read-only field is reserved and always has the value 0.
10 CALOFSI	Calibration Offset Function Implemented This read-only field indicates if a calibration function and trimming control register are implemented. When supported, the CFG[CALOFS] and OFSTRIM[OFSTRIM] fields are available in support of this function. 0b - Offset calibration and offset trimming not implemented. 1b - Offset calibration and offset trimming implemented.
9	Internal LPADC Clock implemented

Table continues on the next page...

Field	Function
IADCKI	This read-only field indicates if this implementation of the LPADC includes an internal clock source. When supported, the CFG[ADCKEN] field is available and documented for enabling the clock source. When available, this clock source is used in clock selection logic external to the module for use within the system. 0b - Internal clock source not implemented. 1b - Internal clock source (and CFG[ADCKEN]) implemented.
8 VR1RNGI	Voltage Reference 1 Range Control Bit Implemented This read-only field indicates if a control bit is implemented for selecting the input voltage range on Voltage Reference Option 1.When range control is required, the CFG[VREF1RNG] field is available and documented for controlling the Voltage Reference Option 1. 0b - Range control not required. CFG[VREF1RNG] is not implemented. 1b - Range control required. CFG[VREF1RNG] is implemented.
7 —	Reserved This read-only field is reserved and always has the value 0.
6-4 CSW	Channel Scale Width This read-only field indicates if channel scaling is supported by this implementation. When supported, each command buffer has a control field (CMDLa[CSCALE]) for setting input scaling. 000b - Channel scaling not supported. 001b - Channel scaling supported. 1-bit CSCALE control field. 110b - Channel scaling supported. 6-bit CSCALE control field.
3 MVI	Multi Vref Implemented This read-only field indicates if multiple Voltage Reference inputs are supported by this implementation. When multiple voltage references are supported, the CFG[REFSEL] field is available and documented for selecting the Voltage Reference High options. 0b - Single voltage reference input supported. 1b - Multiple voltage reference inputs supported.
2 —	Reserved This read-only field is reserved and always has the value 0.
1 DIFFEN	Differential Supported This read-only field indicates if differential operation is supported by this implementation. When supported, each command buffer has control fields (CMDLa[DIFF] and CMDLa[ABSEL]) for configuring for differential operation and expanding the number of supported channels from 32 to 64. 0b - Differential operation not supported. 1b - Differential operation supported. CMDLa[DIFF] and CMDLa[ABSEL] control fields implemented.
0 RES	Resolution This read-only field indicates the maximum accuracy supported by this implementation. When RES=1, each command buffer has a control field (CMDLa[MODE]) for selecting resolution of conversions for the associated command. 0b - Up to 13-bit differential/12-bit single ended resolution supported. 1b - Up to 16-bit differential/15-bit single ended resolution supported.

87.7.1.3 Parameter Register (PARAM)

87.7.1.3.1 Offset

Register	Offset
PARAM	4h

87.7.1.3.2 Function

The Parameter Register (PARAM) indicates the size of several variable integration options for this instance on the device.

87.7.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMD_NUM								CV_NUM							
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOSIZE								TRIG_NUM							
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

87.7.1.3.4 Fields

Field	Function
31-24 CMD_NUM	Command Buffer Number Number of command buffers implemented. 0000_1111b - 15 command buffers implemented
23-16 CV_NUM	Compare Value Number Number of compare value registers implemented. 0000_0100b - 4 compare value registers implemented
15-8 FIFOSIZE	Result FIFO Depth The maximum number of conversion datawords that can be stored in the result FIFO before an overflow occurs. This field is read-only. 0001_0000b - Result FIFO depth = 16 datawords.
7-0 TRIG_NUM	Trigger Number Number of Triggers implemented. 0000_1000b - 8 hardware triggers implemented

87.7.1.4 LPADC Control Register (CTRL)

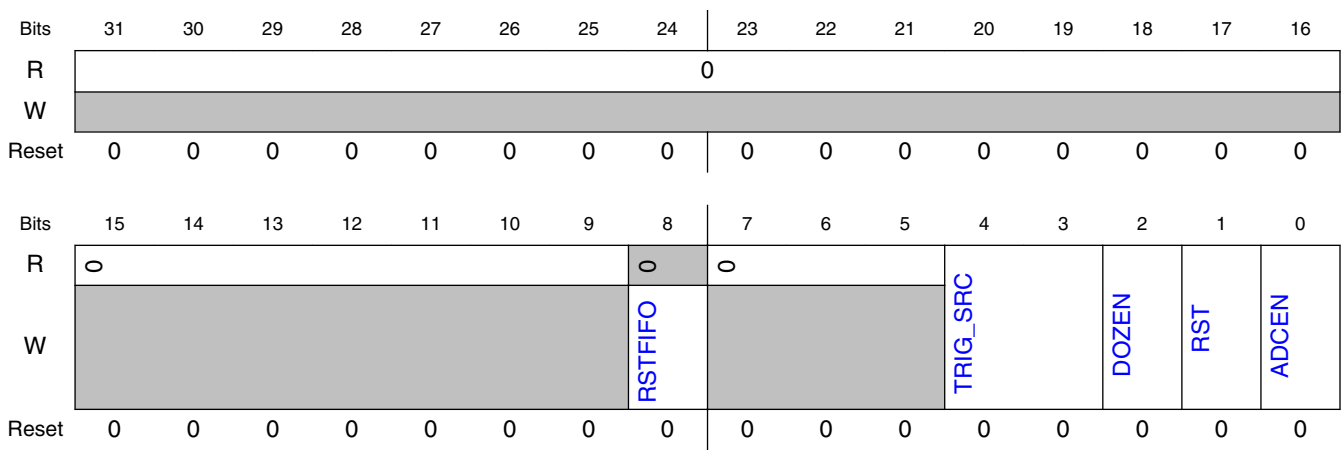
87.7.1.4.1 Offset

Register	Offset
CTRL	10h

87.7.1.4.2 Function

The LPADC Control Register (CTRL) controls various basic functions of the LPADC, such as enables of the ADC and Doze, reset of FIFO and software, and selection of hardware trigger source.

87.7.1.4.3 Diagram



87.7.1.4.4 Fields

Field	Function
31-9 —	Reserved This read-only field is reserved and always has the value 0.
8 RSTFIFO	Reset FIFO 0b - No effect. 1b - FIFO is reset.
7-5 —	Reserved This read-only field is reserved and always has the value 0.
4-3 TRIG_SRC	Hardware trigger source selection 00b - ADC_ETC hw trigger , and HW trigger are enabled 01b - ADC_ETC hw trigger is enabled 10b - HW trigger is enabled

Table continues on the next page...

Field	Function
	11b - Reserved
2 DOZEN	Doze Enable Controls system transition to Stop and Wait power modes while LPADC is converting. When DOZEN is clear, immediate entries to Wait or Stop are allowed. Note that the selected clock source provided from the on-chip clock source must be able to continue operating. When the DOZEN bit is set, the LPADC waits for the current compare iteration/FIFO storage to complete before acknowledging stop or wait mode entry. When the system is entering a Low Leakage Stop mode, the DOZEN bit is ignored and the module waits for the current transfer to complete any pending operation before acknowledging low leakage mode entry. 0b - LPADC is enabled in Doze mode. 1b - LPADC is disabled in Doze mode.
1 RST	Software Reset Resets all internal logic and registers, except the Control Register. Remains set until cleared by software. 0b - LPADC logic is not reset. 1b - LPADC logic is reset.
0 ADCEN	LPADC Enable 0b - LPADC is disabled. 1b - LPADC is enabled.

87.7.1.5 LPADC Status Register (STAT)

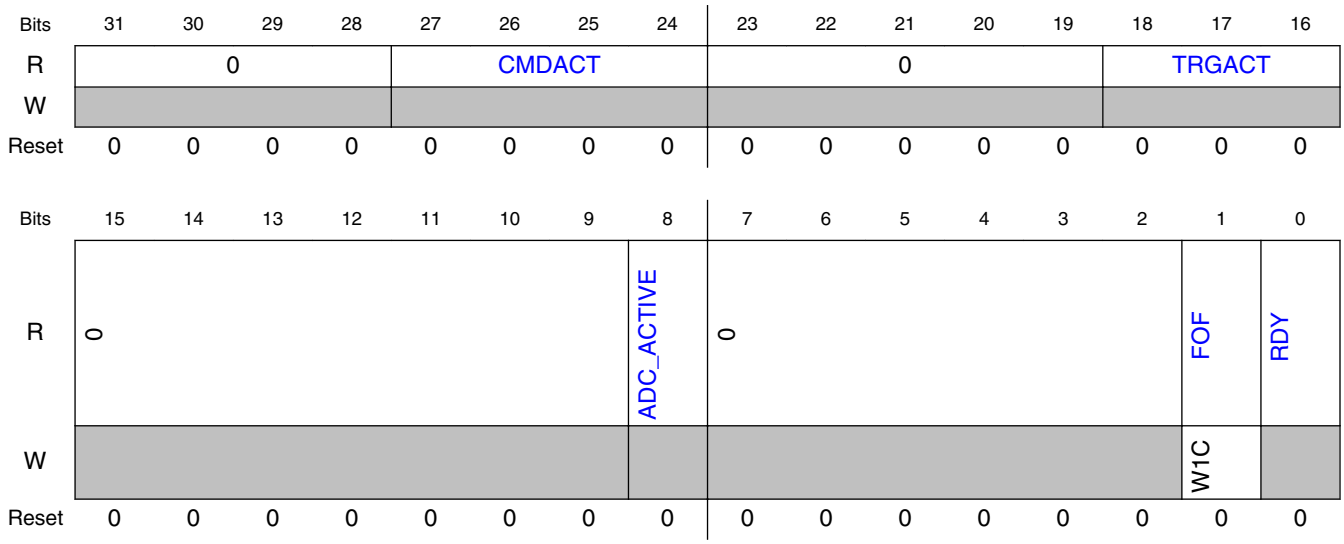
87.7.1.5.1 Offset

Register	Offset
STAT	14h

87.7.1.5.2 Function

The LPADC Status Register (STAT) provides the current status of the LPADC module.

87.7.1.5.3 Diagram



87.7.1.5.4 Fields

Field	Function
31-28 —	Reserved This read-only field is reserved and always has the value 0.
27-24 CMDACT	Command Active CMDACT is a read-only status field indicating the command that is actively being processed. 0000b - No command is currently in progress. 0001b - Command 1 currently being executed. 0010b - Command 2 currently being executed. 0011b-1111b - Associated command number is currently being executed.
23-19 —	Reserved This read-only field is reserved and always has the value 0.
18-16 TRGACT	Trigger Active TRGACT is a read-only status field indicating the trigger associated with the command actively being processed. TRGACT only has meaning when CMDACT is non-zero. 000b - Command (sequence) associated with Trigger 0 currently being executed. 001b - Command (sequence) associated with Trigger 1 currently being executed. 010b - Command (sequence) associated with Trigger 2 currently being executed. 011b-111b - Command (sequence) from the associated Trigger number is currently being executed.
15-9 —	Reserved This read-only field is reserved and always has the value 0.
8 ADC_ACTIVE	ADC Active This flag shows if the module is currently processing a conversion, or has pending triggers to service. When the LPADC is IDLE, this bit will be set to 0b0. 0b - The LPADC is IDLE. There are no pending triggers to service and no active commands are being processed.

Table continues on the next page...

Field	Function
	1b - The LPADC is processing a conversion, running through the power up delay, or servicing a trigger.
7-2 —	Reserved This read-only field is reserved and always has the value 0.
1 FOF	Result FIFO Overflow Flag Indicates that more data has been written to the Result FIFO than it can hold. The newer data is not stored and the FIFO remains holding the original contents. This field asserts regardless of the value of IE[FOFIE]. However, an interrupt request is issued only if IE[FOFIE] is set. This flag is cleared by writing a 1. 0b - No result FIFO overflow has occurred since the last time the flag was cleared. 1b - At least one result FIFO overflow has occurred since the last time the flag was cleared.
0 RDY	Result FIFO Ready Flag Indicates when the number of valid datawords in the result FIFO is greater than the watermark level set in the FCTRL[FWMARK] bitfield. This field asserts regardless of the value of FWMIE. However, an interrupt request or DMA request occurs only when the associated control bit (IE[FWMIE] or DE[FWMDE]) is set. This flag is cleared when the FCOUNT (which decrements on each read of the RESFIFO register) is less than or equal to the watermark level set in the FCTRL[FWMARK] bitfield. 0b - Result FIFO data level not above watermark level. 1b - Result FIFO holding data above watermark level.

87.7.1.6 Interrupt Enable Register (IE)

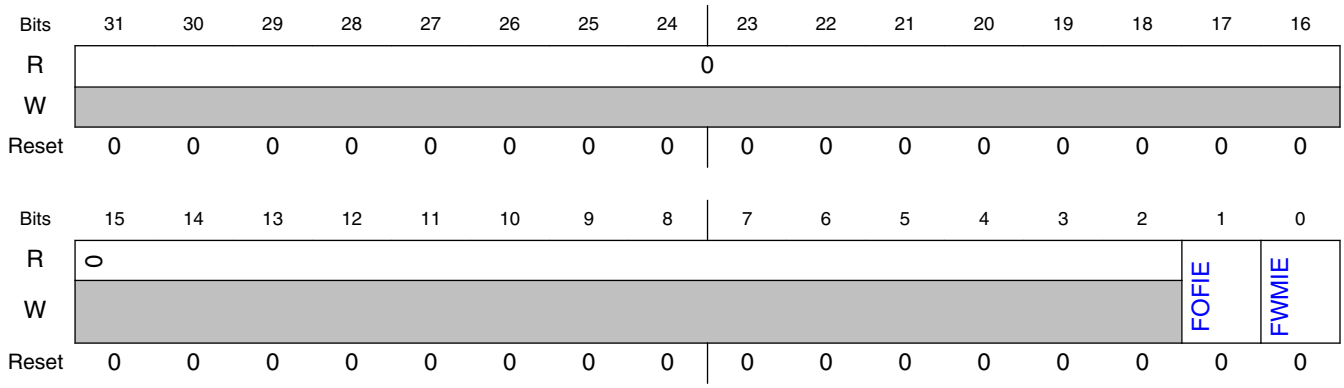
87.7.1.6.1 Offset

Register	Offset
IE	18h

87.7.1.6.2 Function

The Interrupt Enable Register (IE) controls the interrupt enable function of FIFO Watermark and Result FIFO Overflow.

87.7.1.6.3 Diagram



87.7.1.6.4 Fields

Field	Function
31-2 —	Reserved This read-only field is reserved and always has the value 0.
1 FOFIE	Result FIFO Overflow Interrupt Enable Configures ADC to generate overflow interrupt requests when FOF flag is asserted. 0b - FIFO overflow interrupts are not enabled. 1b - FIFO overflow interrupts are enabled.
0 FWMIE	FIFO Watermark Interrupt Enable Configures the module to generate watermark interrupt requests when RDY flag is asserted. 0b - FIFO watermark interrupts are not enabled. 1b - FIFO watermark interrupts are enabled.

87.7.1.7 DMA Enable Register (DE)

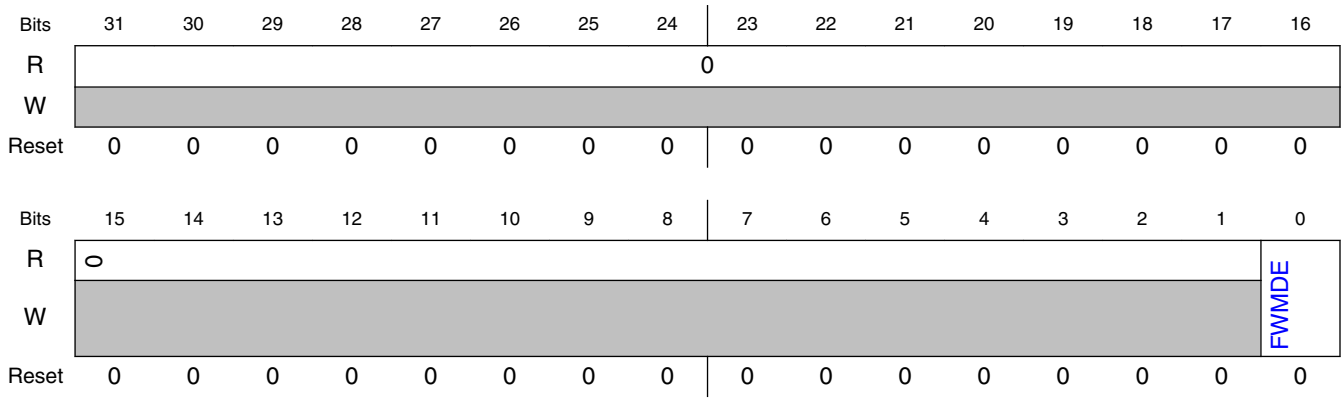
87.7.1.7.1 Offset

Register	Offset
DE	1Ch

87.7.1.7.2 Function

The DMA Enable Register (DE) controls the DMA enable function of FIFO Watermark.

87.7.1.7.3 Diagram



87.7.1.7.4 Fields

Field	Function
31-1 —	Reserved This read-only field is reserved and always has the value 0.
0 FWMDE	FIFO Watermark DMA Enable Configures ADC to generate DMA requests when RDY flag is asserted. 0b - DMA request disabled. 1b - DMA request enabled.

87.7.1.8 LPADC Configuration Register (CFG)

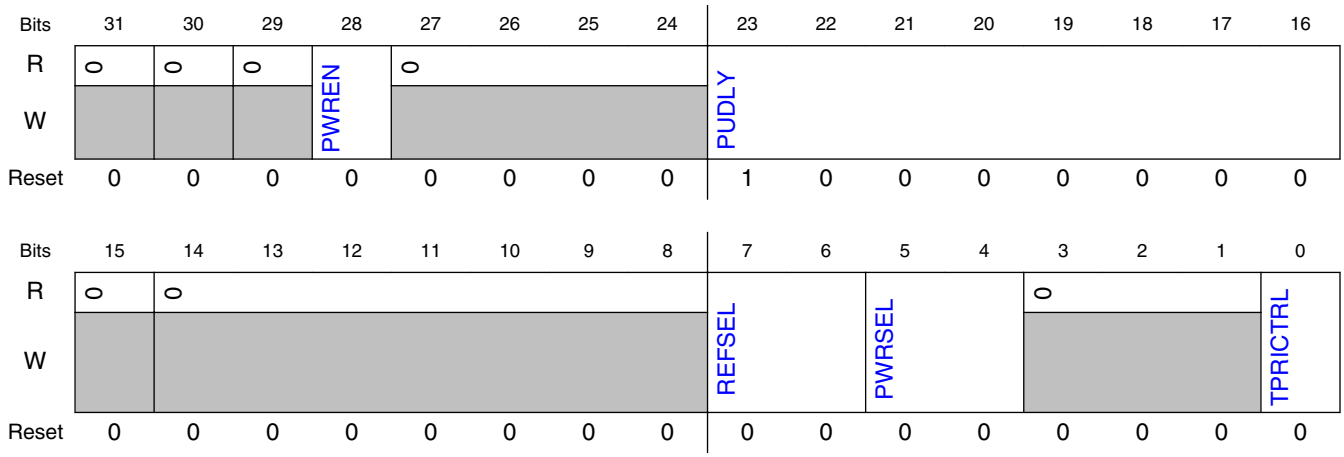
87.7.1.8.1 Offset

Register	Offset
CFG	20h

87.7.1.8.2 Function

The LPADC Configuration Register (CFG) controls ADC functions that are common to all commands. The CFG cannot be changed while the CTRL[ADCEN] bit is set. Writes to CFG while CTRL[ADCEN] is set are ignored.

87.7.1.8.3 Diagram



87.7.1.8.4 Fields

Field	Function
31 —	Reserved This read-only field is reserved and always has the value 0.
30 —	Reserved This read-only field is reserved and always has the value 0.
29 —	Reserved This read-only field is reserved and always has the value 0.
28 PWREN	LPADC Analog Pre-Enable Enables the LPADC analog circuits. When setting CFG[PWREN], user code should delay for a period exceeding the analog startup time of $t_{ADCSTUP}$ before enabling the LPADC for operation. The module is still operational even when CFG[PWREN] is left clear but command execution start is delayed for a period defined by CFG[PUDLY]. 0b - LPADC analog circuits are only enabled while conversions are active. Performance is affected due to analog startup delays. 1b - LPADC analog circuits are pre-enabled and ready to execute conversions without startup delays (at the cost of higher DC current consumption). When PWREN is set, the power up delay is enforced such that any detected trigger does not begin ADC operation until the power up delay time has passed.
27-24 —	Reserved This read-only field is reserved and always has the value 0.
23-16 PUDLY	Power Up Delay When CFG[PWREN]=0b0, the LPADC analog circuits are only powered while the module is active and there is a counted delay defined by CFG[PUDLY] after an initial trigger transitions the LPADC from its Idle state to allow time for the analog circuits to stabilize. The startup delay count of (PUDLY*4) ADCK cycles must result in a longer delay than the analog startup time of $t_{ADCSTUP}$. Accuracy of the initial conversion after activation is degraded if CFG[PUDLY] is set to too small a value. When CFG[PWREN]=0b1 prior to LPADC activation, the analog circuits are pre-enabled and the activation delay defined by CFG[PUDLY] is bypassed.

Table continues on the next page...

Field	Function
15 —	Reserved This read-only field is reserved and always has the value 0.
14-8 —	Reserved This read-only field is reserved and always has the value 0.
7-6 REFSEL	Voltage Reference Selection Selects the voltage reference high used for conversions. **NOTE:** See the chip configuration information on the voltage reference options specific to this packaged device. 00b - (Default) Option 1 setting. 01b - Option 2 setting. 10b - Option 3 setting. 11b - Reserved
5-4 PWRSEL	Power Configuration Select Configures the module for power and performance. In the highest power setting, the highest conversion rates are possible. The highest power setting corresponds to 0b11 with decreasing power down to 0b00. 00b - Level 1 (Lowest power setting) 01b - Level 2 10b - Level 3 11b - Level 4 (Highest power setting)
3-1 —	Reserved This read-only field is reserved and always has the value 0.
0 TPRICTRL	LPADC trigger priority control This bitfield controls how higher priority triggers are handled. See Trigger detect and command execution for a detailed explanation trigger event handling. 0b - If a higher priority trigger is detected during command processing, the current conversion is aborted and the new command specified by the trigger is started. 1b - If a higher priority trigger is received during command processing, the current conversion is completed (including averaging iterations if enabled) and stored to the RESFIFO before the higher priority trigger/command is initiated. Note that compare until true commands can be interrupted prior to resulting in a true conversion.

87.7.1.9 LPADC Pause Register (PAUSE)

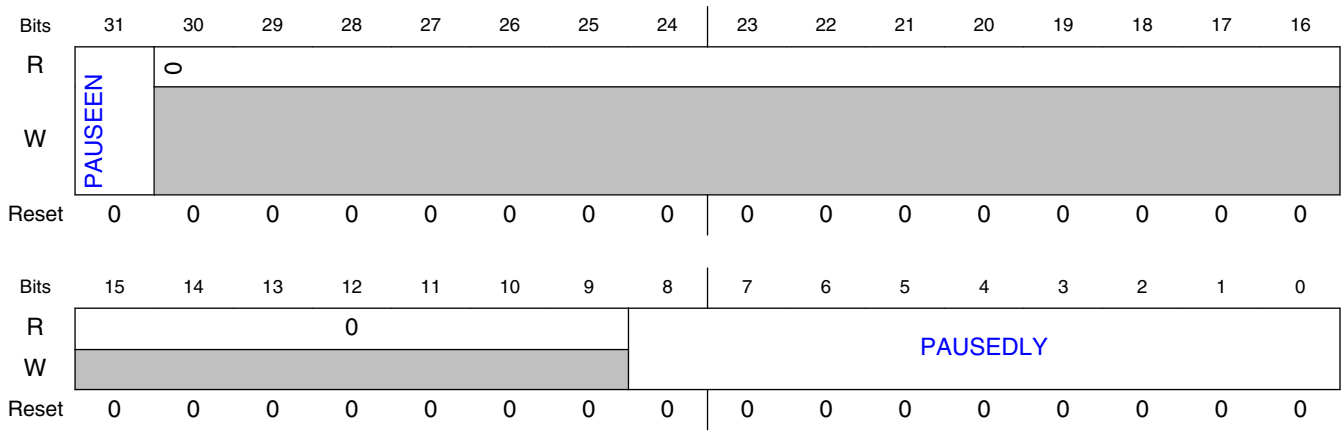
87.7.1.9.1 Offset

Register	Offset
PAUSE	24h

87.7.1.9.2 Function

The LPADC Pause Register (PAUSE) controls an optional inserted delay between conversions. PAUSE cannot be changed while the CTRL[ADCEN] bit is set. Writes to PAUSE while CTRL[ADCEN] is set are ignored.

87.7.1.9.3 Diagram



87.7.1.9.4 Fields

Field	Function
31 PAUSEEN	PAUSE Option Enable Enables the LPADC pausing function. When enabled, a programmable delay is inserted during command execution sequencing between LOOP iterations, between commands in a sequence, and between conversions when command is executing in for "Compare Until True" configuration. 0b - Pause operation disabled 1b - Pause operation enabled
30-9 —	Reserved This read-only field is reserved and always has the value 0.
8-0 PAUSEDLY	Pause Delay When PAUSEEN is set, the PAUSEDLY field controls the duration of pausing during command exection sequencing. The pause delay is a count of (PAUSEDLY*4) ADCK cycles.

87.7.1.10 LPADC FIFO Control Register (FCTRL)

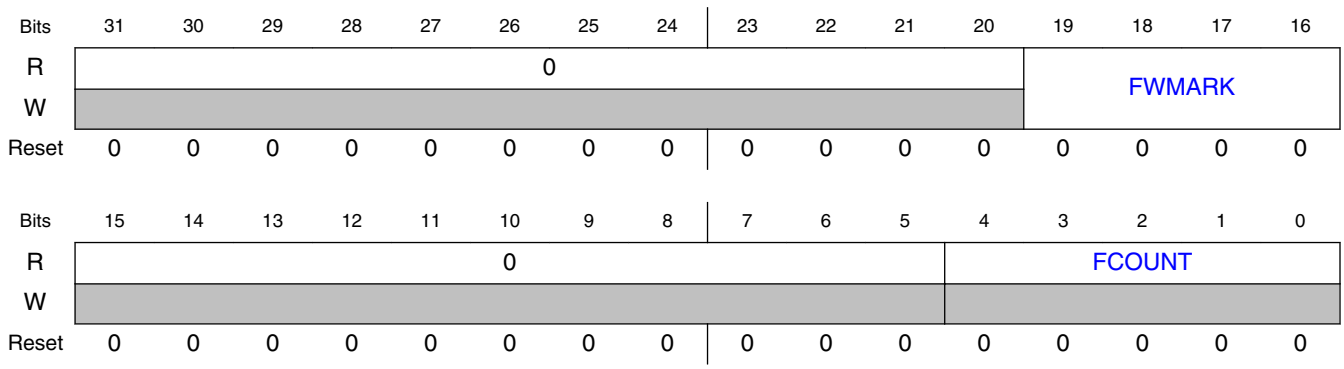
87.7.1.10.1 Offset

Register	Offset
FCTRL	30h

87.7.1.10.2 Function

The LPADC FIFO Control Register (FCTRL) controls the selection of Watermark level and provides the counter for the result FIFO.

87.7.1.10.3 Diagram



87.7.1.10.4 Fields

Field	Function
31-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 FWMARK	Watermark level selection FWMARK is a programmable threshold setting. When the number of datawords stored in the ADC Result FIFO is greater than the value in this field, the STAT[RDY] flag is asserted to indicate stored data has reached the programmable threshold. When IE[FWMIE] is set, an interrupt request is generated. When DE[FWMDE] is set, a DMA request is generated. 0000b - Generates STAT[RDY] flag after 1st successful conversion – single conversion 0001b - Generates STAT[RDY] flag after 2nd successful conversion 0010b - Generates STAT[RDY] flag after 3rd successful conversion 0011b - Generates STAT[RDY] flag after 4th successful conversion 0100b - Generates STAT[RDY] flag after 5th successful conversion 0101b - Generates STAT[RDY] flag after 6th successful conversion 0110b - Generates STAT[RDY] flag after 7th successful conversion 0111b - Generates STAT[RDY] flag after 8th successful conversion 1000b - Generates STAT[RDY] flag after 9th successful conversion 1001b - Generates STAT[RDY] flag after 10th successful conversion 1010b - Generates STAT[RDY] flag after 11th successful conversion 1011b - Generates STAT[RDY] flag after 12th successful conversion 1100b - Generates STAT[RDY] flag after 13th successful conversion 1101b - Generates STAT[RDY] flag after 14th successful conversion 1110b - Generates STAT[RDY] flag after 15th successful conversion 1111b - Generates STAT[RDY] flag after 16th successful conversion
15-5 —	Reserved This read-only field is reserved and always has the value 0.

Table continues on the next page...

Field	Function
4-0 FCOUNT	Result FIFO counter This read-only field indicates the number of datawords that are stored in the result FIFO. This value may be used in conjunction with PARAM[FIFOSIZE] to calculate how much room is left in the result FIFO. FCOUNT is incremented with each store of new data to the result FIFO and decrements with each read of the result FIFO. The FIFO is reset by writing to the CTRL[RSTFIFO] bit, resulting in FCTRL[FCOUNT] initialized to 0x0. 0_0000b - No data stored in FIFO 0_0001b - 1 dataword stored in FIFO 0_0010b - 2 datawords stored in FIFO 0_0100b - 4 datawords stored in FIFO 0_1000b - 8 datawords stored in FIFO 1_0000b - 16 datawords stored in FIFO

87.7.1.11 Software Trigger Register (SWTRIG)

87.7.1.11.1 Offset

Register	Offset
SWTRIG	34h

87.7.1.11.2 Function

The Software Trigger Register (SWTRIG) is written to initiate software triggered conversions. Writes to SWTRIG register are ignored while CTRL[ADCEN] is clear.

NOTE

There is a synchronization delay of 3 ADCK clock cycles when writing to set CTRL[ADCEN]. Writes to SWTRIG will continue to be ignored ultil CTRL[ADCEN] can propagate into the ADCK domain.

87.7.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0	0	0	0	0	0	0
W									SWT7	SWT6	SWT5	SWT4	SWT3	SWT2	SWT1	SWT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

87.7.1.11.4 Fields

Field	Function
31-8 —	Reserved This read-only field is reserved and always has the value 0.
7 SWT7	Software trigger 7 event Write 1 to SWT7 generates a trigger 7 event. Writing 1 to SWT7 is ignored while the trigger 7 event is being serviced or is pending. 0b - No trigger 7 event generated. 1b - Trigger 7 event generated.
6 SWT6	Software trigger 6 event Write 1 to SWT6 generates a trigger 6 event. Writing 1 to SWT6 is ignored while the trigger 6 event is being serviced or is pending. 0b - No trigger 6 event generated. 1b - Trigger 6 event generated.
5 SWT5	Software trigger 5 event Write 1 to SWT5 generates a trigger 5 event. Writing 1 to SWT5 is ignored while the trigger 5 event is being serviced or is pending. 0b - No trigger 5 event generated. 1b - Trigger 5 event generated.
4 SWT4	Software trigger 4 event Write 1 to SWT4 generates a trigger 4 event. Writing 1 to SWT4 is ignored while the trigger 4 event is being serviced or is pending. 0b - No trigger 4 event generated. 1b - Trigger 4 event generated.
3 SWT3	Software trigger 3 event Write 1 to SWT3 generates a trigger 3 event. Writing 1 to SWT3 is ignored while the trigger 3 event is being serviced or is pending. 0b - No trigger 3 event generated. 1b - Trigger 3 event generated.
2 SWT2	Software trigger 2 event

Table continues on the next page...

Field	Function
	Write 1 to SWT2 generates a trigger 2 event. Writing 1 to SWT2 is ignored while the trigger 2 event is being serviced or is pending. 0b - No trigger 2 event generated. 1b - Trigger 2 event generated.
1 SWT1	Software trigger 1 event Write 1 to SWT1 generates a trigger 1 event. Writing 1 to SWT1 is ignored while the trigger 1 event is being serviced or is pending. 0b - No trigger 1 event generated. 1b - Trigger 1 event generated.
0 SWT0	Software trigger 0 event Write 1 to SWT0 generates a trigger 0 event. Writing 1 to SWT0 is ignored while the trigger 0 event is being serviced or is pending. 0b - No trigger 0 event generated. 1b - Trigger 0 event generated.

87.7.1.12 Trigger Control Register (TCTRL0 - TCTRL7)

87.7.1.12.1 Offset

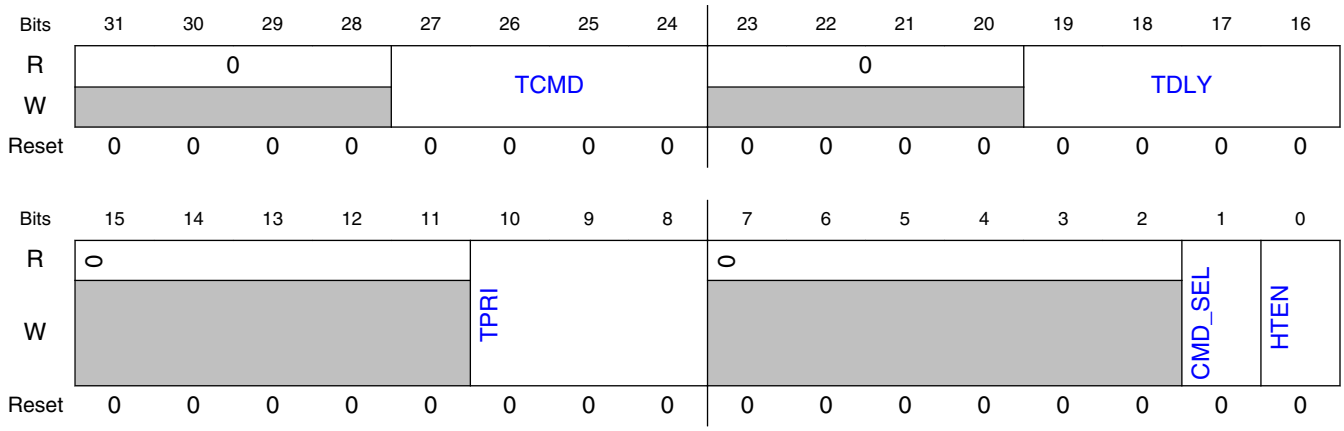
For a = 0 to 7:

Register	Offset
TCTRLa	C0h + (a × 4h)

87.7.1.12.2 Function

The Trigger Control (TCTRL*a*) register implements control fields associated with each implemented trigger source. When the LPADC is actively executing commands, only one of the TCTRL*a* registers is actively controlling ADC conversions. The active trigger can be detected by STAT[TRGACT] and activity of the trigger is dedicated by SWTRIG[SWT0-7] (within ADC module) or by HWTS0-7 (within ADC_ETC module). The actively controlling TCTRL*a* register cannot be updated while the LPADC is active. A write to a TCTRL*a* registers while that trigger control register is controlling the ADC operation is ignored.

87.7.1.12.3 Diagram



87.7.1.12.4 Fields

Field	Function
31-28 —	Reserved This read-only field is reserved and always has the value 0.
27-24 TCMD	Trigger command select Selects the command from command buffer to execute upon detection of the associated trigger event. When a trigger is received while the LPADC is busy converting and the new trigger has higher priority than trigger associated with the current command being executed, the command in this register, associated with the new trigger, is executed under control of CFG[TPRICTRL]. When CFG[TPRICTRL]=0b0, the interruption due to higher priority trigger event is immediate. When CFG[TPRICTRL]=0b1, the current conversion (including any hardware averaging) is allowed to complete and the interruption is at the next loop boundary. 0000b - Not a valid selection from the command buffer. Trigger event is ignored. 0001b - CMD1 is executed 0010b-1110b - Corresponding CMD is executed 1111b - CMD15 is executed
23-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 TDLY	Trigger delay select Selects the trigger delay duration to wait at the start of servicing a trigger event. Each trigger source has an associated programmable delay prior to beginning an initial conversion. When TDLY field is clear, then no delay is incurred. When TDLY is set to a non-zero value, the duration for the delay is 2^{TDLY} ADCK cycles.
15-11 —	Reserved This read-only field is reserved and always has the value 0.
10-8 TPRI	Trigger priority setting This bitfield sets the priority of the associated trigger source. If two or more triggers have the same priority level setting, the lower order trigger event has the higher priority (e. g., if Trigger 0 and Trigger 1 are both pending triggers and TCTRL0[TPRI] is configured the same as TCTRL1[TPRI], then the command associated with Trigger 0 is serviced first). 000b - Set to highest priority, Level 1

Table continues on the next page...

Field	Function
	001b-110b - Set to corresponding priority level 111b - Set to lowest priority, Level 8
7-2 —	Reserved This read-only field is reserved and always has the value 0.
1 CMD_SEL	The command number is selected by software TCMD or hardware tcmd signal 0b - TCTRLa[TCMD] will determine the command 1b - Software TCDM is bypassed , and hardware TCMD from ADC_ETC module will be used. The trigger command is then defined by ADC hardware trigger command selection field in ADC_ETC->TRIGx_CHAINy_z_n[CSEL].
0 HTEN	Trigger enable Enables hardware trigger source to initiate conversion on the rising-edge of the input trigger source. **NOTE:** Enabling hardware trigger does not disable software triggers. 0b - Hardware trigger source disabled 1b - Hardware trigger source enabled

87.7.1.13 LPADC Command Low Buffer Register (CMDL1 - CMDL15)

87.7.1.13.1 Offset

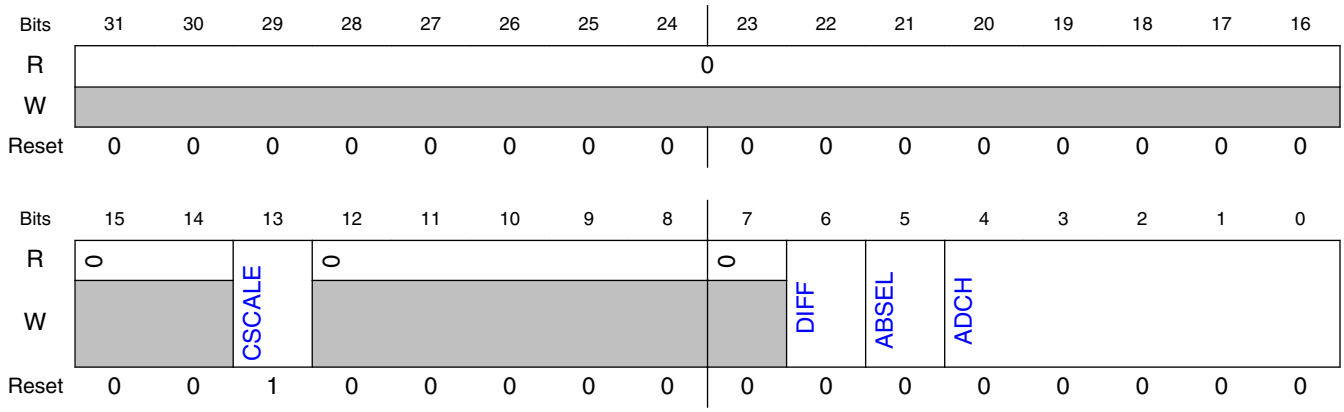
For a = 1 to 15:

Register	Offset
CMDLa	F8h + (a × 8h)

87.7.1.13.2 Function

There are 15 command buffers (CMD*a*), each constructed from two 32-bit registers (CMDL*a*:CMDH*a*) that can be configured for different channel select and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of the TCTRL*a*[TCMD] bitfield. When the LPADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer cannot be updated while the LPADC is active. A write to a CMD buffer while that CMD buffer is controlling the ADC operation is ignored.

87.7.1.13.3 Diagram



87.7.1.13.4 Fields

Field	Function
31-14 —	Reserved This read-only field is reserved and always has the value 0.
13 CSCALE	Channel Scale Reduces the selected ADC analog channel input voltage level by a factor. The maximum possible voltage on the ADC channel input should be considered when selecting a CSCALE value to ensure that the reducing factor always results voltage level at or below the VREFH reference. This reducing capability allows conversion of analog inputs higher than VREFH. A-side and B-side channel inputs are both scaled using the CSCALE field. For example: if VREFH is 1.54V and CSCALE = 0, then the ADC input voltage can be up to 3.3V because 1.54*(64/30)=3.3. 0b - Scale selected analog channel (Factor of 30/64) 1b - (Default) Full scale (Factor of 1)
12-8 —	Reserved This read-only field is reserved and always has the value 0.
7 —	Reserved This read-only field is reserved and always has the value 0.
6 DIFF	Differential Mode Enable Configures the LPADC to operate in differential mode or single-ended mode. 0b - Single-ended mode. 1b - Differential mode.
5 ABSEL	A-side vs. B-side Select When DIFF=0b0, ABSEL selects the channel from either the A-side or B-side channel inputs. When DIFF=0b1, ABSEL configures the ADC result to be (CHnA-CHnB) or (CHnB-CHnA). 0b - When DIFF=0b0, the associated A-side channel is converted as single-ended. When DIFF=0b1, the ADC result is (CHnA-CHnB). 1b - When DIFF=0b0, the associated B-side channel is converted as single-ended. When DIFF=0b1, the ADC result is (CHnB-CHnA).
4-0	Input channel select

Field	Function
ADCH	Each ADCH channel selection has an associated A side and an associated B side input. The ADCH setting in conjunction with the DIFF and ABSEL bitfield settings selects the input from one of the channel inputs or one of the input pairs. See the DIFF and ABSEL bitfield descriptions. **NOTE:** Not all pairs are intended to be converted as differential channels. For the pin pairings available for differential conversions for your device, see the Chip Configuration details. **NOTE:** Some of the input channels are from internal resources such as temperature sensors and voltage sources and may only be connected to individual instances of the ADC module. Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the ADC channel assignments for your device, see the chip-specific information. 0_0000b - Select CH0A or CH0B or CH0A/CH0B pair. 0_0001b - Select CH1A or CH1B or CH1A/CH1B pair. 0_0010b - Select CH2A or CH2B or CH2A/CH2B pair. 0_0011b - Select CH3A or CH3B or CH3A/CH3B pair. 0_0100b-1_1101b - Select corresponding channel CHnA or CHnB or CHnA/CHnB pair. 1_1110b - Select CH30A or CH30B or CH30A/CH30B pair. 1_1111b - Select CH31A or CH31B or CH31A/CH31B pair.

87.7.1.14 LPADC Command High Buffer Register (CMDH1 - CMDH4)

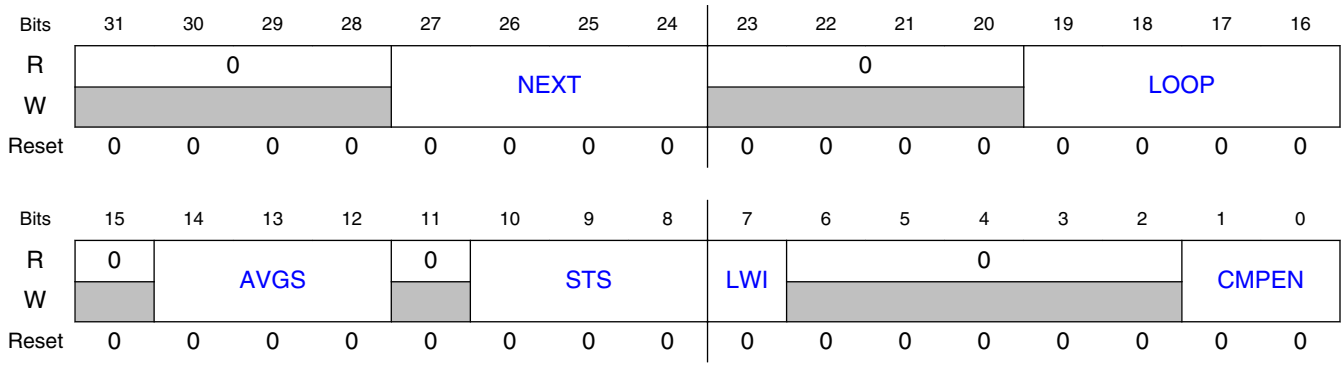
87.7.1.14.1 Offset

Register	Offset
CMDH1	104h
CMDH2	10Ch
CMDH3	114h
CMDH4	11Ch

87.7.1.14.2 Function

There are 15 command buffers (CMD*a*), each constructed from two 32-bit registers (CMDL*a*:CMDH*a*) that can be configured for different channel select and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of the TCTRL*a*[TCMD] bitfield. When the LPADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer cannot be updated while the LPADC is active. A write to a CMD buffer while that CMD buffer is controlling ADC operation is ignored.

87.7.1.14.3 Diagram



87.7.1.14.4 Fields

Field	Function
31-28 —	Reserved This read-only field is reserved and always has the value 0.
27-24 NEXT	Next Command Select Selects the next command to be executed after this command completes. Multiple commands can be configured in a scan configuration by linking the next command in a daisy-chain sequence. The command buffer number is not indicative of any particular order and the order of execution is strictly controlled by the NEXT field (for example, a sequence of commands could be CMD2, CMD1, CMD3). Unending circular command execution is allowed by setting the NEXT field in the last command in a sequence to the first command in the sequence. It is also allowed for a command to set the next command to itself, resulting in a continuous conversion configuration. Setting the next command to 0x0 causes conversions to terminate at the completion of the command. Lower priority trigger events cannot be serviced until a higher priority triggered command (or sequence of commands) completes. 0000b - No next command defined. Terminate conversions at completion of current command. If lower priority trigger pending, begin command associated with lower priority trigger. 0001b - Select CMD1 command buffer register as next command. 0010b-1110b - Select corresponding CMD command buffer register as next command 1111b - Select CMD15 command buffer register as next command.
23-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 LOOP	Loop Count Select Selects how many times this command executes (and stores conversion result to RESFIFO) before finish and transition to the next command or Idle state. LWI field controls whether a single channel is converted on each iteration or auto channel increment results in channel scanning functionality. 0000b - Looping not enabled. Command executes 1 time. 0001b - Loop 1 time. Command executes 2 times. 0010b - Loop 2 times. Command executes 3 times. 0011b-1110b - Loop corresponding number of times. Command executes LOOP+1 times. 1111b - Loop 15 times. Command executes 16 times.
15 —	Reserved This read-only field is reserved and always has the value 0.
14-12	Hardware Average Select

Table continues on the next page...

Field	Function
AVGS	Selects how many ADC conversions are averaged to create the ADC result (2^{AVGS}). An internal storage buffer is used to capture temporary results while the averaging iterations are executed. Hardware averaging is a nested loop control and does not extend across LOOP boundaries. See Functional description for more detailed description on usage of AVGS, LOOP, and NEXT bitfields in command execution sequencing. 000b - Single conversion. 001b - 2 conversions averaged. 010b - 4 conversions averaged. 011b - 8 conversions averaged. 100b - 16 conversions averaged. 101b - 32 conversions averaged. 110b - 64 conversions averaged. 111b - 128 conversions averaged.
11 —	Reserved This read-only field is reserved and always has the value 0.
10-8 STS	Sample Time Select When programmed to 000 the minimum sample time of 3.5 ADCK cycles is selected. When STS is programmed to a non-zero value the sample time is ($3.5 + 2^{STS}$) ADCK cycles. The shortest sample time maximizes conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be accurately sampled. Longer sample times can also be used to lower overall power consumption when command looping and sequencing is configured and high conversion rates are not required. 000b - Minimum sample time of 3.5 ADCK cycles. 001b - $3.5 + 2^1$ ADCK cycles; 5.5 ADCK cycles total sample time. 010b - $3.5 + 2^2$ ADCK cycles; 7.5 ADCK cycles total sample time. 011b - $3.5 + 2^3$ ADCK cycles; 11.5 ADCK cycles total sample time. 100b - $3.5 + 2^4$ ADCK cycles; 19.5 ADCK cycles total sample time. 101b - $3.5 + 2^5$ ADCK cycles; 35.5 ADCK cycles total sample time. 110b - $3.5 + 2^6$ ADCK cycles; 67.5 ADCK cycles total sample time. 111b - $3.5 + 2^7$ ADCK cycles; 131.5 ADCK cycles total sample time.
7 LWI	Loop with Increment When LWI is clear, the LOOP field selects the number of times the selected channel is converted consecutively. When LWI is set, auto channel incrementing is enabled and the LOOP field defines how many consecutive channels are converted as part of the command execution. Example 1: LOOP = 0x8, LWI = 0b0, ABSEL = 0b0, ADCH = 0xD. Convert on channel 13A 8 times. Example 2: LOOP = 0x8, LWI= 0b1, ABSEL = 0b0, ADCH = 0xD. Run channels 13A, 14A, 15A, ... 20A each one time. Maximum channel scanning using a single command buffer then is defined by the maximum value of the LOOP field (16). **NOTE:** When LWI is set, the minimum sample length must be extended to CMDHa[STS] = 0x3. 0b - Auto channel increment disabled 1b - Auto channel increment enabled
6-2 —	Reserved This read-only field is reserved and always has the value 0.
1-0 CMPEN	Compare Function Enable After an ADC channel input is sampled and converted and any averaging iterations are performed, the CMDH*a*[CMPEN] field guides operation of the automatic compare function to optionally only store when the compare operation is true. When compare is enabled, the conversion result is compared to the compare value registers (CV*a*[CVH] and CV*a*[CVL]). There are multiple options on command sequencing related to the compare function. See Compare function for a detailed explanation of the compare options.

Field	Function
	NOTE: Not all Command Buffers have implemented the CMPEN field. The CMPEN field is only available in Command Buffers that have a corresponding Compare Value register. 00b - Compare disabled. 01b - Reserved 10b - Compare enabled. Store on true. 11b - Compare enabled. Repeat channel acquisition (sample/convert/compare) until true.

87.7.1.15 LPADC Command High Buffer Register (CMDH5 - CMDH15)

87.7.1.15.1 Offset

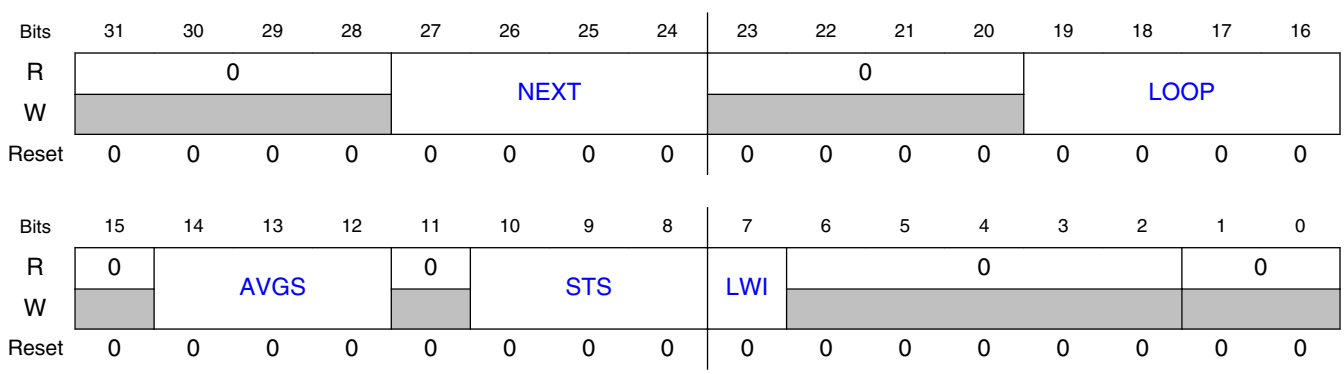
For a = 5 to 15:

Register	Offset
CMDHa	FCh + (a × 8h)

87.7.1.15.2 Function

There are 15 command buffers (CMD*a*), each constructed from two 32-bit registers (CMDL*a*:CMDH*a*) that can be configured for different channel select and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of the TCTRL*a*[TCMD] bitfield. When the LPADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer cannot be updated while the LPADC is active. A write to a CMD buffer while that CMD buffer is controlling ADC operation is ignored.

87.7.1.15.3 Diagram



87.7.1.15.4 Fields

Field	Function
31-28 —	Reserved This read-only field is reserved and always has the value 0.
27-24 NEXT	Next Command Select Selects the next command to be executed after this command completes. Multiple commands can be configured in a scan configuration by linking the next command in a daisy-chain sequence. The command buffer number is not indicative of any particular order and the order of execution is strictly controlled by the NEXT field (for example, a sequence of commands could be CMD2, CMD1, CMD3). Unending circular command execution is allowed by setting the NEXT field in the last command in a sequence to the first command in the sequence. It is also allowed for a command to set the next command to itself, resulting in a continuous conversion configuration. Setting the next command to 0x0 causes conversions to terminate at the completion of the command. Lower priority trigger events cannot be serviced until a higher priority triggered command (or sequence of commands) completes. 0000b - No next command defined. Terminate conversions at completion of current command. If lower priority trigger pending, begin command associated with lower priority trigger. 0001b - Select CMD1 command buffer register as next command. 0010b-1110b - Select corresponding CMD command buffer register as next command 1111b - Select CMD15 command buffer register as next command.
23-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 LOOP	Loop Count Select Selects how many times this command executes (and stores conversion result to RESFIFO) before finish and transition to the next command or Idle state. LWI field controls whether a single channel is converted on each iteration or auto channel increment results in channel scanning functionality. 0000b - Looping not enabled. Command executes 1 time. 0001b - Loop 1 time. Command executes 2 times. 0010b - Loop 2 times. Command executes 3 times. 0011b-1110b - Loop corresponding number of times. Command executes LOOP+1 times. 1111b - Loop 15 times. Command executes 16 times.
15 —	Reserved This read-only field is reserved and always has the value 0.
14-12 AVGS	Hardware Average Select Selects how many ADC conversions are averaged to create the ADC result (2^{AVGS}). An internal storage buffer is used to capture temporary results while the averaging iterations are executed. Hardware averaging is a nested loop control and does not extend across LOOP boundaries. See Functional description for more detailed description on usage of AVGS, LOOP, and NEXT bitfields in command execution sequencing. 000b - Single conversion. 001b - 2 conversions averaged. 010b - 4 conversions averaged. 011b - 8 conversions averaged. 100b - 16 conversions averaged. 101b - 32 conversions averaged. 110b - 64 conversions averaged. 111b - 128 conversions averaged.
11 —	Reserved This read-only field is reserved and always has the value 0.

Table continues on the next page...

Field	Function
10-8 STS	Sample Time Select When programmed to 000 the minimum sample time of 3.5 ADCK cycles is selected. When STS is programmed to a non-zero value the sample time is (3.5 + 2^{STS}) ADCK cycles. The shortest sample time maximizes conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be accurately sampled. Longer sample times can also be used to lower overall power consumption when command looping and sequencing is configured and high conversion rates are not required. 000b - Minimum sample time of 3.5 ADCK cycles. 001b - 3.5 + 2^1 ADCK cycles; 5.5 ADCK cycles total sample time. 010b - 3.5 + 2^2 ADCK cycles; 7.5 ADCK cycles total sample time. 011b - 3.5 + 2^3 ADCK cycles; 11.5 ADCK cycles total sample time. 100b - 3.5 + 2^4 ADCK cycles; 19.5 ADCK cycles total sample time. 101b - 3.5 + 2^5 ADCK cycles; 35.5 ADCK cycles total sample time. 110b - 3.5 + 2^6 ADCK cycles; 67.5 ADCK cycles total sample time. 111b - 3.5 + 2^7 ADCK cycles; 131.5 ADCK cycles total sample time.
7 LWI	Loop with Increment When LWI is clear, the LOOP field selects the number of times the selected channel is converted consecutively. When LWI is set, auto channel incrementing is enabled and the LOOP field defines how many consecutive channels are converted as part of the command execution. Example 1: LOOP = 0x8, LWI = 0b0, ABSEL = 0b0, ADCH = 0xD. Convert on channel 13A 8 times. Example 2: LOOP = 0x8, LWI= 0b1, ABSEL = 0b0, ADCH = 0xD. Run channels 13A, 14A, 15A, ... 20A each one time. Maximum channel scanning using a single command buffer then is defined by the maximum value of the LOOP field (16). **NOTE:** When LWI is set, the minimum sample length must be extended to CMDHa[STS] = 0x3. 0b - Auto channel increment disabled 1b - Auto channel increment enabled
6-2 —	Reserved This read-only field is reserved and always has the value 0.
1-0 —	Reserved This read-only field is reserved and always has the value 0.

87.7.1.16 Compare Value Register (CV1 - CV4)

87.7.1.16.1 Offset

Register	Offset
CV1	200h
CV2	204h
CV3	208h
CV4	20Ch

87.7.1.16.2 Function

The Compare Value Registers (CV*a*) contain values used to compare the conversion result when the compare function is enabled. This register is formatted in the same way as the D field in the RESFIFO registers in different modes of operation for both bit position definition and value format using unsigned or signed 2's complement. There is a direct association of each compare value register to a specific command buffer register (i.e., CV1 is only used during execution of CMD1 command).

NOTE

Not all Command Buffers have an associated Compare Value register. The compare function is only available on Command Buffers that have a corresponding Compare Value register.

When the LPADC is actively executing commands, the CV*a* register associated with the active command (CMD*a*) cannot be updated and writes to associated CV*a* register during this time are ignored.

87.7.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CVH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CVL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

87.7.1.16.4 Fields

Field	Function
31-16 CVH	Compare Value High. The compare function can be configured to check whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVH is used. For example. CVH represents 16-bit high level compare value. In cooperation with CVL and CMDHa[CMPEN] represents limit value for compare function: When CVL < CVH, it represents “outside range” compare function and the compare is true if results is less than CVL value and greater than CVH value. When CVL > CVH, it represents “inside range” compare function and the compare is true if results is less than CVL value and greater than CVH value.

Table continues on the next page...

Field	Function
	When CVL is min value and CVH any other than max value, it represents “greater than” compare function and the compare is true if results is greater than CVH value. When CVL is any other than min and CVH is max value, it represents “less than” compare function and the compare is true if results is greater than CVH value. When CVL is min value and CVH max value, the compare is never true. When CVL is max value and CVH min value, the compare is always true.
15-0 CVL	Compare Value Low The compare function can be configured to check whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVL is used.

87.7.1.17 LPADC Data Result FIFO Register (RESFIFO)

87.7.1.17.1 Offset

Register	Offset
RESFIFO	300h

87.7.1.17.2 Function

The LPADC Data Result FIFO Register (RESFIFO) is a 16 entry FIFO that stores the data result of ADC conversions. In addition, several tag fields of source command and trigger information are stored along with the data. FCTRL[FCOUNT] indicates how many valid datawords are stored in the RESFIFO. Reading RESFIFO provides the oldest unread dataword entry in the FIFO and decrements FCOUNT. The FIFO can be emptied by successive reads of RESFIFO. The FIFO is reset by writing 0b1 to the CTRL[RSTFIFO] bit.

The following table describes the format of data in the result FIFO in the different modes of operation. The sign bit is the (MSB) in signed 2's complement modes. For example, when configured for 12-bit single-ended mode, D[15] and D[2:0] are cleared. When configured for 13-bit differential mode, D[15] is the sign bit and D[2:0] are cleared.

Table 87-7. Data result register format description

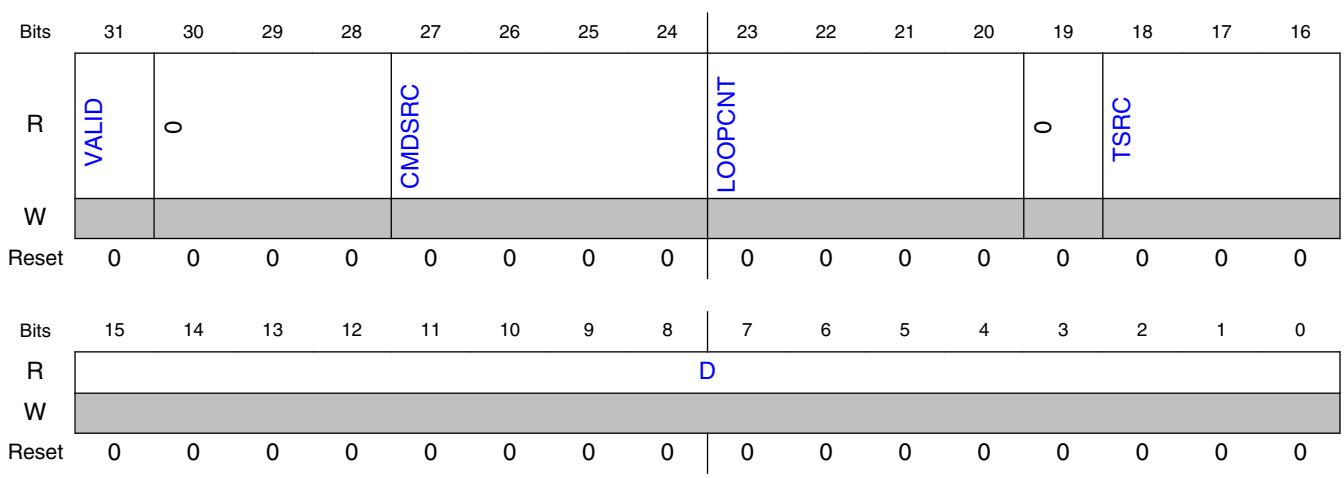
Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
13-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	Signed 2's complement, left-justified, zero extended
12-bit single-ended	0	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	Unsigned, zero in D[15] and D[2:0]

NOTE

S: Sign bit;

D: Data, which is 2's complement data if indicated

87.7.1.17.3 Diagram



87.7.1.17.4 Fields

Field	Function
31 VALID	FIFO entry is valid Indicate the FIFO entry is valid. When the FIFO is holding data the VALID bit is set. When the FIFO is empty the VALID bit is clear. 0b - FIFO is empty. Discard any read from RESFIFO.

Table continues on the next page...

Field	Function
	1b - FIFO record read from RESFIFO is valid.
30-28 —	Reserved This read-only field is reserved and always has the value 0.
27-24 CMDSRC	Command Buffer Source Indicates the command buffer being executed that generated this result. 0000b - Not a valid value CMDSRC value for a dataword in RESFIFO. 0x0 is only found in initial FIFO state prior to an ADC conversion result dataword being stored to a RESFIFO buffer. 0001b - CMD1 buffer used as control settings for this conversion. 0010b-1110b - Corresponding command buffer used as control settings for this conversion. 1111b - CMD15 buffer used as control settings for this conversion.
23-20 LOOPCNT	Loop count value Indicates the loop count value during command execution that generated this result. When CMDH*a*[LOOP] is non-zero, the command execution stores off a result multiple times during command execution (at the loop boundary). 0000b - Result is from initial conversion in command. 0001b - Result is from second conversion in command. 0010b-1110b - Result is from LOOPCNT+1 conversion in command. 1111b - Result is from 16th conversion in command.
19 —	Reserved This read-only field is reserved and always has the value 0.
18-16 TSRC	Trigger Source Indicate the trigger source that initiated a conversion and generated this result. When multiple commands are chained together using the NEXT field, the TSRC field for all datawords stored to the RESFIFO indicate the trigger source that started the sequence of commands. 000b - Trigger source 0 initiated this conversion. 001b - Trigger source 1 initiated this conversion. 010b-110b - Corresponding trigger source initiated this conversion. 111b - Trigger source 7 initiated this conversion.
15-0 D	Data result The formatting for the data in D is summarized in Table 87-7.

Chapter 88
ADC External Trigger Control (ADC_ETC)

88.1 Chip-specific ADC_ETC information

Table 88-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
XBAR resource assignments	XBAR	XBAR Resource Assignments
Touch Screen Control	TSC	TSC

NOTE

No TSC module in this device.

NOTE

The bit 30 in ADC_ETC Global Control Register (CTRL) must be cleared after reset when ADC2 is used.

88.2 Overview

The ADC External Trigger Control (ADC_ETC) module enables multiple trigger sources to share the ADC modules in a TIME-Division-Multiplexing (TDM) way. The triggers of ADC_ETC are from Cross BAR(XBAR). The triggers trig0~trig3 belong to channel0 and control ADC1, while trig4~trig7 belong to channel1 and control ADC2. The ADC_ETC

also supports SyncMode. When TRIGa_CTRL[SYNC_MODE] is set, the ADC_ETC trigger source will control ADC1 and ADC2 synchronously. The ADC_ETC can support interrupt mode and DMA mode (controlled independently).

NOTE

The trigger in this chapter refers the trigger of ADC_ETC, except when referring to the ADC trigger explicitly.

88.2.1 Block diagram

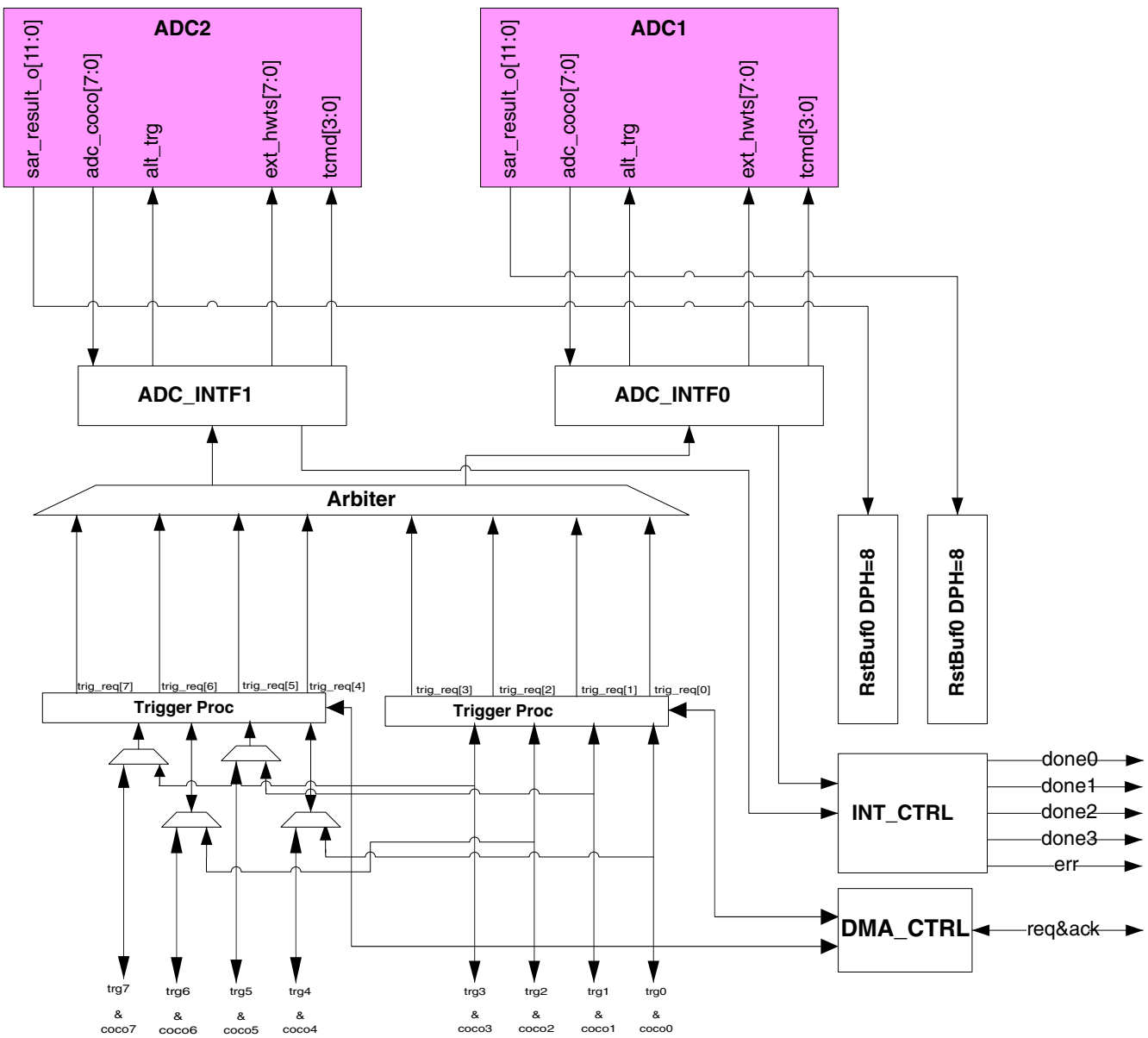


Figure 88-1. ADC_ETC block diagram

NOTE

The ADC1 and ADC2 in ADC_ETC block diagram are functionally same as the module described in ADC chapter, but they have different interface. The ADC control interfaces, which are between ADC_ETC and ADCs, operate as below:

- ADC_ETC puts trigger number of ADC on ext_hwts[7:0], puts the TCMD for this ADC trigger on the port tcmd[3:0]. And meanwhile sets alt_trg to send a trigger to ADC.
- ADC start a conversion after receiving a trigger from the ADC control interface. And the behavior of the conversion is configured by the ext_hwts, the TCMD of ext_hwts and the CMD defined by the command buffer inside ADC.
- When conversion done, ADC will send back sar_result[11:0] as the result of the conversion, together with adc_coco[7:0] to identify which ext_hwts this result is for.

NOTE

The adc_coco signals are internal signals, which are hidden from ADC module and handled by ADC and ADC_ETC automatically.

88.2.2 Features

The ADC_ETC includes the following features:

- External Trigger Interface
  - ADC control interface is connected with 2 ADCs, and support up to 8 Hardware External Trigger (ext_hwts[7:0] in ADC_ETC block diagram) for each ADC
- Capable of triggering dual ADC in SyncMode or AsyncMode:
  - In SyncMode ADC1 and ADC2 are controlled by the same trigger source of ADC_ETC.
  - In AsyncMode ADC1 and ADC2 are controlled by separate trigger source of ADC_ETC.
- Support up to 8 external trigger inputs, input from XBAR (4 for each ADC, TRIG0-TRIG3 for ADC1, and TRIG4-TRIG7 for ADC2):
  - One external trigger of ADC_ETC can launch up to 8 sequential triggers to ADC, which are called a trigger chain, and each trigger is called a segment of the trigger chain.

    - Flexible ADC trigger interval and initial delay control.
    - Each trigger sources of ADC_ETC can be configured as HW or SW trigger mode.
- ADC result holding and status reporting
- ADC_ETC trigger auto hold and arbitration
    - Each trigger of ADC_ETC can be configured with a fixed priority. External trigger with the highest priority is severed first.
    - Hold one trigger event when it fails to win the arbitration or ADC is busy.
- Support ADC trigger interface cascading
- Supports interrupt mode. When interrupt mode is selected and one ADC conversion is done, an interrupt request signal will be generated on one of the Done0~Done3 interrupt outputs.
- Supports DMA mode. When DMA is enabled and one ADC conversion is done, a DMA request will be sent. 2 trigger modes of DMA can be selected by configuring CTRL[DMA_MODE_SEL].

88.3 Functional description

The ADC_ETC block works with XBAR to share with the ADC modules. The following sections describe functional details of the ADC_ETC module.

88.3.1 Operations

This section describes the ADC_ETC module's operations.

The trigger process is used to handle the multiple trigger sources based on the settings in registers. CTRL[TRIG_ENABLE] can be used to mask the input hardware trigger sources. TRIGa_CTRL[TRIG_MODE] selects the hardware trigger source or software trigger source for TRIGa of ADC_ETC. The selected trigger source is double synchronized to a synchronous positive pulse and the pulse will be sent to the arbiter after initial delay defined by TRIGa_COUNTER[INIT_DELAY].

If TRIGa_CTRL[SYNC_MODE] is set, the trigger process will control TRIGa and TRIG(a+4) synchronously. In this mode, the initial delay of TRIGa and TRIG(a+4) is controlled by the settings of TRIGa, while other settings are independent.

The trigger process diagram is indicated as below:

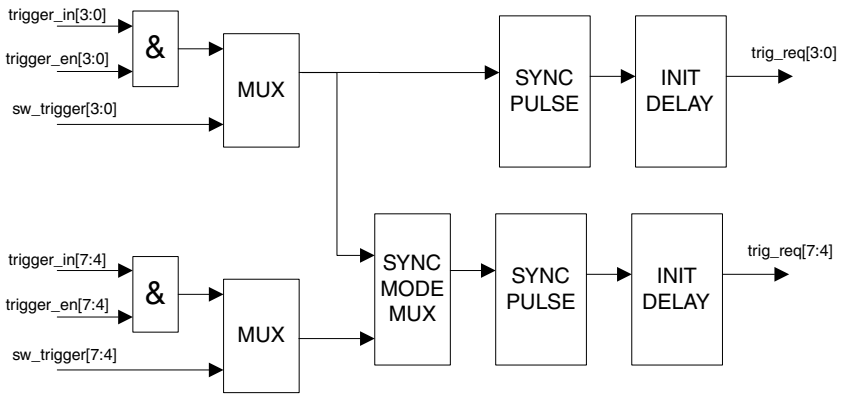


Figure 88-2. Trigger process diagram

As shown in ADC_ETC block diagram, the trig_req[7:0] output from trigger process will go to the arbiter.

The arbiter will approve the request from XBAR which has the highest priority defined in TRIGa_CTRL[TRIG_PRIORITY]. There are total of eight priority levels available for each trigger request, where 7 is the highest priority level and 0 is the lowest priority level. If multiple trigger requests have the same priority level, then the round-robin mechanism will be applied, and the trigger with larger number owns the higher priority. The trigger request is granted only when the previous trigger has finished and the ADC module is IDLE. When in SYNC mode, the trigger request is granted only when both ADC modules are IDLE state.

When there is a trigger request that is granted, the ADC_ETC sends several triggers of ADC as a chain of trigger, with ext_hwts[7:0] as ADC trigger selection signal, and with tcmd[3:0] defining the TCMD to ADC. When the ADC conversion done, the ADC sends back adc_coco to the ADC_ETC, which has the same width as ext_hwts and indicates which trigger the current sar_result_o is for, then the ADC_ETC stores the ADC converted results into the internal registers.

At the same time, the corresponding bit in the CHAINx_DONE bit field of ETC_TRIG Control Register is set, and interrupt asserts according to the software configuration. After that, the next trigger is sent after a interval delay configured by TRIGa_COUNTER[SAMPLE_INTERVAL], until all triggers in the chain are processed. When complete the the last trigger in chain, ADC_ETC sends the coco signal to mark the end of one of ADC_ETC external trigger, and a DMA request will be generated if DMA_CTRL[TIRGa_ENABLE] is set to 1.

Each ADC trigger in the trigger chain, initiated by trigger requests of ADC_ETC, has separate configurations by software. These configurations include Interrupt enable (IE_EN), Interrupt selection (IE), Back to Back ADC trigger (B2B), HWTS and CSEL to ADC. The ADC_ETC have interrupt outputs of Done0, Done1, Done2, Done3 and Error

Interrupt. The Done0/1/2/3 interrupts are controlled by software for each ADC trigger. IEx_EN controls the enable of the IRQ, while IEx selects which port the interrupt occurs on. (If IEx_EN is set, IEx from 2'b00 to 2'b11 select Done0/1/2/3, otherwise IEx is meaningless). Error interrupt occurs when there is an external trigger ignored by ADC_ETC due to the previous trigger not finished yet.

The ETC ADC interface timing is indicated in below diagrams:

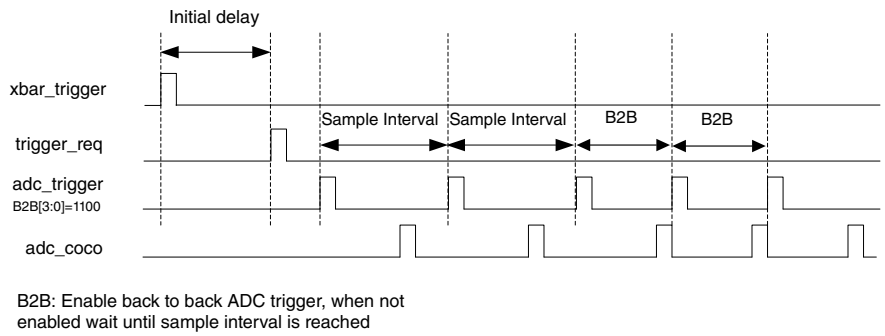


Figure 88-3. ETC ADC interface timing

After one ADC conversion, the next trigger in the chain starts as soon as possible if B2B is enabled. Otherwise, the next trigger does not start until the interval delay is reached.

As shown in ETC-ADC interface timing figure, due to B2B[3:0]=1100, the slots after the first and second adc_trigger are equal to interval delay, but the slots after the third and fourth adc_trigger are shorter.

The length of initial delay and interval delay are configurable, the formulas to calculate initial delay and interval delay are described in the description of the TRIGa_COUNTER.

88.3.2 Clocks

The ADC_ETC module has one global functional clock: ipg_clk. The ipg_clk is used for register read/write operations, also all the functionality inside the ADC_ETC module are synchronized to this clock.

The XBAR may be synchronous or asynchronous with the ADC_ETC clock.

88.3.3 Reset

The ADC_ETC has two resets: ipg_hard_async_reset_b and sw_reset_n. The ipg_hard_async_reset_b is a hardware reset, which resets all registers in ADC_ETC block. The sw_reset_n is a software reset, which can reset all registers except itself by software.

NOTE

sw_reset_n is generated by writting logic 1 into CTRL[SOFTRST], while ipg_hard_async_reset_b is generated by SoC hardware.

88.4 Initialization

Before using the ADC_ETC module, the following initialization procedure must be performed:

1. Configure the Global Control bits, include selecting DMA mode, enabling triggers which will be used, and setting pre-division factor.
2. Set DMA enable bits as needed.
3. Configure control bits for each enabled trigger, include enabling or disabling synchronization mode, assigning trigger priority, setting the length of trigger chain, selecting trigger mode and setting the initial and interval delay of this trigger.
4. According to the length of trigger chain, several segments of the chain should be configured. The configurations of each segment contain:
    a. setting the DONE interrupt for this chain, enabling or disabling B2B mode;
    b. selecting which trigger of ADC will be triggered, setting the ADC command.

After the initialization of ADC_ETC, the initialization of the ADC module should also be confirmed before launching a trigger to ADC_ETC.

88.5 Application information

This section describes applications supported by the ADC_ETC module.

88.5.1 ADC Trigger Concept – Dual PMSM motor control

ADC_ETC's flexible configuration is suitable for complex application timing design. The following describes the configuration scheme of ADC_ETC for dual motor control. There is a 180-degrees phase lag between the PWMs of dual motors, and on this basis ADC_ETC uses the triggers from eFlexPWM to implement time division sampling of analog signals of dual motors.

Figure 88-4 presents the synchronization between ADC and PWMs.

Once eFlexPWM1 counter reaches submodule 1 VAL4, ADC_ETC will be triggered to control ADCs to sampling analog signal of motor 1. After ADC conversation done, ADC_ETC_DONE0 interrupt is enabled to run motor 1 control algorithm. Once eFlexPWM2 counter reaches submodule 0 VAL4, ADC_ETC will be triggered to control ADCs to sampling analog signal of motor 2. After ADC conversation done, ADC_ETC_DONE1 interrupt is enabled to run motor 2 control algorithm.

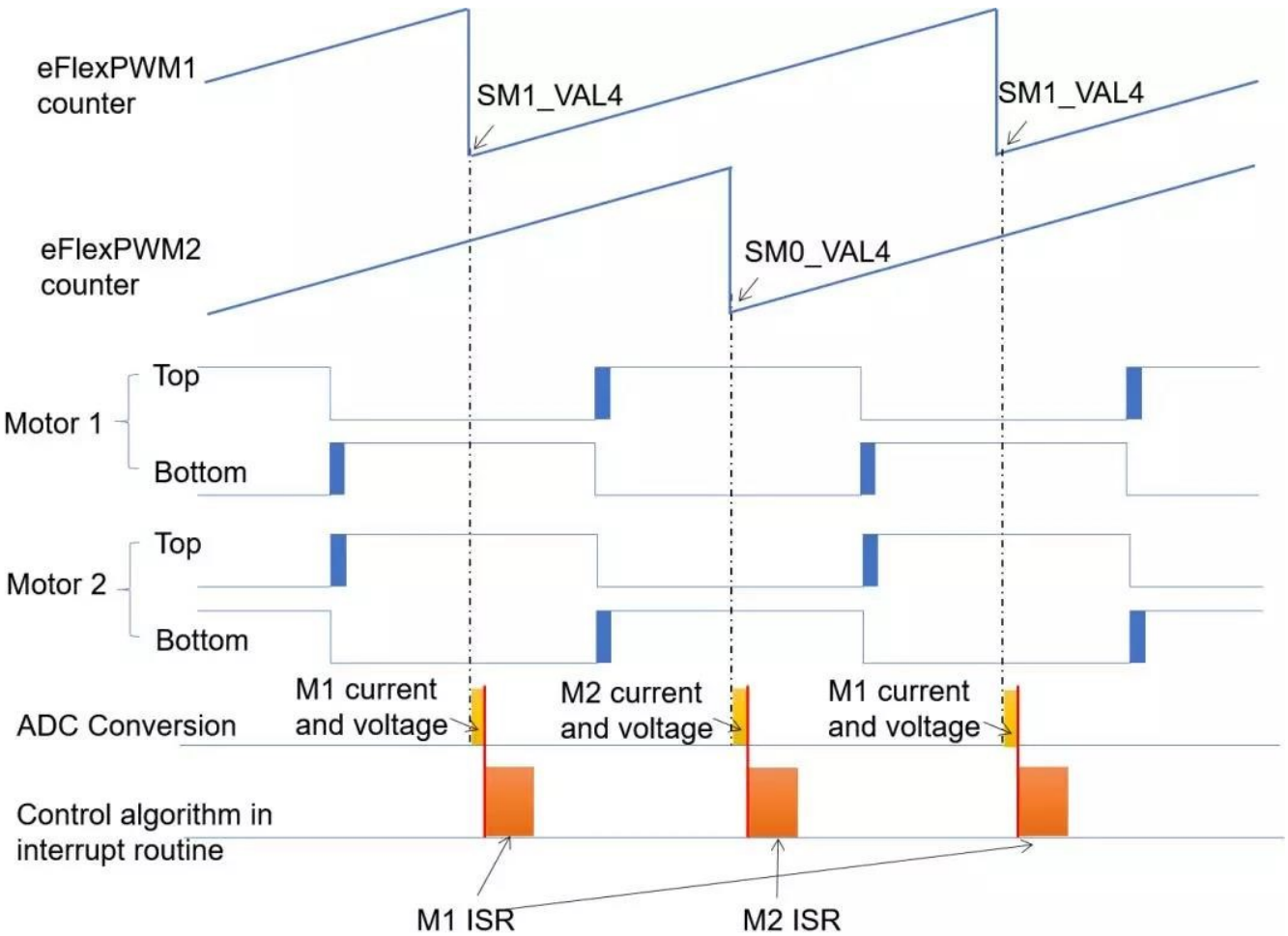


Figure 88-4. Synchronization between ADC and PWMs

The external triggers are generated from the Cross BAR (XBAR) or other sources. The ADC scan is started via ADC_ETC. The following is the key configuration point for this application.

- Enable external XBAR trigger 0 and 1, both ADCs have their own trigger segments.
- The trigger chain length is set to 2. The back-to-back ADC trigger mode is enabled.
- Enable the Sync Mode. In the Sync Mode, ADC1 and ADC2 are controlled by the same trigger source.
- Both trigger chain has their own finished interrupt. When Chain 4 triggered by TRIG0 is finished, enable DONE0 interrupt. When Chain 5 triggered by TRIG1 is finished, enable DONE1 interrupt.

Figure 88-5 shows configuration of ADC_ETC to control dual ADCs in this dual motor application.

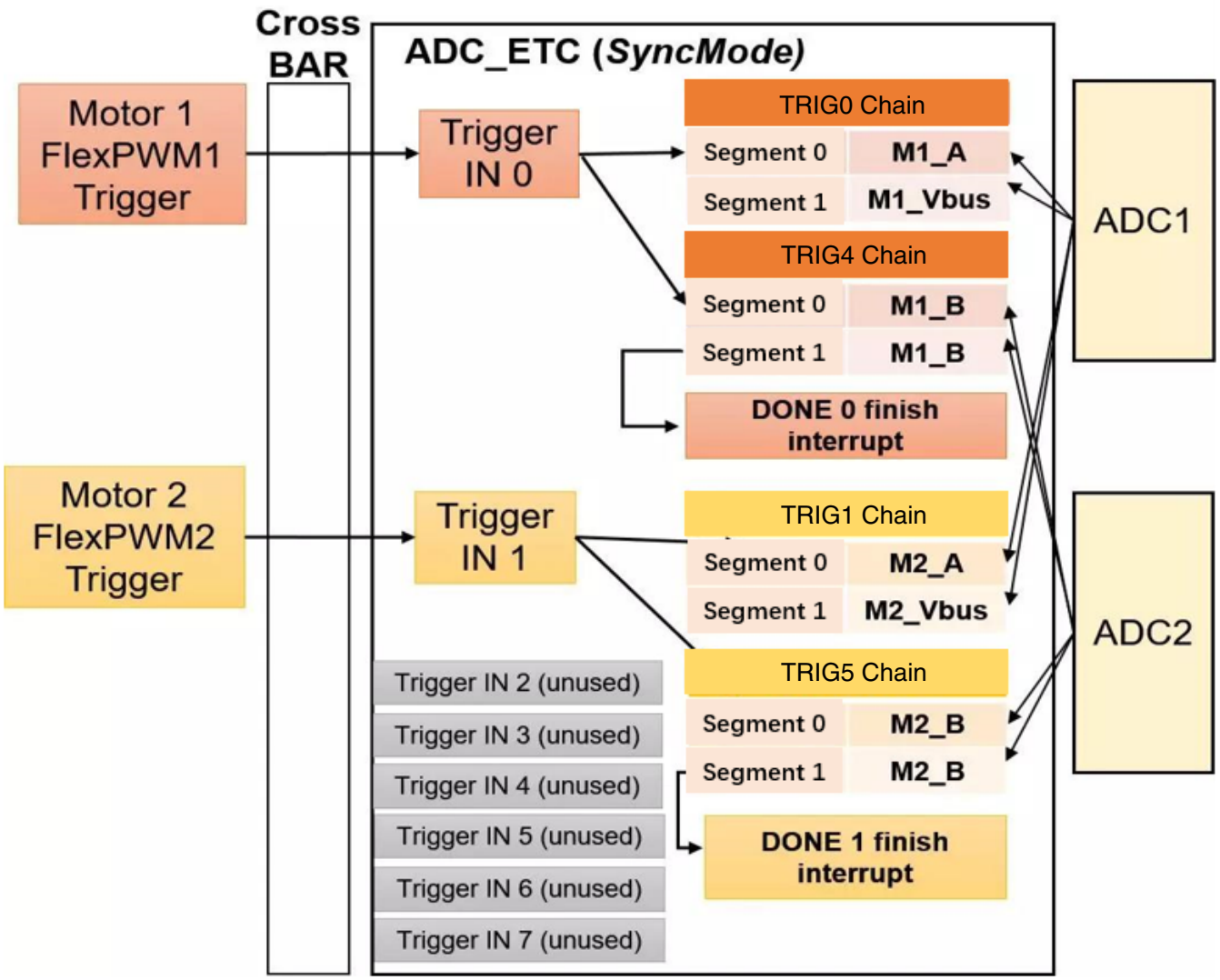


Figure 88-5. The configuration of ADC_ETC for dual motor control

88.6 Memory Map and register definition

This section includes the ADC_ETC module memory map and detailed descriptions of all registers.

88.6.1 ADC_ETC register descriptions

The descriptions below will list the memory map and give detailed explanations for all registers and the bit fields of the registers.

88.6.1.1 ADC_ETC memory map

ADC_ETC base address: 4004_8000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	ADC_ETC Global Control Register (CTRL)	32	RW	8000_0000
4	ETC DONE0 and DONE1 IRQ State Register (DONE0_1_IRQ)	32	W1C	0000_0000
8	ETC DONE_2, DONE_3 and DONE_ERR IRQ State Register (DONE 2_3_ERR_IRQ)	32	RW	0000_0000
C	ETC DMA control Register (DMA_CTRL)	32	RW	0000_0000
10	ETC_TRIG Control Register (TRIG0_CTRL)	32	RW	0000_0000
14	ETC_TRIG Counter Register (TRIG0_COUNTER)	32	RW	0000_0000
18	ETC_TRIG Chain 0/1 Register (TRIG0_CHAIN_1_0)	32	RW	0000_0000
1C	ETC_TRIG Chain 2/3 Register (TRIG0_CHAIN_3_2)	32	RW	0000_0000
20	ETC_TRIG Chain 4/5 Register (TRIG0_CHAIN_5_4)	32	RW	0000_0000
24	ETC_TRIG Chain 6/7 Register (TRIG0_CHAIN_7_6)	32	RW	0000_0000
28	ETC_TRIG Result Data 1/0 Register (TRIG0_RESULT_1_0)	32	RO	0000_0000
2C	ETC_TRIG Result Data 3/2 Register (TRIG0_RESULT_3_2)	32	RO	0000_0000
30	ETC_TRIG Result Data 5/4 Register (TRIG0_RESULT_5_4)	32	RO	0000_0000
34	ETC_TRIG Result Data 7/6 Register (TRIG0_RESULT_7_6)	32	RO	0000_0000
38	ETC_TRIG Control Register (TRIG1_CTRL)	32	RW	0000_0000
3C	ETC_TRIG Counter Register (TRIG1_COUNTER)	32	RW	0000_0000
40	ETC_TRIG Chain 0/1 Register (TRIG1_CHAIN_1_0)	32	RW	0000_0000
44	ETC_TRIG Chain 2/3 Register (TRIG1_CHAIN_3_2)	32	RW	0000_0000
48	ETC_TRIG Chain 4/5 Register (TRIG1_CHAIN_5_4)	32	RW	0000_0000
4C	ETC_TRIG Chain 6/7 Register (TRIG1_CHAIN_7_6)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
50	ETC_TRIG Result Data 1/0 Register (TRIG1_RESULT_1_0)	32	RO	0000_0000
54	ETC_TRIG Result Data 3/2 Register (TRIG1_RESULT_3_2)	32	RO	0000_0000
58	ETC_TRIG Result Data 5/4 Register (TRIG1_RESULT_5_4)	32	RO	0000_0000
5C	ETC_TRIG Result Data 7/6 Register (TRIG1_RESULT_7_6)	32	RO	0000_0000
60	ETC_TRIG Control Register (TRIG2_CTRL)	32	RW	0000_0000
64	ETC_TRIG Counter Register (TRIG2_COUNTER)	32	RW	0000_0000
68	ETC_TRIG Chain 0/1 Register (TRIG2_CHAIN_1_0)	32	RW	0000_0000
6C	ETC_TRIG Chain 2/3 Register (TRIG2_CHAIN_3_2)	32	RW	0000_0000
70	ETC_TRIG Chain 4/5 Register (TRIG2_CHAIN_5_4)	32	RW	0000_0000
74	ETC_TRIG Chain 6/7 Register (TRIG2_CHAIN_7_6)	32	RW	0000_0000
78	ETC_TRIG Result Data 1/0 Register (TRIG2_RESULT_1_0)	32	RO	0000_0000
7C	ETC_TRIG Result Data 3/2 Register (TRIG2_RESULT_3_2)	32	RO	0000_0000
80	ETC_TRIG Result Data 5/4 Register (TRIG2_RESULT_5_4)	32	RO	0000_0000
84	ETC_TRIG Result Data 7/6 Register (TRIG2_RESULT_7_6)	32	RO	0000_0000
88	ETC_TRIG Control Register (TRIG3_CTRL)	32	RW	0000_0000
8C	ETC_TRIG Counter Register (TRIG3_COUNTER)	32	RW	0000_0000
90	ETC_TRIG Chain 0/1 Register (TRIG3_CHAIN_1_0)	32	RW	0000_0000
94	ETC_TRIG Chain 2/3 Register (TRIG3_CHAIN_3_2)	32	RW	0000_0000
98	ETC_TRIG Chain 4/5 Register (TRIG3_CHAIN_5_4)	32	RW	0000_0000
9C	ETC_TRIG Chain 6/7 Register (TRIG3_CHAIN_7_6)	32	RW	0000_0000
A0	ETC_TRIG Result Data 1/0 Register (TRIG3_RESULT_1_0)	32	RO	0000_0000
A4	ETC_TRIG Result Data 3/2 Register (TRIG3_RESULT_3_2)	32	RO	0000_0000
A8	ETC_TRIG Result Data 5/4 Register (TRIG3_RESULT_5_4)	32	RO	0000_0000
AC	ETC_TRIG Result Data 7/6 Register (TRIG3_RESULT_7_6)	32	RO	0000_0000
B0	ETC_TRIG Control Register (TRIG4_CTRL)	32	RW	0000_0000
B4	ETC_TRIG Counter Register (TRIG4_COUNTER)	32	RW	0000_0000
B8	ETC_TRIG Chain 0/1 Register (TRIG4_CHAIN_1_0)	32	RW	0000_0000
BC	ETC_TRIG Chain 2/3 Register (TRIG4_CHAIN_3_2)	32	RW	0000_0000
C0	ETC_TRIG Chain 4/5 Register (TRIG4_CHAIN_5_4)	32	RW	0000_0000
C4	ETC_TRIG Chain 6/7 Register (TRIG4_CHAIN_7_6)	32	RW	0000_0000
C8	ETC_TRIG Result Data 1/0 Register (TRIG4_RESULT_1_0)	32	RO	0000_0000
CC	ETC_TRIG Result Data 3/2 Register (TRIG4_RESULT_3_2)	32	RO	0000_0000
D0	ETC_TRIG Result Data 5/4 Register (TRIG4_RESULT_5_4)	32	RO	0000_0000
D4	ETC_TRIG Result Data 7/6 Register (TRIG4_RESULT_7_6)	32	RO	0000_0000
D8	ETC_TRIG Control Register (TRIG5_CTRL)	32	RW	0000_0000
DC	ETC_TRIG Counter Register (TRIG5_COUNTER)	32	RW	0000_0000
E0	ETC_TRIG Chain 0/1 Register (TRIG5_CHAIN_1_0)	32	RW	0000_0000
E4	ETC_TRIG Chain 2/3 Register (TRIG5_CHAIN_3_2)	32	RW	0000_0000
E8	ETC_TRIG Chain 4/5 Register (TRIG5_CHAIN_5_4)	32	RW	0000_0000
EC	ETC_TRIG Chain 6/7 Register (TRIG5_CHAIN_7_6)	32	RW	0000_0000

Table continues on the next page...

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
F0	ETC_TRIG Result Data 1/0 Register (TRIG5_RESULT_1_0)	32	RO	0000_0000
F4	ETC_TRIG Result Data 3/2 Register (TRIG5_RESULT_3_2)	32	RO	0000_0000
F8	ETC_TRIG Result Data 5/4 Register (TRIG5_RESULT_5_4)	32	RO	0000_0000
FC	ETC_TRIG Result Data 7/6 Register (TRIG5_RESULT_7_6)	32	RO	0000_0000
100	ETC_TRIG Control Register (TRIG6_CTRL)	32	RW	0000_0000
104	ETC_TRIG Counter Register (TRIG6_COUNTER)	32	RW	0000_0000
108	ETC_TRIG Chain 0/1 Register (TRIG6_CHAIN_1_0)	32	RW	0000_0000
10C	ETC_TRIG Chain 2/3 Register (TRIG6_CHAIN_3_2)	32	RW	0000_0000
110	ETC_TRIG Chain 4/5 Register (TRIG6_CHAIN_5_4)	32	RW	0000_0000
114	ETC_TRIG Chain 6/7 Register (TRIG6_CHAIN_7_6)	32	RW	0000_0000
118	ETC_TRIG Result Data 1/0 Register (TRIG6_RESULT_1_0)	32	RO	0000_0000
11C	ETC_TRIG Result Data 3/2 Register (TRIG6_RESULT_3_2)	32	RO	0000_0000
120	ETC_TRIG Result Data 5/4 Register (TRIG6_RESULT_5_4)	32	RO	0000_0000
124	ETC_TRIG Result Data 7/6 Register (TRIG6_RESULT_7_6)	32	RO	0000_0000
128	ETC_TRIG Control Register (TRIG7_CTRL)	32	RW	0000_0000
12C	ETC_TRIG Counter Register (TRIG7_COUNTER)	32	RW	0000_0000
130	ETC_TRIG Chain 0/1 Register (TRIG7_CHAIN_1_0)	32	RW	0000_0000
134	ETC_TRIG Chain 2/3 Register (TRIG7_CHAIN_3_2)	32	RW	0000_0000
138	ETC_TRIG Chain 4/5 Register (TRIG7_CHAIN_5_4)	32	RW	0000_0000
13C	ETC_TRIG Chain 6/7 Register (TRIG7_CHAIN_7_6)	32	RW	0000_0000
140	ETC_TRIG Result Data 1/0 Register (TRIG7_RESULT_1_0)	32	RO	0000_0000
144	ETC_TRIG Result Data 3/2 Register (TRIG7_RESULT_3_2)	32	RO	0000_0000
148	ETC_TRIG Result Data 5/4 Register (TRIG7_RESULT_5_4)	32	RO	0000_0000
14C	ETC_TRIG Result Data 7/6 Register (TRIG7_RESULT_7_6)	32	RO	0000_0000

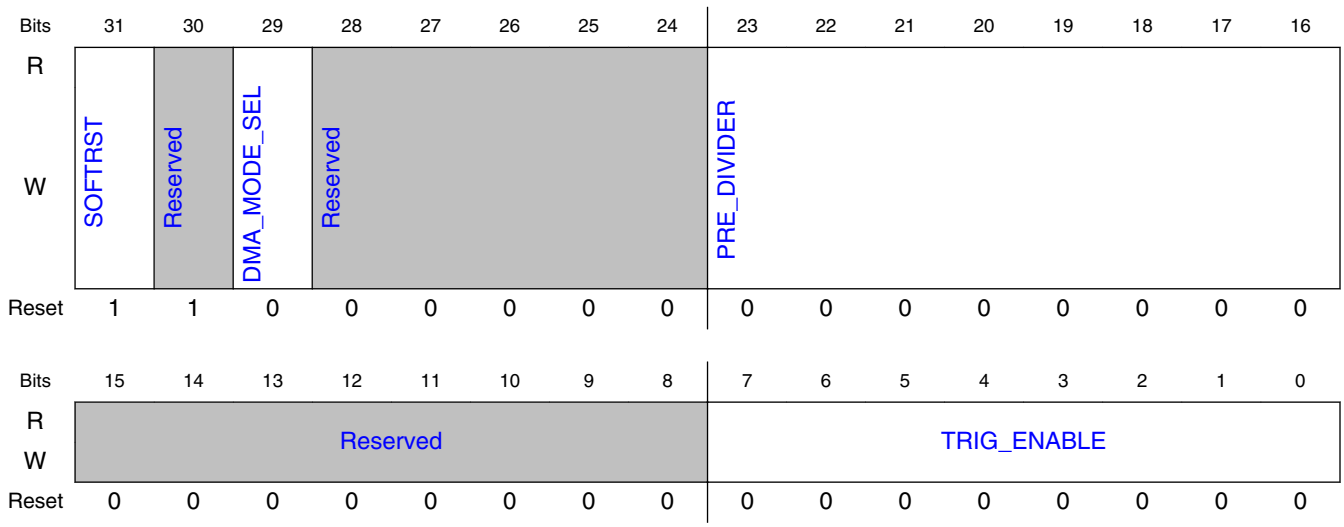
88.6.1.2 ADC_ETC Global Control Register (CTRL)

This register controls various global functions of the ADC_ETC, such as enabling the external XBAR triggers, setting the pre-division factor for trigger delay and interval, and selecting the trigger type of the DMA_REQ. In addition, this register contains SOFTRST to reset all registers inside ADC_ETC.

88.6.1.2.1 Offset

Register	Offset
CTRL	0h

88.6.1.2.2 Diagram



88.6.1.2.3 Fields

Field	Description
31 SOFTRST	Software synchronous reset, active high. 0 - ADC_ETC works normally. 1 - All registers inside ADC_ETC will be reset to the default value.
30 —	Reserved
29 DMA_MODE_SEL	Select the trigger type of the DMA_REQ. 0 - Trig DMA_REQ with latched signal, REQ will be cleared when ACK and source request cleared. 1 - Trig DMA_REQ with pulsed signal, REQ will be cleared by ACK only.
28-24 —	Reserved
23-16 PRE_DIVIDER	Pre-divider for trig delay and interval. The meaning of this field will be explained in the description of TRIGa_COUNTER.
15-8 —	Reserved
7-0 TRIG_ENABLE	TRIG enable register. 00000000 - disable all 8 external XBAR triggers. 00000001 - enable external XBAR trigger0. 00000010 - enable external XBAR trigger1. 00000011 - enable external XBAR trigger0 and trigger1. 11111111 - enable all 8 external XBAR triggers.

88.6.1.3 ETC DONE0 and DONE1 IRQ State Register (DONE0_1_IRQ)

This register has the state for ETC DONE0 and DONE1 interrupts.

NOTE

The IRQs can be cleared by writing 1 to the corresponding bits.

88.6.1.3.1 Offset

Register	Offset
DONE0_1_IRQ	4h

88.6.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								TRIG7_DONE1	TRIG6_DONE1	TRIG5_DONE1	TRIG4_DONE1	TRIG3_DONE1	TRIG2_DONE1	TRIG1_DONE1	TRIG0_DONE1
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TRIG7_DONE0	TRIG6_DONE0	TRIG5_DONE0	TRIG4_DONE0	TRIG3_DONE0	TRIG2_DONE0	TRIG1_DONE0	TRIG0_DONE0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

88.6.1.3.3 Fields

Field	Description
31-24	Reserved

Table continues on the next page...

Field	Description
—	
23 TRIG7_DONE1	TRIG7 done1 interrupt detection. 0 - No TRIG7_DONE1 interrupt detected 1 - TRIG7_DONE1 interrupt detected
22 TRIG6_DONE1	TRIG6 done1 interrupt detection. 0 - No TRIG6_DONE1 interrupt detected 1 - TRIG6_DONE1 interrupt detected
21 TRIG5_DONE1	TRIG5 done1 interrupt detection. 0 - No TRIG5_DONE1 interrupt detected 1 - TRIG5_DONE1 interrupt detected
20 TRIG4_DONE1	TRIG4 done1 interrupt detection. 0 - No TRIG4_DONE1 interrupt detected 1 - TRIG4_DONE1 interrupt detected
19 TRIG3_DONE1	TRIG3 done1 interrupt detection. 0 - No TRIG3_DONE1 interrupt detected 1 - TRIG3_DONE1 interrupt detected
18 TRIG2_DONE1	TRIG2 done1 interrupt detection. 0 - No TRIG2_DONE1 interrupt detected 1 - TRIG2_DONE1 interrupt detected
17 TRIG1_DONE1	TRIG1 done1 interrupt detection. 0 - No TRIG1_DONE1 interrupt detected 1 - TRIG1_DONE1 interrupt detected
16 TRIG0_DONE1	TRIG0 done1 interrupt detection. 0 - No TRIG0_DONE1 interrupt detected 1 - TRIG0_DONE1 interrupt detected
15-8 —	Reserved
7 TRIG7_DONE0	TRIG7 done0 interrupt detection. 0 - No TRIG7_DONE0 interrupt detected 1 - TRIG7_DONE0 interrupt detected
6 TRIG6_DONE0	TRIG6 done0 interrupt detection. 0 - No TRIG6_DONE0 interrupt detected 1 - TRIG6_DONE0 interrupt detected
5 TRIG5_DONE0	TRIG5 done0 interrupt detection. 0 - No TRIG5_DONE0 interrupt detected 1 - TRIG5_DONE0 interrupt detected
4 TRIG4_DONE0	TRIG4 done0 interrupt detection. 0 - No TRIG4_DONE0 interrupt detected 1 - TRIG4_DONE0 interrupt detected

Table continues on the next page...

Field	Description
3 TRIG3_DONE0	TRIG3 done0 interrupt detection. 0 - No TRIG3_DONE0 interrupt detected 1 - TRIG3_DONE0 interrupt detected
2 TRIG2_DONE0	TRIG2 done0 interrupt detection. 0 - No TRIG2_DONE0 interrupt detected 1 - TRIG2_DONE0 interrupt detected
1 TRIG1_DONE0	TRIG1 done0 interrupt detection. 0 - No TRIG1_DONE0 interrupt detected 1 - TRIG1_DONE0 interrupt detected
0 TRIG0_DONE0	TRIG0 done0 interrupt detection. 0 - No TRIG0_DONE0 interrupt detected 1 - TRIG0_DONE0 interrupt detected

88.6.1.4 ETC DONE_2, DONE_3 and DONE_ERR IRQ State Register (DONE2_3_ERR_IRQ)

This register has the state for ETC DONE2, DONE3 and DONE_ERR interrupts.

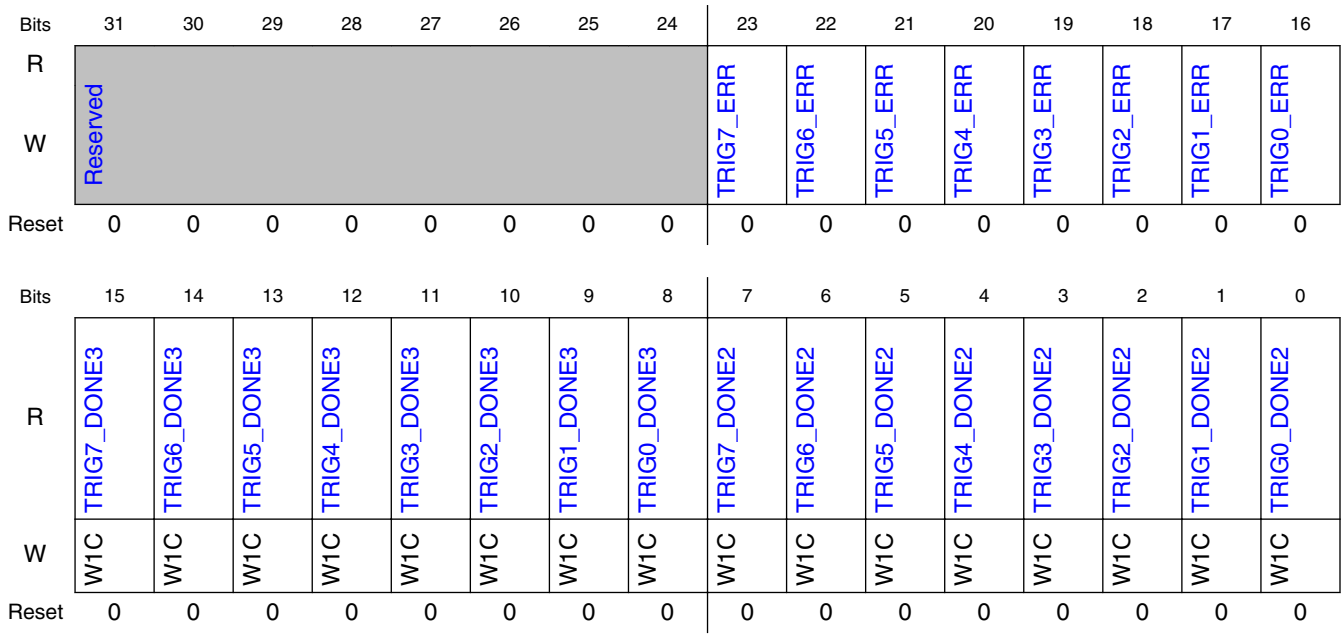
NOTE

The IRQs can be cleared by writing 1 to the corresponding bits.

88.6.1.4.1 Offset

Register	Offset
DONE2_3_ERR_IRQ	8h

88.6.1.4.2 Diagram



88.6.1.4.3 Fields

Field	Description
31-24 —	Reserved
23 TRIG7_ERR	TRIG7 error interrupt detection. 0 - No TRIG7_ERR interrupt detected 1 - TRIG7_ERR interrupt detected
22 TRIG6_ERR	TRIG6 error interrupt detection. 0 - No TRIG6_ERR interrupt detected 1 - TRIG6_ERR interrupt detected
21 TRIG5_ERR	TRIG5 error interrupt detection. 0 - No TRIG5_ERR interrupt detected 1 - TRIG5_ERR interrupt detected
20 TRIG4_ERR	TRIG4 error interrupt detection. 0 - No TRIG4_ERR interrupt detected 1 - TRIG4_ERR interrupt detected
19 TRIG3_ERR	TRIG3 error interrupt detection. 0 - No TRIG3_ERR interrupt detected 1 - TRIG3_ERR interrupt detected
18 TRIG2_ERR	TRIG2 error interrupt detection. 0 - No TRIG2_ERR interrupt detected

Table continues on the next page...

Field	Description
	1 - TRIG2_ERR interrupt detected
17 TRIG1_ERR	TRIG1 error interrupt detection. 0 - No TRIG1_ERR interrupt detected 1 - TRIG1_ERR interrupt detected
16 TRIG0_ERR	TRIG0 error interrupt detection. 0 - No TRIG0_ERR interrupt detected 1 - TRIG0_ERR interrupt detected
15 TRIG7_DONE3	TRIG7 done3 interrupt detection. 0 - No TRIG7_DONE3 interrupt detected 1 - TRIG7_DONE3 interrupt detected
14 TRIG6_DONE3	TRIG6 done3 interrupt detection. 0 - No TRIG6_DONE3 interrupt detected 1 - TRIG6_DONE3 interrupt detected
13 TRIG5_DONE3	TRIG5 done3 interrupt detection. 0 - No TRIG5_DONE3 interrupt detected 1 - TRIG5_DONE3 interrupt detected
12 TRIG4_DONE3	TRIG4 done3 interrupt detection. 0 - No TRIG4_DONE3 interrupt detected 1 - TRIG4_DONE3 interrupt detected
11 TRIG3_DONE3	TRIG3 done3 interrupt detection. 0 - No TRIG3_DONE3 interrupt detected 1 - TRIG3_DONE3 interrupt detected
10 TRIG2_DONE3	TRIG2 done3 interrupt detection. 0 - No TRIG2_DONE3 interrupt detected 1 - TRIG2_DONE3 interrupt detected
9 TRIG1_DONE3	TRIG1 done3 interrupt detection. 0 - No TRIG1_DONE3 interrupt detected 1 - TRIG1_DONE3 interrupt detected
8 TRIG0_DONE3	TRIG0 done3 interrupt detection. 0 - No TRIG0_DONE3 interrupt detected 1 - TRIG0_DONE3 interrupt detected
7 TRIG7_DONE2	TRIG7 done2 interrupt detection. 0 - No TRIG7_DONE2 interrupt detected 1 - TRIG7_DONE2 interrupt detected
6 TRIG6_DONE2	TRIG6 done2 interrupt detection. 0 - No TRIG6_DONE2 interrupt detected 1 - TRIG6_DONE2 interrupt detected
5 TRIG5_DONE2	TRIG5 done2 interrupt detection. 0 - No TRIG5_DONE2 interrupt detected

Table continues on the next page...

Field	Description
	1 - TRIG5_DONE2 interrupt detected
4 TRIG4_DONE2	TRIG4 done2 interrupt detection. 0 - No TRIG4_DONE2 interrupt detected 1 - TRIG4_DONE2 interrupt detected
3 TRIG3_DONE2	TRIG3 done2 interrupt detection. 0 - No TRIG3_DONE2 interrupt detected 1 - TRIG3_DONE2 interrupt detected
2 TRIG2_DONE2	TRIG2 done2 interrupt detection. 0 - No TRIG2_DONE2 interrupt detected 1 - TRIG2_DONE2 interrupt detected
1 TRIG1_DONE2	TRIG1 done2 interrupt detection. 0 - No TRIG1_DONE2 interrupt detected 1 - TRIG1_DONE2 interrupt detected
0 TRIG0_DONE2	TRIG0 done2 interrupt detection. 0 - No TRIG0_DONE2 interrupt detected 1 - TRIG0_DONE2 interrupt detected

88.6.1.5 ETC DMA control Register (DMA_CTRL)

This register has the control and status bit for the DMA of ADC_ETC. Each trigger has separate enabling and status bit in this register. The control bit enables the DMA request for each trigger, and the status bit indicates whether the corresponding trigger request was detected.

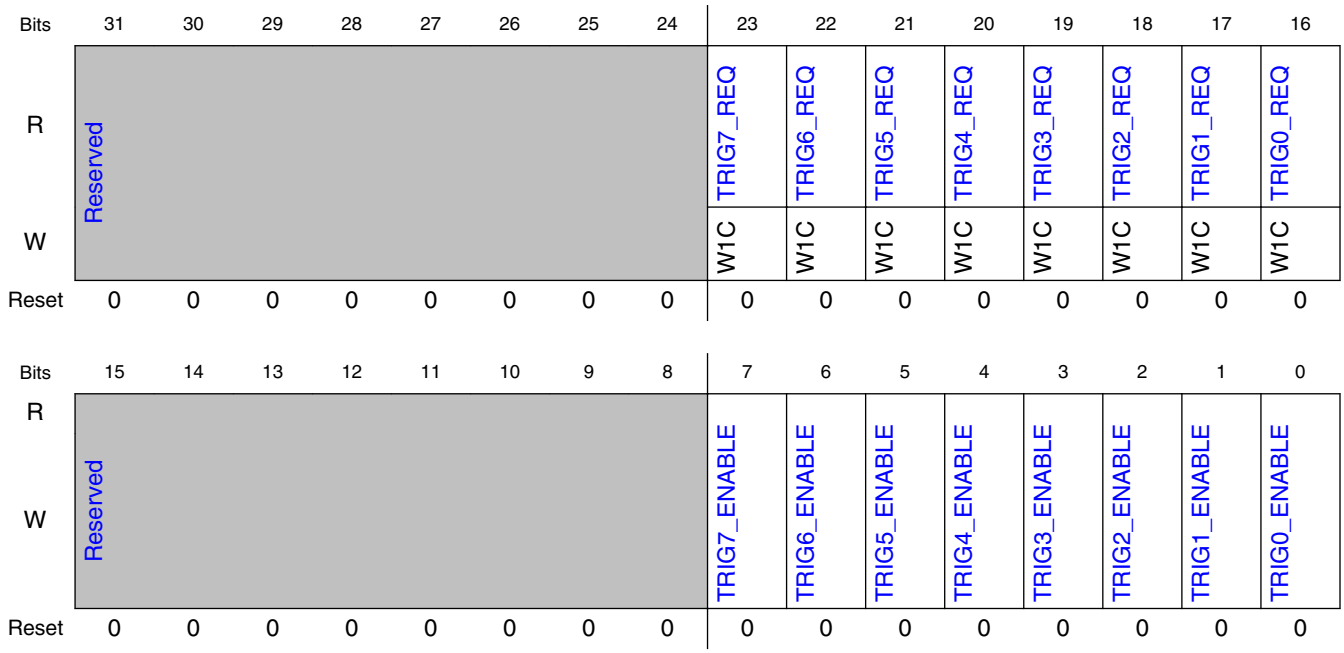
NOTE

TRIGa_REQ can be cleard by writing 1 to the corresponding bit.

88.6.1.5.1 Offset

Register	Offset
DMA_CTRL	Ch

88.6.1.5.2 Diagram



88.6.1.5.3 Fields

Field	Description
31-24 —	Reserved
23 TRIG7_REQ	Flag bit for DMA request. TRIG7_REQ being active indicates trigger7 done and a DMA transmission is needed. If TRIG7_ENABLE is also set, a DMA request will be sent. 0 - TRIG7_REQ not detected. 1 - TRIG7_REQ detected.
22 TRIG6_REQ	Flag bit for DMA request. TRIG6_REQ being active indicates trigger6 done and a DMA transmission is needed. If TRIG6_ENABLE is also set, a DMA request will be sent. 0 - TRIG6_REQ not detected. 1 - TRIG6_REQ detected.
21 TRIG5_REQ	Flag bit for DMA request. TRIG5_REQ being active indicates trigger5 done and a DMA transmission is needed. If TRIG5_ENABLE is also set, a DMA request will be sent. 0 - TRIG5_REQ not detected. 1 - TRIG5_REQ detected.
20 TRIG4_REQ	Flag bit for DMA request. TRIG4_REQ being active indicates trigger4 done and a DMA transmission is needed. If TRIG4_ENABLE is also set, a DMA request will be sent. 0 - TRIG4_REQ not detected. 1 - TRIG4_REQ detected.
19 TRIG3_REQ	Flag bit for DMA request. TRIG3_REQ being active indicates trigger3 done and a DMA transmission is needed. If TRIG3_ENABLE is also set, a DMA request will be sent.

Table continues on the next page...

Field	Description
	0 - TRIG3_REQ not detected. 1 - TRIG3_REQ detected.
18 TRIG2_REQ	Flag bit for DMA request. TRIG2_REQ being active indicates trigger2 done and a DMA transmission is needed. If TRIG2_ENABLE is also set, a DMA request will be sent. 0 - TRIG2_REQ not detected. 1 - TRIG2_REQ detected.
17 TRIG1_REQ	Flag bit for DMA request. TRIG1_REQ being active indicates trigger1 done and a DMA transmission is needed. If TRIG1_ENABLE is also set, a DMA request will be sent. 0 - TRIG1_REQ not detected. 1 - TRIG1_REQ detected.
16 TRIG0_REQ	Flag bit for DMA request. TRIG0_REQ being active indicates trigger0 done and a DMA transmission is needed. If TRIG0_ENABLE is also set, a DMA request will be sent. 0 - TRIG0_REQ not detected. 1 - TRIG0_REQ detected.
15-8 —	Reserved
7 TRIG7_ENABLE	Enable DMA request when TRIG7 done. 0 - TRIG7 DMA request disabled. 1 - TRIG7 DMA request enabled.
6 TRIG6_ENABLE	Enable DMA request when TRIG6 done. 0 - TRIG6 DMA request disabled. 1 - TRIG6 DMA request enabled.
5 TRIG5_ENABLE	Enable DMA request when TRIG5 done. 0 - TRIG5 DMA request disabled. 1 - TRIG5 DMA request enabled.
4 TRIG4_ENABLE	Enable DMA request when TRIG4 done. 0 - TRIG4 DMA request disabled. 1 - TRIG4 DMA request enabled.
3 TRIG3_ENABLE	Enable DMA request when TRIG3 done. 0 - TRIG3 DMA request disabled. 1 - TRIG3 DMA request enabled.
2 TRIG2_ENABLE	Enable DMA request when TRIG2 done. 0 - TRIG2 DMA request disabled. 1 - TRIG2 DMA request enabled.
1 TRIG1_ENABLE	Enable DMA request when TRIG1 done. 0 - TRIG1 DMA request disabled. 1 - TRIG1 DMA request enabled.
0 TRIG0_ENABLE	Enable DMA request when TRIG0 done. 0 - TRIG0 DMA request disabled. 1 - TRIG0 DMA request enabled.

88.6.1.6 ETC_TRIG Control Register (TRIG0_CTRL - TRIG7_CTRL)

This register contains the control bits for trigger, such as selecting synchronization mode, setting trigger priority and setting the length of trigger chain. It also contains a bit for selecting trigger mode between hardware trigger and software trigger, and a self-clearing bit for launching a software trigger.

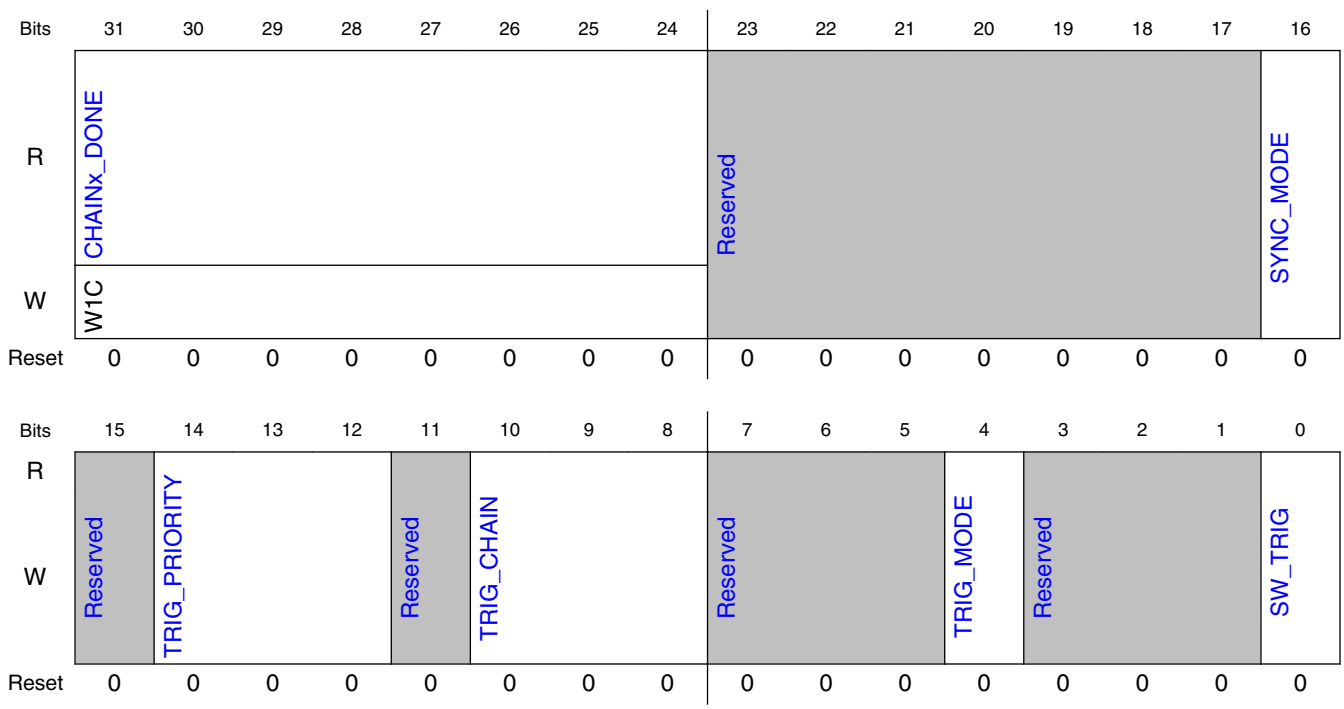
The maximum length of trigger chain is 8, so there are 8 status bits in this register.

88.6.1.6.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_CTRL	10h + (a × 28h)

88.6.1.6.2 Diagram



88.6.1.6.3 Fields

Field	Description
31-24 CHAINx_DONE	Segment x done detection. Each trigger of ADC_ETC supports up to 8 segments, and each segment will launch a trigger of ADC to start an ADC convertion. Bit 24 is for segment 0 and bit 31 is for segment 7. **NOTE:** Each bit in this field is cleared by writing a logic 1 to it. 00000000 - segment x done not detected. 00000001 - segment x done detected.
23-17 —	Reserved
16 SYNC_MODE	Trigger synchronization mode selection. **NOTE:** Setting SYNC_MODE = 1 for TRIGa or TRIG(a+4) represents the same meaning. 0 - Synchronization mode disabled, TRIGa and TRIG(a+4) are triggered independently. 1 - Synchronization mode enabled, TRIGa and TRIG(a+4) are triggered by TRIGa source synchronously.
15 —	Reserved
14-12 TRIG_PRIORITY	External trigger priority, 7 is highest priority, while 0 is lowest. **NOTE:** When more than one triggers use the same level setting of priority, then round-robin mechanism implemented and the trigger with larger number owns the higher priority.
11 —	Reserved
10-8 TRIG_CHAIN	The number of segments inside the trigger chain of TRIGa. 000 - Trigger chain length is 1 001 - Trigger chain length is 2 010 - Trigger chain length is 3 011 - Trigger chain length is 4 100 - Trigger chain length is 5 101 - Trigger chain length is 6 110 - Trigger chain length is 7 111 - Trigger chain length is 8
7-5 —	Reserved
4 TRIG_MODE	Trigger mode selection. 0 - Hardware trigger. The softerware trigger will be ignored. 1 - Software trigger. The hardware trigger will be ignored.
3-1 —	Reserved
0 SW_TRIG	Software trigger. **NOTE:** This field is self-clearing. 0 - No software trigger event generated. 1 - Software trigger event generated.

88.6.1.7 ETC_TRIG Counter Register (TRIG0_COUNTER - TRIG7_COUNTER)

This register controls the initial delay ahead the first ADC trigger and interval delay between ADC triggers launched by the same ADC_ETC trigger.

88.6.1.7.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_COUNTER	14h + (a × 28h)

88.6.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SAMPLE_INTERVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_DELAY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

88.6.1.7.3 Fields

Field	Description
31-16 SAMPLE_INTERVAL	TRIGGER sampling interval counter. When B2B is unset: Interval_delay = (SAMPLE_INTERVAL+1)*(PRE_DIVIDER+1)*ipg_clk
15-0 INIT_DELAY	TRIGGER initial delay counter. Initial_delay = (INIT_DELAY+1)*(PRE_DIVIDER+1)*ipg_clk

88.6.1.8 ETC_TRIG Chain 0/1 Register (TRIG0_CHAIN_1_0 - TRIG7_CHAIN_1_0)

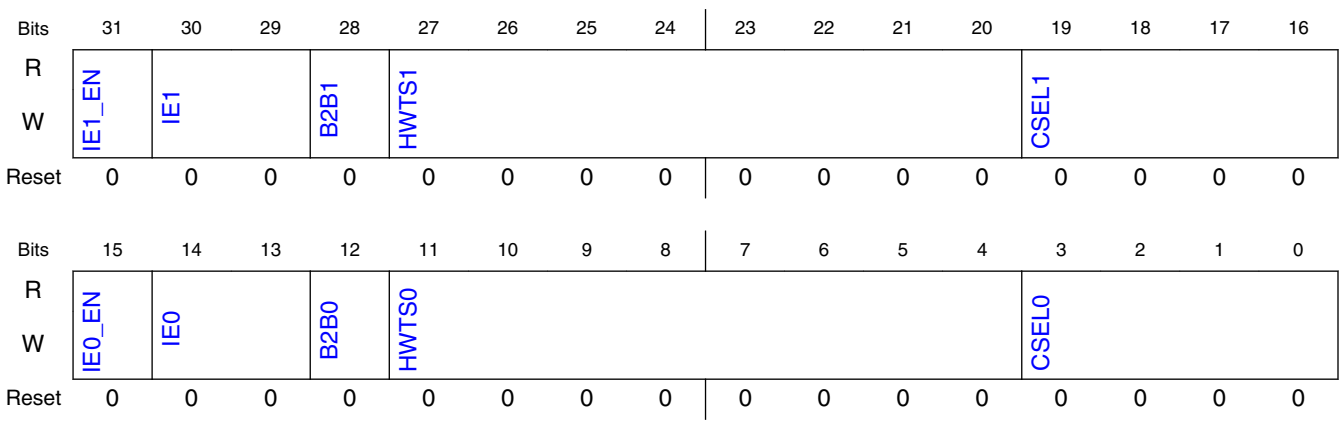
This register controls ETC_TRIG segment 0/1 configuration, such as enabling and selecting interrupt, setting back-to-back mode,selecting ADC trigger and trigger command.

88.6.1.8.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_CHAIN_1_0	18h + (a × 28h)

88.6.1.8.2 Diagram



88.6.1.8.3 Fields

Field	Description
31 IE1_EN	IRQ enable of segment 1. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 1 finish, an interrupt will be generated on the specific port configured by the IE1.
30-29 IE1	Segment 1 interrupt enable. (This bit field is meaningful only when IE1_EN is set) 00 - Generate interrupt on Done0 when Segment 1 finish. 01 - Generate interrupt on Done1 when Segment 1 finish. 10 - Generate interrupt on Done2 when Segment 1 finish.

Table continues on the next page...

Field	Description
	11 - Generate interrupt on Done3 when Segment 1 finish.
28 B2B1	Segment 1 B2B 0 - Disable B2B. Wait until delay value defined by TRIG1_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
27-20 HWTS1	Segment 1 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
19-16 CSEL1	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.
15 IE0_EN	IRQ enable of segment 0. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 0 finish, an interrupt will be generated on the specific port configured by the IE0.

Table continues on the next page...

Field	Description
14-13 IE0	Segment 0 interrupt enable. (This bit field is meaningful only when IE0_EN is set) 00 - Generate interrupt on Done0 when segment 0 finish. 01 - Generate interrupt on Done1 when segment 0 finish. 10 - Generate interrupt on Done2 when segment 0 finish. 11 - Generate interrupt on Done3 when segment 0 finish.
12 B2B0	Segment 0 B2B 0 - Disable B2B. Wait until delay value defined by TRIG0_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
11-4 HWTS0	Segment 0 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
3-0 CSEL0	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.

88.6.1.9 ETC_TRIG Chain 2/3 Register (TRIG0_CHAIN_3_2 - TRIG7_CHAIN_3_2)

This register controls ETC_TRIG segment 2/3 configuration, such as enabling and selecting interrupt, setting back-to-back mode,selecting ADC trigger and trigger command.

88.6.1.9.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_CHAIN_3_2	1Ch + (a × 28h)

88.6.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IE3_EN	IE3		B2B3	HWTS3								CSEL3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IE2_EN	IE2		B2B2	HWTS2								CSEL2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

88.6.1.9.3 Fields

Field	Description
31 IE3_EN	IRQ enable of segment 3. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 3 finish, an interrupt will be generated on the specific port configured by the IE3.
30-29	Segment 3 interrupt enable. (This bit field is meaningful only when IE3_EN is set)

Table continues on the next page...

Field	Description
IE3	00 - Generate interrupt on Done0 when segment 3 finish. 01 - Generate interrupt on Done1 when segment 3 finish. 10 - Generate interrupt on Done2 when segment 3 finish. 11 - Generate interrupt on Done3 when segment 3 finish.
28 B2B3	Segment 3 B2B 0 - Disable B2B. Wait until delay value defined by TRIG3_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
27-20 HWTS3	Segment 3 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
19-16 CSEL3	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.

Table continues on the next page...

Field	Description
15 IE2_EN	IRQ enable of segment 2. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 2 finish, an interrupt will be generated on the specific port configured by the IE2.
14-13 IE2	Segment 2 interrupt enable. (This bit field is meaningful only when IE2_EN is set) 00 - Generate interrupt on Done0 when segment 2 finish. 01 - Generate interrupt on Done1 when segment 2 finish. 10 - Generate interrupt on Done2 when segment 2 finish. 11 - Generate interrupt on Done3 when segment 2 finish.
12 B2B2	Segment 2 B2B 0 - Disable B2B. Wait until delay value defined by TRIG2_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
11-4 HWTS2	Segment 2 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
3-0 CSEL2	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected.

Field	Description
	1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.

88.6.1.10 ETC_TRIG Chain 4/5 Register (TRIG0_CHAIN_5_4 - TRIG7_CHAIN_5_4)

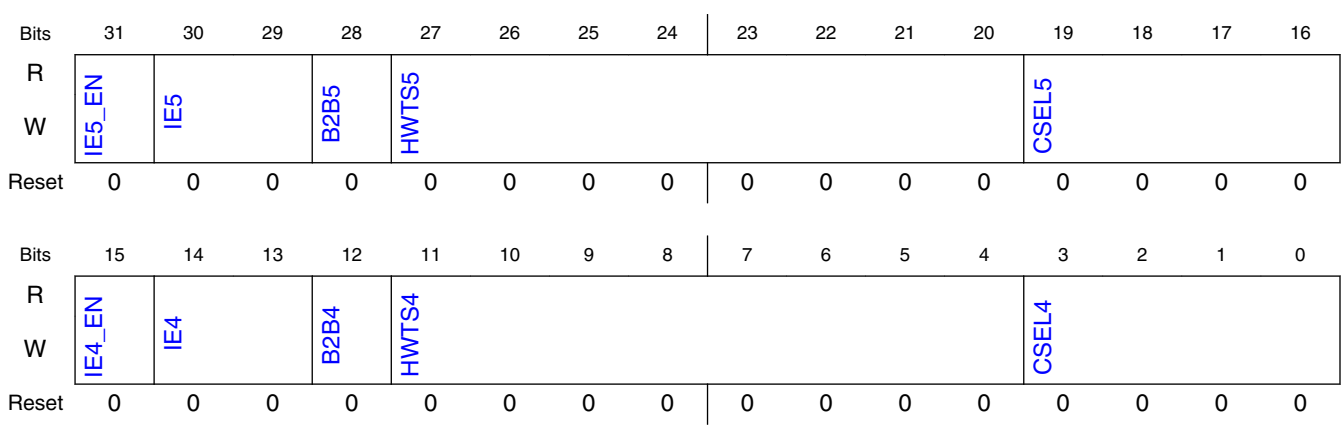
This register controls ETC_TRIG segment 4/5 configuration, such as enabling and selecting interrupt, setting back-to-back mode,selecting ADC trigger and trigger command.

88.6.1.10.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_CHAIN_5_4	20h + (a × 28h)

88.6.1.10.2 Diagram



88.6.1.10.3 Fields

Field	Description
31 IE5_EN	IRQ enable of segment 5. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 5 finish, an interrupt will be generated on the specific port configured by the IE5.
30-29 IE5	Segment 5 interrupt enable. (This bit field is meaningful only when IE5_EN is set) 00 - Generate interrupt on Done0 when segment 5 finish. 01 - Generate interrupt on Done1 when segment 5 finish. 10 - Generate interrupt on Done2 when segment 5 finish. 11 - Generate interrupt on Done3 when segment 5 finish.
28 B2B5	Segment 5 B2B 0 - Disable B2B. Wait until delay value defined by TRIG5_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
27-20 HWTS5	Segment 5 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
19-16 CSEL5	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected.

Table continues on the next page...

Field	Description
	1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.
15 IE4_EN	IRQ enable of segment 4. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 4 finish, an interrupt will be generated on the specific port configured by the IE4.
14-13 IE4	Segment 4 interrupt enable. (This bit field is meaningful only when IE4_EN is set) 00 - Generate interrupt on Done0 when segment 4 finish. 01 - Generate interrupt on Done1 when segment 4 finish. 10 - Generate interrupt on Done2 when segment 4 finish. 11 - Generate interrupt on Done3 when segment 4 finish.
12 B2B4	Segment 4 B2B 0 - Disable B2B. Wait until delay value defined by TRIG4_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
11-4 HWTS4	Segment 4 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
3-0 CSEL4	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected.

Field	Description
	0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.

88.6.1.11 ETC_TRIG Chain 6/7 Register (TRIG0_CHAIN_7_6 - TRIG7_CHAIN_7_6)

This register controls ETC_TRIG Chain 6/7 configuration, such as enabling and selecting interrupt, setting back-to-back mode,selecting ADC trigger and trigger command.

88.6.1.11.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_CHAIN_7_6	24h + (a × 28h)

88.6.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IE7_EN	IE7		B2B7	HWTS7								CSEL7			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IE6_EN	IE6		B2B6	HWTS6								CSEL6			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

88.6.1.11.3 Fields

Field	Description
31 IE7_EN	IRQ enable of segment 7. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 7 finish, an interrupt will be generated on the specific port configured by the IE7.
30-29 IE7	Segment 7 interrupt enable. (This bit field is meaningful only when IE7_EN is set) 00 - Generate interrupt on Done0 when segment 7 finish. 01 - Generate interrupt on Done1 when segment 7 finish. 10 - Generate interrupt on Done2 when segment 7 finish. 11 - Generate interrupt on Done3 when segment 7 finish.
28 B2B7	Segment 7 B2B 0 - Disable B2B. Wait until delay value defined by TRIG7_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
27-20 HWTS7	Segment 7 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
19-16 CSEL7	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected. 0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected.

Table continues on the next page...

Field	Description
	1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.
15 IE6_EN	IRQ enable of segment 6. 0 - Interrupt DONE disabled. 1 - Interrupt DONE enabled. When segment 6 finish, an interrupt will be generated on the specific port configured by the IE6.
14-13 IE6	Segment 6 interrupt enable. (This bit field is meaningful only when IE6_EN is set) 00 - Generate interrupt on Done0 when segment 6 finish. 01 - Generate interrupt on Done1 when segment 6 finish. 10 - Generate interrupt on Done2 when segment 6 finish. 11 - Generate interrupt on Done3 when segment 6 finish.
12 B2B6	Segment 6 B2B 0 - Disable B2B. Wait until delay value defined by TRIG6_COUNTER[SAMPLE_INTERVAL] is reached 1 - Enable B2B. When Segment 0 finished (ADC COCO) then automatically trigger next ADC conversion, no need to wait until interval delay reached.
11-4 HWTS6	Segment 6 HWTS ADC hardware trigger selection. This field will control the ext_hwts of ADC (see the ADC_ETC block diagram). **NOTE:** Please keep only one bit set at the same time. 00000000 - no trigger selected 00000001 - ADC TRIG0 selected 00000010 - ADC TRIG1 selected 00000100 - ADC TRIG2 selected 00001000 - ADC TRIG3 selected 00010000 - ADC TRIG4 selected 00100000 - ADC TRIG5 selected 01000000 - ADC TRIG6 selected 10000000 - ADC TRIG7 selected
3-0 CSEL6	ADC hardware trigger command selection. For more information, see the ADC chapter. **NOTE:** TCRTLx[CMD_SEL] of ADC should be set to 1. 0000 - Not a valid selection from the command buffer. Trigger event is ignored. 0001 - ADC CMD1 selected. 0010 - ADC CMD2 selected. 0011 - ADC CMD3 selected.

Field	Description
	0100 - ADC CMD4 selected. 0101 - ADC CMD5 selected. 0110 - ADC CMD6 selected. 0111 - ADC CMD7 selected. 1000 - ADC CMD8 selected. 1001 - ADC CMD9 selected. 1010 - ADC CMD10 selected. 1011 - ADC CMD11 selected. 1100 - ADC CMD12 selected. 1101 - ADC CMD13 selected. 1110 - ADC CMD14 selected. 1111 - ADC CMD15 selected.

88.6.1.12 ETC_TRIG Result Data 1/0 Register (TRIG0_RESULT_1_0 - TRIG7_RESULT_1_0)

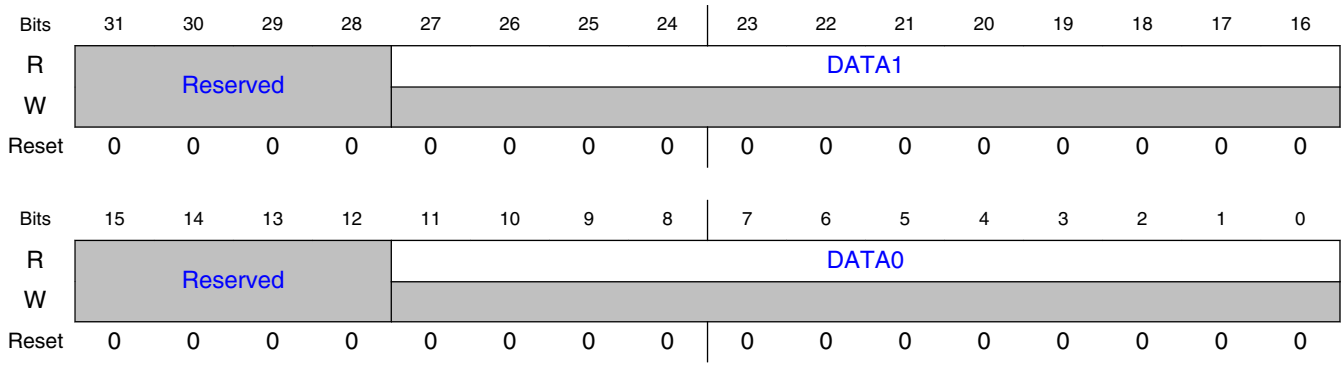
This register stores the results of the ADC conversions corresponding to segment 0 and segment 1 of TRIGa.

88.6.1.12.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_RESULT_1_0	28h + (a × 28h)

88.6.1.12.2 Diagram



88.6.1.12.3 Fields

Field	Description
31-28 —	Reserved
27-16 DATA1	Result DATA1 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.
15-12 —	Reserved
11-0 DATA0	Result DATA0 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.

88.6.1.13 ETC_TRIG Result Data 3/2 Register (TRIG0_RESULT_3_2 - TRIG7_RESULT_3_2)

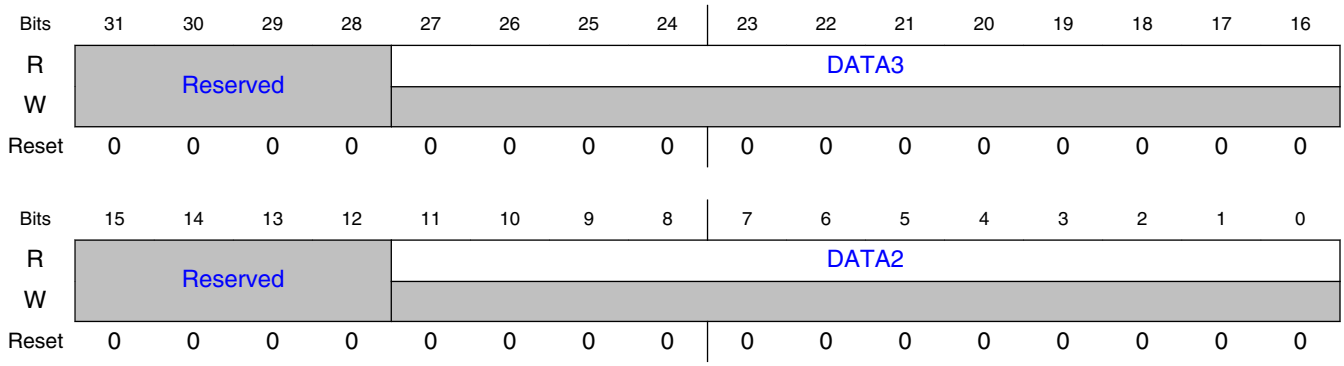
This register stores the results of the ADC conversions corresponding to segment 2 and segment 3 of TRIGa.

88.6.1.13.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_RESULT_3_2	2Ch + (a × 28h)

88.6.1.13.2 Diagram



88.6.1.13.3 Fields

Field	Description
31-28 —	Reserved
27-16 DATA3	Result DATA3 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.
15-12 —	Reserved
11-0 DATA2	Result DATA2 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.

88.6.1.14 ETC_TRIG Result Data 5/4 Register (TRIG0_RESULT_5_4 - TRIG7_RESULT_5_4)

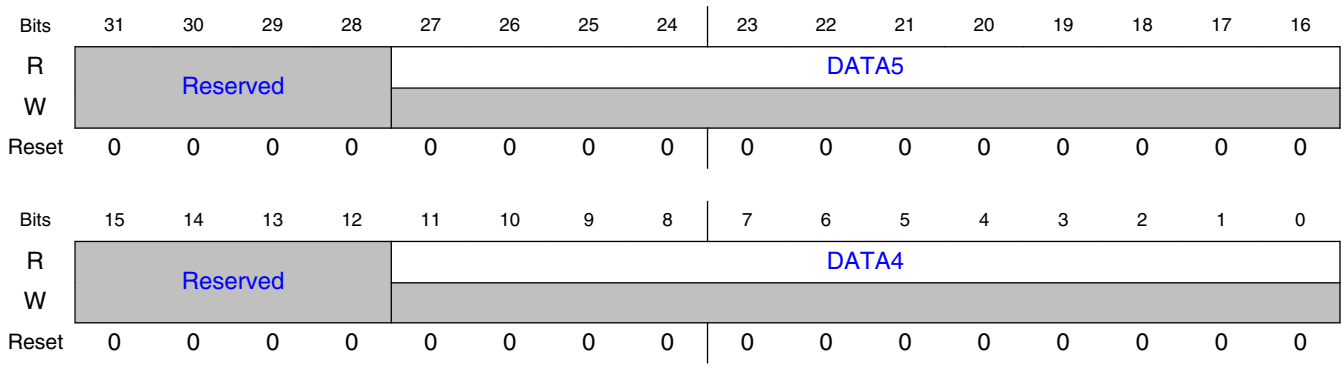
This register stores the results of the ADC conversions corresponding to segment 4 and segment 5 of TRIGa.

88.6.1.14.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_RESULT_5_4	30h + (a × 28h)

88.6.1.14.2 Diagram



88.6.1.14.3 Fields

Field	Description
31-28 —	Reserved
27-16 DATA5	Result DATA5 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.
15-12 —	Reserved
11-0 DATA4	Result DATA4 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.

88.6.1.15 ETC_TRIG Result Data 7/6 Register (TRIG0_RESULT_7_6 - TRIG7_RESULT_7_6)

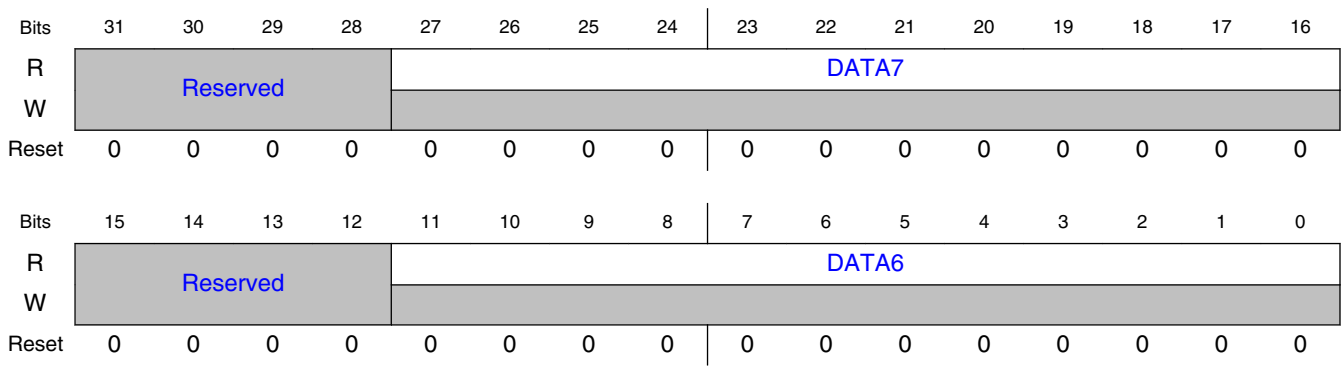
This register stores the results of the ADC conversions corresponding to segment 6 and segment 7 of TRIGa.

88.6.1.15.1 Offset

For a = 0 to 7:

Register	Offset
TRIGa_RESULT_7_6	34h + (a × 28h)

88.6.1.15.2 Diagram



88.6.1.15.3 Fields

Field	Description
31-28 —	Reserved
27-16 DATA7	Result DATA7 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.
15-12 —	Reserved
11-0 DATA6	Result DATA6 **NOTE:** The sign bit from ADC result FIFO is ignored by ETC_TRIG result, so only 12-bit unsigned results is supported by ADC_ETC module.

Chapter 89
Digital-to-Analog Converters (DAC)

89.1 Chip-specific DAC information

Table 89-1. Reference links to related information

Topic	Related module(s)	Reference
System memory map	-	System Memory Map
Clocking	CCM	Clock Management Clock Control Module (CCM)
Power management	PMU	Power Management Power Management Unit
Signal multiplexing	IOMUX	External Signals and Pin Multiplexing IOMUX
Interrupts, DMA Events and XBAR Assignments	-	Interrupts, DMA Events and XBAR Assignments

The only DAC reference option for this chip is VREFH, which uses the ADC_VREFH pin. VREFL uses the ADC_VREFL pin. DACRFS must be set to 1.

There is no hw trigger option defined for DAC on this chip.

IREF2 is not available on this chip.

89.2 Overview

The 12-bit digital-to-analog converter (DAC) is a low-power, general-purpose DAC. The output of the DAC can be placed on an external pin or set as one of the inputs to the analog comparator, op-amps, or ADC.

89.2.1 Block diagram

The block diagram of the DAC module is as follows:

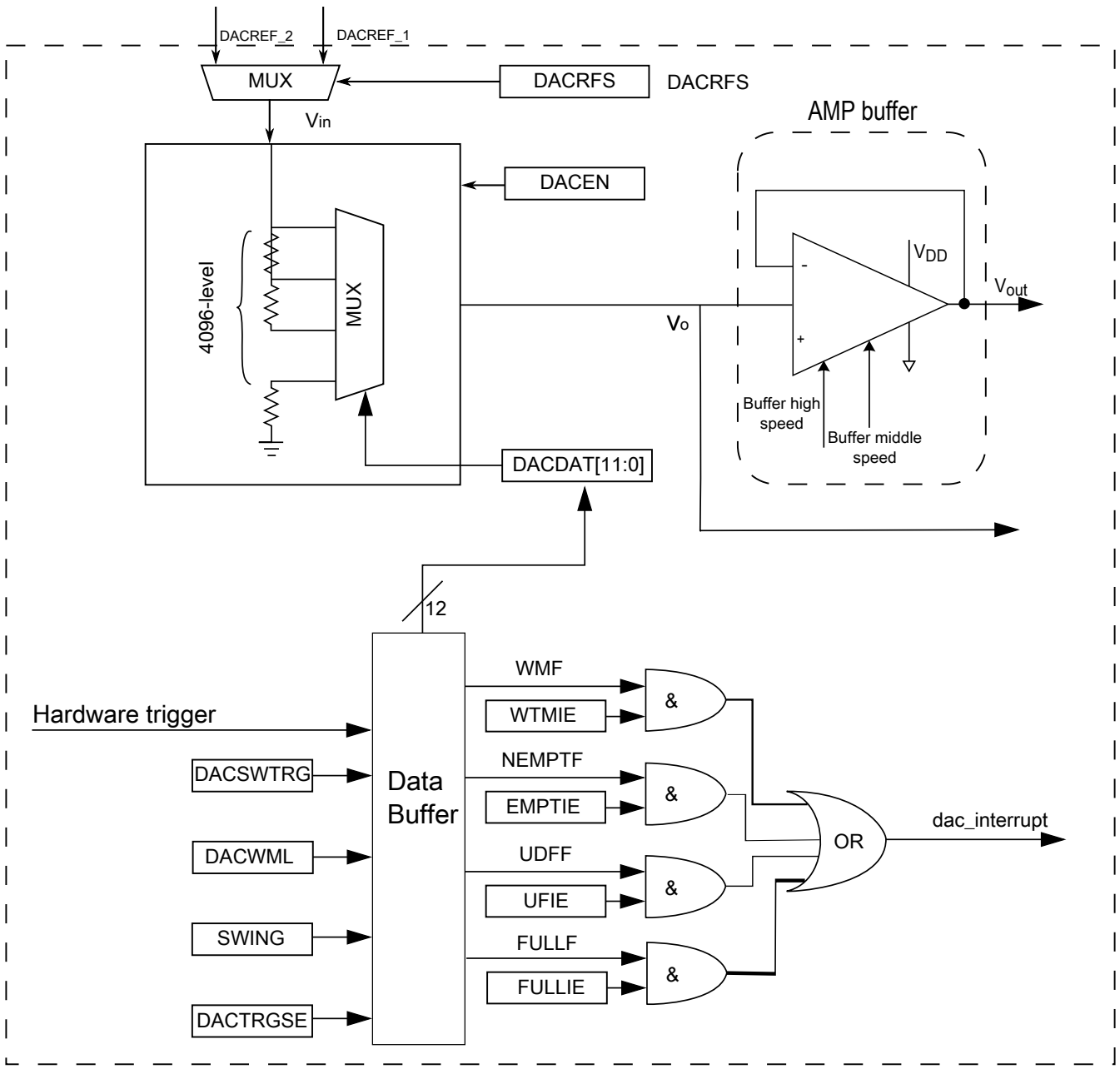


Figure 89-1. DAC block diagram

89.2.2 Features

The features of the DAC module include:

- On-chip programmable reference generator output. The voltage output range is from 1⁄4096 V_{in} to V_{in}, and the step is 1⁄4096 V_{in}, where V_{in} is the input voltage.
- V_{in} can be selected from two reference sources. See Chip-Specific Information section.
- 16 -word data buffer supported with configurable watermark and multiple operation modes
- DMA support

89.3 Functional description

The 12-bit DAC module can select one of the two reference inputs - DACREF_1 and DACREF_2 as the DAC reference voltage, V_{in} by C0[DACRFS]. See the chip-specific DAC information to determine the source options for DACREF_1 and DACREF_2.

When the DAC is enabled, it converts the data in DATA[11:0] or the data from the DAC data buffer to a stepped analog output voltage. The output voltage range is from V_{in} to V_{in}/4096, and the step is V_{in}/4096.

89.3.1 FIFO mode

In FIFO mode, the buffer is organized as a FIFO. For a valid write to the DATA register, the data is put into the FIFO, and the write pointer is automatically incremented. The module is connected internally to a 32-bit interface. For any 16-bit or 8-bit FIFO access, address bit[1] needs to be 0; otherwise, the write is ignored. A successful 32-bit FIFO write increases the write pointer by one.

For an 8-bit write, the LSB determines which byte lane is written to. Only if both the byte lanes are written to, will the write pointer increase. The user needs to ensure that an 8-bit access is done in pair, and both the upper and lower bytes are written. There is no priority for which byte is written to first.

FIFO mode needs to be enabled with DACCR[FIFOEN].

89.3.2 DAC data buffer operation

When the DAC FIFO is disabled, and the one entry buffer is enabled, the DAC converts the data in the buffer to analog output voltage. Any write to the DATA register will replace the data in the buffer and push data to analog conversion without trigger support.

When the IP interface bus access (IPS) is in write state, the write speed of DATA[DATA0] is high. However, the analog part of the DAC cannot convert data at the same speed. Therefore, the multi write speed limit needs to be defined in the firmware.

89.3.3 DAC interrupts

There are several interrupts and associated flags that can be configured for the DAC buffer.

FIFO request flag

The FIFO request flag is set based on the number of entries in the FIFO and the FIFO watermark configuration. This flag is set when the number of entries in any of the enabled FIFOs is less than or equal to the FIFO watermark configuration and is automatically cleared when the number of entries in FIFO is greater than the FIFO watermark configuration. The FIFO request flag can generate an interrupt or a DMA request.

FIFO warning flag

The FIFO warning flag is set based on the number of entries in the FIFO. The empty flag and full flags are both warning flags. The FIFO warning flag can generate an interrupt, or a DMA request (for empty flag only).

FIFO error flag

The FIFO error flag is set when overflow and underflow happens.

89.3.4 DMA operation

When DMA is enabled, DMA requests are generated instead of interrupt requests. The DMA Done signal clears the DMA request.

The status register flags are still set and are cleared automatically when the DMA completes.

89.3.5 Resets

During reset, the DAC is configured in the default mode and is disabled.

89.3.6 Low-Power mode operation

The following table shows the wait mode and the stop mode operation of the DAC module.

Table 89-2. Modes of operation

Modes of operation	Description
Wait mode	The DAC will operate normally, if enabled.
Stop mode	If enabled, the DAC module continues to operate normally, with the hardware trigger initiating the conversion.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

89.3.7 Background Debug Mode Operation

When the microcontroller is in active background debug mode, the DAC continues to operate normally.

89.4 Memory map/register definition

The DAC has registers to control analog comparator and programmable voltage divider to perform the digital-to-analog functions.

89.4.1 DAC register descriptions

89.4.1.1 DAC memory map

DAC base address: 4006_4000h

Offset (hex)	Register	Width (In bits)	Access	Reset value (hex)
0	Version Identifier Register (VERID)	32	RO	0100_0000
4	Parameter Register (PARAM)	32	RO	0000_0003
8	DAC Data Register (DATA)	32	WO	0000_0000
C	DAC Status and Control Register (CR)	32	RW	0000_0002
10	DAC FIFO Pointer Register (PTR)	32	RO	0000_0000
14	DAC Status and Control Register 2 (CR2)	32	RW	0000_0000

89.4.1.2 Version Identifier Register (VERID)

The Version ID register indicates the version integrated for this instance on the device.

89.4.1.2.1 Offset

Register	Offset
VERID	0h

89.4.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

89.4.1.2.3 Fields

Field	Description
31-24	Major version number
MAJOR	This read-only field returns the major version number for the module specification.
23-16	Minor version number

Table continues on the next page...

Field	Description
MINOR	This read-only field returns the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number This read-only field returns the feature set number. 0000000000000000 - Standard feature set 0000000000000001 - C40 feature set 0000000000000010 - 5V DAC feature set 0000000000000100 - ADC BIST feature set

89.4.1.3 Parameter Register (PARAM)

The Parameter register indicates the feature parameters for this instance on the device.

89.4.1.3.1 Offset

Register	Offset
PARAM	4h

89.4.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													FIFOSZ		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

89.4.1.3.3 Fields

Field	Description
31-3 —	Reserved
2-0	FIFO size

Field	Description
FIFOSZ	The number of words in the DAC FIFO is 2^(FIFOSZ+1). 000 - FIFO depth is 2 001 - FIFO depth is 4 010 - FIFO depth is 8 011 - FIFO depth is 16 100 - FIFO depth is 32 101 - FIFO depth is 64 110 - FIFO depth is 128 111 - FIFO depth is 256

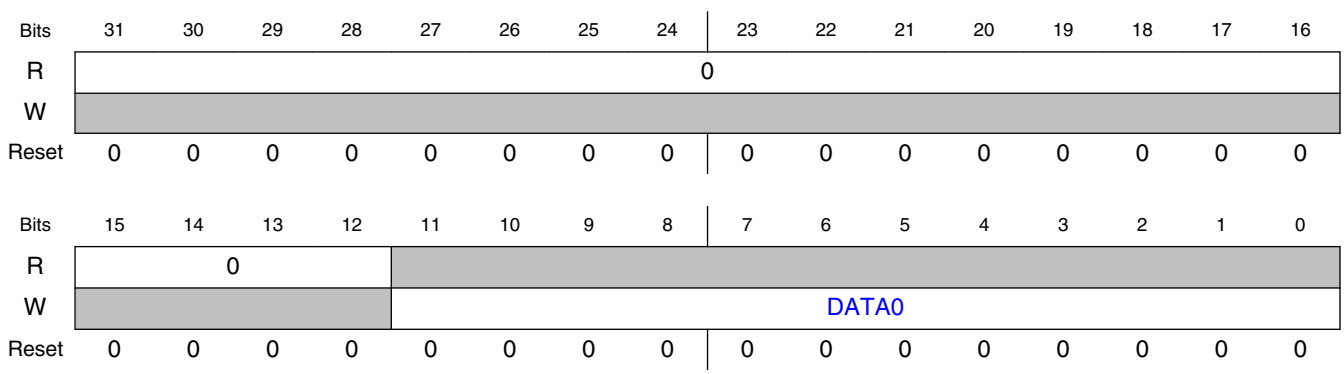
89.4.1.4 DAC Data Register (DATA)

The DAC Data Register holds the entry of FIFO in FIFO mode. It also works as the data buffer for buffer mode.

89.4.1.4.1 Offset

Register	Offset
DATA	8h

89.4.1.4.2 Diagram



89.4.1.4.3 Fields

Field	Description
31-12	Reserved

Table continues on the next page...

Field	Description
—	
11-0 DATA0	FIFO DATA0

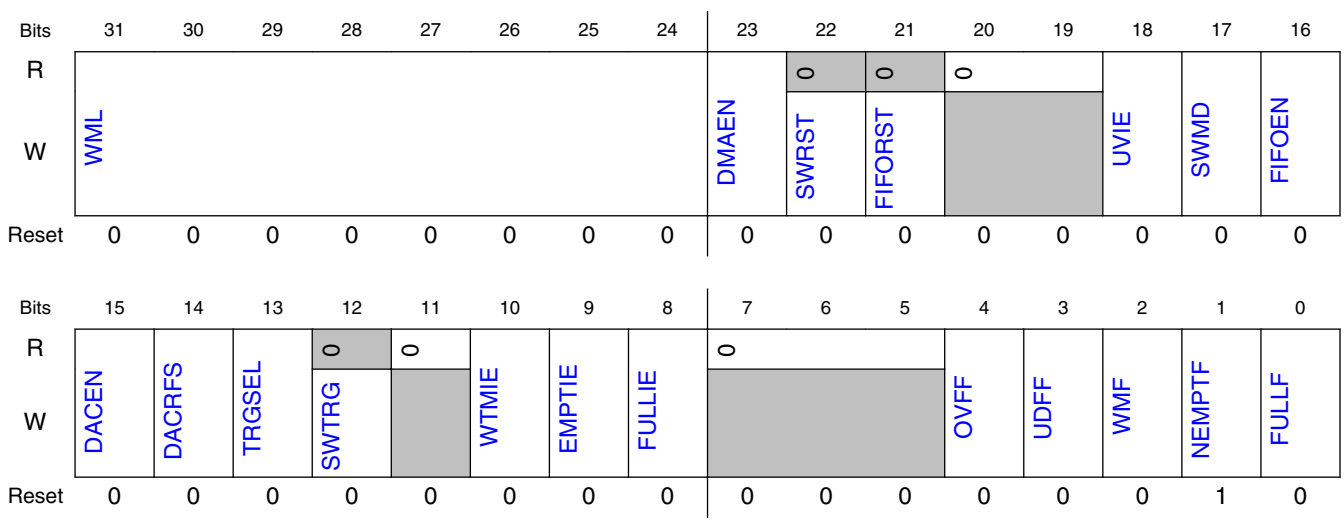
89.4.1.5 DAC Status and Control Register (CR)

The DAC Status and Control Register contains configuration options for the DAC operation, such as DMA and interrupt, trigger select and FIFO mode select. It also includes the FIFO status flags.

89.4.1.5.1 Offset

Register	Offset
CR	Ch

89.4.1.5.2 Diagram



89.4.1.5.3 Fields

Field	Description
31-24 WML	Watermark Level Select MAX is the FIFO size.

Table continues on the next page...

Field	Description
	NOTE: If the FIFO depth is 2, then all settings are treated the same. That is, if the FIFO is not full, the watermark status bit and NEMPTF will be set. One of the status bits maybe needed for DMA or IRQ request.
23 DMAEN	DMA Enable Select 0 - DMA is disabled. 1 - DMA is enabled. When DMA is enabled, the DMA request will be generated by original interrupts. The interrupts will not be presented on this module at the same time.
22 SWRST	Software reset Resets all internal logic and registers, in case of any failure during the Low-power mode. This field is written to only to trigger reset, self clear, or read 0.
21 FIFORST	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. FIFO pointers should only be reset when the DAC is disabled. This field can be used to configure both pointers to the same address to reset the FIFO as empty. This field is written to only to trigger reset, self clear, or read zero. 0 - No effect 1 - FIFO reset
20-19 —	Reserved
18 UVIE	Underflow and overflow interrupt enable 0 - Underflow and overflow interrupt is disabled. 1 - Underflow and overflow interrupt is enabled.
17 SWMD	DAC FIFO Mode Select In Swing back mode, the FIFO must be set up such that the FIFO is full. In Swing back mode, a trigger changes the read pointer to make it swing between the FIFO Full and Nearly Empty state. That is, the trigger increases the read pointer till FIFO is nearly empty and decreases the read pointer till the FIFO is full, and so on. 0 - Normal mode 1 - Swing back mode
16 FIFOEN	FIFO Enable 0 - FIFO is disabled and only one level buffer is enabled. Any data written from this buffer goes to conversion. 1 - FIFO is enabled. Data will first read from FIFO to buffer then go to conversion.
15 DACEN	DAC Enable Starts the Programmable Reference Generator operation. 0 - The DAC system is disabled. 1 - The DAC system is enabled.
14 DACRFS	DAC Reference Select 0 - The DAC selects DACREF_1 as the reference voltage. 1 - The DAC selects DACREF_2 as the reference voltage.
13 TRGSEL	DAC Trigger Select 0 - The DAC hardware trigger is selected. 1 - The DAC software trigger is selected.

Table continues on the next page...

Field	Description
12 SWTRG	DAC Software Trigger Active high. This is a write-only field, which always reads 0. If DAC software trigger is selected and FIFO is enabled, writing 1 to this field will advance the FIFO read pointer once. **NOTE:** It is not suggested to adopt hardware trigger and software trigger in mixed use, during continuous conversions. 0 - The DAC soft trigger is not valid. 1 - The DAC soft trigger is valid.
11 —	Reserved
10 WTMIE	Watermark Interrupt Enable 0 - Watermark interrupt is disabled. 1 - Watermark interrupt is enabled.
9 EMPTIE	Nearly Empty Interrupt Enable 0 - FIFO Nearly Empty interrupt is disabled. 1 - FIFO Nearly Empty interrupt is enabled.
8 FULLIE	Full Interrupt Enable 0 - FIFO Full interrupt is disabled. 1 - FIFO Full interrupt is enabled.
7-5 —	Reserved
4 OVFF	Overflow Flag This is the FIFO overflow status flag. This flag indicates that more data has been written into FIFO than it can hold. This field asserts regardless of the value of UVIE. However, an interrupt is issued to the host only if UVIE is set. This flag is cleared by writing a 1 to it. 0 - No overflow has occurred since the last time the flag was cleared. 1 - At least one FIFO overflow has occurred since the last time the flag was cleared.
3 UDFF	Underflow Flag This is the FIFO underflow status flag. If this field is set, it means that there is a new trigger after NEMPTF is set. This field asserts regardless of the value of UVIE. However, an interrupt is issued to the host only if UVIE is set. This flag is cleared by writing a 1 to it. 0 - No underflow has occurred since the last time the flag was cleared. 1 - At least one trigger underflow has occurred since the last time the flag was cleared.
2 WMF	FIFO Watermark Status Flag This field is set if the remaining FIFO data is less than or equal to the watermark setting. It is cleared automatically by writing data into FIFO by DMA or CPU. Write to this bit is ignored in FIFO mode. 0 - The DAC buffer read pointer has not reached the watermark level. 1 - The DAC buffer read pointer has reached the watermark level.
1 NEMPTF	Nearly Empty Flag This is the FIFO nearly empty flag. It is set when only one data remains in FIFO. 0 - More than one data is available in the FIFO. 1 - One data is available in the FIFO.

Table continues on the next page...

Field	Description
0 FULLF	Full Flag This is the FIFO Full status bit. This means that the FIFO read pointer equals the write pointer, because of the write pointer increase. This field is cleared if any DAC trigger is making the DAC read pointer increase. Any write to this field is ignored in FIFO mode. 0 - FIFO is not full. 1 - FIFO is full.

89.4.1.6 DAC FIFO Pointer Register (PTR)

The DAC FIFO Pointer Register contains the FIFO read pointer and write pointer.

89.4.1.6.1 Offset

Register	Offset
PTR	10h

89.4.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DACRFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DACWFP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

89.4.1.6.3 Fields

Field	Description
31-24 —	Reserved
23-16 DACRFP	DACRFP This is the FIFO read pointer.

Table continues on the next page...

Field	Description
	NOTE: The read pointer is later than the trigger for one cycle. After the conversion finishes and the next trigger comes, then the read pointer updates in the hardware trigger mode.
15-8 —	Reserved
7-0 DACWFP	DACWFP This is the FIFO write pointer. It's value is initially set to equal the read pointer value automatically, and the FIFO status is empty. Both write pointer and read pointer in FIFO mode do not change by IPS read access.

89.4.1.7 DAC Status and Control Register 2 (CR2)

The DAC Status and Control Register 2 contains configuration options for the analog DAC.

NOTE

For CR2[6:4] (IREF and IREF1), it is suggested to enable one of the bits to reduce power. If more than one bit is selected, the power consumption will increase but the conversion rate will also speed up.

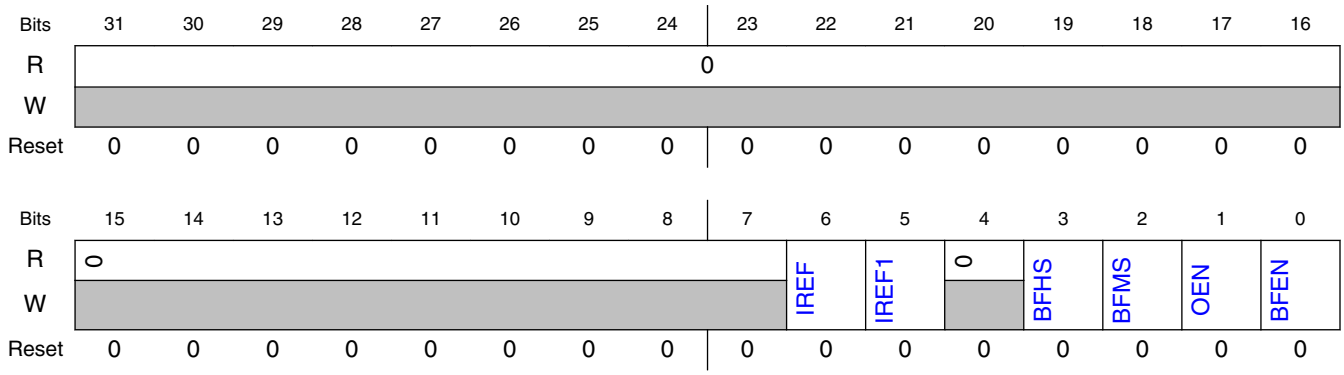
NOTE

IREF works in all modes, but the operation mode of IREF1 depends on PMC.

89.4.1.7.1 Offset

Register	Offset
CR2	14h

89.4.1.7.2 Diagram



89.4.1.7.3 Fields

Field	Description
31-7 —	Reserved
6 IREF	Internal Current Reference Select 0 - Internal Current Reference not selected 1 - Internal Current Reference selected
5 IREF1	Internal ZTC (Zero Temperature Coefficient) Current Reference Select 0 - Internal ZTC Current Reference not selected 1 - Internal ZTC Current Reference selected
4 —	Reserved
3 BFHS	Buffer High Speed Select 0 - Buffer high speed not selected 1 - Buffer high speed selected
2 BFMS	Buffer Middle Speed Select 0 - Buffer middle speed not selected 1 - Buffer middle speed selected
1 OEN	Optional Enable 0 - Output buffer is not bypassed 1 - Output buffer is bypassed
0 BFEN	Buffer Enable 0 - Opamp is not used as buffer 1 - Opamp is used as buffer

Appendix A
Revision History

A.1 CCM Revision History

Topic Cross-Reference	Change Description
MISC register description	• Updated MISC_DIFPROG register.

A.2 ADC_ETC Revision History

Topic Cross-Reference	Change Description
Chip-specific ADC_ETC information	• Added a note for ADC_ETC Global Control Register(CTRL)

A.3 QDC Revision History

Topic Cross-Reference	Change Description
QDC register descriptions	• Removed FILT_PRSC bit from Input Filter (FILT) register.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.
ColdFire+ — is a trademark of NXP B.V.
CoolFlux — is a trademark of NXP B.V.
EdgeLock — is a trademark of NXP B.V.
eIQ — is a trademark of NXP B.V.
GreenChip — is a trademark of NXP B.V.
HITAG — is a trademark of NXP B.V.
Immersiv3D — is a trademark of NXP B.V.
Mantis — is a trademark of NXP B.V.
MIFARE — is a trademark of NXP B.V.
MIFARE Classic — is a trademark of NXP B.V.
MIFARE4Mobile — is a trademark of NXP B.V.
MIFARE Plus — is a trademark of NXP B.V.
NTAG — is a trademark of NXP B.V.
PEG — is a trademark of NXP B.V.
Processor Expert — is a trademark of NXP B.V.
QorIQ — is a trademark of NXP B.V.
QorIQ Qonverge — is a trademark of NXP B.V.
RoadLINK — wordmark and logo are trademarks of NXP B.V.
SafeAssure — is a trademark of NXP B.V.
SafeAssure — logo is a trademark of NXP B.V.
SmartMX — is a trademark of NXP B.V.
VortiQa — is a trademark of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2025 All rights reserved.

For more information, please visit: http://www.nxp.com

Date of release: May 2025
Document identifier: IMXRT1170RM